ParkPredict+: Multimodal Intent and Motion Prediction for Vehicles in Parking Lots with CNN and Transformer

Xu Shen, Matthew Lacayo, Nidhir Guggilla, and Francesco Borrelli

Abstract—The problem of multimodal intent and trajectory prediction for human-driven vehicles in parking lots is addressed in this paper. Using models designed with CNN and Transformer networks, we extract temporal-spatial and contextual information from trajectory history and local bird’s eye view (BEV) semantic images, and generate predictions about intent distribution and future trajectory sequences. Our methods outperform existing models in accuracy, while allowing an arbitrary number of modes, encoding complex multi-agent scenarios, and adapting to different parking maps. To train and evaluate our method, we present the first public 4K video dataset of human driving in parking lots with accurate annotation, high frame rate, and rich traffic scenarios.

I. INTRODUCTION

While the rapid advancement of self-driving technology in the past decade has brought people much closer to an era with automated mobility, autonomous vehicles (AVs) still face great challenges in interacting with other road users safely and efficiently. In addition to having a robust perception system, the ability to make predictions and infer the potential future intents and trajectories of other vehicles will be essential for AVs to make optimal decisions.

Researchers have made great strides in the field of motion prediction. Physics model-based methods [1] such as the Kalman Filter and its variants leverage the dynamics of the vehicle to propagate the state of the vehicle forward for intuitive short-term predictions. Reachability study [2] also provides a formal way to measure the uncertainties of vehicle behavior for the control design.

When vehicles operate in complex environments, model-based approaches tend to suffer from the difficulty of accurate modeling, along with the burden of heavy computation. In contrast, deep learning methods have demonstrated great potential to incorporate various forms of information and generalize to new environments. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks [3] are widely known for learning sequential data, and various research papers have also focused on adapting the networks to account for the multi-agent interactions [4]. There is also interest in using Convolutional Neural Networks (CNN) to make predictions [5] where vehicle trajectories and local environments can be embedded in images efficiently.

In recent years, Transformer networks [6] have achieved great success in Natural Language Processing tasks. Their attention mechanism helps to keep track of global dependencies in input and output sequences regardless of the relative position, which overcomes the limitations of RNNs in learning from long temporal sequences [7]. For trajectory prediction tasks [8], Transformer networks have been shown to outperform the LSTMs in many aspects, including accuracy [9] and interaction modeling [7].

Most of the existing work mentioned above focuses on pedestrians or vehicles driving in a road network. These environments feature simple dynamics or clear lane markings and traffic rules. However, for vehicles driving in a parking lot, we are faced with the following challenges:

1) There is no strict enforcement of traffic rules regarding lane directions and boundaries.
2) The vehicles need to perform complex parking maneuvers to drive in and out of the parking spots.
3) There are few public datasets for the motion of vehicles in parking lots. The existing ones such as CNRPark+EXT [10] and CARPK [11] are only for car detection in images and do not provide continuous trajectories in ground coordinates.

The ParkPredict [12] work addresses the vehicle behavior prediction problem in parking lots by using LSTM and CNN networks. However, in [12] the driving data was collected using a video game simulator to model just a single vehicle. Also, the problem formulation was restricted to a specific global map and could not be generalized. In this work, we are presenting ParkPredict+, an extensible approach that generates multimodal intent and trajectory prediction with CNN and Transformer. We also present the Dragon Lake Parking (DLP) dataset, the first human driving dataset in a parking lot environment. We offer the following contributions:

1) We propose a CNN-based model to predict the probabilities of vehicle intents in parking lots. The model is agnostic to the global map and the number of intents.
2) We propose a Transformer and CNN-based model to predict the future vehicle trajectory based on intent, image, and trajectory history. Multimodality is achieved by coupling the model with the top-k intent prediction.
3) We release the DLP dataset and its Python toolkit for autonomous driving research in parking lots.

