Prototypical Cross Domain Self-Supervised Learning for Few-shot Unsupervised Domain Adaptation in Semantic Segmentation

GM Harshvardhan
Boston University
Graduate School of Arts and Sciences
gmharsh@bu.edu

Harshil Gandhi
Boston University
Graduate School of Arts and Sciences
harshilg@bu.edu

Ashwin Daswani
Boston University
Graduate School of Arts and Sciences
ashwind@bu.edu

Rohan Sawant
Boston University
Graduate School of Arts and Sciences
rohan16@bu.edu

1. Introduction

Deep learning has grown significantly in the past decade and is now commonly used in many areas. This is because better computers have allowed us to work with more data and use advanced algorithms. However, finding high-quality data still remains a challenge, and it is often difficult and very costly to obtain labeled data. Moreover, the problem of "domain shift" arises when machine learning models that are trained on a specific data domain are deployed in a domain foreign to its knowledge base. This severely degrades model performance, and various techniques have been employed to tackle this issue in the past [9]. In this project, we will attempt to implement a few-shot unsupervised method, based on previous approaches [12], that will minimize model error when tested on new domains. We will only keep a few data labels from the source, and keep the rest of the source data and the target domain data unlabeled. We will use this model specifically for **semantic segmentation** which remains to be an active area of research, and to the best of our knowledge, has not been solved by other state-of-the-art methods.

Our current implementation, which we will see in further detail later, utilizes a UNet [8] with a pretrained encoder with ImageNet weights. High-level features at the bottom of the encoder are extracted and fed to a Cosine classifier [3] which labels each region. Then these region vectors are used for performing K-means and finding an in-domain and cross-domain loss. Each of these losses is used to train the UNet, along with its own segmentation loss. Currently, we are running into memory issues during training, wherein we are facing abrupt termination of training due to process exits (see Fig. 1). We are working on making the pipeline more efficient by searching for memory leaks, optimization, and vectorization solutions.

2. Datasets and Approach

For this project, we used the Cityscapes dataset [2] which contains scenery images from urban areas, and is a standard benchmark dataset for semantic segmentation tasks. Primarily, this dataset is used as the source domain, and the GTA 5 dataset [7] is used as the target dataset. While we had planned to use the Office-Home dataset [10] before and use the Meta Segment Anything API, we found that using this API would be very time consuming, and hence we switched to industry standard datasets for domain adaptation in semantic segmentation.

Fig. 2 shows the proposed model architecture. Following are the steps to train our model:

1) Train a cosine classifier separately on Unet encoder's H x W x C features to get region-level labels on the downsampled segmentation features
2) For each label, get a 1 x C averaged embedding (by averaging across all regions of the same label for each training instance)
3) Use these embeddings as initial centroid guesses for K-means
4) Compute in domain and cross domain features using the implementation of [12]
5) With the updated prototypes and shifted samples (with momentum), update the memory bank and get source and target weight vectors w^s and w^t 6) Perform the adaptive cosine classifier learning on average region embeddings using implementation of [12]
7) Combine all losses: in-domain, cross-domain, cosine classifier loss, and few-shot segmentation loss, and backpop

them through the Unet. Additionally, we seek to implement the max squares loss [1] that is useful when computing cross-domain loss in domain adaptation for semantic segmentation.

```
!python PCS-FUDA-v2/PCS-FUDA-master/run.py --config PCS-FUDA-v2/PCS-FUDA-master/config/cynthia/cynthia_json.json
  File "/share/pkg.7/pytorch/1.10.2/install/lib/SCC/../python3.7/site-packages/torch/autograd/grad_mode.py", line 28, in decor
ate_context
    return func(*args, **kwargs)
  File "/projectnb/cs585bp/gmharsh/PCS-FUDA-v2/PCS-FUDA-master/pcs/agents/CDSAgent.py", line 987, in compute_train_features
    for batch_i, (images, labels) in enumerate(train_loader):
  File "/share/pkg.7/pytorch/1.10.2/install/lib/SCC/../python3.7/site-packages/torch/utils/data/dataloader.py", line 521, in _
_next__
    data = self._next_data()
  File "/share/pkg.7/pytorch/1.10.2/install/lib/SCC/../python3.7/site-packages/torch/utils/data/dataloader.py", line 1186, in
_next_data
    idx, data = self._get_data()
  File "/share/pkg.7/pytorch/1.10.2/install/lib/SCC/../python3.7/site-packages/torch/utils/data/dataloader.py", line 1142, in
_get_data
    success, data = self._try_get_data()
  File "/share/pkg.7/pytorch/1.10.2/install/lib/SCC/../python3.7/site-packages/torch/utils/data/dataloader.py", line 1003, in
_try_get_data
    raise RuntimeError('DataLoader worker (pid(s) {}) exited unexpectedly'.format(pids_str)) from e
RuntimeError: DataLoader worker (pid(s) 29317) exited unexpectedly
[Compute train features of source]:  19%|▉    | 88/457 [20:20<1:25:17, 13.87s/it]
```

