Final Project

DLCV Fall 2021

Important Deadlines (tentative)

- **Poster Submission Deadline**

  2022/1/14 13:00 GMT+8

- **Kaggle/Eval.ai Challenge Deadline**

  2022/1/17 23:59 GMT+8

- **Poster Presentation**

  2022/1/18 1-5pm GMT+8

- **GitHub Commit Deadline**

  2022/1/18 23:59 GMT+8

Introduction

1. About the final project
2. Challenges
   a. **Challenge 1** Medical images - skull fracture detection
   b. **Challenge 2** Medical images - nodule detection
   c. **Challenge 3** Fine-grained long-tailed food classification

About Final Project

- GitHub
- Kaggle / EvalAI
- Poster
- Presentation
- Code / Report*

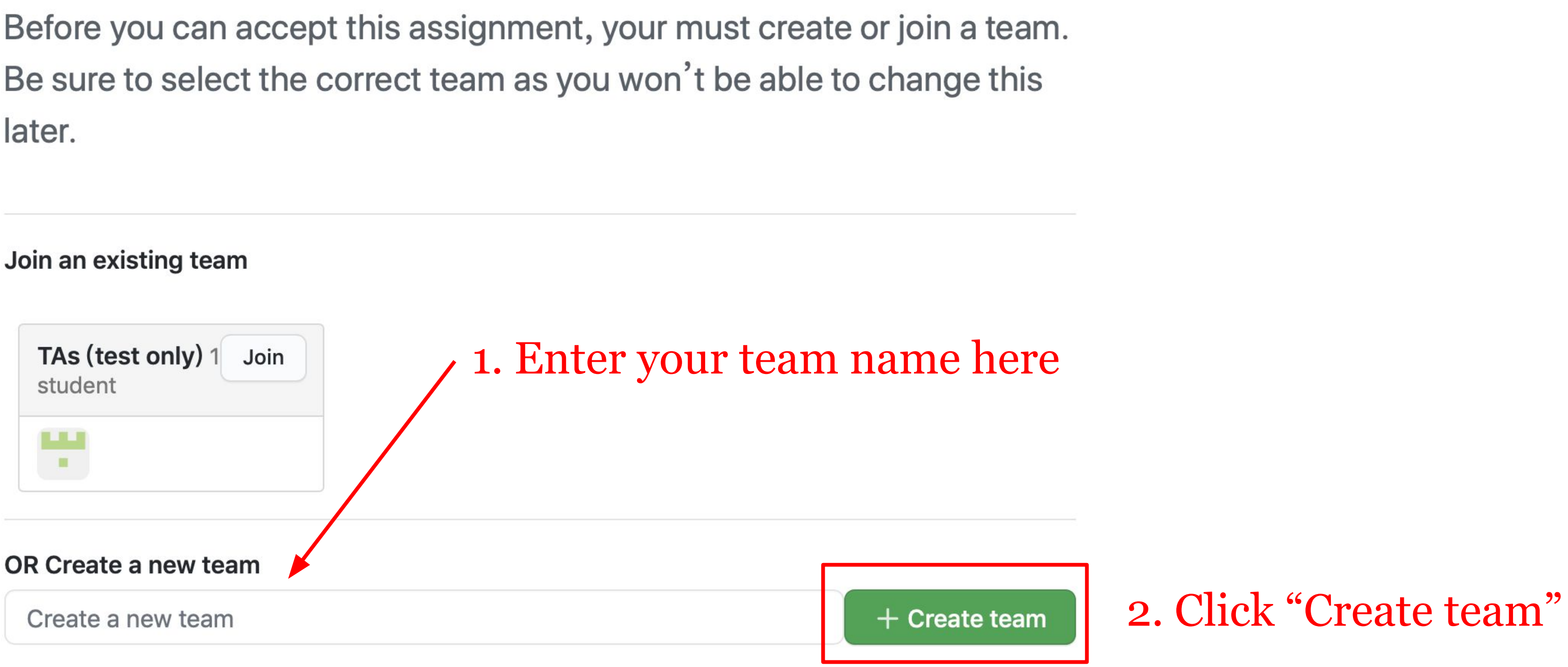
- Grading
- Intra/Inter-Team Evaluation
- Rules
- Challenge Awards (please refer to the “Award” part in each challenge)

GitHub

- Each team needs to come up with a team name...
  - English alphabets and numbers only
  - You need to use **the same team name for Kaggle/Eval.ai.**
- The team leader **creates** the team first,

  and the team members **join** the team afterwards.

If you are a **team leader**,



Before you can accept this assignment, your must create or join a team. Be sure to select the correct team as you won’t be able to change this later.

