Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 21(4): 346-350, Dec. 2023 Regular paper

Effects of CNN Backbone on Trajectory Prediction Models for Autonomous Vehicle

Seoyoung Lee*, Hyogyeong Park, Yeonhwi You, Sungjung Yong, and Il-Young Moon*, *Member, KIICE*

Department of Computer Science and Engineering, Korea University of Technology and Education, Cheonan 31253, Republic of Korea

Abstract

Trajectory prediction is an essential element for driving autonomous vehicles, and various trajectory prediction models have emerged with the development of deep learning technology. Convolutional neural network (CNN) is the most commonly used neural network architecture for extracting the features of visual images, and the latest models exhibit high performances. This study was conducted to identify an efficient CNN backbone model among the components of deep learning models for trajectory prediction. We changed the existing CNN backbone network of multiple-trajectory prediction models used as feature extractors to various state-of-the-art CNN models. The experiment was conducted using nuScenes, which is a dataset used for the development of autonomous vehicles. The results of each model were compared using frequently used evaluation metrics for trajectory prediction. Analyzing the impact of the backbone can improve the performance of the trajectory prediction task. Investigating the influence of the backbone on multiple deep learning models can be a future challenge.

Index Terms: Autonomous Driving, CNN, Deep Learning, Trajectory Prediction

I. INTRODUCTION

Trajectory prediction has become an increasingly critical task owing to rapid advancements in the research and development of autonomous vehicles. With the development of deep learning technologies, various models are continuously being developed for predicting the paths of autonomous vehicles.

Existing models for path prediction perform future predictions based on an agent's previous displacement or state values, such as in terms of velocity and acceleration. However, recent models are increasing their real-world potential by reflecting complex elements such as interactions with peripheral agents and surrounding situations. Owing to the development of various high-precision sensors such as LiDAR and radars, as well as the continuous emergence of large datasets for autonomous vehicles, models with more sophisticated methods and advanced performances are expected to be developed [1]. Depending on the type of input and output, type of deep learning model, response of the surroundings, and number of trajectories to be predicted, several types of deep learning-based trajectory prediction models have emerged, each model with advantages and disadvantages.

When considering the structure of complex elements and various models, a convolutional neural network (CNN) is a neural network structure commonly used for the feature extraction of image data from surrounding situations. CNN models are known to have superior performances. In addition, CNN have been used as backbone models for deep learning-based path prediction models [2, 3]. This study aimed to find an efficient CNN backbone to improve the performance of a trajectory prediction model.

We address the existing research on trajectory prediction

Received 8 July 2023, **Revised** 23 October 2023, **Accepted** 23 October 2023
***Corresponding Author** Il-Young Moon (E-mail: iymoon@koreatech.ac.kr)
Department of Computer Science and Engineering, Korea University of Technology and Education, Cheonan 31253, Republic of Korea

Open Access https://doi.org/10.56977/jicce.2023.21.4.346 print ISSN: 2234-8255 online ISSN: 2234-8883

methods based on nuScenes, which is a dataset for autonomous vehicle development, and Multiple-Trajectory Prediction (MTP) models, which are a type of deep learning-based multi-mode trajectory prediction models. As shown in Fig. 1, we used a rasterized bird's eye view (BEV) image from the nuScenes dataset as input. We observed performance changes when changing MobileNetV2, which is a backbone model used for feature extraction in the MTP model, to other state-of-the-art performing CNN models:ResNet18, ResNet50, ResNeXt50, and WideResNet50.