The paper is organized as follows: Section II formulates the multimodal intent and trajectory prediction problem, Section III elaborates on the model design of ParkPredict++, Section IV discusses the dataset, experiment setting, and results, and finally Section V concludes the paper.

II. PROBLEM FORMULATION

We aim to generate multimodal intent and trajectory predictions for vehicles driving in parking lots.
A. Inputs

We make predictions based on two types of input:

1) Trajectory history: Denote the current time step as 0, the trajectory history \( Z_{\text{hist}} = \{z(t)\}_{t=-(N_{\text{hist}}+1)}^{0} \in \mathbb{R}^{N_{\text{hist}} \times 3} \) is the sequence of target vehicle states \( z(t) = (x(t), y(t), \psi(t)) \in \mathbb{R}^{3} \) sampled backward in time from 0 with horizon \( N_{\text{hist}} \) and step interval \( \Delta t \). For convenience of notation, we denote the index set of history time steps by \( T_{\text{hist}} = \{- (N_{\text{hist}} - 1), \ldots, 0\} \).

To obtain better numerical properties and generalize the model to different maps, all states mentioned in this paper are local with respect to the vehicle body frame at \( t = 0 \), therefore indicating that \( z(0) \equiv (0, 0, 0) \).

2) Local Contextual Information: Similarly to [5], we generate a rasterized bird’s eye view (BEV) image of size \( n \times n \) and resolution \( r \) that encodes the local environment and neighboring agents of the target vehicle. This image reflects the farthest sensing limit \( L \) of the target vehicle in four directions to make decisions. Each color represents a distinct type of objects. As shown in Fig. 1a, the driving lanes are local with respect to the vehicle body frame at \( t = 0 \), neighboring agents of the target vehicle. This image reflects the farthest sensing limit \( L \) of the target vehicle in four directions to make decisions. Each color represents a distinct type of objects. As shown in Fig. 1a, the driving lanes are plotted in gray, static obstacles in blue, open spaces in green, the target vehicle in red, and other agents in yellow.

Since we are interested in representing the local context of the target vehicle, we always position the target vehicle in red, and other agents in yellow.

B. Outputs

The outputs are the probability distributions over intents and future trajectory sequence:

1) Intent: We define the intent to be a location \( \eta = (x, y) \) in the local context that the vehicle is “aiming at” in the long term. The vehicle does not need to reach \( \eta \) at the end of the prediction horizon, but the effort to approach it would be implicitly contained along the trajectory.

In this work, we are interested in two types of intents around the target vehicle: empty parking spots and lanes. We assume the existence of certain algorithms to detect all possible intents through BEV or other on-board sensor outputs, as shown in Fig. 2. Given \( M_e \) empty spots and \( M_d \) lanes, the intent prediction will output a probability distribution for the \( M = M_e + M_d \) detected intents

\[
\hat{p} = \left[ p_s^{[1]}, \ldots, p_s^{[M_e]}, p_d^{[1]}, \ldots, p_d^{[M_d]} \right] \in \Delta^{M-1}
\]  

where \( p_s^{[i]}, i \in \mathcal{N}_e = \{1, \ldots, M_e\} \) represents the probability that the vehicle will choose the \( i \)-th parking spot centered at \( \eta_s^{[i]} = (x_s^{[i]}, y_s^{[i]}) \), and \( p_d^{[j]}, j \in \mathcal{N}_d = \{1, \ldots, M_d\} \) represents the probability that the vehicle will bypass all spots and continue driving along the \( j \)-th lane towards \( n_d^{[j]} = (x_d^{[j]}, y_d^{[j]}) \), which is at the boundary of its sensing limit.

2) Future trajectories: The trajectory output \( Z_{\text{pred}} = \{\hat{z}(t)\}_{t=1}^{N_{\text{pred}}} \in \mathbb{R}^{N_{\text{pred}} \times 3} \) is the sequence of future vehicle states along the prediction horizon \( N_{\text{pred}} \).