Join an existing team

TAs (test only) 1 student

Join

OR Create a new team

Create a new team

+ Create team

1. Enter your team name here

2. Click “Create team”

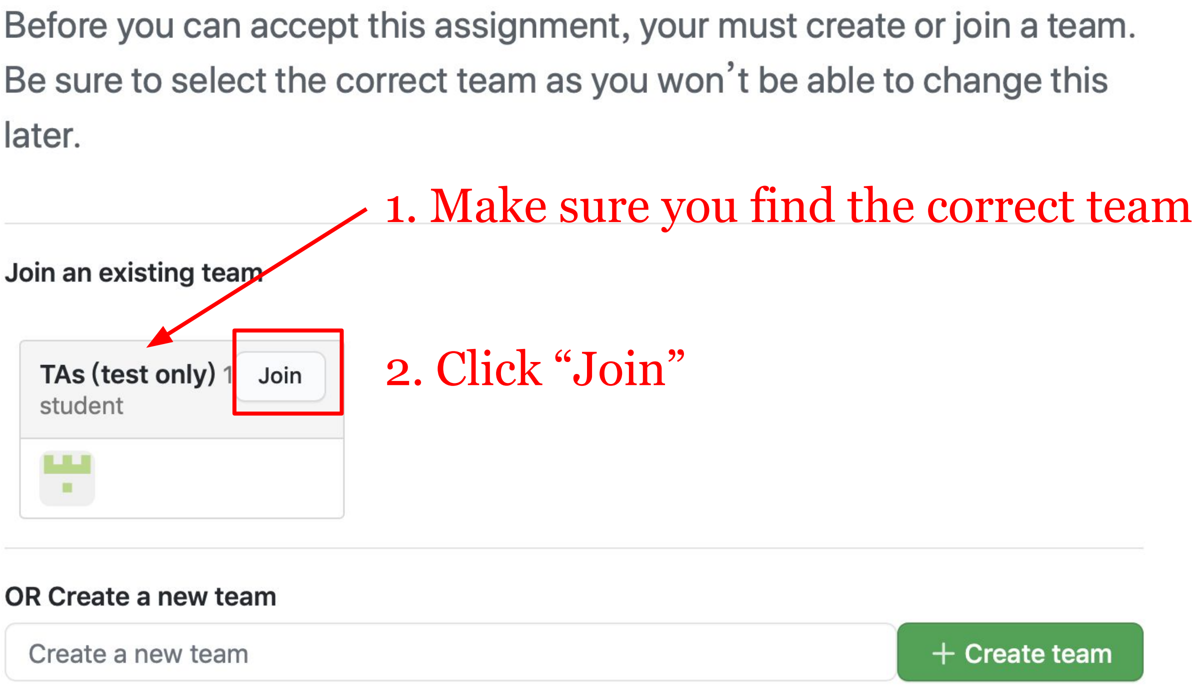
If you are a **team member**,

WARNING

Do **not** join any team unless your team leader has already created the team.

Do **not** join a team without asking its leader or members.

Do **not** join the TA team.



Before you can accept this assignment, your must create or join a team. Be sure to select the correct team as you won't be able to change this later.

Join an existing team

TAs (test only) 1 student

OR Create a new team

1. Make sure you find the correct team

2. Click "Join"

Kaggle / Eval.ai

- You need to participate the kaggle / Eval.ai challenge and compete with your classmates!
- Maximum Daily Submissions: **5** times (for each team)
    - Kaggle: Reset at 08:00 AM (GMT+8) every day
    - Eval.AI: Reset at 00:00 AM (GMT+8) every day
- Challenge deadline: 2022/1/17 23:59 GMT+8 (tentative)