II. BACKGROUND

A. MTP Model

An MTP is a model used in deep learning-based autonomous driving. A rasterized BEV image and the current state of the agent vehicle (speed, acceleration, and direction change rate) are used as input. The vehicle coordinates and probabilities of H seconds for multiple M modes are the output [4]. The image input in the form of a rasterized BEV enables us to consider the impact of the interaction between the autonomous vehicle and the surrounding environment; however, it limits the vehicle's cognitive module performance [5]. Fig. 1 shows the shape of the input image. The final loss function used in the model is given by (1):

$$L_{ij}^{MTP} = L_{ij}^{class} + \alpha \sum_{m=1}^{M} I_{m=m*} L(\tau_{ij}, \tilde{\tau}_{imj}), \quad (1)$$

where α is a hyperparameter that balances two losses, and I_p is a binary index function that returns a value of zero or one, depending on condition p. The single-mode loss function L is given by (2). This is the average displacement error of an autonomous vehicle; τ_{ij} denotes an actual trajectory, and $\tilde{\tau}_{imj}$ denotes a trajectory predicted in the *m*th mode.

$$L(\tau_{ij}, \tilde{\tau}_{imj}) = \frac{1}{H} \sum_{h=1}^{H} \left\| \tau_{ij}^{h} - \tilde{\tau}_{imj}^{h} \right\|_2 \quad (2)$$

In addition, L_{ij}^{class} is a classification cross-entropy loss function, as given by (3).

$$L_{ij}^{class} = -\sum_{m=1}^{M} I_{m=m} * \log p_{im} \quad (3)$$

In addition, the model trains the CNN parameter θ to minimize the loss over the training data, as shown in (4).

$$\theta^* = \underset{\theta}{argmin} \sum_{j=1}^{T} \sum_{i=1}^{N_j} L_{ij}^{\circ} \quad (4)$$

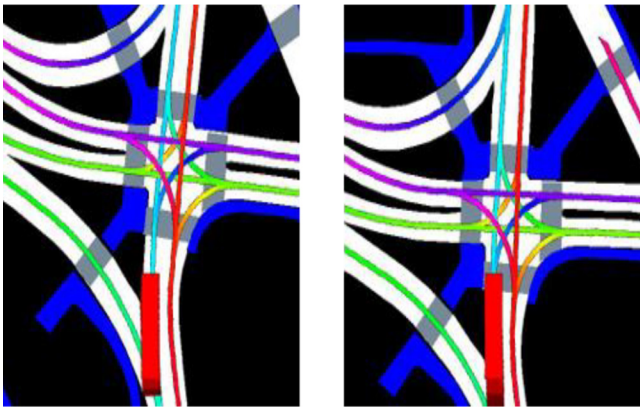

Fig. 1. Rasterized bird's eye view input image.

B. Networks

The backbone models used in this study are as follows.

1) ResNet

ResNet can employ up to 152 layers networks by learning a residual representation function. It uses skip or shortcut connections to pass the input from a previous layer to the next, allowing deep neural networks to be constructed [6].

2) ResNeXt

ResNeXt is a model that introduces the concept of cardinality used in inception to the ResNet architecture. It is characterized by the addition of a split-transform-merge, which splits the convolution operations, obtains different weights, and then merges them [7].

3) WideResNet

This model increases the area and depth of ResNet. The dimming feature-reuse problem was solved by increasing the areas of the two types of residual blocks and applying a dropout between the convolutional layers within the residual block. This significantly increases the learning speed and performance of the 16-layer model [8,9].

4) MobileNetV2

MobileNetV2 was designed for use in environments with less computational power than personal computers, such as mobile and embedded devices. MobileNetV2 is recognized for its light weight, low number of parameters, and high accuracy. A modified depthwise separable convolution, a concept introduced in MobileNetV1, is also used. The main feature of MobileNetV2 is the inverted residual block concept, which is differs from the residual structure used in ResNet [10].

III. METHODS

A. Model and Experiments

In this study, trajectory prediction was performed using deep learning-based models that perform feature extraction based on a CNN backbone.

Fig. 2 shows the overall structure of the corresponding model for the trajectory prediction. The model receives the rasterized image, speed of the agent, and state of the direction change rate as input. The feature values extracted from the corresponding image through the CNN-based model are combined with the state vector values to pass through the neural network and then predict the trajectory information (x and y coordinates) and probabilities of autonomous vehicles for future N seconds and M modes. The results were compared with those of structures employing different backbone models.