As to be discussed in Section III-B, the trajectory prediction model takes the intent as input, so given a probability \( p^{[m]} \) of choosing intent \( \eta^{[m]}, m \in \{1, \ldots, M\} \), the predicted trajectory \( Z_{\text{pred}}^{[m]} \) is also distributed with probability \( p^{[m]} \).

III. ParkPredict+

Our model has two main components: 1) a Convolutional Neural Network (CNN) model to predict vehicle intent based on local contextual information, and 2) a CNN-Transformer model to predict future trajectory based on the trajectory history, image history, and the predicted intent.

A. Intent Prediction

We first consider a distribution

\[
\tilde{p} = \left[ p_s^{[1]}, \ldots, p_s^{[M_e]}, p_d^{[1]}, \ldots, p_d^{[M_d]} \right] \in \Delta^{M_e}
\]

where there are only the probabilities of choosing the spots \( 1, \ldots, M_e \) and bypassing all of them (denoted \( p_d \)). Compared to Eq. (1), it is intuitive to have \( p_{-d} = \sum_{j=1}^{M_d} p_d^{[j]} \).

Given the \( i \)-th spot intent \( p_s^{[i]} = (x_s^{[i]}, y_s^{[i]}) \), we can generate a spot-specific image \( I^{[i]}(0) \) by painting the corresponding spot a different color in the BEV image \( I(0) \), as shown in Fig. 3. We could also generate supplementary features for
Intent Predict CNN

Flatten

Feed Forward

Score

Encoder

Add & Norm

Feed Forward

Add & Norm

Multi-Head Attention

Add & Norm

Masked Multi-Head Attention

Feed Forward

Concatenation

Base CNN

Add & Norm

Intent Multi-Head Attention

Add & Norm

Intent

Add & Norm

FEED Forward

Decoder

Intent

Add & Norm

FEED Forward

Encoder

Add & Norm

FEED Forward

Score

Intent

Fig. 3: Mapping $f$ constructed by CNN and feed forward layers. The $i$-th spot is painted in purple on image $I^{i}_{(0)}$.

this intent, such as the distance to it $||y_{k}^{i}||$, and the difference of heading angle $|\Delta\psi_{i}| = |\arctan2(y_{k}^{i}, x_{a}^{i})|$.

Denoting by $f : \mathbb{R}^{n \times n} \times \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ a mapping that assigns a “score” to an intent, we can compute the scores for all $M_s$ spot intents as

$$s^{[i]} = f \left( I^{i}_{(0)}, ||y_{k}^{i}||, |\Delta\psi_{i}| \right), i \in \mathcal{N}_s,$$  \hspace{1cm} (2)

Without painting any particular spot on $I(0)$ and setting the distance and angle to 0, the score for bypassing all spots is

$$s^{[0]} = f \left( I(0), 0, 0 \right).$$  \hspace{1cm} (3)

Getting a higher score means that the corresponding intent is more likely.

We train a CNN based model as the mapping $f$. The model architecture is demonstrated in Fig. 3. The loss function is binary cross-entropy (BCE) between the predicted score $\hat{s}^{[i]}$ and the ground truth label $s^{[i]}$.

$$\ell = - \left[ s^{[i]} \log \hat{s}^{[i]} + \left( 1 - s^{[i]} \right) \log \left( 1 - \hat{s}^{[i]} \right) \right],$$  \hspace{1cm} (4)

where $s^{[i]} = 1$ if the vehicle chooses the $i$-th spot, and $s^{[i]} = 0$ if not.