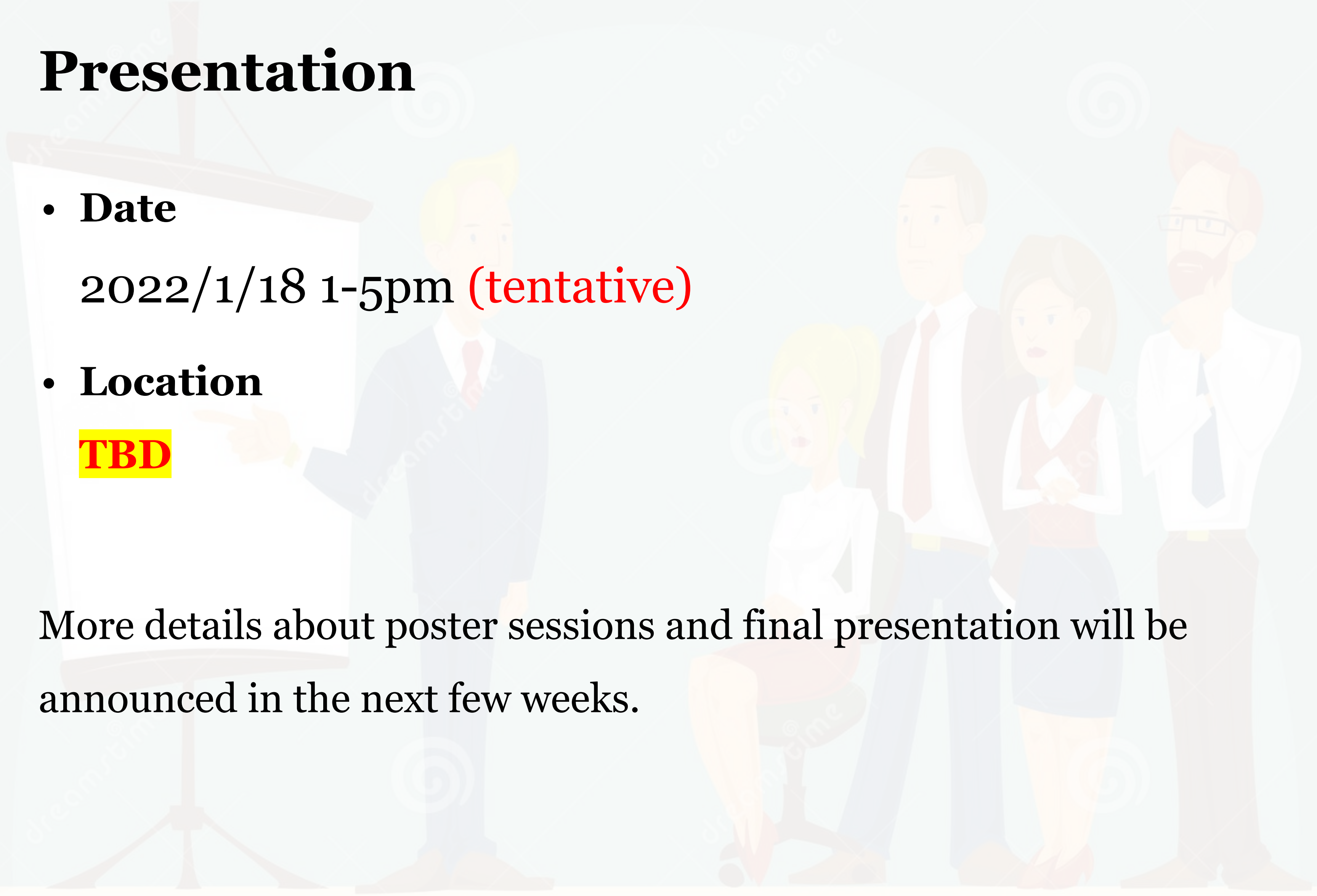
Poster

Format : A1 (Portrait, 84.1 cm x 59.4 cm)

Submission:

- You should convert your file into **PDF** format for submission.
- The poster should be submitted in **the team's Github repository** (should be in the format **poster.pdf**).
- You may leave some blank areas on your poster (e.g., recognition rates) and fill them up before the final presentation.
- If you fail to submit your poster before the submission deadline, you need to print it out by yourself (i.e., using your own budget).

Presentation



- **Date**

  2022/1/18 1-5pm (tentative)

- **Location**

  TBD

More details about poster sessions and final presentation will be announced in the next few weeks.

Code/Report*

- The code should be submitted in **the team's Github repository**.
- You need to modify the "How to run your code?" part of the **README** file, so that TAs will be able to reproduce your results.
- If TAs cannot reproduce your results, you will receive **0** points in the code part. (unless minor errors)

Code/Report*

- For the top 3 teams selected for cash awards,

  an additional technical report needs to be submitted to Github as well

  (by 2022/1/25 11:59 GMT+8 (tentative); single-column, 4 pages maximum)
    - Approach (e.g., data preprocessing, model architecture, implementation details, hyperparameter choices, etc.)
    - Experiments
    - A template will be provided.

Grading

- **Final 35%** (Bonus up to 3%)
  - **Code / Kaggle / Eval.ai 10%**

    Kaggle / Eval.ai for references, final accuracy evaluated by TAs
    - Please refer to the “Grading" section in each challenge for more details.
  - **Approach & Presentation 25%**
    - Novelty and technical contribution 10%
    - Completeness of experiments 10%

      (e.g., comparisons to baseline and recent models, ablation studies, visualization, etc.)
    - Presentation (Oral + Poster) 5% + bonus up to 3% (top 3 teams voted by class)

Intra/Inter-Team Evaluation

- **Intra-Team Evaluation**
  - You must participate and work with your team member.
  - In the past few semesters,

    we did assign different scores for team members within the same team.

- **Inter-Team Evaluation**
  - The most voted team from each challenge will receive bonus points.
  - Will provide an online form during final presentation (one form per team)

Rules

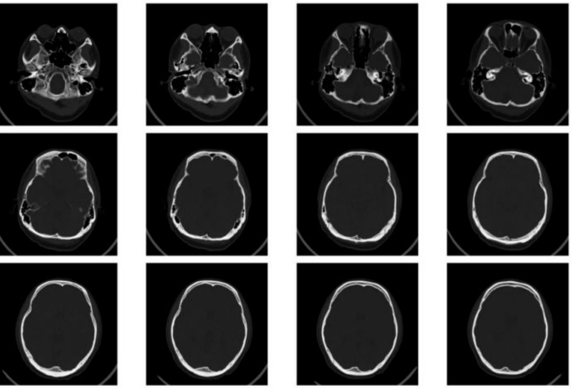
- Please submit your final project to the GitHub repository by the deadline. You will **not have any late days**, so **late submission is not allowed**. Besides, **any mistake or violation will result in severe penalty** (which means your results are questionable).
- For each team, if no members show up for the final presentation, all team members will receive 0 points for this part (0 out of 25 points).

