Transfer Learning in Semantic Segmentation
Team 9

Iou-Sheng Chang ichang9@jhu.edu
Ching-Yang Huang chuan120@jhu.edu
Ayush Gupta agupt120@jhu.edu
Arthur Beyer cbeyer4@jhu.edu

Johns Hopkins University
3400 N. Charles St., Baltimore, MD 21218, United States

Abstract

Deep Learning models have shown great success in various computer vision tasks, but their performance can be affected by domain differences between the pre-training dataset and the target dataset. In this paper, we investigate how domain differences affect the performance of existing pre-trained models for semantic segmentation. We then propose a transfer learning approach to address this issue. Specifically, we utilize a pre-trained model trained on the Cityscapes dataset and fine-tune it on a smaller, locally collected dataset, ***JHUStreet****, to evaluate the effectiveness of transfer learning. We analyze the domain differences between the two datasets and compare the performance of the pre-trained model before and after fine-tuning. Our experiments show that existing pre-trained models suffer under domain shifts, leading to a decrease in performance on the target domain. However, our transfer learning approach can effectively mitigate this issue and significantly improve the segmentation accuracy on the smaller out of domain dataset. The proposed approach is especially useful in scenarios where collecting large-scale training data is challenging, and the pre-trained model on a related dataset can be leveraged to improve the performance on a smaller dataset. Overall, our findings suggest that transfer learning can be an effective technique for adapting pre-trained models to new domains and improving their performance in real-world applications where collecting large amount of data in the target domain is not viable.*

1. Introduction

1.1. Motivation

Semantic segmentation is a crucial computer vision task that categorizes each pixel of an image to a certain class or object. The importance of semantic segmentation has increased in recent years, particularly in the context of autonomous vehicles, as it allows vehicles to operate safer by providing a better understanding of the surroundings.

However, one of the major challenges of semantic segmentation, or any deep learning-based method for that matter, is domain difference [12]. When the model is trained on one dataset and then tested on another, the model's performance can deteriorate significantly. The issue of domain difference is especially essential for autonomous vehicles, as they have to function in diverse environments and conditions. As a result, the ability to adapt to different domains – including variations in sensor types and environmental factors – is a crucial requirement of deep learning based semantic segmentation models.

1.2. Prior Work

Several algorithms have been proposed to tackle this problem. It can be observed in literature that CNN-based approaches are the most popular for the segmentation task [10]. The differences between different models lie in the specific CNN architecture, some examples being the encoder-decoder structure of U-net [11], dilated/atrous convolutions of DeepLabv3 [3], or the pyramid structure of ESPNet [9].

Several datasets have been collected to train models for this task. Some examples include CamVid [1,2], Cityscapes [5], Berkeley DeepDrive [13] and PASCAL VOC 2012 [6]. Many datasets are focused only on the task of autonomous driving, but there are generalized datasets for segmentation as well [6].

Currently, DeepLabv3+ [4] is one of the state-of-the-art models for semantic segmentation. It uses dilated convolutions at multiple scales to form a pyramid-like structure, enabling it to see features at multiple visual scales using a ResNet-like backbone.

Since these models need to work in a wide variety of domains, the techniques of domain adaptation [12] can be

utilized to make these models deployable in practical, real-life scenarios. We discuss this in Sec. 1.3.

1.3. Proposed Solution, Significance and Innovation

In this paper, we propose a solution for semantic segmentation in the context of autonomous driving across domains, with a focus on detecting two important categories: cars and traffic signs. Our proposed solution is to use the pre-trained DeepLabv3+ [4], a state-of-the-art deep learning model for semantic segmentation, on a smaller dataset with domain differences, which we collected around the streets near the Johns Hopkins Homewood campus, named as **JHUStreet**.

It was expected that the model would not perform well on the JHUStreet dataset due to the large domain difference with the dataset that the model was pre-trained on, which is Cityscapes [5]. To enhance the model's performance, we planned to utilize transfer learning to fine-tune the pre-trained DeepLabv3+ model by updating its parameters using the JHUStreet dataset.

Our objective is to enhance the model's capacity to generate more precise car and traffic sign segmentation in Baltimore through the use of transfer learning. This advancement can have a significant impact on the autonomous driving industry and can lead to the creation of transportation systems that are safer and more efficient.