The probability outputs are normalized scores

$$p^{[i]}_{a} = \frac{s^{[i]}_{a}}{\sum_{j=0}^{M_s} s^{[j]}_{a}}, i \in \mathcal{N}_s,$$  \hspace{1cm} (5a)

$$p_{-a} = \frac{\sum_{j=0}^{M_s} s^{[j]}_{-a}}{\sum_{j=0}^{M_s} s^{[j]}_{a}}.$$  \hspace{1cm} (5b)

To split $p_{-a}$ into the probabilities of continuing to drive through different lanes, we generate a set of weights $\{w^{[j]} \in [0, 1] | j \in \mathcal{N}_d\}$ as an arithmetic sequence and reorder them based on a simple heuristic: an intent $\eta^{[j]}$ which requires more steering and longer driving distance will have lower weight $w^{[j]}$. Then, the probabilities are split as

$$p^{[j]}_{a} = \frac{w^{[j]}}{\sum_{k=1}^{M_d} w^{[k]}}, j \in \mathcal{N}_d.$$  \hspace{1cm} (6)

We would like to highlight here that since $f$ is only a mapping from a single intent to its score, the intent prediction model proposed above is invariant to the number of detected intents $M_s$ and $M_d$ in the local context.

B. Trajectory Prediction

We leverage the Multi-Head Attention mechanism and Transformer architecture [6] to construct the trajectory prediction model $\mathcal{F} : \mathbb{R}^{N_{hist} \times 3} \times \mathbb{R}^{N_{hist} \times n \times n} \times \mathbb{R}^{2} \rightarrow \mathbb{R}^{N_{pred} \times 3}$, which predicts future trajectory $Z_{pred}$ with the trajectory history $Z_{hist}$, image history $I_{hist}$, and intent $\eta$. The model structure is illustrated in Fig. 4 and will be elaborated below.

1) Positional Encoding: As pointed out in [6], the attention mechanism does not contain recurrence or convolution, so we need to inject unique position encoding to inform the model about the relative position of data along the horizon. The positional encoding mask $PE$ is calculated as sinusoidal waves [13]:

$$PE_{k,i} = \begin{cases} 
\sin \left( \frac{t}{10000^{i/D}} \right), & i = 2k \\
\cos \left( \frac{t}{10000^{(i-1)/D}} \right), & i = 2k + 1 
\end{cases},$$  \hspace{1cm} (7)

where $t$ denotes the time step along the input horizon, $D$ denotes the model dimension, $i \in \{1, \ldots, D\}$.

2) Transformer Encoder: The image history $I$ is first processed by a base CNN network $g : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{d_{img}}$ to encode contextual information: $X_{img}^{in}(i) = g(I(t)), t \in T_{hist}$.

Subsequently the processed image history is concatenated with the trajectory history, projected to the model dimension
We define a linear encoder \( \Phi_{en} : \mathbb{R}^{d_{\text{imgs}}+3} \rightarrow \mathbb{R}^D \), and summed with the positional encoding as in Eq. (7):

\[
X^{\text{in}}_{\text{en}}(t) = \Phi_{en} \left[ \text{Concat} \left( X^{\text{in}}_{\text{imgs}}(t), z(t) \right) \right] \oplus PE_t, t \in T_{\text{last}} \tag{8}
\]

We then apply a classical Transformer Encoder \( \mathcal{F}_{\text{en}} : \mathbb{R}^{N_{\text{hist}} \times D} \rightarrow \mathbb{R}^{N_{\text{hist}} \times D} \) that consists of \( n_{\text{head}} \) self-attention layers and a fully connected layer: \( X^{\text{out}}_{\text{en}} = \mathcal{F}_{\text{en}}(X^{\text{in}}_{\text{en}}) \). Residual connections are employed around each layer.

3) Intent Embedding: Given a certain intent \( \eta \in \mathbb{R}^2 \), we apply a fully connected layer \( \Phi_{it} : \mathbb{R}^2 \rightarrow \mathbb{R}^D \) to embed it as latent state for the Transformer Decoder: \( X^{\text{out}}_{it} = \Phi_{it}(\eta) \).

We use the ground truth intent to train the model, and at run time we obtain multimodal trajectory predictions by picking intents with high probabilities.

