Selected Topics in Visual Recognition using Deep Learning
Homework 3 announcement

TA: 楊証琨, Jimmy

Ph.D. student at National Taiwan Universitiy

d08922002@csie.ntu.edu.tw

HW2 Reminder

- **Deadline: Nov. 25, 23:59**

1. Finish the competition (Check if your ID on the leaderboard)

Results					
#	User	Entries	Date of Last Entry	Team Name	mAP ▲
1	luluhoooo	2	11/03/21	baseline	0.39199 (1)

2. Benchmark your model on Colab
   - Check the inference code for more details
1. Upload your reports in PDF format to **E3 system**
   - Naming rule: VRDL_HW2_{**STUDENT ID**}_Report.pdf

HW3 Timeline

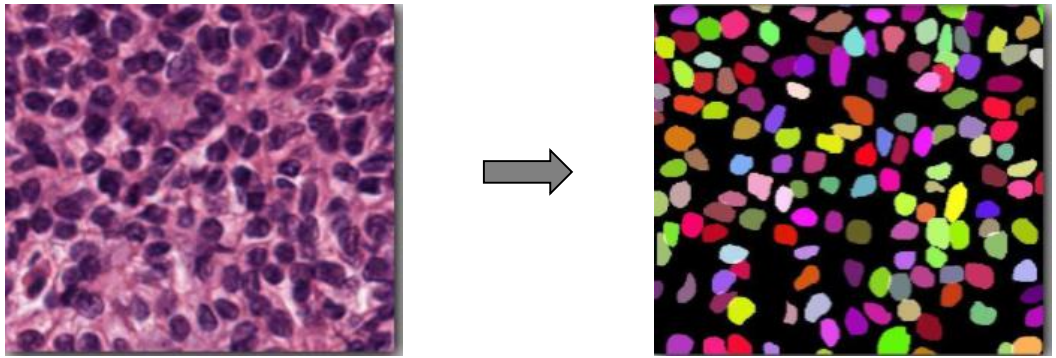
- **Deadline: Dec. 16, 23:59**
  1. Finish the competition (check if your ID on the leaderboard)

Results					
#	User	Entries	Date of Last Entry	Team Name	mAP ▲
1	luluhoooo	2	11/03/21	baseline	0.39199 (1)

  2. Upload your reports in PDF format to **E3 system**
     - Naming rule: VRDL_HW3_{**STUDENT ID**}_Report.pdf

HW3 Introduction: Nuclei segmentation

- Nuclear segmentation dataset contains 24 training images with 14,598 nuclear and 6 test images with 2,360 nuclear
- Train an instance segmentation model to detect and segment all the nuclei in the image
- **No external data should be used!**



CodaLab server is updated

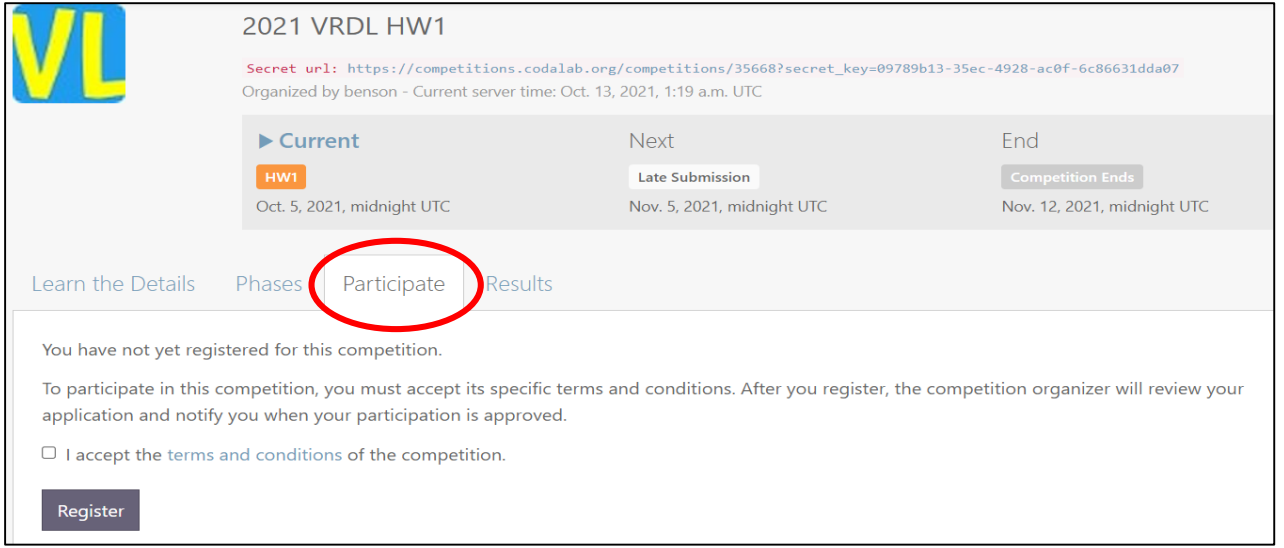
- Users are required to re-create a new account
- **Please sign up again**

11 Sep 2021: Welcome to the new Python3 version of Codalab. It includes a few novel features including: updates of all packages Django and others; possibility to set submission size quotas to keeping storage growth under control; and, a novel admin interface for organizers.

