DL Lab8: Semantic Segmentation

Lab Objective:

In this homework, you are required to implement a U-Net neural network using the PyTorch framework on your own, and use the CCAgT dataset.

Rules:

(1) This assignment should be **done individually**. Plagiarism is strictly prohibited.

(2) Only PyTorch frameworks are allowed in this lab, beyond that you can only use numpy, matplotlib, and library built in with python.

(3) **Only CCAgT dataset are allowed** in this lab.

(4) The assignment format and files are not in accordance with the regulations, the report **score × 0.9**.

(5) If the assignment is missing or incomplete training for any item (for example, implementing semantic segmentation as other segmentation or not including certain measurement values or not dividing the dataset according to the requirements), the assignment score will be deducted proportionally to the incompleteness.

(6) If you submit your assignment late, your score will **be multiplied by 0.9 for each day of delay**.

Submission:

(1) Please **use a jupyter notebook as the carrier for your program execution** (training, testing, and presentation of chart results must be included in the file, and some other functions such as drawing can still be written as a **.py file but must be imported by the .ipynb file**) so that we can clearly see the process of your model training and the final performance of the model.

(2) You should detail any clever ideas in your program design and the final performance of the model, even graphical results, in a report of **no more than Six A4 pages**. If you have different methods, please detail the performance comparison between the various methods. Finally, you need to attach your insights, but it is strictly forbidden to post the entire code.

(3) Upload the compressed file (**.zip**) of "the report (**.pdf**), all program files (**.ipynb**), and all the best weight of 1 kind methods which you wrote." to the e3 platform. The file name is **A8_studentID_studentName.zip**.

Deadline: 2024/12/23 (Mon.) 12:00

Requirements:

(1) You are required to **implement your own CCAgT dataset**. If you are unable to complete it, you may refer to and modify others' written ones on the internet

(but you must clearly state and cite the website in the report, if not it will be considered as plagiarism).

(2) You need to divide the CCAgT dataset into 3 parts: training, validation, and testing, occupying **7:1:2** respectively. You need to **plot the training loss and validation loss during training** and present the final results through **several test methods (mAP@[.5:.95:.05], PA, cPA, mPA, Dice, IoU, mIoU)** and clearly **present 10 of the best results** through graphical methods using matplotlib.

(3) You need to implement **at least one simple U-Net neural network** as the model for training and present the results. If you have implemented other models or some special data increment improvement methods, they will be an additional bonus.

(4) **Set "torch.manual_seed(12)" and "torch.backends.cudnn.deterministic = True"** in your code for model's training reproducibility.

(5) **In short, your homework must at least include the following parts**:

I. Your own implementation of the CCAgT dataset.

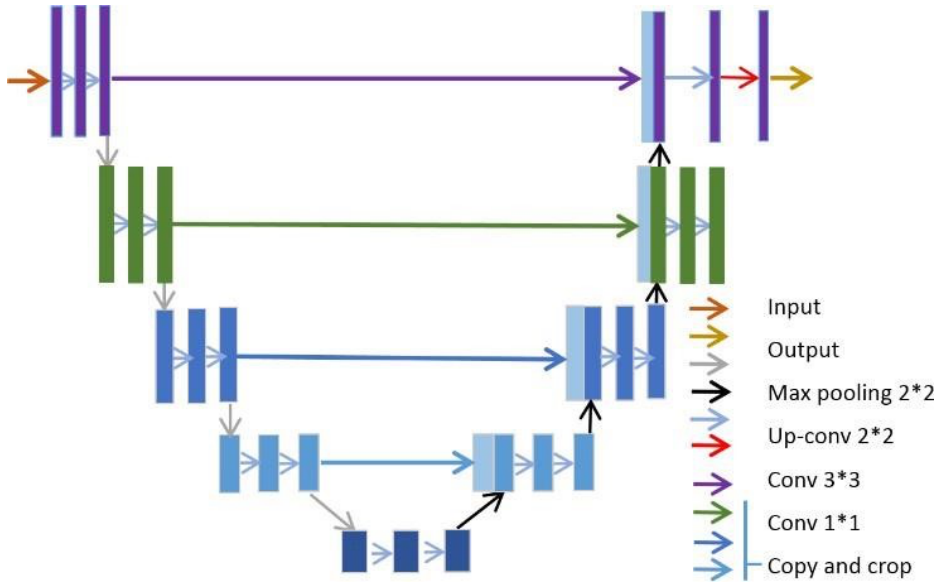
II. Your own implementation of the U-Net neural model for semantic segmentation.