We designed the experiments for predicting the trajectory of two modes (M=2) for the next 6 s (N=6). As the image data of the nuScenes dataset were configured at 2 Hz, the output consisted of 50 values. In addition, the state vector, given as another input, includes the velocity, acceleration, and heading change rates of the agent.

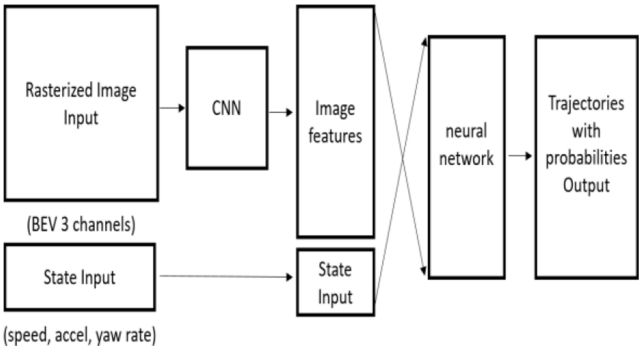

Fig. 2. Overall trajectory prediction process used in MTP.

The model was implemented in PyTorch using the nuScenes software development kit [11]. For the MTP model, we set the hyperparameter regression loss weight to 1 and angle threshold to 5. In addition, we set the epoch to 10 and batch size to 4. We trained the model with the Adam optimizer and set the learning rate to 0.0001 [12].

B. Dataset

The Motional team developed the nuScenes dataset as a large-scale public dataset for autonomous driving applications. This dataset was collected using six cameras in addition to one LiDAR, five radar, GPS, and IMU (Inertial Measurement Unit) sensors. Scenes of 20 s were manually selected to show a diverse and interesting set of driving maneuvers, traffic situations, and unexpected behaviors [13].

We used the nuScenes dataset for the model training. We split the dataset into a mini-training set and a mini-validation set. Because nuScenes does not provide public annotations for a test set, we used a subset of the validation set as a mini-test set. The mini-training, mini-validation, and mini-test sets had a scene ratio of 8:2:2 and contained 742, 61, and 71 observations, respectively.

C. Metrics

To evaluate and compare the performances, we used the following metrics. These are often used in trajectory prediction tasks.

1) $minADE_K$

The minimum average displacement error (ADE) is the average of the L2 distances at the points between the predicted trajectory and measured data. This is the calculated value of the k most likely predictions, which is generally unacceptable.

2) $minFDE_K$

The minimum final displacement error (FDE) is the L2 distance between the final point of the prediction and the reference point. This implies that the minimum FDE is obtained for the most likely prediction k and average value for all agents.

3) $MissRate@2m_K$

The miss rate at 2 m over k defines a prediction as a failure if the maximum point-by-point L2 distance between the prediction and survey data is greater than 2 m. In addition, for each agent, the k most likely predictions are used to evaluate whether they are real. This is the ratio of the measurements of all agents.

D. Experimental Environment

Table 1 details the experimental environment used in this study.

Table 1. Experimental environment

OS	CPU	RAM	GPU
Windows 10	Intel ® Core™ i7-7700	32.0 GB	NVIDIA GeForce GTX 1060 6GB

IV. RESULTS

Table 2 lists the indicators of the prediction results for different backbones that the CNN model used for feature extraction.

