Advancing Autonomous Driving through Direct Perception and Attention

2 Team Members

Note: There is an inconsistency between the number of team members stated in the proposal (3) and in this report (2) , as one member dropped the class.

Abstract

This project explores the use of self-attention mechanisms to improve depth perception models for autonomous driving systems. The proposed model was trained on an image dataset to estimate the distance of the closest vehicle. Results show improved performance compared to traditional direct perception methods, but limitations were observed in cases where no car was present in the image or when predicting coordinates for vehicles within the region. Future work includes addressing these limitations and using more training data to improve performance. Overall, the project demonstrates the potential of self-attention mechanisms in improving depth perception for autonomous driving.

Keywords: direct perception, depth perception, self-attention, distance estimation, convolutional neural network (CNN), mean absolute error (MAE), autonomous driving, machine learning.

1 Introduction

Autonomous driving is an area of research that has gained significant traction over the past few years, and one of its fundamental requirements is the ability to accurately estimate distances in real-time. While existing paradigms, such as mediated perception and behavior reflex approaches, have their respective limitations, recent research has shown that direct perception methodologies offer a promising alternative [1][2][3].

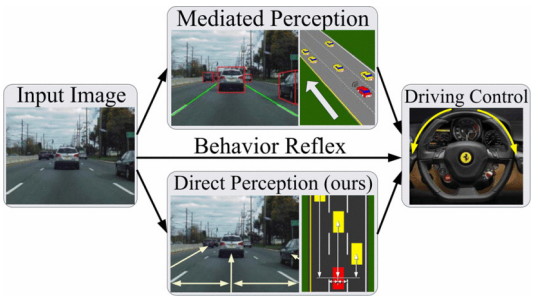

Figure 1: Three paradigms for autonomous driving [1]

The three paradigms for autonomous driving are different ways of making a car move on its own without a human driver. Mediated perception uses sensors to detect the environment around the car

and then uses a computer to make decisions about how the car should move. Behavior reflex works by programming the car to react to certain situations in a specific way, without needing to understand the environment. Finally, direct perception, which is the focus of this project, works by training a computer program to directly understand the environment around the car and make decisions based on that understanding.

In this work, we propose to extend the direct perception methodology by incorporating self-attention, with the aim of improving distance estimation in autonomous driving. The objective of this project is to develop a model capable of estimating the distance of the closest vehicle in an image, which is essential for autonomous driving systems. Specifically, given an input image I, our model aims to estimate the distance D of the closest vehicle, where $D \in [0, 50]$ meters. We aim to minimize the mean absolute error (MAE) between the estimated distance and the ground truth distance. Our proposed approach builds on the direct perception methodology proposed by Chen *et al.* [1] and extends it by incorporating self-attention to enhance the model's performance in detecting close obstacles or abruptly moving nearby vehicles, thereby offering a more robust and efficient solution for distance estimation in autonomous driving.

In this paper, we will provide an overview of the direct perception methodology and its limitations, followed by a detailed description of our proposed approach, including the incorporation of self-attention into the network architecture. We will also describe the data pre-processing and cleaning steps, as well as the training and evaluation procedures. Finally, we will present our results and discuss the strengths and weaknesses of our approach.

2 Problem Statement

2.1 Conceptual Problem Statement

The task of estimating the distance of the closest vehicle in an image is a critical component of autonomous driving systems. Without accurate distance estimation, autonomous vehicles may fail to take appropriate actions in response to other vehicles on the road, leading to safety hazards. The goal of this project is to develop a model that can accurately estimate the distance of the closest vehicle in an image, which will enable autonomous vehicles to make informed decisions about their actions on the road.

2.2 Mathematical Problem Statement

Given an input image I, our model aims to estimate the distance D of the closest vehicle, where $D \in [0, 50]$ meters. We aim to train a model $f(I; \theta)$, parameterized by θ, that can estimate the distance D from the input image I such that the MAE between the estimated distance and the ground truth distance is minimized. More formally, we aim to minimize the following loss function $L(\theta)$:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} |f(I_i; \theta) - D_i| \tag{1}$$

where n is the number of samples in the training set, and I_i and D_i denote the i-th input image and the ground truth distance of the closest vehicle in the image, respectively.

3 Related Work

In this section, we review existing approaches to vision-based autonomous driving: mediated perception and behavior reflex approaches. Mediated perception involves parsing entire driving scenes, constructing a high-dimensional world representation, and relying on the accuracy and field-of-view of sensors such as radar.