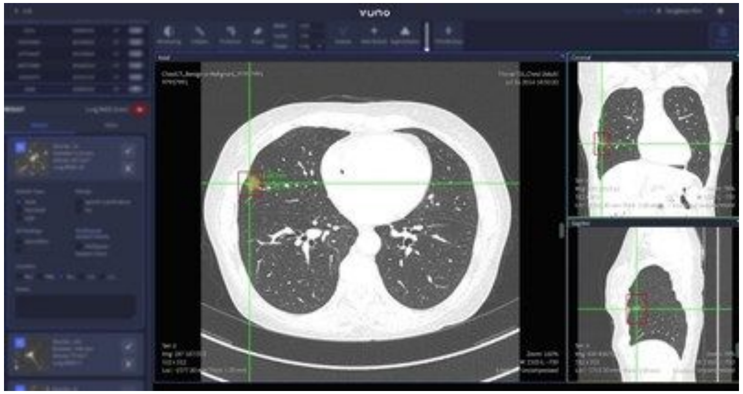
Tasks

1. **Challenge 1** Skull fracture detection

2. **Challenge 2** Nodule detection

3. **Challenge 3** Fine-grained long-tailed food classification





Challenge I
Skull fracture detection

Deep01
Artificial Intelligence

Sponsered by Deep01

Outline

- Introduction
- Challenges
- Dataset
- Evaluation
- Grading
- Challenge Awards
- Rules

Introduction

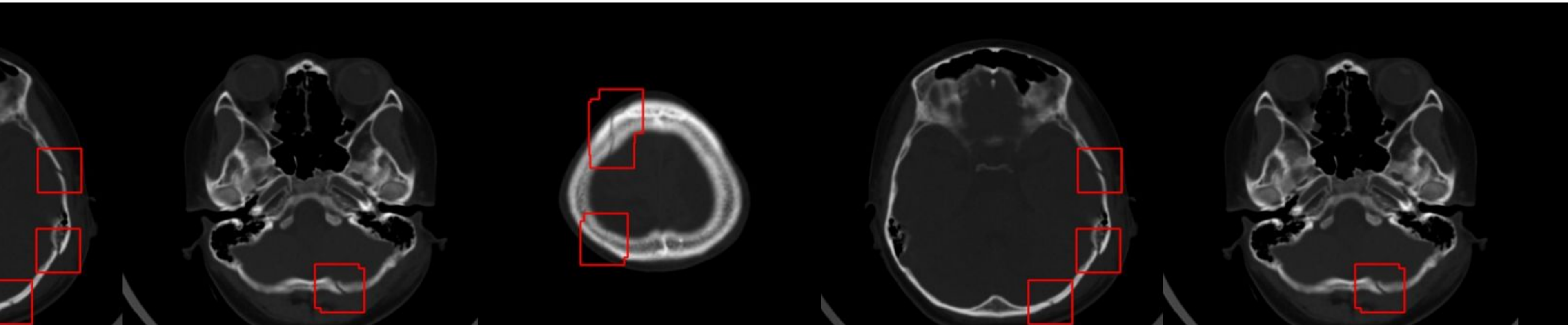
- Dataset

  Computer tomography (電腦斷層影像) of different patients with skull Fractures (腦部骨折)

- Goal

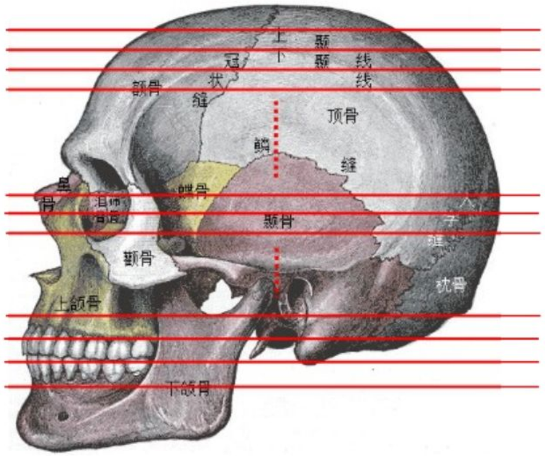
  Train a neural network to **detect** whether the patient has skull fractures and where they are by utilizing data annotated by physicians.

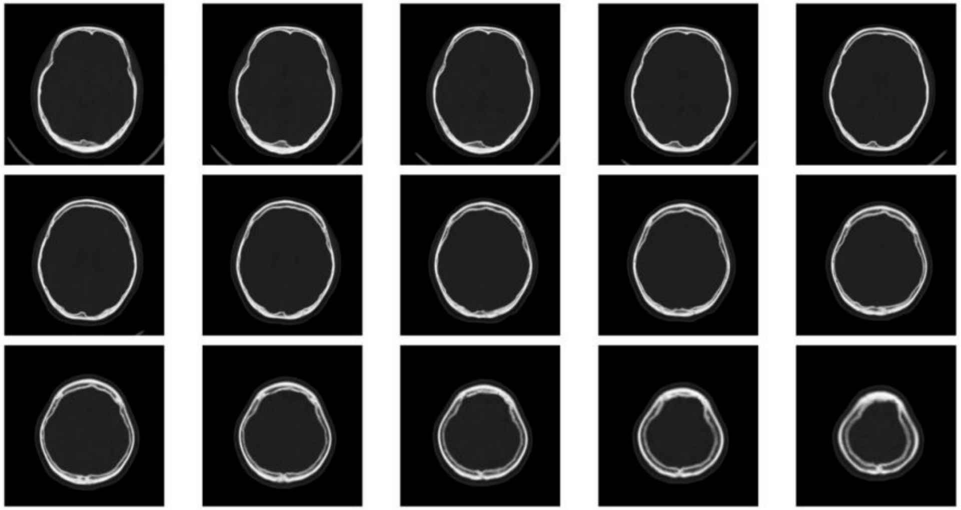
- Task **Object detection**



Challenges - Things are not that easy!

