Selected Topics in Visual Recognition using Deep Learning
Homework 2 announcement

TA: 楊証琨, Jimmy

Ph.D. student at National Taiwan Universitiy

d08922002@csie.ntu.edu.tw

HW1 Reminder

- **Deadline: Nov. 4, 23:59**
  1. Finish the competition (your ID on the leaderboard)

Results					
#	User	Entries	Date of Last Entry	Team Name	Accuracy ▲
1	ChenHsuanTai	1	10/07/21	baseline	0.65579 (1)

1. Upload your reports in PDF format to E3 system
- **Naming rule: VRDL_HW1_{STUDENT ID}_Report.pdf**
- Deduct 3 points if the file naming is wrong

HW2 Timeline

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

1. Finish the competition (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)

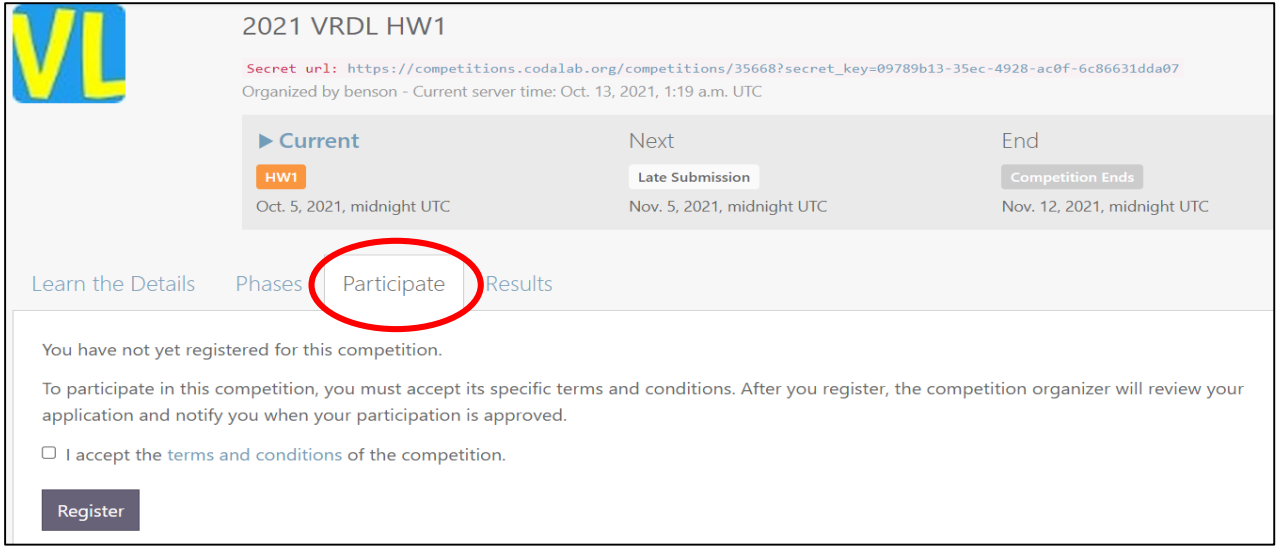
2. Benchmark your model on Colab
   - Check the inference code for more details
1. Upload your reports in PDF format to **E3 system**
   - Naming rule: VRDL_HW2_{**STUDENT ID**}_Report.pdf

HW2 Introduction: Street View House Numbers detection

- SVHN dataset contains 33,402 trianing images, 13,068 test images
- Train a not only **accurate** but **fast** digit detector!

