110-1 NTU DBME5028

Application of Deep Learning in Medical Imaging, Fall 2021

Final project

Due: 5th Jan. 2022 (Wed.) 09:00 AM (GMT+8)
Total Score: 140 points
Contact TA (Ming-Yang Ho) by r08945027@g.ntu.edu.tw

Rules:

1. Any form of plagiarism (e.g., copy a string of consecutive 7 or more words from an original work) is not allowed. Students involved in plagiarism would receive an ***F*** for this course (not just for this final project).
2. Codes should be written by yourself. Open-source codes (except libraries that can be installed by `pip install`) cannot be utilized unless you provide the source (e.g., GitHub link and corresponding #line).
3. The final project has two parts: the *non-programming part* and the *programming part*. For the *non-programming part*. You should write critiques based on four selected papers and submit before the deadline **independently**. For *the programming part*, you should collaborate with teammates to accomplish a machine learning competition held on Kaggle. Both reports and codes should be submitted. A 20-minute presentation for each group will be hosted on 5th Jan. 2022.
4. We strongly encourage you to find solutions by reading journal and conference papers. However, you should cite those researches *in a formal format* (such as APA-7th).
5. For the programming part, we strongly encourage you to discuss it with your classmates. However, their names and contributions should be listed in the acknowledgement section in the submitted report.
6. ALL THE REPORTS should be written in English, except your Chinese name, or you will not gain any point from the report part.
7. You cannot leverage any public dataset to train and improve your model. Those who violate this policy would get an ***F***. However, the pretrained models provided by `torchvision.models` are allowed.
8. **Late submissions are not accepted.**

Submission and deadline:

1. Critiques should be submitted to NTU COOL in pdf format. Deadline:

- <Object detection 1> 15th Dec. 2021 09:00 (GMT+8)

- <Object detection 2> 22nd Dec. 2021 09:00 (GMT+8)
- <Self-supervised learning> 29th Dec. 2021 09:00 (GMT+8)
- <Domain adaption> 5th Jan. 2022 09:00 (GMT+8).

2. [Project] The prediction results should be submitted to Kaggle. Deadline: 4th Jan. 2022 23:59 (GMT+8).
3. [Project] The report should be submitted to NTU COOL in pdf format. Deadline: 5th Jan. 2022 09:00 (GMT+8).
4. [Project] The presentation slide should be submitted to NTU COOL in pdf format. Deadline: 5th Jan. 2022 09:00 (GMT+8).
5. [Project] The codes should be uploaded to your GitHub repository. Deadline: 5th Jan. 2022 09:00 (GMT+8).

Grading policy:

1. The final project accounts for 50 percent of your overall grade. Namely, if you get *x* points in the final project, it will account for *x/140*0.5* point of your overall grade.

2. The following conditions will make you fail this class; please be careful.

a. Any form of plagiarism (e.g., copy-paste, translation of copied sentences), especially for your report.
b. Use other public/private datasets to improve your models or attempt to find oracles to obtain the ground truth labels.
c. Create multiple accounts to bypass the limitation of Kaggle's daily submission.

3. The following conditions will also lead to penalties; please be careful.

a. The prediction results cannot be reproduced (-100% Kaggle & -100% Code)
b. Codes with poor quality (e.g., not modularized, poor time/space complexity)
c. Reports with poor quality (e.g., too many grammar errors and formatting errors)

A. Non-programming part (40 points)

[THIS PART SHOULD BE DONE INDIVIDUALLY]

Please select one paper from the **`<Object detection I>`**, one paper from the **`<Object detection II>`**, one paper from the **`<Self-supervised learning>`**, and one paper from the **`<Domain adaption>`** and write critiques for them. Each will account for 10 points. The critique should be **one page** (12 pt, single line, Times New Roman, English only) and contain sections of Summary, Strength, Weakness, and Reflection.

You can download the template here:
https://docs.google.com/document/d/1lv5srIU19749k5MfDWolD-pKDzORe7TM/edit?usp=sharing&ouid=102460530069025567619&rtpof=true&sd=true

<Object detection I>

1. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).
2. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 764-773).
3. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 658-666).
4. Tian, Z., Shen, C., Chen, H., & He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9627-9636).

<Object detection II>

1. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., ... & Lin, D. (2019). Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4974-4983).
2. Cao, J., Cholakkal, H., Anwer, R. M., Khan, F. S., Pang, Y., & Shao, L. (2020). D2det: Towards high quality object detection and instance segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11485-11494).
3. Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS--improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision* (pp. 5561-5569).
4. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

<Self-supervised learning>

1. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G. (2020). Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*.
2. Chen, X., Fan, H., Girshick, R., & He, K. (2020). Improved baselines with

momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.
3. Grill, J. B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., ... & Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.
4. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021). Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*.