The behavior reflex approach uses blind mapping of images to control commands, but struggles to handle rare events and complicated driving conditions. Chen *et al.* [1] proposed a direct perception methodology, which learns/estimates affordances and maps input images to a limited set of essential perception indicators linked to road conditions. The paper by Weinzaepfel *et al.* [4] demonstrates

how ConvNet works well on the KITTI Dataset for driving images, which is the dataset we used in the project.

Another approach in vision-based autonomous driving is the End-to-End deep learning approach, which aims to learn robust perception-action models from diverse training data. Xu *et al.* proposed a generic vehicle motion model learned from large-scale crowd-sourced video data [5]. They presented an end-to-end trainable architecture that predicts a distribution over future vehicle egomotion from instantaneous monocular camera observations and previous vehicle state. Their model incorporates a novel FCN-LSTM architecture that can be learned from large-scale crowd-sourced vehicle action data and takes advantage of scene segmentation side tasks to improve performance in a privileged learning paradigm.

In addition, there has been a growing need for machine learning models that can run on mobile devices with limited computational power [6] to enable further applications in this field.

4 Method

We implemented a CNN-based model for estimating the distance of the closest vehicle in an image, which is essential for autonomous driving systems. The model uses the AlexNet architecture and incorporates a self-attention mechanism to capture important relationships between different parts of the input image.

The KITTI Dataset is used for training and testing, and the model is evaluated using the MAE metric. The goal is to minimize the error between the estimated distance and the ground truth distance, which is limited to the range of 0-50 meters.

The model is preprocessed by resizing and normalizing the input images, and the output is six distance estimates. The code is designed to be run with a script that loads a trained model checkpoint and uses it to estimate distances for a given set of images.

4.1 Dataset

In this project, we are using KITTI dataset [7] as our training data. A Volkswagen Passat mounted with various sensors was used to obtain information from rides that took place in rural areas and highways in Europe. Grayscale, color images from cameras and point clouds from a lidar can be found in the dataset. We used only the color images and the officially labeled tracklets, which is an XML file of the coordinates of all objects in the images, for the purpose of this project. Figure 2 below is an example of one frame from a video.

Figure 2: One video frame from KITTI dataset [7]

In this project, the input will be multiple images from different videos. The height and width of these images are 375×1242.

4.2 Preprocessing

Prior to using the images as input for the CNN model, certain preprocessing steps must be taken. This involves resizing the images to a standardized size of 150×497 and then normalizing them. This not only saves training time but also ensures that important features in the image are retained while making the model more robust to outlier and noise in the data. The normalization process helps to scale the pixel values to a range of 0-1, which is a common requirement for deep learning models.

4.3 Model

AlexNet is a type of neural network architecture that is commonly used for image recognition tasks. In this project, we are using the standard AlexNet architecture [3], the structure is shown in Figure 3 below.

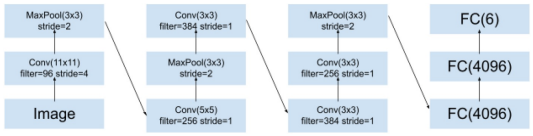

Figure 3: AlexNet architecture

AlexNet consists of 5 convolutional layers followed by 3 fully connected layers. The first convolutional layer takes an input of size 3 (representing RGB channels) and produces 96 feature maps, with a kernel size of 11×11, stride of 4, and padding of 1. The next four convolutional layers have 256, 384, 384, and 256 feature maps respectively, with varying kernel sizes, stride, and padding. The network also has three max-pooling layers to reduce the spatial size of the feature maps, and three fully connected layers for classification. The output of the network has six nodes, representing the predicted distances to the closest vehicle in the input image.

Despite there were comments on our proposal pointing out AlexNet no longer being the state-of-the-art model, we decided to stick with this model as our goal is to reproduce the result of the paper. We initialized the parameters of the convolution layers randomly using the Gaussian distribution and used Relu as the activation function. The authors used stochastic gradient decent to optimize the loss function when performing backward pass.

4.4 Output

This model has six outputs, which represents three sets of coordinates. The model splits each image into three sections and determines the distance between the closest vehicle for each section. The first section is the left area, with x coordinates in the region $[-12, -1.6]$. The second section is the central area, with x coordinates in the region $(-1.6, 1.6)$. Finally the third section is the right area, with x coordinates in the region $[1.6, 12]$. Figure 4 below illustrates the setup.

