2D Semantic Segmentation in Urban Scenes

11911612 Haoyu Wang
12012524 Zhangjie Chen
12011923 Xudong Zhang
12011231 Xuyuan Li

Dec 2023

Introduction

Background

Autonomous driving, also known as self-driving or driverless technology, represents a revolutionary advancement in the automotive industry. It involves vehicles capable of navigating and operating without human intervention. Autonomous vehicles leverage a combination of sensors, cameras, radar and artificial intelligence to perceive their environment and make real-time decisions, enabling them to navigate roads independently. The development of autonomous vehicles is based on key elements, including environmental perception, data processing, path planning, and control systems. These elements work cohesively to ensure a safe and comfortable autonomous driving experience.

Environmental perception is a pivotal function in autonomous driving, focusing on vehicles' ability to interpret and understand its surrounding. However, the uncertainty in environmental states creates challenges for autonomous driving systems. Factors like unpredictable road conditions and dynamic scenarios require advanced perception technologies to adapt and make accurate decisions.

In order to meet the challenges, we apply semantic segmentation for environmental perception. Semantic segmentation is a perception method that categorizes pixels in an image into different semantic classes, such as road, pedestrians, vehicles, and obstacles, enabling the vehicle to understand its environment. The primary purpose of semantic segmentation in autonomous driving is to provide a detailed understanding of the surrounding environment, aiding decision-making processes for self-driving vehicles.

Motivation

Semantic segmentation is a crucial process for the environmental perception of autonomous driving, involving associating each pixel of an image with predefined class. In this task, we will apply U-net model to complete semantic segmentation.

U-Net is a popular architecture for semantic segmentation. The down/up-sampling technique became an important design idea and later was adopted by AIGC as well. Although initially designed for biomedical applications, its powerful performance goes beyond. U-net can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures [1]. To effectively leverage label information, we will use the fully-supervised batch contrastive approach, which will pull clusters of points belonging to the same class together in embedding classes while pushing apart clusters of samples from different classes [2].

Meanwhile, in order to improve the performance under the situation without any human labels, an approach, named Cut-and-LEaRn (CutLER) will be used for training unsupervised object detection and segmentation models. CutLER first generate coarse masks for multiple objects in an image and then learns a detector on theses masks using robust loss function [3].

Related Work

U-Net is one of the early algorithms that utilizes fully convolutional networks for semantic segmentation. It aims to identify cell boundaries in images, and its innovation lies in the symmetrical U-shaped structure, comprising a compressive path and an expansive path. The nomenclature of the network is derived from the U-shaped configuration it exhibits. The compressive and expansive paths conduct max-pooling downsampling and transposed convolution operations, respectively, culminating in a two-dimensional segmentation map, as the task at hand pertains to binary classification. The inventive design of U-Net had a discernible impact on the subsequent development of several segmentation networks.

Due to the substantial memory consumption of model weights, particularly when employing a very large batch size, a significant portion of GPU memory is allocated for this purpose, resulting in considerable GPU memory wastage. Consequently, the U-Net team opted for larger individual images with a batch size set to 1. However, this choice introduces a challenge where gradient estimation becomes highly dependent on a single image, leading to increased noise. To address this issue, a high momentum value is employed, ensuring that early training samples exert a considerable influence on the gradient descent process.

The experiments conducted with U-Net utilized a relatively straightforward ISBI cell tracking dataset. Given the inherently uncomplicated nature of the task, U-Net achieved remarkably low error rates by training on only 30 images, complemented by the implementation of data augmentation strategies.

Improvements

Architecture: Addressing Rigidity

The architecture proposed by U-Net, with its distinctive U-shaped structure, poses a challenge when attempting modifications. Exploring alternative architectures that can accommodate variations in image features and structures while maintaining the essence of the original design is a noteworthy avenue for improvement.

Optimizer: Navigating Resource Constraints

Considering the computational resources required during training, there exists a delicate trade-off between the optimization algorithm's sophistication and the available hardware capabilities. Investigating optimization techniques that strike a balance between efficiency and performance gains could potentially enhance the overall training process.