III. Plot the training loss and validation loss during training IV. Model performance evaluation through methods such as (**mAP@[.5:.95:.05], PA, cPA, mPA, Dice, IoU, mIoU**).

VI. Graphical presentation of 10 results of semantic segmentation through graphical methods.

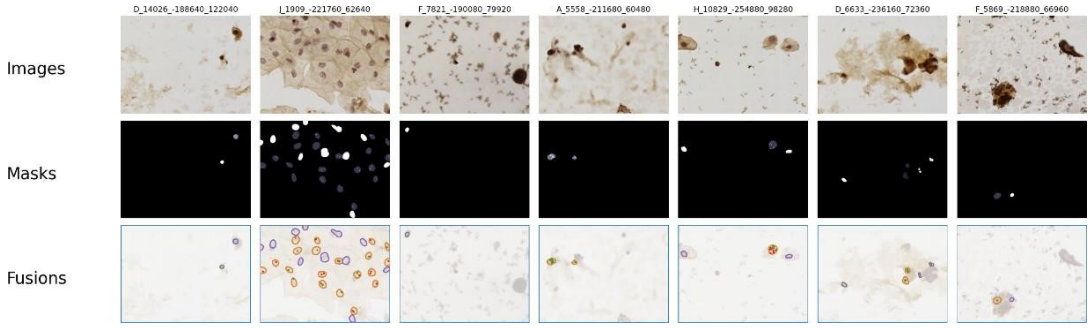
Descriptions:

(1) Model Architecture



(2) Dataset (CCAgT)

- Datasets: https://data.mendeley.com/datasets/wg4bpm33hj/2



Contains 9339 images with resolution of 1600×1200 where each pixel is 0.111µmX0.111µm from 15 different slides stained with AgNOR technique, having at least one label per image.

(3) Metrics Method

I. **Pixel Accuracy**：Defined as the ratio of the number of correctly predicted category pixels to the total number of pixels.

II. **class Pixel Accuracy**：Defined as the accuracy of the pixels that truly belong to category i among the predicted values of category i, in other words: the model has many predicted values for category i, and some of them are wrong, and the ratio of the correct predicted values to the total predicted values is cPA.

III. **mean Pixel Accuracy**：Definition is to calculate the ratio of correctly classified pixels for each category, that is, CPA, and then add them up to find the average.

IV. **Sørensen–Dice coefficient**：Defined as two times the area of the intersection of A and B, divided by the sum of the areas of A and B.

V. **Intersection over Union**：Defined as the ratio of the intersection and union of the model's predicted results and the true values for a category.

VI. **mean Intersection over Union**：Defined as the sum and average of the ratio of the intersection and union of the model's predicted results and the true values for each category.