- Medical images are quite different from those in COCO or Visual Genome datasets. (e.g., RGB images)
- Some cases of skull fracture are very similar to the normal skull sutures (non-fractures) in the CT imaging.
- CT images can be viewed as sequential data:
    - If you treat each frame as independent inputs, the performance may be limited.





Dataset

- Training set (with label)
    - 32,665 images in total, with 1,116 study series (patients)
- Testing set (without label)
    - 3,775 images in total, with 130 study series (patients)
- Download link

Dataset -- File Structure

```
skull/
 ├── train/
 │   ├── 醫院_Study_Series/      training dataset
 │   │   ├── 醫院_Study_Series_SliceID.npy
 │   │   └── ...
 │   └── .../
 ├── test/                       testing dataset
 │   ├── 醫院_Study_Series/
 │   │   ├── 醫院_Study_Series_SliceID.npy
 │   │   └── ...
 │   └── .../
 │
 ├── readme.txt                 dataset introduction
 ├── records_train_readme.txt annotation introduction
 └── records_train.json         training data annotation
```

Evaluation

- **Case-Level Accuracy**: identify if the patient has skull fractures
- **Centroid-Level F1 Score**: detect where the skull fractures are in each CT slice

Patient ID	CT Slice ID	**Skull Fracture Coordinates (if any)**	**Label Annotation in the dataset**	**Case-Level Ground Truth**
001	1		**-1** (no fracture in this slice)	**1 (positive)**
	2	**(12, 34), (56, 78)**	**1**	
	3	**(23, 45)**	**1**	
	4		**-1** (no fracture in this slice)	
002	1		**0**	**0 (negative)**
	2		**0**	

Your code should output predicted label (-1, 0, 1) and coordinates for each CT slice, and we have provided the code that can calculate the case-level accuracy and F1 score based on your output.

Evaluation - Centroid-Level F1-Score

- Evaluate the predicted skull fracture **points (coordinates)**
- We provide the code for F1 score evaluation in **for_students_eval.py**
  - Usage: python3 for_students_eval.py --pred_file {pred.csv} --gt_file {gt.csv}
  - {pred.csv} / {gt.csv} should follow the sample output format (csv) in the following pages
  - For validation, you can generate the gt.csv on your own by the provided training annotation

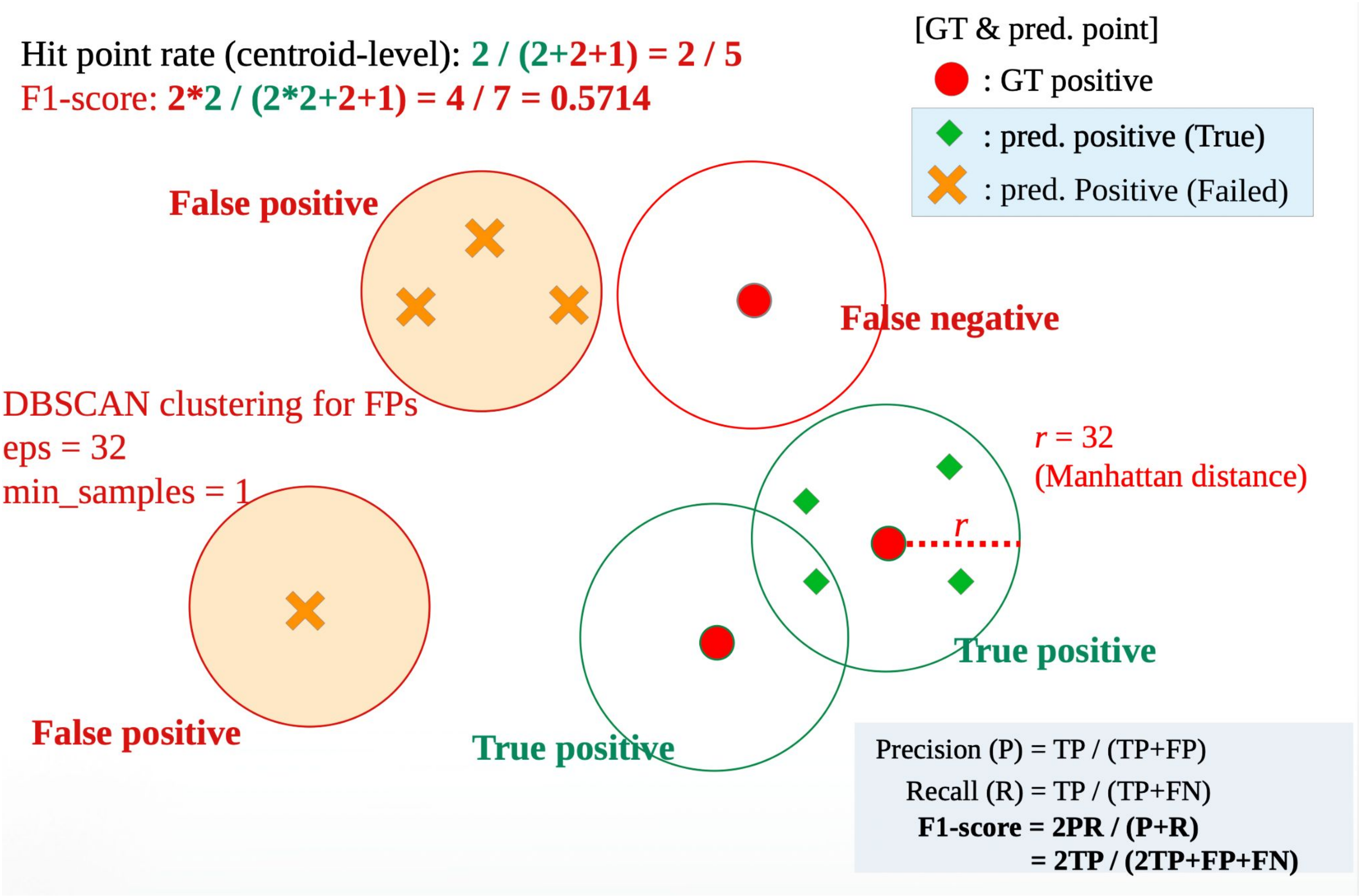
True Positive, TP **（正確偵測）**	Given a GT positive point, a TP exists when there are one or more predicted points inside the circle (radius=32) centered at the GT point.
False Negative, FN **（遺漏偵測）**	Given a GT positive point, a FN exists when there is no predicted point inside the circle (radius=32) centered at the GT point.
False Positive, FP **（錯誤偵測）**	For those predicted points that are not inside any circles centered at GT positive points, we perform **DBSCAN clustering** and get the cluster centers. Each cluster center counts for one FP.