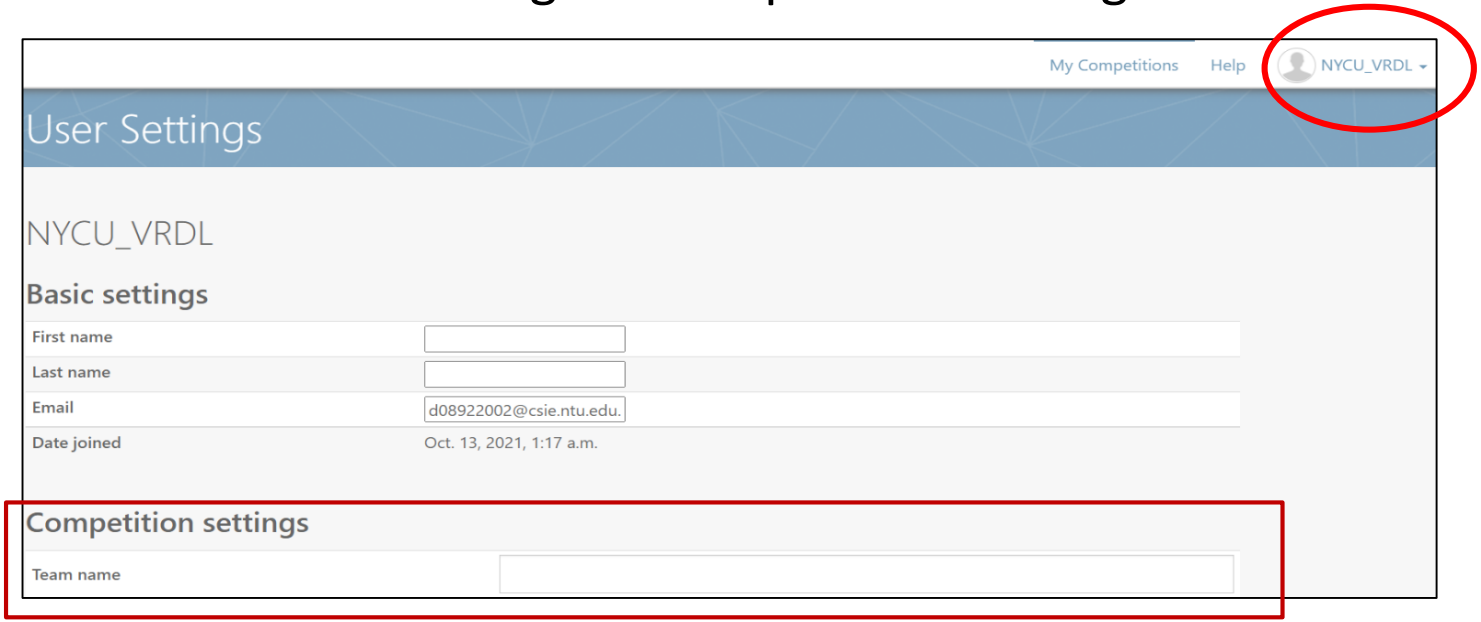
CodaLab competition: Sign In

- Competition link
- Sing in and participate the competition



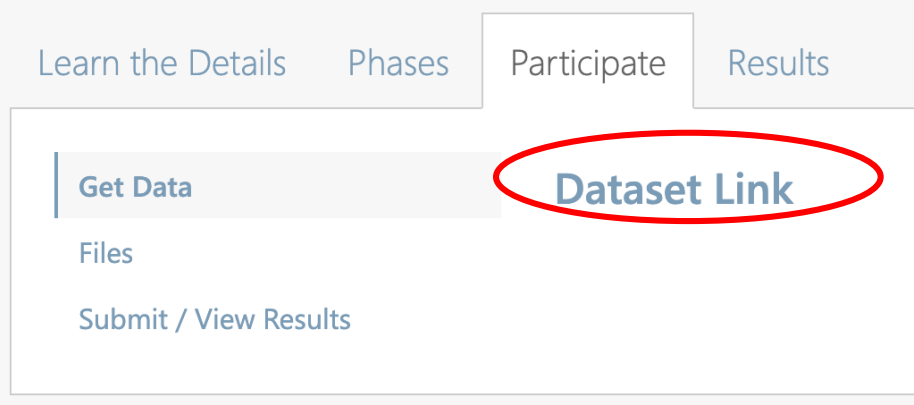
CodaLab competition: Team name

- 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 the .mat file
  - label 10 represent digit “0”



