Computer Vision II, Fall 2023
Assignment 1

Prof. Jingya Wang

October 23, 2023

Acknowledgement

1. Deadline: 2023/11/26 23:59:00. Late Policy please refers to the course slides.
2. Giving your report in English, report in Chinese is not accepted. Besides, handwritten homework is not accepted and we highly recommend you to use LaTeX. LaTeX template has been upload to Blackboard.
3. Please submit your assignment in Gradescope with PDF format. Registration code: ZWP2JB.
4. Please upload your code zip to ShanghaiTech cloud disk. `https://epan.shanghaitech.edu.cn/l/9Frhjy`. All source files and readme should be included but remember to remove the datasets. Your zip should be named as CS272_NAME_ID_hw1.zip.
5. Plagiarism or cheat is strictly prohibited. Do not share your assignment or code! No fake solution is allowed! Make sure that your codes can run and are consistent with your solutions

1 [10 points] Train your classification model using PyTorch (source-only training for domain adaptation)

- Learn PyTorch: `https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html`.
- Download the Office-Home dataset. You are required to show some instances per domain.
- Using PyTorch to build your data loader (including data augmentation), model (feature extractor + classifier), optimizer in your code. For the backbone selection of feature extractor, we recommend you to use ResNet50, but any other vision backbone is also welcomed. Then build your training and testing pipeline. You are required to specify your implement details, including data augmentation, network structure, batch size, optimizer, learning rate, training steps etc.
- Choose one domain as train set (source domain), and choose another different domain as test set (target domain). Then train your classification model using cross-entropy loss. You are required to show the line chart of loss decay, train accuracy (source domain) and test accuracy (target domain) in the training process for 3 transfer tasks: Art→Product, Art→Real-World-Images, Clipart→Art. Performance should excess 50%.

2 [30 points] Deep correlation alignment (CORAL) loss for domain adaptation

- Revisit Deep correlation alignement (CORAL) and go deeper for mathmetical details of CORAL loss, then try to figure out how to implement CORAL loss in your code. You are required to give a detailed description about the implementation of CORAL loss and specify how CORAL helps in domain adaptation from your view.

- Implement CORAL loss on your own, then add CORAL loss into the loss in Sec. 1, i.e. cross-entropy and CORAL loss.
- Choose your source and target domain, then train your domain adaptation model. You are required to show the line chart of loss decay, source domain accuracy and target domain accuracy in the training process for 3 transfer tasks: Art→Product, Art→Real-World-Images, Clipart→Art. At least two tasks should have performance improvement.

3 [30 points] Local Maximum Mean Discrepancy (LMMD) loss for domain adaptation

- We have learned MMD loss in class. Now you are required to explore LMMD loss proposed in [2] and go deeper for mathmetical details of LMMD loss, then try to figure out how to implement LMMD loss in your code. You are required to give a detailed description about the implementation of LMMD loss and specify how LMMD helps in domain adaptation from your view.
- Implement LMMD loss on your own, then add LMMD loss into the loss in Sec. 1, i.e. cross-entropy and LMMD loss.
- Choose your source and target domain, then train your domain adaptation model. You are required to show the line chart of loss decay, source domain accuracy and target domain accuracy in the training process for 3 transfer tasks: Art→Product, Art→Real-World-Images, Clipart→Art. Performance should excess 60%.

4 [30 points] Neural Style Transfer

Neural Style Transfer (NST) is one of fundamental applications in computer vision. The NST merges a content image $\boldsymbol{x}_c$ and a style image $\boldsymbol{x}_s$ to create a generated image $\boldsymbol{x}_g$. The generated image should combine the content of one image (the content image) with the artistic style of another image (the style image). The classical neural style transfer approach proposed in [1] iteratively optimizes a content loss and style loss:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{content}} + \beta \mathcal{L}_{\text{style}}, \tag{1}$$

Assume the feature maps for $\boldsymbol{x}_g, \boldsymbol{x}_c, \boldsymbol{x}_s$ in layer l are denoted by $\boldsymbol{F}_l \in \mathbb{R}^{N_l \times M_l}, \boldsymbol{P}_l \in \mathbb{R}^{N_l \times M_l}, \boldsymbol{S}_l \in \mathbb{R}^{N_l \times M_l}$, respectively, where N_l denotes the number of feature maps, and M_l equals to the width $\times$ height of the feature map. Now:

- Derive the formula of $\mathcal{L}_{\text{content}}$ and explain the choice of layers to capture the content representation.
- Derive the formula of $\mathcal{L}_{\text{style}}$ and briefly discuss the role of gram matrix.
- Analyse the $\mathcal{L}_{\text{style}}$ from the perspective of domain adaptation. (Hint: Derive an equivalent form of $\mathcal{L}_{\text{style}}$ and the MMD loss.)

References

[1] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2

[2] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE transactions on neural networks and learning systems*, 32(4):1713–1722, 2020. 2