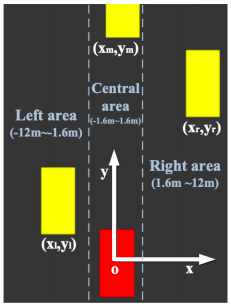

Figure 4: Output configuration

The first output represents the x position of the closest vehicle in the left area of the image. The second output represents the y position of the closest vehicle in the left area of the image and so on. If there is no vehicle in the left area, we label the result as $(-50, 50)$ which means the vehicle is far away. If there is no vehicle in the central area, we label it as $(0, 50)$, meaning the vehicle is ahead of us and far away. If there is no vehicle in the right area, we label the result as $(50, 50)$.

With this setup in mind, the loss function we used in this model is the MAE loss. The formula is given as follows:

$$L = \sum_{i=1}^{6} |\hat{y}_i - y_i| \tag{2}$$

Where $\hat{y_i}$ is the predicted output and y_i is the label or ground truth.

4.5 Self-Attention

To improve the accuracy of our depth perception model, we incorporated a self-attention mechanism into the network architecture. The self-attention module aims to capture important relationships between different parts of an image, and it does this by calculating a query, key, and value for each pixel in the image.

In our implementation, the self-attention module contains three convolutional layers: a query layer, a key layer, and a value layer. These layers produce tensors with the same spatial dimensions as the input tensor, but with a reduced number of channels. The query tensor and key tensor are then used to compute a measure of similarity between all pairs of pixels.

We take the dot product between the query tensor and the transpose of the key tensor, and then normalize the result with a softmax function to obtain attention weights. These weights are then used to weight the value tensor, producing a weighted sum of the values that highlights the most relevant pixels in the image.

Finally, we add the input tensor to the weighted sum of values to obtain the output of the self-attention module, which helps preserve important information from the original image. By incorporating self-attention, we can better focus on relevant regions in the input image, providing more accurate and efficient distance estimations under the same number of training iterations.

5 Results and Discussions

This section presents the evaluation and performance of the proposed model for distance estimation in autonomous driving. The results indicate that the model performs reasonably well but has limitations that need to be addressed in future work.

5.1 Evaluation

The MAE metric was chosen as it is commonly used to measure the performance of distance estimation models in autonomous driving. We calculated the absolute difference between the predicted and ground truth distances for each input image, and then took the mean of these differences to obtain the MAE. Since our model predicts three pairs of coordinates, we separately monitored the average MAE for the x and y axes.

To compare the performance of our model with that of the model proposed by Chen *et al.* [1], we used their reported results as a baseline. We aimed to achieve similar or better accuracy in our model compared to theirs. It is important to note that the evaluation of our model was done on a separate test set of images that were not used during training to avoid overfitting.

In addition to the MAE metric, we also visually inspected the model's predictions on a set of sample images to ensure that the model was correctly estimating the distances. We analyzed the predicted coordinates of the closest vehicle and compared them with the ground truth coordinates. We also examined cases where the model made significant errors to identify any patterns or limitations in the model's performance.

5.2 Results

Figure 5 and 6 show the MAE for the x and y axes of the model in meters. We observe that the training loss is consistently decreasing over epochs, indicating that the model is effectively learning from the data. This suggests that the model is capable of capturing the essential features of the image that are necessary for accurate distance estimation.

The MAE for the x axis of the test data is 5.654. The MAE loss for the y axis of the test data is 7.645. While the model proposed by Chen *et al.* [1] has the MAE of 1.565 and 5.832 for the x and y axes (in meters). The DPM car detector mentioned in the paper has even better performance, as depicted in Table 1. As an example, Figure 7 is a sample image from the test dataset.

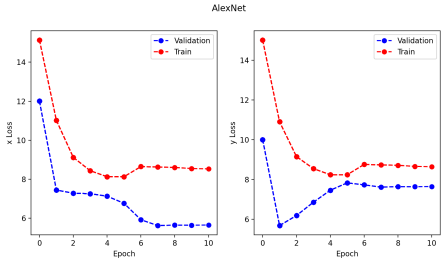

Figure 5: AlexNet training loss

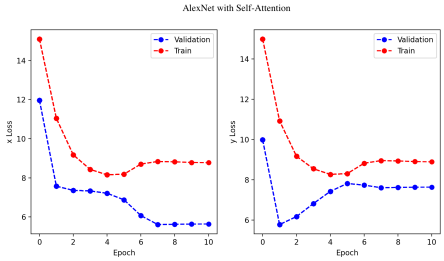

Figure 6: AlexNet with self-attention training loss

The predicted output is: $(-43.9, 46.4)$, $(-0.02, 44.6)$, and $(11.1, 20.2)$, while the true label for this image is: $(-50.0, 50.0)$, $(0.0, 50.0)$, and $(3.8, 13.3)$. The error between the predicted output and the true label matches the error in Table 1.