Variables – digitStruct(137).bbox

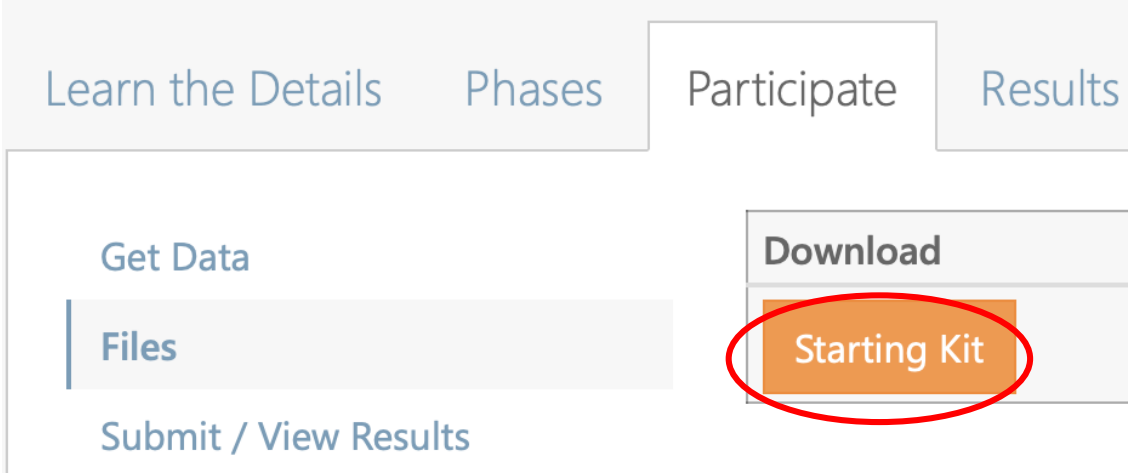
digitStruct | digitStruct(137).bbox

digitStruct(137).bbox

Fields	height	left	top	width	label
1	43	50	10	14	1
2	43	70	10	17	10

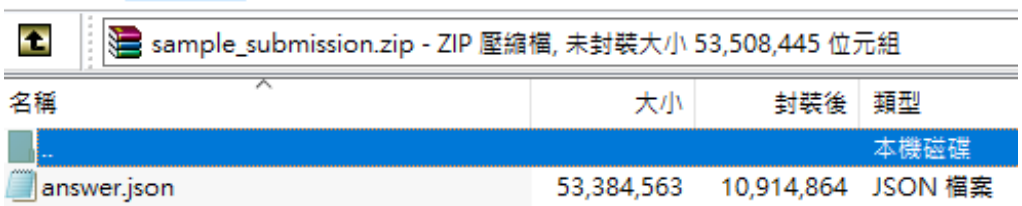
CodaLab competition: Create submission

- We provide a sample submission file (.zip) and the sample code to generate the sample submission
  - Files -> Starting Kit