- This is a NEW server, you must re-create your Codalab account if you previously had one.
- To help build our community we created this Google group, please join! You can use it to advertise your own competitions or read news from Codalab's users.
- To report bugs and problems you face, please open Github issues.

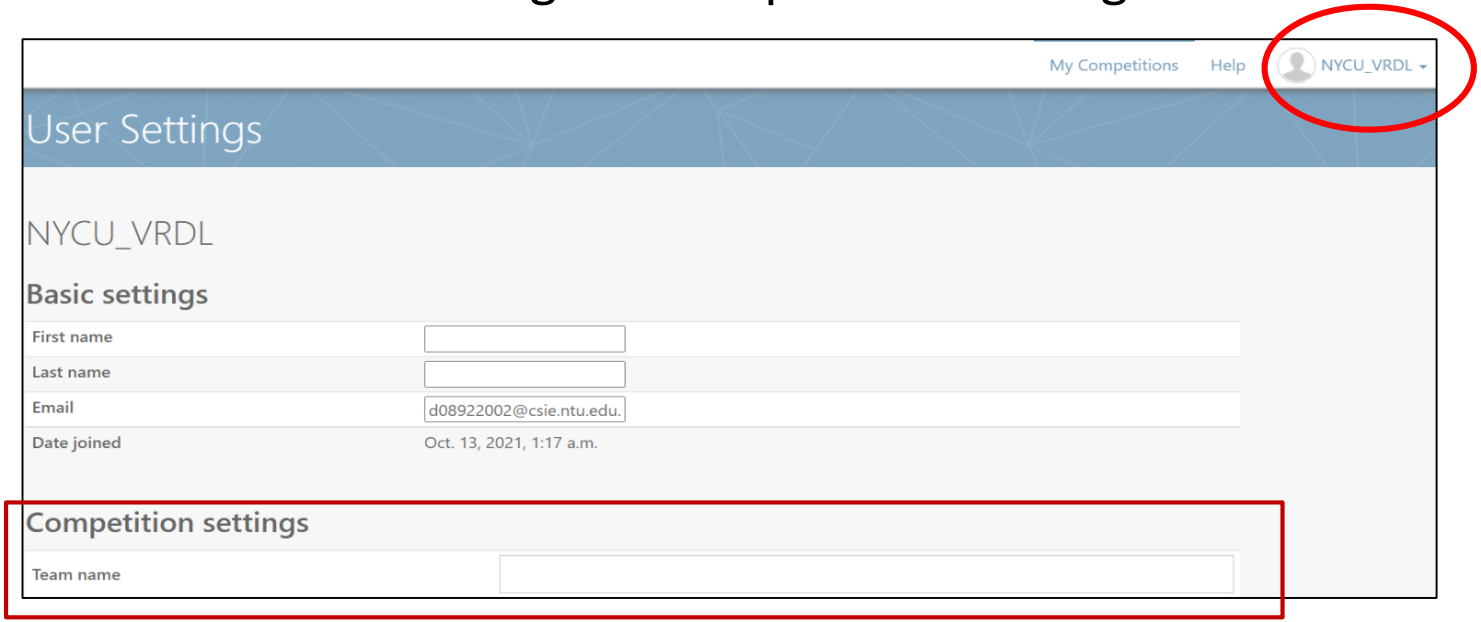
CodaLab competition: Sign in

- Competition link
- Sing in and participate the competition



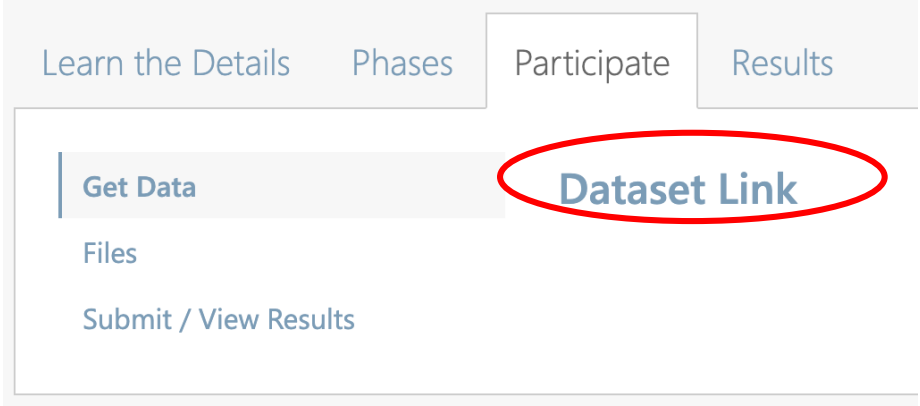
CodaLab competition: Team name

- Change your team name into your **Student ID**!
  - Account -> Settings -> Competition settings -> Team name