2. Methods

This section describes the data, algorithm, and evaluation metrics used in our approach.

2.1. Data

The original dataset, Cityscapes [5], contains thousands of images taken from streets in Europe. Cityscapes recognizes 19 different classes of interest relevant for driving scenes captured by cameras placed on a car. The dataset has 2 types of labels: fine and coarse. The fine subset of the labels contains 5,000 images with corresponding pixel-precise labels. The coarse dataset contains nearly 25,000 images but each image has bounding polygons instead of pixel-precise labels. Fig. 1 shows examples of a finely and coarsely labeled image in the Cityscapes dataset. The base model we use to do transfer learning is pre-trained on the fine version of Cityscapes.

To perform transfer learning on the model, we first collected our own data by taking videos around the JHU-Homewood campus with an iPhone 12 Pro Max and wrote a Python script to subsample images from the video at a uniform frequency, forming our image data. We then manually labeled the data using *paint.net* image labeling software to serve as the ground truth. Fig. 2 shows examples of these images, as well as the corresponding labels. Since creating

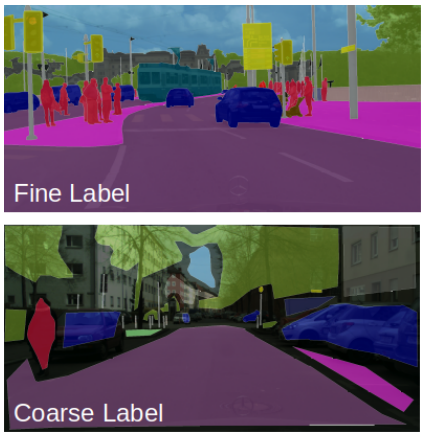

Figure 1. Example images of Cityscapes fine and coarse labels

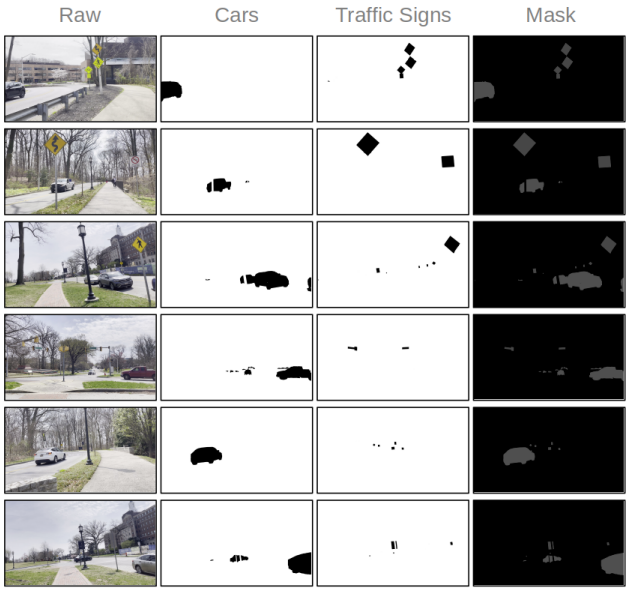

Figure 2. Examples of captured images and corresponding labels

segmentation masks is a time-consuming task, we were only able to label 46 images with 3 classes: Cars, Traffic Signs, and Background; as opposed to Cityscapes' 19 labels. We split these 46 images into a standard 70/10/20 train/val/test split for a total of 32 train images, 5 validation images, and 9 test images.

2.2. Algorithms

The model we use is the DeepLabv3+ Mobilenet architecture built in Python [4, 7] for segmentation, which has been pre-trained on the Cityscapes dataset. The network was fine-tuned on the self-collected JHUStreet dataset, utilizing the pre-trained model as initialization. To overcome the challenge of limited data, we incorporated various data augmentation techniques during training, such as random