Loss Function: Enhancing Boundary Recognition

One critical aspect is the challenge of accurately recognizing "border" regions in semantic segmentation tasks. Exploring and refining loss functions tailored to improve the model's ability to precisely identify and delineate object boundaries is crucial for achieving more accurate and fine-grained segmentation results.

Data Preprocessing: Elevating Robustness through Augmentation

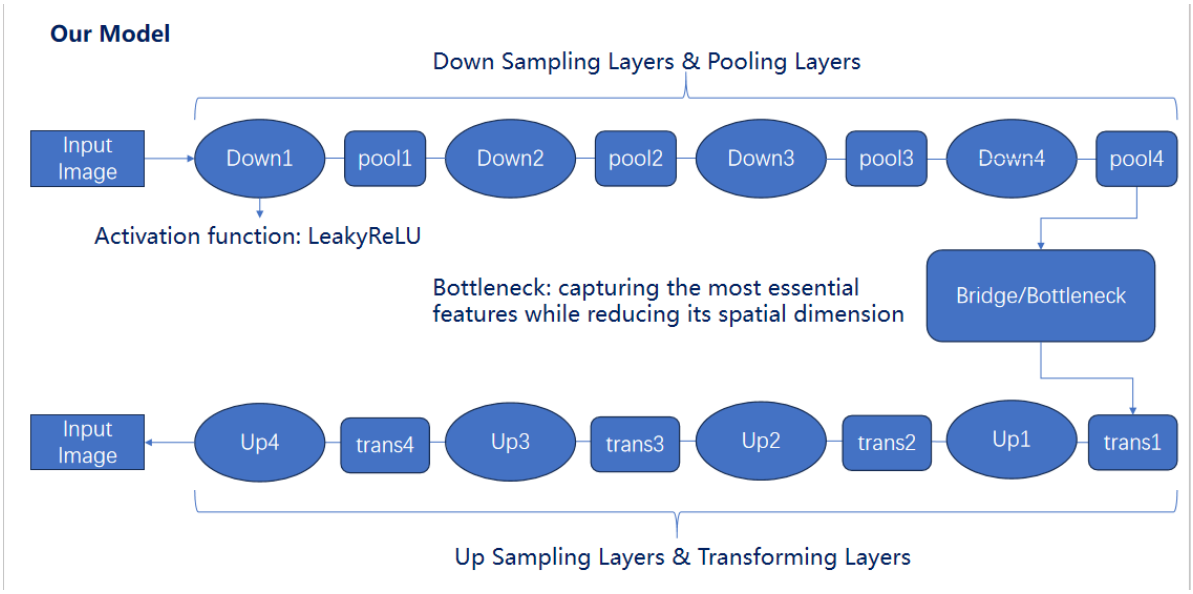
To further enhance the robustness of the model, augmenting the dataset through diverse preprocessing techniques becomes pivotal. Experimenting with advanced data augmentation strategies, such as geometric transformations, color variations, or introducing synthetic data, can contribute to training a more resilient model capable of handling a broader range of real-world scenarios and variations in input data.

Proposed Method

Extending U-Net for Multi-Class Classification

In adapting U-Net for multi-class classification, a fundamental modification involves extending its architecture to accommodate the intricacies of multiple classes rather than the binary segmentation it was initially designed for (background and foreground).

For the transition to multi-class segmentation, a pivotal adjustment is made in the output layer, necessitating an increase in the number of channels. Specifically, each class requires a dedicated channel for prediction. For example, in a scenario with 5 classes, the output layer is expanded to incorporate 5 channels, with each channel dedicated to predicting the presence and characteristics of a specific class. This expansion enables the model to provide nuanced and class-specific segmentation outputs, allowing it to discern and delineate between different objects or entities within the input data. This modification empowers U-Net to extend its utility beyond binary segmentation tasks and lends itself effectively to the demands of multi-class classification challenges.