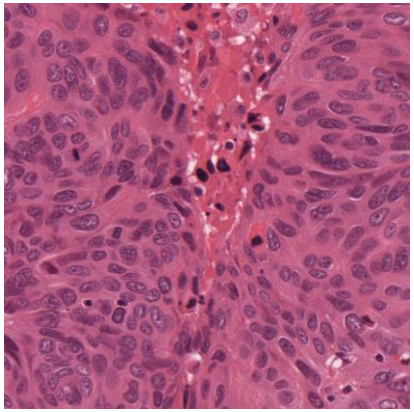


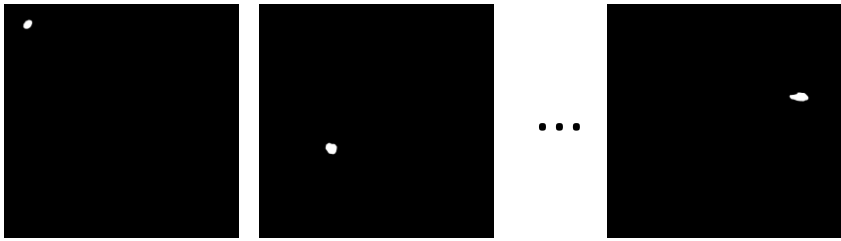
CodaLab competition: Download dataset

- Download the provided dataset
  - Participate -> Get Data
  - Read binary mask of each nuclei in .png
  - Number of .png = number of nuclei



480 nuclei in image

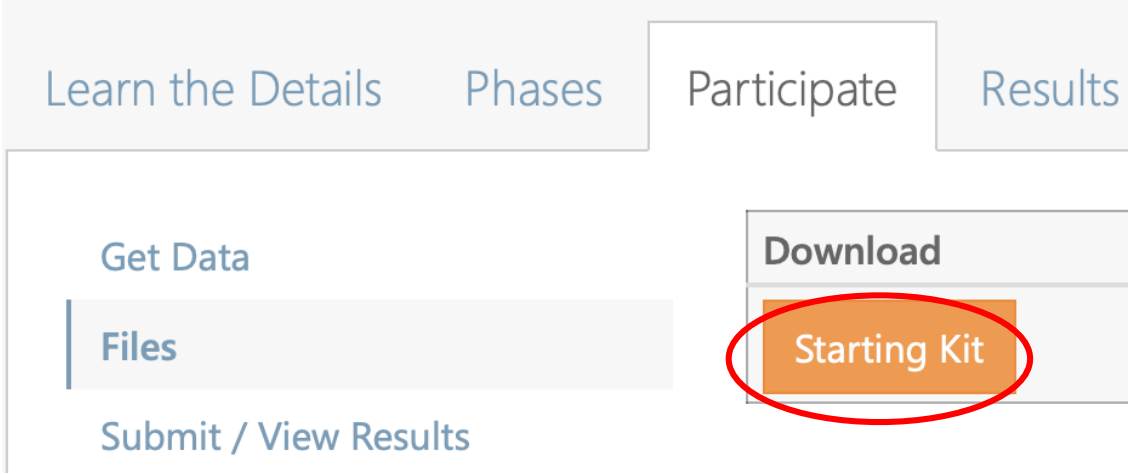




480 masks

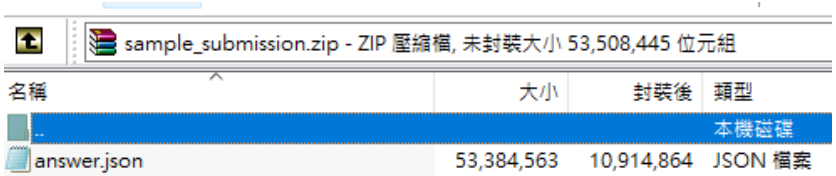
CodaLab competition: Create submission

- We provide a sample submission file (.zip)
  - Files -> Starting Kit



CodaLab competition: Create submission

- The submission file is a single .json file compressed in zip
- The .json file (list of dictionaries) should be named as answer.json
  - Format is the same as COCO resul
  - Set the category_id = 1

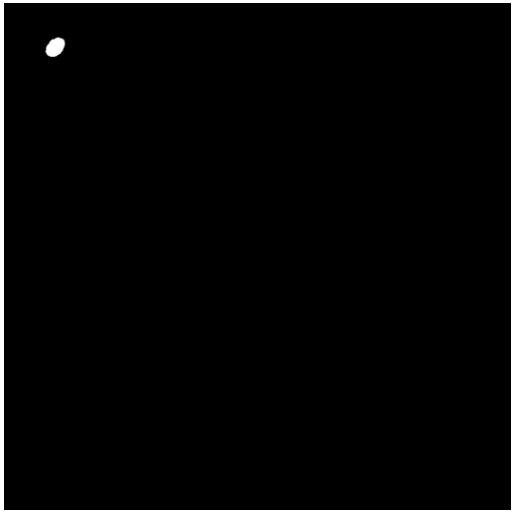



```
[
  {"image_id": 1,
   "bbox": [342.7453918457031,
            123.58514404296875,
            19.428497314453125,
            19.685256958007812],
    "score": 0.9901022911071777,
    "category_id": 1,
    "segmentation":
      {"size": [1000, 1000],
       "counts": "mR_:7nn05M2N1O1O1O100O10000000O101N1O2N2N3Kbm^c
      },

  {"image_id": 1,
   "bbox": [245.0045928955078,
            326.7176818847656,
            24.647659301757812,
            18.889068603515625],
    "score": 0.9896910190582275,
    "category_id": 1,
    "segmentation":
      {"size": [1000, 1000],
       "counts": "fb_76on05M2N101OO1O1O000O10000000000100O100O2O
```