Figure 1. Memory issues during training

9

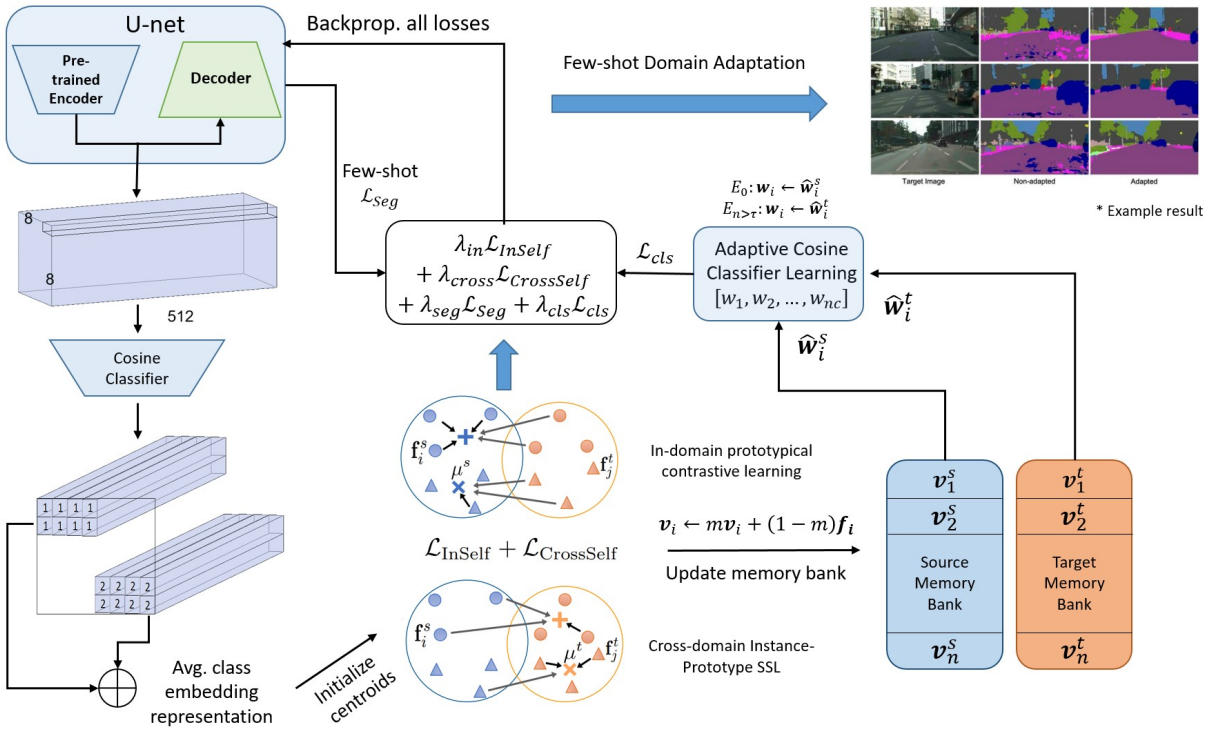

Figure 2. Proposed model architecture. Portions of image adapted from [12]

3. Results

Since we are currently working on finishing the training of the model (and running into memory issues), we do not have concrete results. However, we are confident that the UNet when trained separately on the top left of Fig. 2, will come up with interesting segmentation masks that will be domain independent. We have talked with the Professor, and we intend to **continue our work over the summer** and collaborate with him to submit our work at a major CV conference.