Experiment Resources & Platform

We conducted the experiment based on the server with following info provided by course:

Tool	Detail/Version
GPU	NVIDIA GeForce RTX 2080 Ti
CPU	Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz
CUDA	12.2
python	3.7
pytorch+torchvision	1.13.1/0.14.1
tensorboardx	2.2

Initial Results

After data processing, we applied a U-net model to manage the street scene semantic segmentation task. Here we chose 'train' folder of Cityscapes dataset to be the source domain. Meanwhile, 'test' folder of Cityscapes dataset is used as target domain to evaluate the model. Our basic experiment evaluated the model.

We trained our model with the following parameters:

Parameters	Explanation	Values
batch size per GPU	Train batch size(Modified according to different GPU)	4

Parameters	Explanation	Values
epochs	Number of epochs during training	100
lr	Learning rate	0.0002
img_size	Size of image(resized) in training and evaluating	128*256

We observed the descending loss by epoch:

Below is an example of semantic segmentation result:

We evaluated the model in two ways: one is pixel-level accuracy, and the other is IoU (Intersection over Union) for each class.

Pixel-Level Accuracy: *0.8443*

IoU result of different labels:

ID	IoU	Object Type
0	N/A	unlabeled
1	**0.8170**	**ego vehicle**
2	0.5637	rectification border
3	0.0193	out of roi
4	0.0851	static
5	0.0547	dynamic
6	0.0610	ground
7	**0.8951**	**road**
8	0.5937	sidewalk
9	0.2419	parking
10	N/A	rail track
11	**0.7786**	**building**
12	0.2226	wall
13	0.1992	fence
14	N/A	guard rail
15	0.2725	bridge
16	0.0000	tunnel
17	0.3443	pole
18	0.0000	polegroup
19	0.2746	traffic light

ID	IoU	Object Type
20	0.3951	traffic sign
21	**0.8210**	**vegetation**
22	0.4440	terrain
23	**0.8090**	**sky**
24	**0.4578**	**person**
25	0.0114	rider
26	**0.8112**	**car**
27	0.1822	truck
28	0.3999	bus
29	N/A	caravan
30	N/A	trailer
31	0.0055	train
32	0.0643	motorcycle
33	0.4022	bicycle

In summary, our model has a high accuracy, it performs well on several categories which takes up a large proportion of street view images, for example, car, road and sky. However, it shows room for improvement in other categories.

We identify the cause of poor segmentation performance in certain categories due to small input image size. When the input image size is too small, the model may struggle to capture intricate details, leading to lower IoU scores, especially for objects or structures that require finer spatial resolution.

Further enhancements in performance could be achieved through applying other model, augmenting training data, and tuning train parameters.

Task Assignment

All the staffing information are provided in the following.

- Haoyu Wang：
    - Survey on 2D semantic segmentation
    - Construct cityscapes dataset
    - Train the model
- Zhangjie Chen
    - Train the model
    - Evaluate test performance
    - Try better Performance
- Xudong Zhang
    - Choose model and estimate memory consumption
    - Evaluate test performance
    - Try better Performance
- Xuyuan Li

- Construct self-sampled SUSTech dataset
- Data preprocessing
- Try better Performance

Project Schedule

Week	Task
12	- Survey on 2D semantic segmentation - Starting training U-Net
13	- Construct cityscapes dataset - Train the initial results of semantic segmentation
14	- Improve the accuracy of U-Net - Collect SUSTech datasets for U-Net - Clean and label the SUSTech datasets
15	- Get the results on SUSTech datasets - Analyze the final results - No-predefined label
16	- Visualize the final results - PPT, Report and Presentation

Reference

1. Olaf Ronneberger, Philipp Fisher and M. Kozubek. U-Net: Convolutional Networks for Biomedical Image Segmentation[J] .arXiv e-prints arXiv:1505.04597,2015. ↩

2. Prannay Khosla, Piotr Teterwak and Chen Wang. Supervised Contrastive Learning[J] .arXiv e-prints arXiv:2004.11362,2021. ↩

3. Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and Learn for Unsupervised Object Detection and Instance Segmentation[J]. arXiv pre-prints arXiv:2301.11320,2023. ↩