CodaLab competition: Create submission

- The segmentation result should be the RLE encoded format. See the encode function from pycocotools HERE



RLE encode

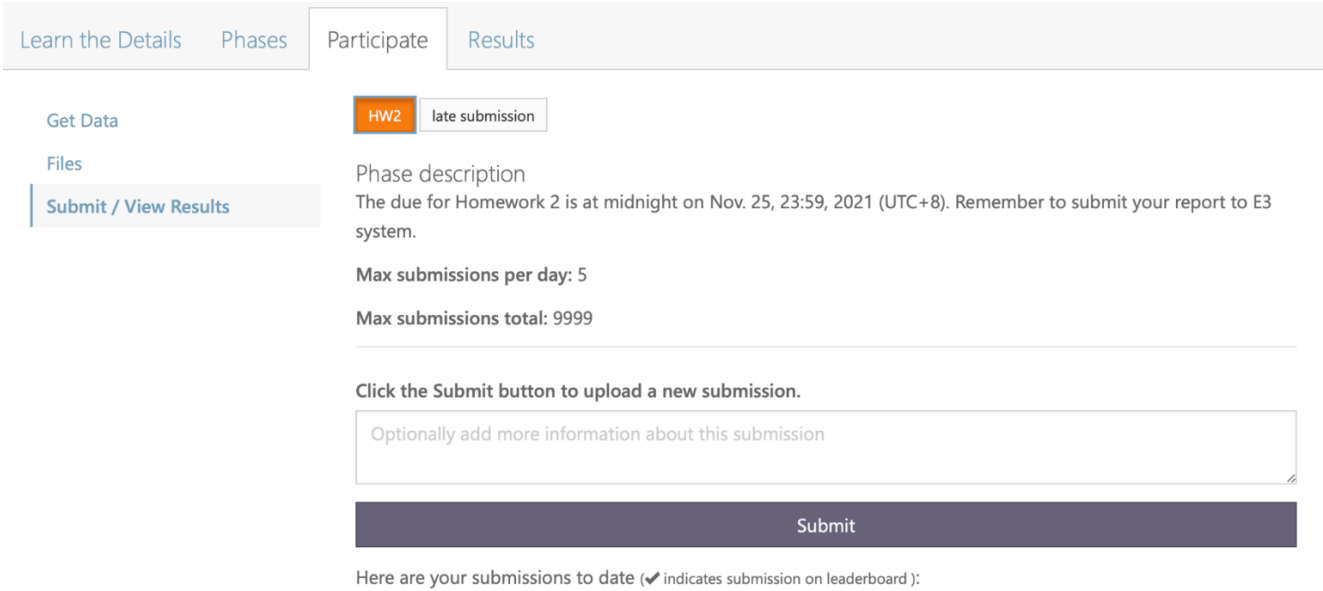
```
{"size": [1000, 1000],
 "counts": "mR_:7nn05M2N1O1O1O1O001000000O101N1O2N2N3Kbm^c0"}
},
```

Put the result under the "segmentation" key

```
"category_id": 1,
"segmentation":
  {"size": [1000, 1000],
   "counts": "mR_:7nn05M2N1O1O1O1O001000000
  },
```