4) Transformer Decoder: The decoder predicts the future trajectory in an autoregressive fashion. The input of the decoder \( X^{\text{in}}_{\text{de}} \in \mathbb{R}^{N_{\text{pred}} \times D} \) is the trajectory prediction shifted one step to the right, together with a linear projection and positional encoding as the encoder in Sect. III-B.2. The Masked Multi-Head Attention block also prevents the decoder from looking ahead to future time steps.

The Encoder Multi-Head Attention block uses the Transformer Encoder output \( X^{\text{out}}_{\text{en}} \in \mathbb{R}^{N_{\text{hist}} \times D} \) as the key (\( K \)) and value (\( V \)), and the output of the Masked Multi-Head Attention block \( X^{\text{out}}_{\text{mm}} \in \mathbb{R}^{N_{\text{pred}} \times D} \) as the query (\( Q \)) to compute cross-attention

\[
\text{attn}_{\text{en}} = \text{softmax} \left( \frac{X^{\text{out}}_{\text{en}} X^{\text{out}}_{\text{mm}}^\top}{\sqrt{D}} \right) X^{\text{out}}_{\text{en}}. \tag{9}
\]

We add the third block, Intent Multi-Head Attention, to compute the attention weights using intent so that the final trajectory prediction will be affected by intent. Here, we use intent embedding \( X^{\text{out}}_{it} \in \mathbb{R}^D \) as key (\( K \)) and value (\( V \)), and the output of the previous Multi-Head Attention block \( X^{\text{out}}_{\text{ma}} \in \mathbb{R}^{N_{\text{pred}} \times D} \) as the query (\( Q \))

\[
\text{attn}_{it} = \text{softmax} \left( \frac{X^{\text{out}}_{\text{ma}} X^{\text{out}}_{it}^\top}{\sqrt{D}} \right) X^{\text{out}}_{it}. \tag{10}
\]

Finally, we apply fully connected layers at the end to generate the trajectory output \( \mathcal{Z}_{\text{pred}} = \{ z(t) \}_{t=1}^{N_{\text{pred}}} \in \mathbb{R}^{N_{\text{pred}} \times 3} \). Residual connections are also employed in the decoder.

The loss function is L1 loss since 1) the gradient does not decrease as the prediction gets closer to the ground truth, and 2) L1 is more resistant to outliers in the dataset:

\[
L_1 = \sum_{t=1}^{N_{\text{pred}}} \sum_{i=1}^{3} | z(t)_i - \hat{z}(t)_i |
\]

\[
\hat{z}(t) = \text{attn}_{\text{en}, it, de} X^{\text{out}}_{\text{en}, it, de} + \text{attn}_{\text{en, it, de}} X^{\text{out}}_{\text{en, it, de}}
\]

The entire DLP dataset contains 30 scenes, 317,873 frames, 5,188 agents, and 15,383,737 instances.

In this work, we are using the sensing limit \( L = 10 \text{ m} \) and resolution \( r = 0.1 \text{ m/pixel} \), so that the semantic BEV image \( I \) is of size \( 200 \times 200 \). The sampling time interval \( \Delta t = 0.4s \), \( N_{\text{hist}} = N_{\text{tail}} = N_{\text{pred}} = 10 \). In other words, there are total of 8s’ information history encoded in the inputs and we are predicting the vehicle trajectory over the next 4s.

After data cleaning, filtering, and random shuffling, we obtain a training set of size 51750 and a validation set of size 5750. The models are trained on an Alienware Area

\[\text{https://sites.google.com/berkeley.edu/dlp-dataset}\]

\[\text{https://github.com/XuShenLZ/ParkSim/}\]
TABLE I: Hyperparameters of the Intent Prediction Model

| Types             | Parameters |
|-------------------|------------|
| Conv2d → BatchNorm | 8 × (7 × 7) → 8 |
| Dropout → LeakyReLU → MaxPool | 0.2 → 0.01 → 2 |
| Conv2d → BatchNorm | 8 × (5 × 5) → 8 |
| Dropout → LeakyReLU → MaxPool | 0.2 → 0.01 → 2 |
| Conv2d → BatchNorm | 3 × (3 × 3) → 3 |
| Dropout → LeakyReLU → MaxPool | 0.2 → 0.01 → 2 |
| Linear            | 6629 × 100 |
| Linear & Sigmoid  | 100 × 1    |

Fig. 7: Top-k accuracy of intent prediction

51m PC with 9th Gen Intel Core i9-9900K, 32GB RAM, and NVIDIA GeForce RTX 2080 GPU.