Table 2. Comparison of trajectory prediction performance indicators for different backbone networks

Backbone Network	$minFDE_1$	$minFDE_5$	$minFDE_{10}$	$minADE_1$	$minADE_5$	$minADE_{10}$	Miss Rate @ 2m (K=1)	Miss Rate @ 2m (K=5)	Miss Rate @ 2m (K=10)	OffRoad-Rate	Time (s)
ResNet18	11.9768	11.1553	11.1553	5.9370	5.6793	5.6793	0.9577	0.9577	0.9577	0.0423	99.92
ResNet50	13.3336	11.0664	11.0664	7.5461	6.5413	6.5413	0.9859	0.9859	0.9859	0.0	106.71
ResNeXt50	16.2913	15.4996	15.4996	9.0213	8.7106	8.7106	1.0	1.0	1.0	0.0	104.15
WideResNet50	15.3408	11.9301	11.9301	8.2955	6.7309	6.7309	0.9437	0.9437	0.9437	0.0070	104.55
MobileNetV2	10.1748	7.6100	7.6100	5.2987	4.1070	4.1070	1.0	1.0	1.0	0.0211	100.49

A. Quantitative results

The $minFDE_1$ metric exhibited the best results for MobileNetV2 (10.1748), followed by ResNet18 and ResNet50 with values of 11.9768 and 13.3336, respectively. The model that achieved the best values for $minFDE_5$ and $minFDE_{10}$ was MobileNetV2, with a value of 7.6100. ResNet50 and ResNet18 followed MobileNetV2 with values of 11.0664 and 11.1553, respectively. The other models exhibited the same values for $minFDE_5$ and $minFDE_{10}$. For the $minFDE_K$ metrics, MobileNetV2 exhibited the best performance at 5.2987, followed by ResNet18 and ResNet50 at 5.6793 and 6.5413, respectively.

ResNeXt50 exhibited the worst performance in terms of the $minFDE_K$ and $minADE_K$. For $minFDE_5$, ResNeXt50 achieved 15.4996, which is 2.04 times the value of MobileNetV2 and 1.39 times that of ResNet18 (11.1553). WideResNet50 exhibited the worst performance after ResNext50 for both $minFDE_K$ and $minADE_K$. In terms of the $minFDE_5$, it scored 1.57 times that of MobileNetV2, and for $minADE_5$, it scored 1.64 times that of MobileNetV2.

For the Miss Rate @ 2m and OffRoadRate metrics, almost all results were derived similar for all models. Therefore, a meaningful analysis was difficult to conduct because the differences in the results were relatively small.

The time measure refers to the time that the model required to perform the trajectory prediction. The execution time was measured through ten experiments for each model, and Table II lists the average execution times. ResNet18 and MobileNetV2 exhibited relatively short execution times. The execution times of the remaining three models were similarly measured.

V. DISCUSSION AND CONCLUSIONS

This study aimed to determine an efficient CNN structure among the feature extractors used in deep learning models for multimode trajectory prediction in autonomous vehicles. To this end, we changed MobileNetV2, which has been used as the primary backbone model of the MTP model and is a traditional deep learning-based multimode trajectory prediction model, to ResNet18, ResNet50, ResNeXt50, and WideResNet50, which exhibit state-of-the-art performances. For the experiments, we used the nuScenes dataset, which is an autonomous vehicle dataset. We compared the results in terms of metrics commonly used in trajectory prediction tasks.

We used nuScenes dataset when training and testing the deep learning models for real trajectory prediction tasks. However, while traditional state of the art models require training and validation of the model with large amounts of data for a fair comparison, we used mini-data to conduct our experiments. This should be considered when discussing the results. Nonetheless, analyzing the impact of the CNN backbone can be useful for improving the performance of trajectory prediction tasks. In addition, because the trajectory prediction of autonomous vehicles is accomplished in real time, the temporal aspect of the model is an important factor. This measure does not seem to have been reflected well thus far. In the future, we plan to conduct experiments on models other than MTP to make more general and meaningful observations on the effects of the backbone, and we leave the temporal aspects for future studies.

ACKNOWLEDGEMENTS

This research was supported by the Basic Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (No. 2021R1I1A3057 800), and the results were supported by the Regional Innovation Strategy (RIS) through the NRF funded by the Ministry of Education (MOE) (2021RIS-004).