Precision (P) = TP / (TP+FP)

Recall (R) = TP / (TP+FN)

F1 Score = 2PR / (P+R)

Evaluation - Centroid-Level Hit Rate



Hit point rate (centroid-level): 2 / (2+2+1) = 2 / 5

F1-score: 2*2 / (2*2+2+1) = 4 / 7 = 0.5714

[GT & pred. point]

● : GT positive

◆ : pred. positive (True)

✕ : pred. Positive (Failed)

DBSCAN clustering for FPs
eps = 32
min_samples = 1

r = 32
(Manhattan distance)

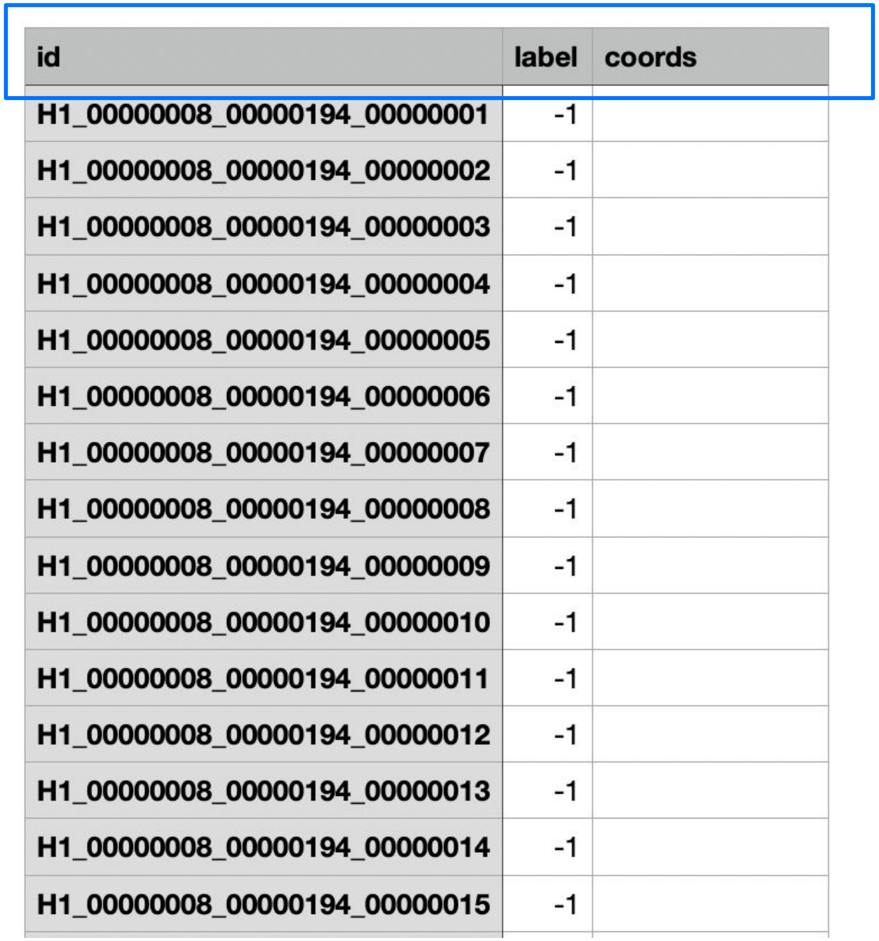
Precision (P) = TP / (TP+FP)