Since most well-known motion prediction benchmarks are not for the parking scenario, we choose the physics-based EKF model presented in the ParkPredict [12] paper as our baseline. The CNN-LSTM approach requires a different global map encoding thus cannot establish a fair comparison.

B. Intent Prediction

1) Hyperparameters: The hyperparameters of the intent prediction model in Fig. are outlined in Table. We use the Adam optimizer with learning rate 0.001 and stop early at convergence.

2) Evaluation Metrics: We use the top-k accuracy to evaluate our model: Let the set \( \mathcal{N}^{(i)}_k \subseteq \{1, \ldots, M^{(i)}\} \) include the \( k \) most likely intent indices in the \( i \)-th predicted distribution \( \hat{p}^{(i)} \) and the ground truth intent be at index \( t^{(i)} \), then the top-k accuracy \( A_k \) is computed as

\[
A_k = \frac{1}{M_D} \sum_{i=1}^{M_D} I\left(t^{(i)} \in \mathcal{N}^{(i)}_k \right).
\]

The variable \( M_D \) here corresponds to the cardinality of the validation set and \( I(\cdot) \) is the indicator function.

3) Results: Fig. 7 shows the Top-k prediction accuracy up to \( k = 5 \). The proposed method outperforms EKF methods for all values of \( k \) and achieves almost 100% accuracy in top-3 results, which means we can reliably cover the ground truth intent when making multimodal predictions.

C. Trajectory Prediction

1) Hyperparameters: We use the same Convolutional layers as Table. for the Base CNN in trajectory prediction model. The subsequent feed forward layers reshape the inputs \( X_{in}^{en} \) and \( X_{in}^{de} \) to the model dimension \( D = 52 \). We construct 16 encoder layers, 8 decoder layers, and 4 heads for all Multi-Head Attention blocks. The dropout rates are 0.14 for all blocks. We use the SGD optimizer with learning rate 0.0025 and stop early at convergence.

2) Evaluation Metrics: Given the \( i \)-th unimodal prediction \( \mathcal{Z}^{(i)}_{\text{pred}} = \left\{ \hat{z}^{(i)}(t) = (\hat{x}^{(i)}(t), \hat{y}^{(i)}(t), \hat{\psi}^{(i)}(t)) \right\}_{t=1}^{N_{\text{pred}}} \) corresponding to the ground truth intent, we compute the mean position error \( e_p(t) \) and mean heading error \( e_a(t) \) to the ground truth label \( \left\{ \hat{z}^{(i)}(t) = (x^{(i)}(t), y^{(i)}(t), \psi^{(i)}(t)) \right\}_{t=1}^{N_{\text{pred}}} \) as a function of the time step \( t \):

\[
e_p(t) = \frac{1}{M_D} \sum_{i=1}^{M_D} \left\| \hat{z}^{(i)}(t) - x^{(i)}(t), y^{(i)}(t) - y^{(i)}(t) \right\|_2,
\]

\[
e_a(t) = \frac{1}{M_D} \sum_{i=1}^{M_D} |\hat{\psi}^{(i)}(t) - \psi^{(i)}(t)|.
\]

3) Results: We report the performance of our trajectory prediction model and the results of an ablation study in Fig. 8. For positional displacement error in Fig. 8a, we can see that our model outperform the EKF approach [12] in both short- and long-term. However, by removing the intent information, the model performs almost the same as the EKF, and by removing the image, the performance is severely worsened. The error of heading angle in Fig. 8b has less difference among models but reflects the same trend. The comparison results demonstrate that the model learns a lot of information from image and intent inputs to generate predictions with high accuracy, especially in the long-term.