CodaLab competition: Create submission

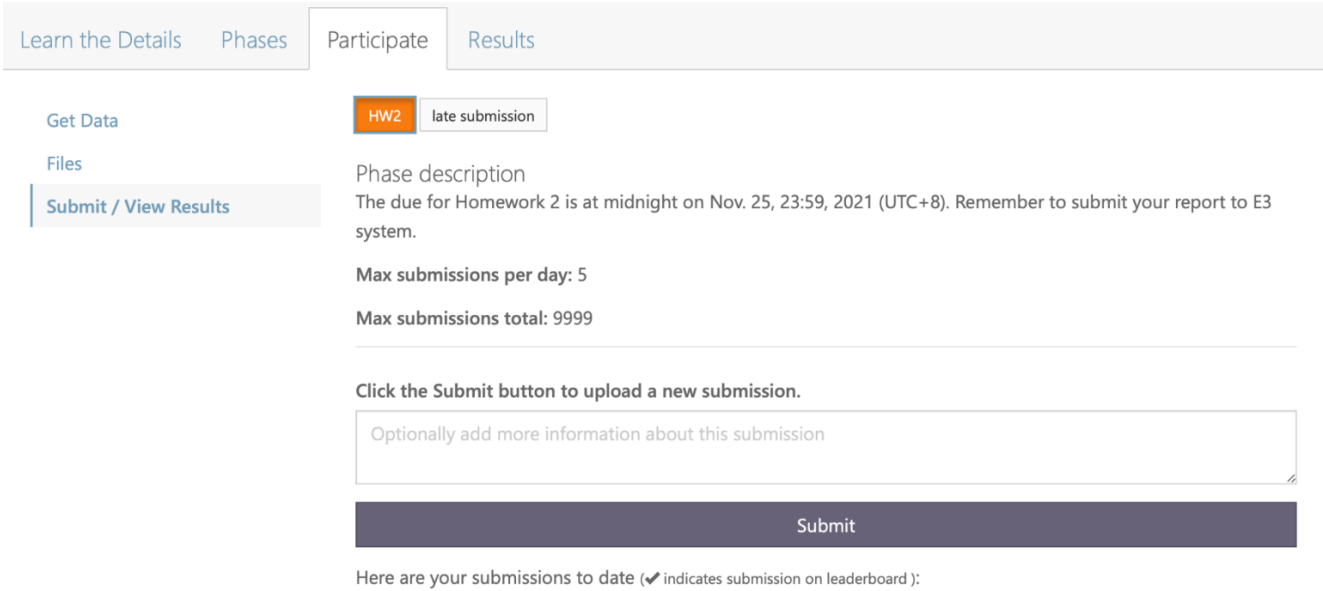
- The submission file is a single .json file compressed in zip
- The .json file (list of dictionaries) should be named as answer.json
  - Format is same as COCO results
  - Digit “0” -> category_id: 0




```
{
    "image_id": 11918,
    "bbox": [
        101.51773834228516,
        33.335304260253906,
        2.8105697631835938,
        14.105010986328125
    ],
    "score": 0.05053979158401489,
    "category_id": 0
},
{
    "image_id": 11918,
    "bbox": [
        93.37824249267578,
        31.939891815185547,
        7.4753875732421875,
        16.854549407958984
    ],
    "score": 0.5855768322944641,
    "category_id": 1
},
```

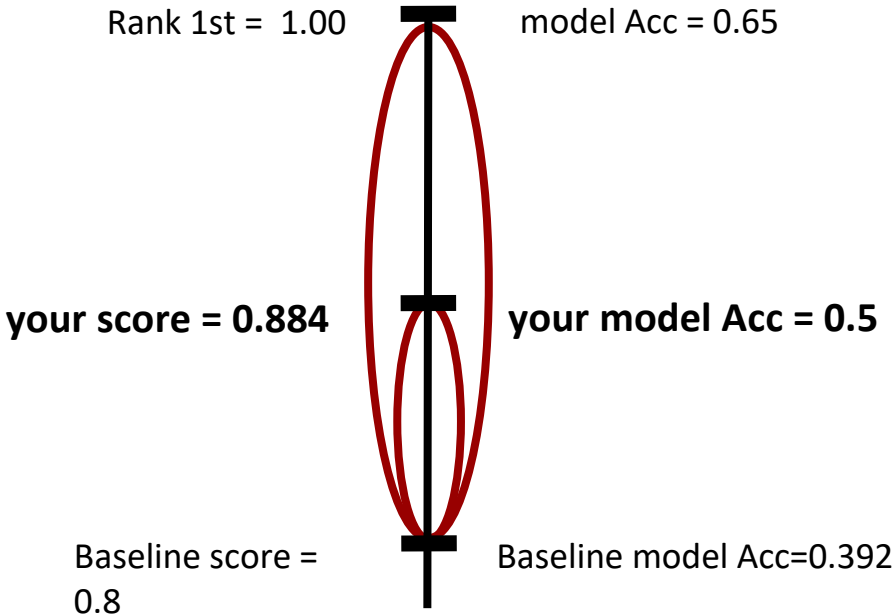
CodaLab competition: Submit results

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



Grading policy: Model performance (70 points)

- 50 points for the accuracy ranking
- 20 points for the speed benchmark ranking
- Pass the each baseline will get 80% of that points



Benchmark your detection model

- Develop efficient and accurate detection model
- Practice employing your model on new environment (**Colab**)
  a. Git clone your project code
  b. Install required packages
  c. Download test-set and your model weights
  d. Run inference and benchmark
  e. Screenshot your benchmark results and generate submissions
- See the inference.ipynb for detail instructions
  - Make a copy of this notebook and modify it in your own