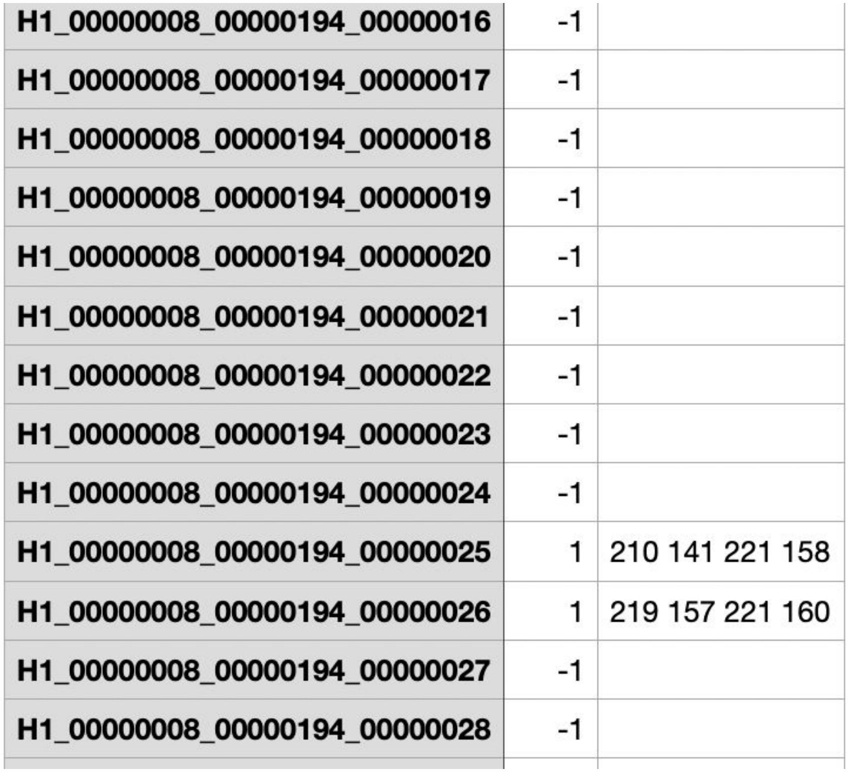
Recall (R) = TP / (TP+FN)

F1-score = 2PR / (P+R)
= 2TP / (2TP+FP+FN)

Sample Output (csv)

slice id slice label coords: x1 y1 x2 y2 …





id	label	coords
H1_00000008_00000194_00000001	-1	
H1_00000008_00000194_00000002	-1	
H1_00000008_00000194_00000003	-1	
H1_00000008_00000194_00000004	-1	
H1_00000008_00000194_00000005	-1	
H1_00000008_00000194_00000006	-1	
H1_00000008_00000194_00000007	-1	
H1_00000008_00000194_00000008	-1	
H1_00000008_00000194_00000009	-1	
H1_00000008_00000194_00000010	-1	
H1_00000008_00000194_00000011	-1	
H1_00000008_00000194_00000012	-1	
H1_00000008_00000194_00000013	-1	
H1_00000008_00000194_00000014	-1	
H1_00000008_00000194_00000015	-1	
H1_00000008_00000194_00000016	-1	
H1_00000008_00000194_00000017	-1	
H1_00000008_00000194_00000018	-1	
H1_00000008_00000194_00000019	-1	
H1_00000008_00000194_00000020	-1	
H1_00000008_00000194_00000021	-1	
H1_00000008_00000194_00000022	-1	
H1_00000008_00000194_00000023	-1	
H1_00000008_00000194_00000024	-1	
H1_00000008_00000194_00000025	1	210 141 221 158
H1_00000008_00000194_00000026	1	219 157 221 160
H1_00000008_00000194_00000027	-1	
H1_00000008_00000194_00000028	-1	

Grading

- **Final 35%** (Bonus up to 3%)
    - **Code / Eval.ai 10%**

      Eval.ai for references, final accuracy evaluated by TAs

        - Baseline 5% (case-level 2.5% / F1 score 2.5%)
        - Relative ranking in the class 5%

          (case-level 2.5% / F1 score 2.5%)
    - **Approach & Presentation 25%**
        - Novelty and technical contribution 10%
        - Completeness of experiments 10%

          (e.g., comparisons to baseline and recent models, ablation studies, visualization, etc.)
        - Presentation (Oral + Poster) 5% + bonus up to 3% (top 3 teams voted by class)

Relative ranking (case-level / F1 score)

Score	Points	Topic 1
highest ↓ lowest	2.5%	3
	2.0%	3
	1.5%	2
	1.0%	2
	0.5%	2

(number of teams)

Challenge Awards

- Top three teams for overall performance (accuracy, presentation, novelty…)
  - 1st: NTD $10k
  - 2nd: NTD $5k
  - 3rd: NTD $3k
- Extra two winners for model performance (case level 50% / F1 score 50%)
  - 1st: NTD $ TBD
  - 2nd: NTD $ TBD

EvalAI & Github links

- EvalAI Competition Link:
  - LINK
- Github Link:
  - LINK
- The team leader should create a team on EvalAI **with the same team name as in the GitHub Classroom**.
- It is **strictly prohibited** for members in the same team to create multiple teams on EvalAI.