REFERENCES

[1] B. D. Kim, S. H. Park, S. H. Lee, E. Khoshimjonov, D. S. Kum, J. S. Kim, J. S. Kim, and J. W. Choi, "LaPred: Lane-Aware Prediction of Multi-Modal Future Trajectories of Dynamic Agents," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville: TN, pp. 14631-14640, 2021. DOI: 10.1109/CVPR46437.2021.01440.

[2] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction," *Proceedings of the Conference on Robot Learning*, vol. 100, pp. 86-99, 2020. DOI: 10.48550/arXiv.1910.05449.

[3] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "CoverNet: Multimodal Behavior Prediction Using Trajectory Sets," in *2020 IEEE/CVF Conference on Computer Vision and*

Pattern Recognition (CVPR), Seattle: WA, pp. 14062-14071, 2020. DOI: 10.1109/CVPR42600.2020.01408.

[4] H. Cui, Vl. Radosavljevic, F-C. Chou, T-H. Lin, T. Nguyen, T-K. Huang, J. Schneider, and N. Djuric, "Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal: QC, pp. 2090-2096, 2019. DOI: 10.1109/ICRA.2019.8793868.

[5] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33-47, Jan. 2022. DOI: 10.1109/TITS.2020.3012034.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, Dec. 2015. DOI: 10.48550/arXiv.1512.03385.

[7] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," *arXiv preprint arXiv:1611.05431*, Nov. 2016. DOI: 10.48550/arXiv.1611.05431.

[8] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," *arXiv preprint arXiv:1605.07146*, May 2016. DOI: 10.48550/arXiv.1605.07146.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, Dec. 2015. DOI: 10.48550/arXiv.1512.03385.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmognimov, and L-C. Chen, "MobileNetV2: Inverted Residual and Linear Bottlenecks," *arXiv preprint arXiv:1801.04381*, Jan. 2018. DOI: 10.48550/arXiv.1801.04381.

[11] nuScenes Contributors, nuScenes [Internet], Available: https://www.nuscenes.org/.

[12] D. P. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, Dec. 2014. DOI: 10.48550/arXiv.1412.6980.

[13] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, Mar. 2019. DOI: 10.48550/arXiv.1903.11027.

Seoyoung Lee

received her B.S. degree in computer science and engineering in 2022 from Korea University of Technology and Education, Cheonan, Republic of Korea. She is currently pursuing a M.S. degree from the Department of Computer Science and Engineering at Korea University of Technology and Education. Her current research interests include artificial intelligence, web services, and computer vision.

Hyogyeong Park

received her B.S. degree in computer science and engineering in 2021 from Korea University of Technology and Education, Cheonan, Republic of Korea. She is currently pursuing a M.S. degree from the Department of Computer Science and Engineering at Korea University of Technology and Education. Her current research interests include artificial intelligence, web services, big data, and recommendation systems.

Yeonhwi You

received his B.S. degree in computer science and engineering in 2022 from Korea University of Technology and Education, Cheonan, Republic of Korea. He is currently pursuing a M.S. degree from the Department of Computer Science and Engineering at Korea University of Technology and Education. His current research interests include artificial intelligence, big data, and recommendation systems.

Sungjung Yong

received his M.S. degree in computer science and engineering in 2020 from Korea University of Technology and Education, Cheonan, Republic of Korea. He is currently pursuing a Ph.D. from the Department of Computer Science and Engineering at Korea University of Technology and Education. His current research interests include artificial intelligence, web services, and recommendation systems.

Il-Young Moon

has been a professor at the Department of Computer Science and Engineering, Korea University of Technology and Education, Cheonan, Republic of Korea since 2005. In 2005, he received his Ph.D. from the Department of Aeronautical Communication and Information Engineering, Korea Aerospace University. His current research interests include artificial intelligence, wireless internet applications, wireless internet, and mobile IP.