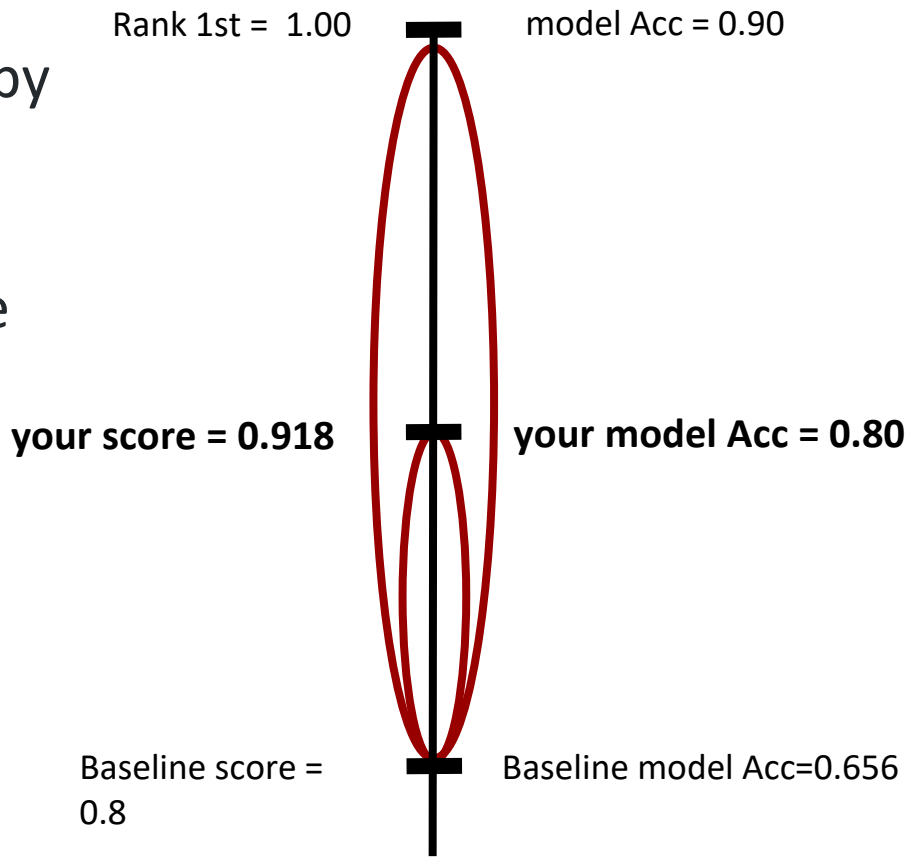
CodaLab competition: Submit results

- Upload your submission and see the performance on Results!
  - Participate -> Submit / View Results



Grading policy: Model performance (70 points)

- Get at least 56 points (70x0.8) by scoring over the baseline
- Your score will be interpolated with the model accuracy by the 1st rank and the baseline



Grading policy: Reports (20 points)

- Document your work (**in PDF**)
  - GitHub/ GitLab link of your code
  - **Reference if you used any code from other resources**
  - Brief introduction
  - Methodology (Data pre-process, Model architecture, Hyperparameters, ...)
  - Table of experiment results
  - Analyze your result
  - Summary
- **Meet requirements above can get 80% of the points (16 points)**

Reports bonus

- Thorough experimental results
- Comprehensive related work survey
- Interesting findings or summary

Good report examples

- Left picture shows the good example of “interesting findings”.
- Right picture shows the good example of “innovative augmentation”.

1. Interesting finding or summary

(1) Different loss function will enhance the robustness

I tried to finetune the first model, that’s the 4th column of the table above, but the model will diverge even though I set a very tiny learning rate. In the opposite, if I finetuned the second model(i.e. the third column of the table) with a tiny learning rate, then the model will keep converging with higher training accuracy as well as robust dev(val) accuracy.(set dev set as training set, and set training set as dev set)

(2) The more complicated data argumentation, the higher accuracy

For example, random erasing will enhance the generalization ability of model.

(3) Transfer learning maybe not enhance the performance of the model

For example, refert o the column7 and 8, in densenet121, I freeze all the layer parameters except for classification layer, and train the model, but the performance of the model is not convincing, maybe that’s because the data source of densenet121 and homework dataset is totally different.

(4) Maybe initial CNN is not the main solution for getting good grades in image classification competition

Recently, many new methods show up, most of them outperformed the initial CNN architecture, such as TransFG[6], Attentive Pairwise Interaction Net(APInet)[7]

(5) Maybe ViT is not suitable for this dataset

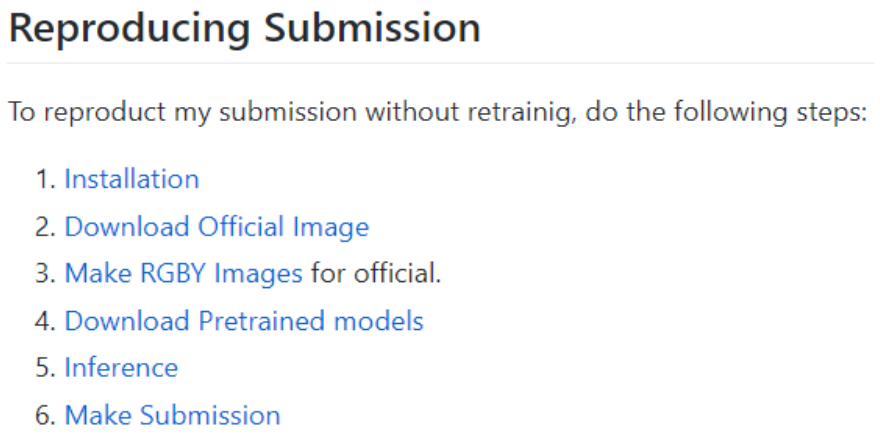
As I just mentioned, ViT is extremely sensitive to change of learning rate, even if the change of learning rate is tiny, you can refer to the last column of the table. In fact, I tried many types of learning rate, for example, I tried 1e-4, but the performance of model is not robust, it will keep changing, it didn’t have the tendency of convergence. And I also tried more smaller learning rate, but the loss will reduce very slowly, as well as accuracy. I think 1e-5 is the best learning rate for me to use. Although 1e-5 is the ‘normal’ learning rate for us to choose, the accuracy will overfit after thousands of epochs. For example, the training accuracy will achieve about 80%, but the testing accuracy only achieved about 20%.