D. Case Study

Fig. 9 presents our prediction for four representative scenarios in the parking lot. The model generates accurate trajectory predictions when compared to the ground truth, while providing other possible behaviors according to the local context. Video link: https://youtu.be/rj9deuvRXp8

In Fig. 9a the vehicles have already slowed down and started steering towards parking spots. In the left example, it is still hard to tell the actual intention of the vehicle, therefore
In this work, we investigate the problem of multimodal intent and trajectory prediction for vehicles in parking lots. CNN and Transformer networks are leveraged to build the prediction model, which takes the trajectory history and vehicle-centered BEV images as input. The proposed model can be flexibly applied to various types of parking maps with an arbitrary number of intents. The result shows that the model learns to predict the intent with almost 100% accuracy in top-3 candidates, and generates multimodal trajectory rollouts. Furthermore, we release the DLP dataset with its software kit. As the first human driving dataset in parking lot, it can be applied to a wide range of prospective applications.

ACKNOWLEDGMENT

We would like to thank Michelle Pan and Dr. Vijay Govindarajan for their contribution in building DLP dataset.

REFERENCES

[1] S. Lefèvre, D. Vasquez, and C. Laugier, “A survey on motion prediction and risk assessment for intelligent vehicles,” ROBOMECH Journal, vol. 1, no. 1, p. 1, Dec. 2014.
[2] K. Leung, E. Schmerling, M. Zhang, M. Chen, J. Talbot, J. C. Gerdes, and M. Pavone, “On infusing reachability-based safety assurance within planning frameworks for human–robot vehicle interactions,” International Journal of Robotics Research, vol. 39, no. 10-11, pp. 1326–1345, 2020.
[3] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, “TrafficPredict: Trajectory Prediction for Heterogeneous Traffic-Agents,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6120–6127, July 2019.
[4] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human Trajectory Prediction in Crowded Spaces,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 81. IEEE, June 2016, pp. 961–971.
[5] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, and J. Schneider, “Short-term Motion Prediction of Traffic Actors for Autonomous Driving using Deep Convolutional Networks,” Aug. 2018.
[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proceedings of the 31st International Conference on Neural Information Processing Systems, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 6000–6010.
[7] L. L. Li, B. Yang, M. Liang, W. Zeng, M. Ren, S. Segal, and R. Urtasun, “End-to-end contextual perception and prediction with interaction transformer,” IEEE International Conference on Intelligent Robots and Systems, pp. 5784–5791, 2020.
[8] A. Quintanar, D. Fernandez-Llorca, I. Parra, R. Izquierdo, and M. A. Sotelo, “Predicting vehicles trajectories in urban scenarios with transformer networks and augmented information,” IEEE Intelligent Vehicles Symposium, Proceedings, vol. 2021-July, no. Iv, pp. 1051–1056, 2021.
[9] F. Giuliani, I. Hasan, M. Cristani, and F. Galasso, “Transformer networks for trajectory forecasting,” Proceedings - International Conference on Pattern Recognition, pp. 10 335–10 342, 2020.
[10] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, “Car parking occupancy detection using smart camera networks and deep learning,” in Computers and Communication (ISCC), 2016 IEEE Symposium on. IEEE, 2016, pp. 1212–1217.
[11] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, “Drone-based object counting by spatially regularized regional proposal network,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 4145–4153.
[12] X. Shen, I. Batkovic, V. Govindarajan, P. Falcone, T. Darrell, and F. Borrelli, “ParkPredict: Motion and Intent Prediction of Vehicles in Parking Lots,” in 2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, Oct. 2020, pp. 1170–1175.
[13] Z. Sui, Y. Zhou, X. Zhao, A. Chen, and Y. Ni, “Joint Intention and Trajectory Prediction Based on Transformer,” pp. 7082–7088, 2021.