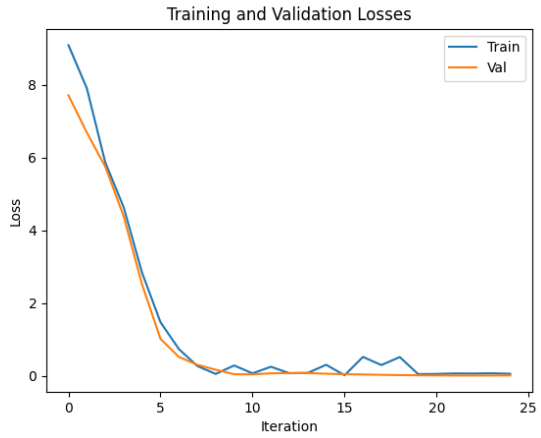

Figure 3. Training and validation losses over time using Cross-Entropy classification loss per pixel. It can be observed that both the training and validation losses have stabilized.

cropping, random color jittering, and random horizontal flip. The model was trained for 24 iterations (or 6 epochs) with a batch size of 8, using the cross entropy classification loss.

The original DeepLabv3+ model was trained with cross entropy loss as well, however, the pixels classified as background were ignored in the loss calculation. This made sense for the densely labelled Cityscapes dataset, however, this formulation did not work for our dataset with only two other classes apart from background. As a result, we implemented our own version of cross entropy loss which incorporated background pixels as well during training.

Fig. 3 depicts the loss curves, indicating that the training and validation losses have reached a steady state.

2.3. Evaluation Metrics

We employed two primary evaluation metrics, namely, pixel-wise accuracy and mean Intersection over Union (IoU), which are standard for segmentation tasks [3, 10]. Additionally, to visually assess the model's performance over time, we ran the model on the validation set after each iteration. We then overlaid an opaque version of the labeled image onto the original image and saved it as a new image. Fig. 4 illustrates some of these images, and we also created a short clip that displays them in sequence, starting from iteration 1 until the end, to showcase how the model improves over time.

After training, we assess the model's final performance on the test set using the same metrics as before, pixel-wise accuracy and mean IoU, to obtain a comprehensive evaluation of the model's effectiveness.

	Pixel Wise Accuracy	Mean IoU
Before	0.6642	0.6097
After	**0.8128**	**0.7819**

Table 1. Performance of the pre-trained DeepLabv3+ model before and after fine-tuning on the JHUStreet dataset.

Class	Before	After
Traffic sign	0.0717	**0.5753**
Cars	0.7847	**0.7882**
Background	0.9728	**0.9823**

Table 2. A comparison of the class IoU scores before and after fine-tuning on the JHUStreet dataset.

3. Results

This section discusses the main results of our approach, along with qualitative visualizations and failure cases.

3.1. Main Results

We conducted a performance comparison of our model before and after transfer learning on the JHUStreet dataset. Our expectation was that the model would exhibit better performance after training. As depicted in Tab. 1, the results confirm that the model indeed improved, which aligns with our initial expectations.

We performed IoU score calculations for each class separately. As there are three classes in our dataset, namely cars, traffic signs, and background, we made three such comparisons. Since the pre-trained model outputs 19 classes instead of only 3, we filtered out the output to only these three classes before calculating the IoU. The resulting comparisons are displayed in Table 2.

Notably, while there are improvements in the cars and background classes, the most significant enhancement in performance is observed in the traffic signs class, which we will discuss in Sec. 3.2.

3.2. Qualitative Comparison

In Fig. 4, we present visualizations of the model output before and after training, revealing that the outputs are notably improved after training, particularly for traffic signs. As the pre-trained model is trained on Cityscapes data captured on European streets, which feature significantly different traffic signs from those in the JHUStreet dataset, the model initially struggles since it lacks familiarity with these new types of signs. Nonetheless, given that cars appear similar in both domains, the performance on cars remains relatively unchanged.