Considering the reliability and the quality of the dataset, I finally cropped all the 6,033 images by using my cropping script : https://github.com/Chialiang86/Bird-Image-Classification/blob/master/crop.py. To crop an image, I only needed to click the left-top and the right-bottom position of the bird image patch. It took me about six hours to crop the whole dataset, and I would show how drastically it influenced the prediction performance on validation set and testing set in later sections.

fig.2 Before cropping (top) and after cropping (bottom) the images by myself

Code readability (10 points)

- Write beautiful Python code with PEP8 guidelines for readability
- Must provide
  - Downloadable link of your model weights on GitHub README
  - A inference.py/.ipynb to reproduce your submission file
- Get only half points of model performance if fail on reproducing your submission



Reproducing Submission

To reproduct my submission without retrainig, do the following steps:

1. Installation
2. Download Official Image
3. Make RGBY Images for official.
4. Download Pretrained models
5. Inference
6. Make Submission

Code readability bonus

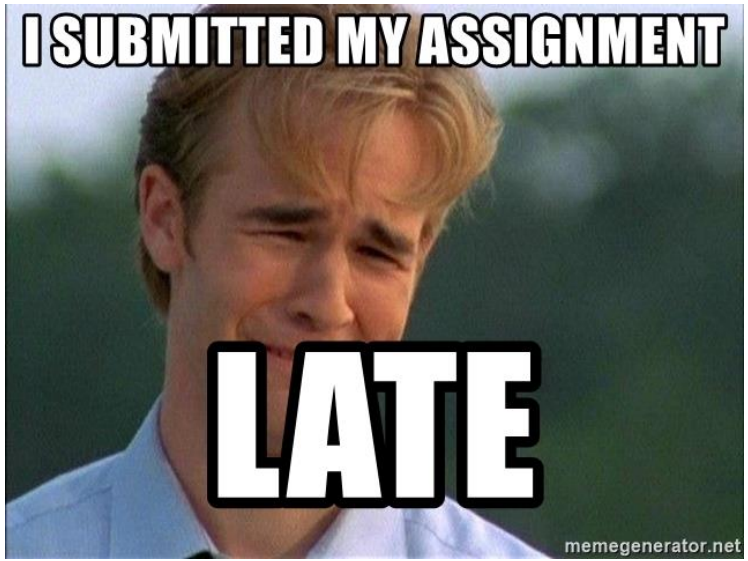
- Clear structure and README of all your steps to reproduce the submission
- Good example: https://github.com/paperswithcode/releasing-research-code

The ML Code Completeness Checklist consists of five items:

1. Specification of dependencies
2. Training code
3. Evaluation code
4. Pre-trained models
5. README file including table of results accompanied by precise commands to run/produce those results

Late policy

- We will deduct a late penalty of 20% per additional late day
- For example, If you get 90% of HW but delay for two days, your will get only 90% - (20% x 2) = 50%!



Keywords

- Beat the baseline
  - Mask R-CNN with proper configuration

- Rank Top 3!
  - Read some new instance segmentation paper from CVPR’2021, ICCV’2021 and try to implement it!
  - Read some useful tips for nuclei segmentation on Kaggle discussion forum