Benchmark your detection model

- Screenshot your benchmark result and patse it on your Report
- Provide the notebook link on your GitHub README.md

```
start_time = time.time()
for img in tqdm(test_img_list):
    # your model prediction
    pred = inference_detector(model, img)

end_time  = time.time()
print("\nInference time per image: ", (end_time - start_time) / len(test_img_list))

# Remember to screenshot!
```

```
  0%|          | 0/100 [00:00<?, ?it/s]/content/mmdetection/mmdet/datasets/utils.py:69: UserWarning: "ImageToTensor"
  'data pipeline in your config file.', UserWarning)
100%|██████████| 100/100 [00:29<00:00,  3.35it/s]
Inference time per image:  0.29895930528640746
```

Grading policy: Reports (20 points)

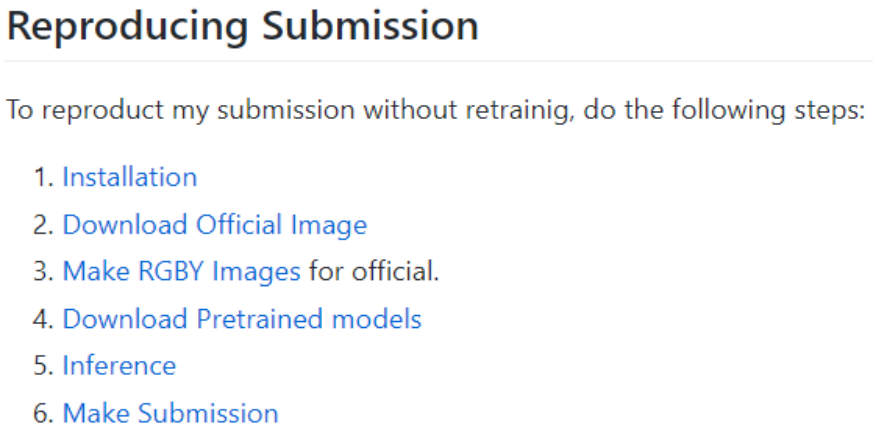
- Document your work (**in PDF**)
  - GitHub/ GitLab link of your code
  - **Reference if you used any code from other resources**
  - Brief introduction
  - Methodology (Data pre-process, Model architecture, Hyperparameters, ...)
  - Summary
- **Meet requirements above can get 80% of the points (16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

Reports bonus

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

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

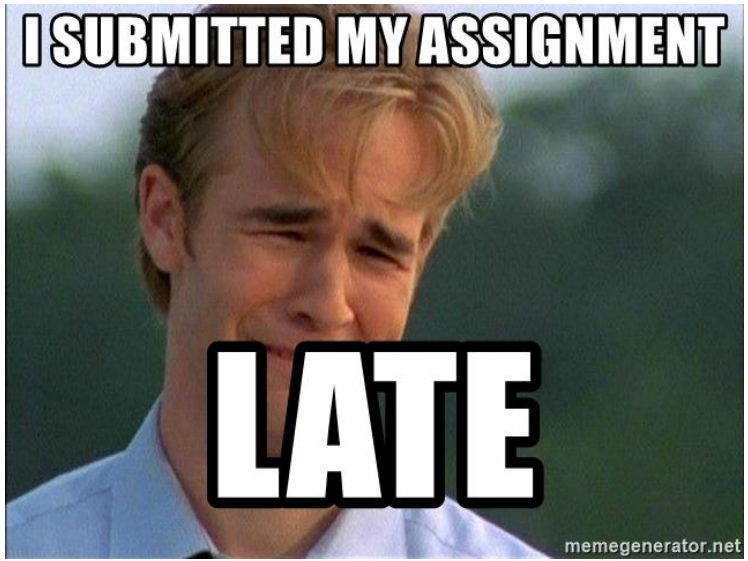
- Clear structure and README of all your steps to reproduce the submission
- Good example: https://github.com/paperswithcode/releasing-research-code

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
  - YOLO, Retina-Net, **Faster RCNN**

- Rank Top 3!
  - Read some new object detection paper from CVPR'2021, ICCV'2021 and try to implement it!