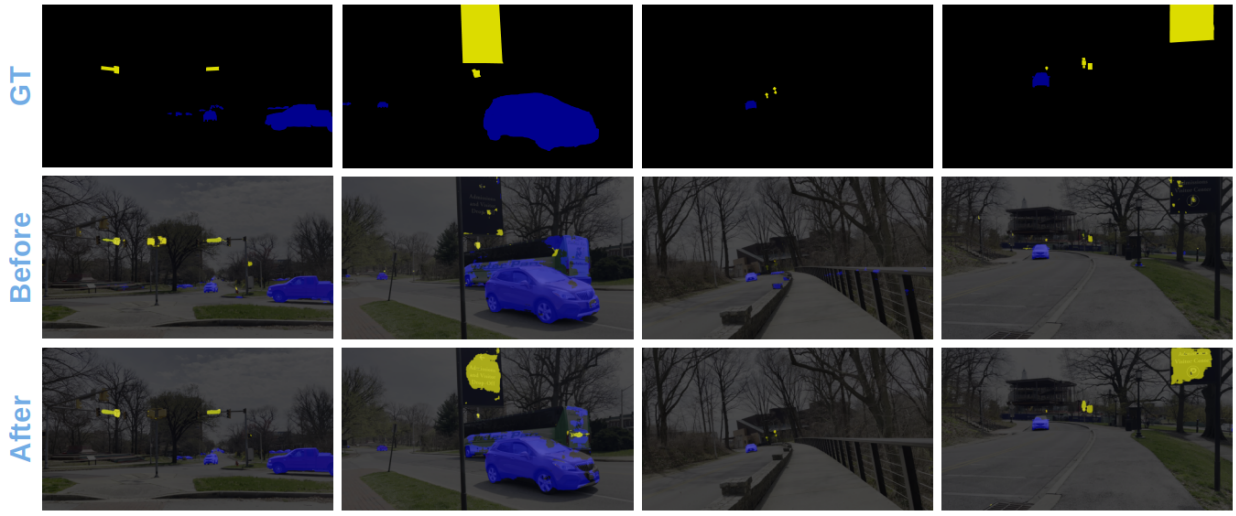

Figure 4. Some visualizations of the output of the network before and after training. The first row shows the self-labeled ground truth, the second row shows the output before training (filtered to show only three classes), and the last row shows the output after training. The last row has better segmentation outputs, especially for traffic signs. The second column also demonstrates a failure case of the network when it classifies some pixels of a bus as a car.

3.3. Anomalies and Failure Cases

The pre-trained model was originally trained to distinguish between cars and buses, but in our case, we trained it to only recognize cars while considering buses as background. This results in the model being confused during inference, as depicted in the second column of Fig. 4. Even after training, the model is not able to fully rectify the issue, as a few pixels of the bus are still misclassified as belonging to a car. We hypothesize that this is due to our limited ability to train the model on a diverse set of classes.

4. Discussion

This section provides an overview of the results and their connection to prior research in the field, while also proposing potential avenues for future investigation.

4.1. Summary of Results

We observe that the original network is not effective in generalizing well to our own dataset due to significant distribution shifts, caused by differences in camera sensor and geographic location. Since the original network was trained on images captured in Europe, it fails to adapt to the images collected in Baltimore, and this disparity is most evident in the case of traffic signs, which have distinct differences between the United States and European roads.

The results show that after transfer learning, the model's performance improves for the classes of interest. Specifically, the segmentation outputs for cars remain qualitatively similar, but the class IoU score increases as some incorrectly classified pixels are now correctly recognized as background. Moreover, the model can now identify blue JHU traffic signs, which were previously absent, as demonstrated in the second and fourth column of Fig. 4. Overall, the transfer learning process enhances the outputs of the model.

In our work, we highlight a limitation of our method stemming from the lack of diversity of classes used for training. The original network was trained on 19 classes, but we only use three classes in our work due to the significant manual labeling effort required to accurately create segmentation masks. As a result, the model fails to distinguish between similar but distinct objects, such as cars and buses, and classifies them as belonging to the same class (cars), which ultimately harms the final performance.

4.2. Pertaining to Previous Work

Semantic segmentation is a crucial computer vision task, especially for autonomous vehicles. Semantic segmentation models used for autonomous driving must be highly robust to a large variety of conditions to be deployable. However, most deep learning methods driven by data are not adept at performing well across various domains. This issue has spawned a whole field of study called domain adaptation [12], which seeks to address this problem. The simplest form of domain adaptation, or transfer learning, involves training a model on a large source dataset to capture common low-level features and subsequently fine-tuning it on a smaller target dataset to enable adaptation to the new domain. In our work, we follow this approach by collecting a target dataset around the JHU-Homewood campus. Our findings align with the existing research on domain adapta-

tion, indicating that deep learning models typically struggle with distribution shifts. However, a small amount of data from the new distribution can assist the model in quickly generalizing to the new domain.