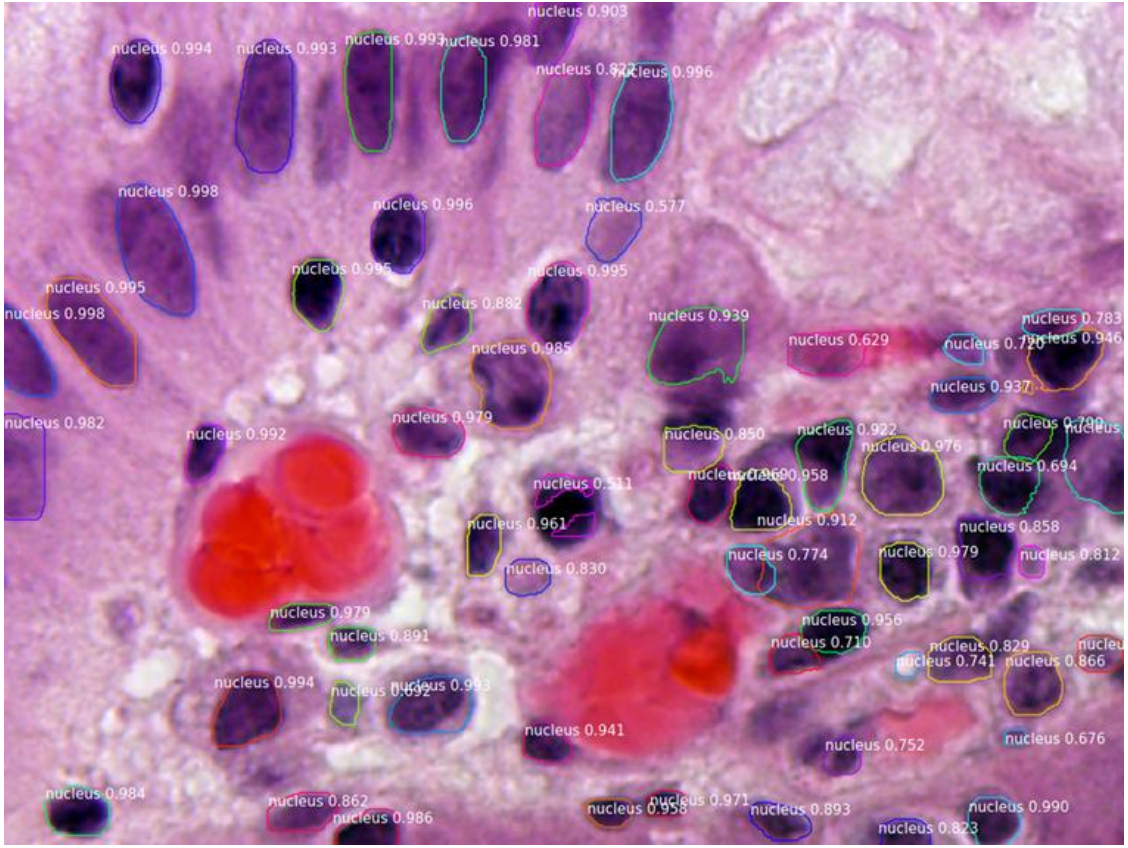
FAQ

- Can I use any code/tools/Library from GitHub or other resources?
  - Yes! We encourage you to learn how to apply existing tools on your own task, such as Keras- MaskRCNN, Pytorch-mmdetection,
    DO NOT copy code from your classmate!
  - Pre-trained model is **usable** for this homework
- Why my testing results are so bad?
  - If you have done any image translation (resize, padding), you will need to transfer the coordinates into original image dimension
- How to filter the overlapping predicitions?
  - Set a proper parameter for non-maximum suppression

Notice

- Check your email regularly, we will mail you if there are any updates or problems of the homework
- If you have any questions or comments for the homework, feel free to mail me and cc Prof. Lin or post it on E3 forum
  - Prof. Lin: lin@cs.nctu.edu.tw
  - TA Jimmy: d08922002@csie.ntu.edu.tw
  - TA 柏聲: bensonliu0904@gmail.com
  - TA 晨軒: derekt.cs06@nctu.edu.tw
  - TA 政儒: ace52751208@gmail.com

Have fun!


nucleus 0.903
nucleus 0.994
nucleus 0.993
nucleus 0.993
nucleus 0.981
nucleus 0.996
nucleus 0.998
nucleus 0.996
nucleus 0.577
nucleus 0.995
nucleus 0.995
nucleus 0.995
nucleus 0.998
nucleus 0.882
nucleus 0.939
nucleus 0.985
nucleus 0.629
nucleus 0.783
nucleus 0.946
nucleus 0.937
nucleus 0.982
nucleus 0.992
nucleus 0.979
nucleus 0.850
nucleus 0.922
nucleus 0.976
nucleus 0.694
nucleus 0.958
nucleus 0.511
nucleus 0.961
nucleus 0.912
nucleus 0.858
nucleus 0.774
nucleus 0.979
nucleus 0.812
nucleus 0.830
nucleus 0.979
nucleus 0.956
nucleus 0.891
nucleus 0.710
nucleus 0.829
nucleus 0.741
nucleus 0.866
nucleus 0.994
nucleus 0.941
nucleus 0.752
nucleus 0.676
nucleus 0.984
nucleus 0.862
nucleus 0.986
nucleus 0.971
nucleus 0.893
nucleus 0.990
nucleus 0.823

Selected Topics in Visual Recognition using Deep Learning
Preview the final project

TA: 楊証琨, Jimmy

Ph.D. student at National Taiwan Universitiy

d08922002@csie.ntu.edu.tw

Final project: Join a real-world competition

- Team up
  - **3 persons per team**
  - We will randomly match up if you can not form a team
- Select one of the provided competitions by your team
- Prepare presentation and submit the reports