FAQ

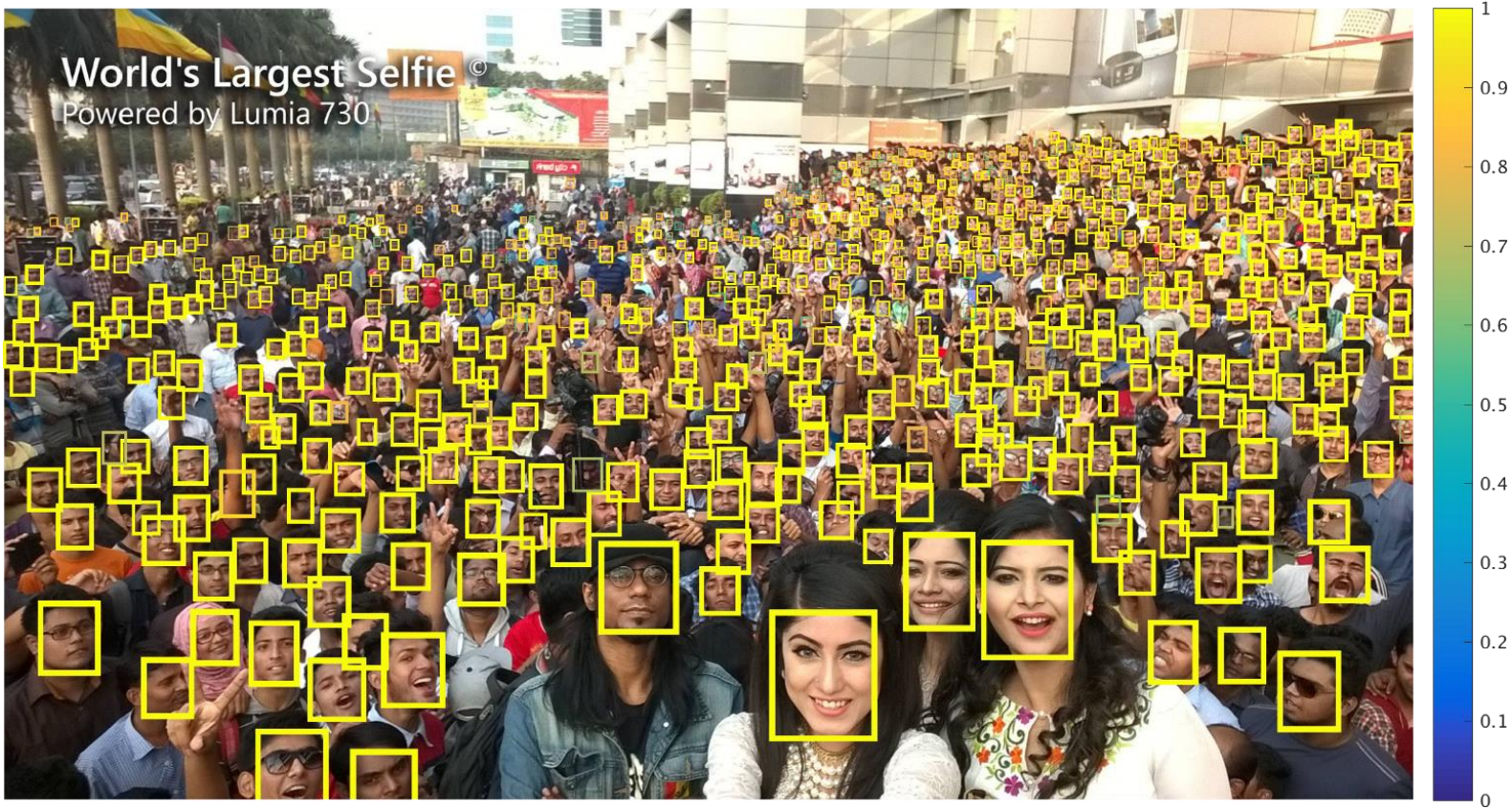
- Can I use any code/tools/Library from GitHub or other resources?
  - Yes! We encourage you to learn how to apply existing tools on your own task, such as Keras-Retinanet, Pytorch-mmdetection, TF-object-detection-API

    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 do I set the score threshold for box predictions?
  - Figure out how mAP is computed, you will get the answer!

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!


World's Largest Selfie ©
Powered by Lumia 730
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0