4.3. Future Work

As a potential avenue for future research, an assessment of the generalizability of the trained model on the Cityscapes dataset could be conducted. This would provide additional insight into whether the transfer learning process resulted in the model overfitting to the JHUStreet data.

Additionally, efforts to increase the size and diversity of the dataset may improve the performance of the model's segmentation outputs. To achieve this goal, data collection and labeling could be pursued.

Given the limited amount of training data available, it may also be advantageous to explore approaches such as few-shot learning or meta-learning, which have demonstrated superior performance compared to standard fine-tuning techniques on target datasets.

5. Conclusion

Semantic segmentation has seen many advances in recent years [8, 10], with the proposal of numerous new architectures [3, 9, 11]. However, for practical deployment and use in autonomous vehicles, it is crucial that these models exhibit robustness to a wide range of environmental conditions and work effectively in various geographic locations. In this paper, we examine whether a state-of-the-art semantic segmentation technique [3], trained on a large-scale public dataset [5], is capable of performing well under domain shift. To achieve this, we collect our own dataset, **JHUStreet**, from a geographically distinct area, and evaluate the performance of the model across different continents. On seeing a significant drop in performance, we use transfer learning to further adapt the model to the new dataset and observe that the model is able to perform better than before. Our results show that the model performs better after transfer learning, which we confirm through visualization of the network outputs. However, the limited diversity of class labels and small dataset size still impose restrictions on the model's performance. Future work could involve increasing the dataset size and diversity, as well as implementing more efficient techniques to learn from limited data to further enhance performance.

6. Contribution

For a comprehensive overview of group contributions, please refer to Tab. 3.

References

[1] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video. a high-definition ground truth database. 30(2):88 – 97, 2009. Available online 22 April 2008. 1

[2] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, number Part 1 in Lecture Notes in Computer Science, pages 44 – 57. Springer, 2008. 10th European Conference on Computer Vision; Conference Date: October 10-12, 2008; . 1

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017. 1, 3, 5

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018. 1, 2

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. 06 2016. 1, 2, 5

[6] Mark Everingham, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 06 2010. 1

[7] Gongfan Fang. Deeplabv3plus-pytorch. https://github.com/VainF/DeepLabV3Plus-Pytorch, 2022. 2

[8] Christopher J. Holder and Muhammad Shafique. On efficient real-time semantic segmentation: A survey, 2022. 5

[9] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation, 2018. 1, 5

[10] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, 44(07):3523–3542, jul 2022. 1, 3, 5

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1, 5

[12] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, oct 2018. 1, 4

[13] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning, 2020. 1

Member	Coding		Presentation		Write-up	
	Description	%	Description	%	Description	%
Danny	Collected and labeled the JHUStreet dataset. Corrected falsely labeled data. Proposed solution to align code with project's transfer learning objectives. Collected transfer learning results. Improved printing and visualization methods for information collection.	25	Formatted the PowerPoint template. Drafted the Methods and Results sections. Presented the Results section. Finalized the presentation slides.	27	Created the LaTeXtemplate. Drafted the write-up (bullet points). Drafted the Abstract, Results, Discussion, and References section. Proofread and Finalized the write-up.	27
Austin	Collected and labeled the JHUStreet dataset. Corrected falsely labeled data. Created script for processing raw video to image data and visualizing validation data during training. Collected transfer learning results. Resolved memory issues and corrected code mistakes.	25	Drafted the Results section. Presented the Introduction and part of the Method section. Finalized the presentation slides.	27	Drafted the Abstract, Introduction and Methods sections. Proofread and Finalized the write-up.	27
Ayush	Researched methods for labeling and labeled the JHUStreet dataset. Proposed solution to align code with project's transfer learning objectives. Implemented code for loss curve plotting, and modified the cross entropy loss function to include background.	25	Drafted the Introduction, Discussion, Conclusion, and Future Work sections. Presented the Discussion and Conclusion sections. Proofread the final version.	27	Drafted the Introduction, Results, Discussion, and Conclusion sections. Proofread and Finalized the write-up.	27
Arthur	Processed the JHUStreet dataset with the pre-trained DeepLabv3+ model. Created the script for combining class labels to mask. Implemented code for evaluation metrics.	25	Presented the Data and methods sections. Proofread the final version.	19	Drafted the Methods section. Proofread the write-up.	19

Table 3. Group Contribution