Final project: Join a real-world competition

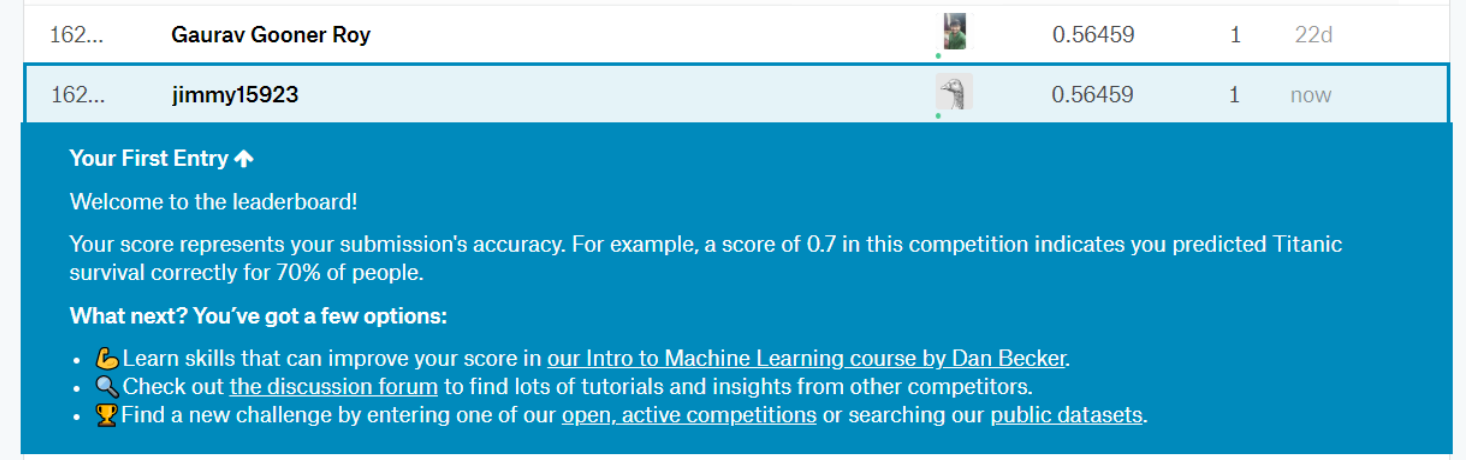
1. Beat the baseline
2. Make a 10-min presentation in English of your methodology and have a 2-min Q&A session
3. Upload your report and slides for a team (one report for each team)

Tasks of competition

- The competition used in last year, only for reference
  - APTOS 2019 Blindness Detection: Image classficiation
  - Handwritten Grapheme Classification: Image classficiation
  - Global Wheat Detection: Object detection
  - Severstal: Steel Defect Detection: Semantic segmentation

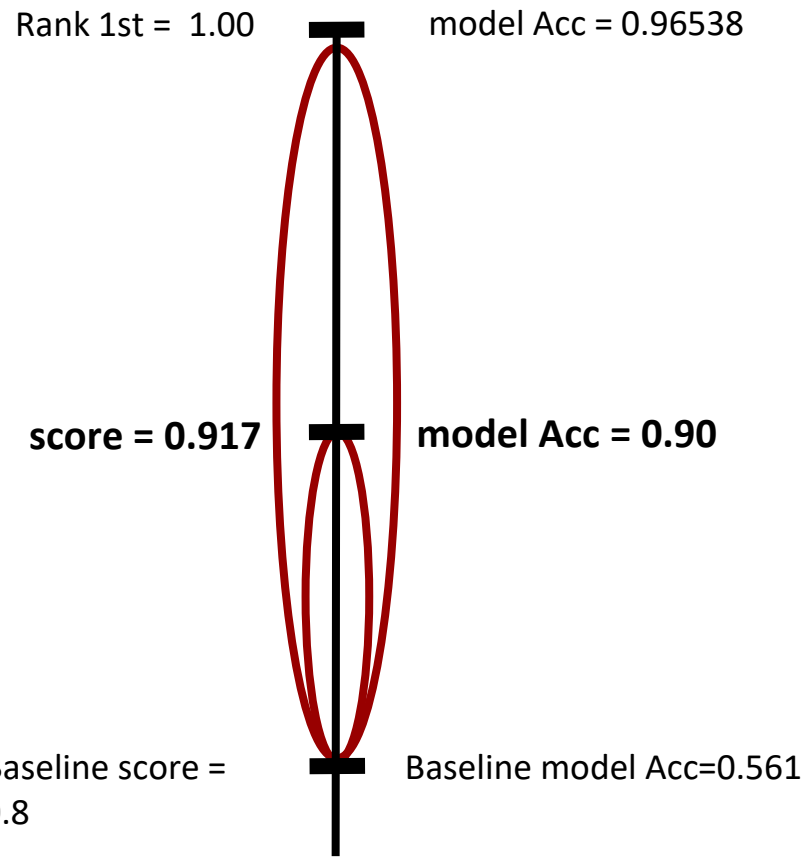
Grading policy

- Model performance: **40 points**
  - Screenshot your rank and post it in the reports
- Presentation & Report & Code: **50 points**
- Teammate contribution: **10 points**



Grading policy: Model performance (40 points)

- Your score will be interpolated with the model accuracy by rank 1th and baseline



Grading policy: Presentation (50 points)

- Your presentation/reports should include
  - GitHub/ GitLab link of your code
  - Introduction
  - Related work
  - Proposed approach
  - Experiment results
  - Conclusion
- Meet all requirements can get 80% points (40 points)

Grading policy: Contribution (10 points)

- Specify the teamwork of each task from your team **in the reports**

Tasks	**contributors (%)**
Literature survey	0856065 (100%)
Approach design	0856078 (50%), 0856605 (50%)
Approach implementation (experiment)	0856078 (30%), 0856605 (70%)
Report writing	0856065 (80%), 0856078 (20%)
Slide making and oral presentation	0856605 (33%), 0856065 (33%), 0856078 (33%)