In this case, for the left section of the image, the model estimates a vehicle exist around (-43.9 46.4). The truth is there is no vehicle at the left section (The vehicle is not fully in view so it is not treated as in the left section). Despite the inconsistency, both values suggest the vehicle is still far away from the camera.

For the central section of the image, $(-0.02, 44.6)$ was predicted, with $(0.0, 50.0)$ as the ground truth. This implies that no vehicle is close to the camera at the section.

For the right section of the image, $(11.1, 20.2)$ was predicted while the true label was $(3.8, 13.3)$, there is some difference between the model output and the ground truth, but the model is able to tell if there is a vehicle in a certain section and whether the vehicle is far or not from the camera.

Figure 7: Sample of test data

Overall, the results suggest that the proposed model performs reasonably well but is not the best performing model. Further research and improvements could be made to improve its performance, especially in comparison to other state-of-the-art models.

6 Conclusions

6.1 Strengths and Contributions

Our project demonstrated the feasibility of training models for depth perception using images or videos as input, which has significant applications in autonomous driving systems. Moreover, the authors anticipated the incorporation of self-attention mechanism will allow the model to focus on

Model	x **(m)**	y **(m)**
Ours (AlexNet)	5.654	7.645
Ours (w/ self-attention)	5.643	7.633
Chen	1.565	5.832
DPM	1.502	5.824

Table 1: Comparison of MAE

relevant regions of the input image, resulting in more accurate and efficient distance estimations within the same number of training iterations. This did not come into effect as a consequence in this project.

6.2 Limitations

However, the current model has certain limitations that need to be addressed in future work. For instance, the model does not perform well when there is no car present in the image. As demonstrated in Figure 6, the model may predict the presence of a vehicle even when there is none in the middle section of the image. Ideally, the model should be trained to predict (0, 50) if there is no vehicle in the center region.

Additionally, the model's performance for predicting the coordinates of a vehicle within the region is not as good as the models mentioned in the literature. One reason for this might be the limited amount of training data used due to memory constraints in the Great Lakes Clusters. Future work can explore using more training data and increasing memory resources to improve the model's performance.

6.3 Conclusion and Future Directions

In conclusion, our project demonstrated the potential of using self-attention mechanism in direct perception for distance estimation in autonomous driving. While the current model has certain limitations, our work provides insights for future research directions to improve the performance of the model. Some potential future works for this project could include:

1. Fine-tuning the model: In this project, we trained the model on a limited amount of data due to memory constraints. Fine-tuning the model with a larger dataset could potentially improve its performance.
2. Exploring different architectures: While our model utilized the self-attention mechanism, there are many other architectures that could be explored to improve the accuracy and efficiency of distance estimation.
3. Testing on different datasets: In this project, we only used the KITTI dataset. Testing the model on other datasets could provide more insights into its strengths and weaknesses, as well as its generalizability to different driving scenarios.
4. Improving the model's performance in the absence of vehicles: As mentioned in section 6.2, the current model does not perform well when there are no vehicles in the image. Developing a more robust solution to this problem could improve the overall performance of the model in real-world applications.
5. Developing a real-time implementation: In this project, we focused on training the model and evaluating its performance offline. Developing a real-time implementation that can be integrated into an autonomous driving system would be a valuable next step.

7 Contribution by Group Members

Author 1 proposed the main idea, coded the CNN, and conducted the training and testing. Author 2 handled the pre-processing of the data. Both Author 1 and Author 2 contributed to the finalization of the codes, interpretation of the results, and preparation of this report. Author 3 dropped the class one week after the proposal submission.

References

[1] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. *IEEE International Conference on Computer Vision.*, 2015.

[2] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *International Conference on Computer Vision (ICCV)*, 2017.

[4] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. *IEEE International Conference on Computer Vision.*, 2013.

[5] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.