<Domain adaption>

1. Wei, G., Lan, C., Zeng, W., & Chen, Z. (2021). MetaAlign: Coordinating Domain Alignment and Classification for Unsupervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16643-16653).
2. Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., & Zhang, W. (2020, April). Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 04, pp. 6502-6509).
3. Tachet des Combes, R., Zhao, H., Wang, Y. X., & Gordon, G. J. (2020). Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, *33*.
4. Yue, X., Zheng, Z., Zhang, S., Gao, Y., Darrell, T., Keutzer, K., & Vincentelli, A. S. (2021). Prototypical Cross-domain Self-supervised Learning for Few-shot Unsupervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13834-13844).

B. Programming part (100 points)

[THIS PART SHOULD BE DONE WITH YOUR TEAMMATE]

The report should be in two columns and comprise at least **abstract, introduction, methods, results, discussions, and conclusion sections**. We recommend you utilize the CVPR LATEX template (e.g., https://www.overleaf.com/latex/templates/cvpr-2018-template/qgmrfntfbqns) to easily deal with formatting problems. The report is limited to **eight** pages.

Please follow the instructions to initialize everything:

1. Create a Private GitHub repository named 110-1-NTU-DBME5028-Final and add **Kaminyou** as the collaborator.
2. Please register in the following google form (with your teammate). Please also team up in the Kaggle with the team name you specified in the google form. You cannot submit to Kaggle until finishing these steps!

https://forms.gle/4RoQkN6WLKF2TK9C9

3. Team up in the Kaggle.
   https://www.kaggle.com/t/74d120ec07144012aa72a12aa1d08e8a
4. Download the datasets and train the models. The links are provided in the Kaggle.
5. The maximum daily submission count is 5.

TASK: Histology images query competition

https://www.kaggle.com/t/74d120ec07144012aa72a12aa1d08e8a

Dataset description: image query evaluated by the c-index

	Training set	**Testing set (public)**	**Testing set (hidden)**
# patches	6187	3427 (49807 queries)	
# whole slide	20	-	-

No label will not be provided in this task.

Given a query image pair $\{x_i,\ x_j\}$, you should predict *whether x_i and x_j belong to the same category*. Suppose all images are sampled from $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}} = \{\boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2, \ldots, \boldsymbol{\mathcal{X}}_k\}$, a query pair $\{x_i,\ x_j\}$ is randomly sampled from $x_i \leftarrow \boldsymbol{\mathcal{X}}_i$ and $x_j \leftarrow \boldsymbol{\mathcal{X}}_j$, where i and j are randomly selected from $\{1,2,\ldots,k\}$. However, k is unknown here.

C-index will be adopted to evaluate your prediction.

$$\mathrm{c-index} = \frac{correct\ predicitons}{\#\ of\ query\ pairs}$$

Due to the limits of computation, C_2^{3427} queries would be impossible for the Kaggle submission. We randomly sampled 50000 queries from all the query pairs and enumerated them in the `queries.csv`.

a. Kaggle (40 points) **baselines will be release before 14th Dec. 2021*
   - Simple baseline (+ 15 points)
   - Strong baseline (+ 10 points)
   - Ranking (+ 0~15 points by linear interpolation)
     - Please note that if two teams get the same scores on the evaluation metric, we will adopt the same inferior ranking for these two teams (e.g., if there are only team A, B, and C with accuracies of 99%, 99%, and 98%, team A and B will be graded as if they are all in the 2nd prize). Hence, try your best to defeat any opponent!

b. Report (35 points)

c. Presentation (20 points)

d. Code (5 points)

- Please provide a script called `train.py` with an argument `--data $data_path`, which assume the training data are in `$data_path/train/`
- TAs will arrange their file structure as:

```
$data_path\
    ├── queries.csv
    ├── test\
    ├── train\
    └── train_whole_slide\
```

- Please provide a script called `download.sh` if there is any model that should be loaded for inference. You can upload your models to any accessible cloud drive.
- Please provide a script called `inference.py` with an argument `--data $data_path`, which assumes the testing data are in `$data_path/test/`. This script should be able to generate a `prediction.csv` with headers of `query, prediction` (please refer to the `example_submission.csv` in the Kaggle), which can be submitted to obtain the same prediction results of your best submission record.
- TA will execute your script in the environment of Ubuntu 18.04.5 + PyTorch 1.7.0. If your program requires some special packages, please provide a `requirements.txt` with a specified version.

Hints for the Kaggle competitions

1. Try to leverage the unlabeled slide by some segmentation strategies.
2. If you do not have enough computational resources, try something that can alleviate this issue.
3. You are encouraged to survey the latest papers and implement those methods to defeat your opponents.