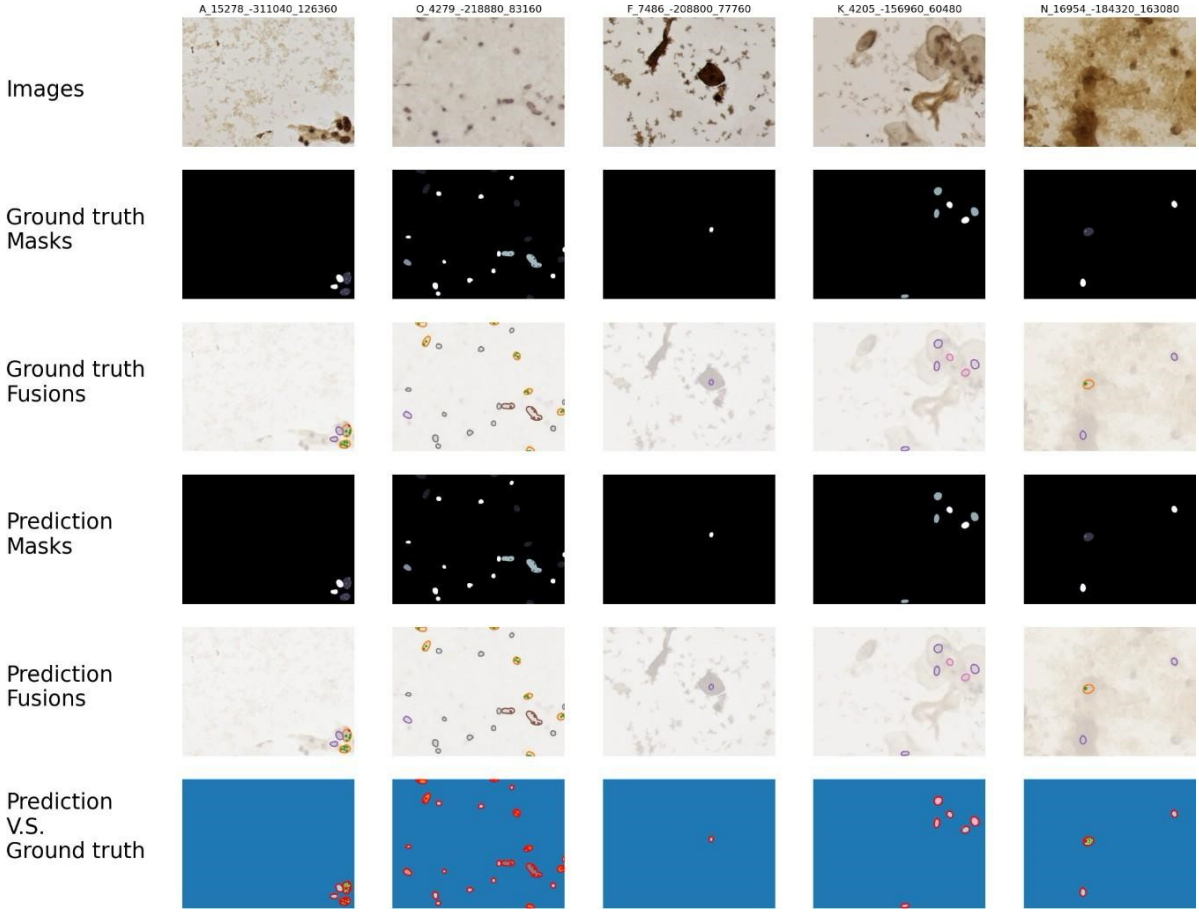
VII. **mAP@[.5:.95:.05]**：Unless otherwise specified, AP(AP is averaged over all categories. Traditionally, this is called “mean average precision” (mAP). We make no distinction between AP and mAP (and likewise AR and mAR) and assume the difference is clear from context.) and AR are averaged over multiple Intersection over Union (IoU) values. Specifically, we use 10 IoU thresholds of .50:.05:.95. This is a break from tradition, where AP is computed at a single IoU of .50 (which corresponds to our metric APIoU=.50). Averaging over IoUs rewards detectors with better localization.

(4) Sample code

U-Net (Can only refer to, **not copy**.) https://github.com/milesial/Pytorch-UNet

(5) Output graphical presentation examples

Please use your implemented network to generate semantic segmentation images of the CCAgT dataset.



Assignment Evaluation:

(1) Code & model performances (60%)

(2) Report (40%)

(3) Bonus (5%)

Please contact TA if you have any questions.

sk774325.ai12@nycu.edu.tw andy90123.ai12@nycu.edu.tw