4. Discussion and conclusion

In our approach, we consider **certain novel strategies and improvements** that address the shortcomings of other state-of-the-art implementations such as the PCS. One such improvement is providing a **warm start** to the K-means clustering that obtains in-domain and cross-domain losses. In the PCS implementation [12], the authors overcome the shotcomings of CDS [5] by using instance-prototype matching instead of instance-instance. In our approach, we overcome the problem of non-convergence or non-optimal solution given by K-means that would be inevitable in higher dimensions with the PCS approach by using smartly initialized centroids produced by source labeled instances and high confidence unlabeled instances fed to the cosine classifier. Moreover, as suggested by [1], we use the max squares loss for semantic segmentation in domain adaptation which is better suited for our problem setting.

So far, using the code built upon our primary reference PCS, we have added ¿400 lines in data loading, region-wise indexing, and U-net and Cosine Classifier train loops and 500 lines in Jupyter notebooks for unit-testing and debugging.

We have learnt a lot of things from this implementation. Following are some insights:

1) Attempting a state-of-the-art implementation for a new problem setting requires significant changes to model architecture, since the existing implementation will be very specifically designed for its native problem setting. Due to this fact, we spent more than 80% of the semester time in the ideation process, which required a lot of research and brainstorming.
2) The codebase of PCS is very specific to its own implementations, and changing anything leads to a chain of error solving and troubleshooting.
3) After everything is resolved, there may still be problems during runtime, such as memory issues during training which we are facing now.

5. Individual contributions

Most part of the work were done in groups and in a collaborative manner. Each member contributed in more or less the same amount. The individual contributions are as follows:

GM Harshvardhan. The task of implementing the PCS GitHub repo and translating ideas into code, made significant progress in this area by successfully implementing the formulation of semantic segmentation problem setting and devising ways to convert from classification to segmentation setting. Additionally, documented the implementation process in detail and shared it with the team for future reference. 2) Research: Read, review and analyze Prototypical Contrastive Learning [6] which is a concept used in PCS. 3) Optimized many parts of the code so that we it could run within limitations of our GPU memory and computational power

Harshil Gandhi. 1) Worked on implementation of the formulation of semantic segmentation problem setting and think of ways to convert from classification to segmentation setting. 2) Research: Read, review and analyze Momentum Contrast for Unsupervised Visual Representation Learning [4] which is a concept used in PCS. 3) Worked on preprocessing of the dataset, target and source (Over 220 GB of data) 4) Optimized many parts of the code so that we it could run within limitations of our GPU memory and computational power

Ashwin Daswani. 1) Worked on implementation of Meta AI's segment anything engine to generate segmentation masks as channels for each class for the domains "art" and "clipart". 2) Research: Read, review and analyze CDS [5] as it is related closely to PCS. 3) Worked on preprocessing of the dataset 4) Actual execution of the implemented code on SCC and solving for errors, bugs and giving feedback to the one who has written the code

Rohan Sawant. 3) Worked on preprocessing of the dataset, target and source (Over 220 GB of data) 2) Research: Read, review and analyze [11] which helped in the intuition of "memory banks" used in PCS. 3) Actual execution of the implemented code on SCC and solving for errors, bugs and giving feedback to the one who has written the code 4) Making the final report as well as the presentation for the work done.

References

[1] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2090–2099, 2019. 2, 3

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*, volume 2. sn, 2015. 1

[3] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375, 2018. 1

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. 3

[5] Donghyun Kim, Kuniaki Saito, Tae-Hyun Oh, Bryan A. Plummer, Stan Sclaroff, and Kate Saenko. Cds: Cross-domain self-supervised pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9123–9132, October 2021. 3

[6] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 3

[7] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016. 1

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1

[9] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 213–226. Springer, 2010. 1

[10] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 1

[11] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 3

[12] Xiangyu Yue, Zangwei Zheng, Shanghang Zhang, Yang Gao, Trevor Darrell, Kurt Keutzer, and Alberto Sangiovanni-Vincentelli. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 1, 2, 3