(Super Important) Rules

- **Do not disclose the dataset! Dataset 請勿外流！**
- You are **NOT** allowed to use any pretrained models or data related to **computer tomography**, but other types of pretrained models or datasets (e.g., MRI, ImageNet, etc.) are OK.
- Your models and scores need to be reproducible.
- Any violation would result in 0 score for your final project!!

Challenge II
Nodule detection

Sponsered by Inventec

英業達
Inventec

Outline

- Introduction
- Challenges
- Dataset
- Evaluation
- Grading
- Challenge Awards
- Rules

Introduction

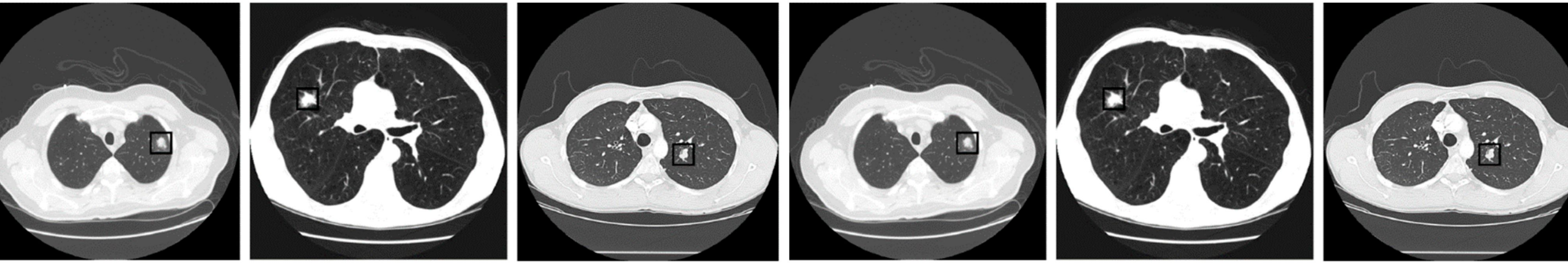
- Dataset

  Computer tomography (電腦斷層影像) of different patients with pulmonary nodules (肺結節)

- Goal

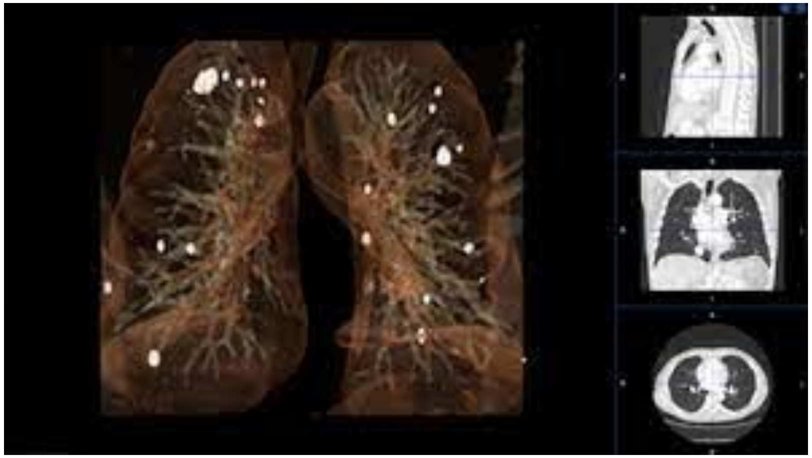
  Train a neural network to **Identify locations of the possible nodules that is larger than 3 mms** by utilizing data annotated by several experienced radiologists.

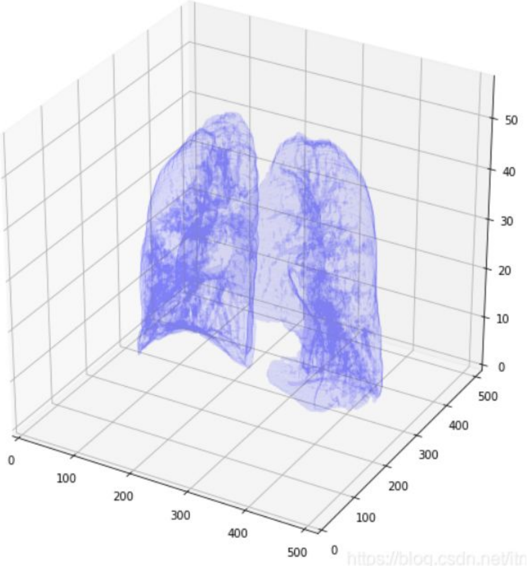
- Task **object detection**



Challenges - Things are not that easy!

- Medical images are quite different from those in COCO or Visual Genome datasets. (e.g., RGB images)
- The given data may be in the 3D voxel format. Therefore, Use the 3D lung nodule feature might make the interpretation of nodule easier.
- The similarity in the morphological appearance of some false positive candidates to that of true pulmonary nodules further increases the difficulty in detection



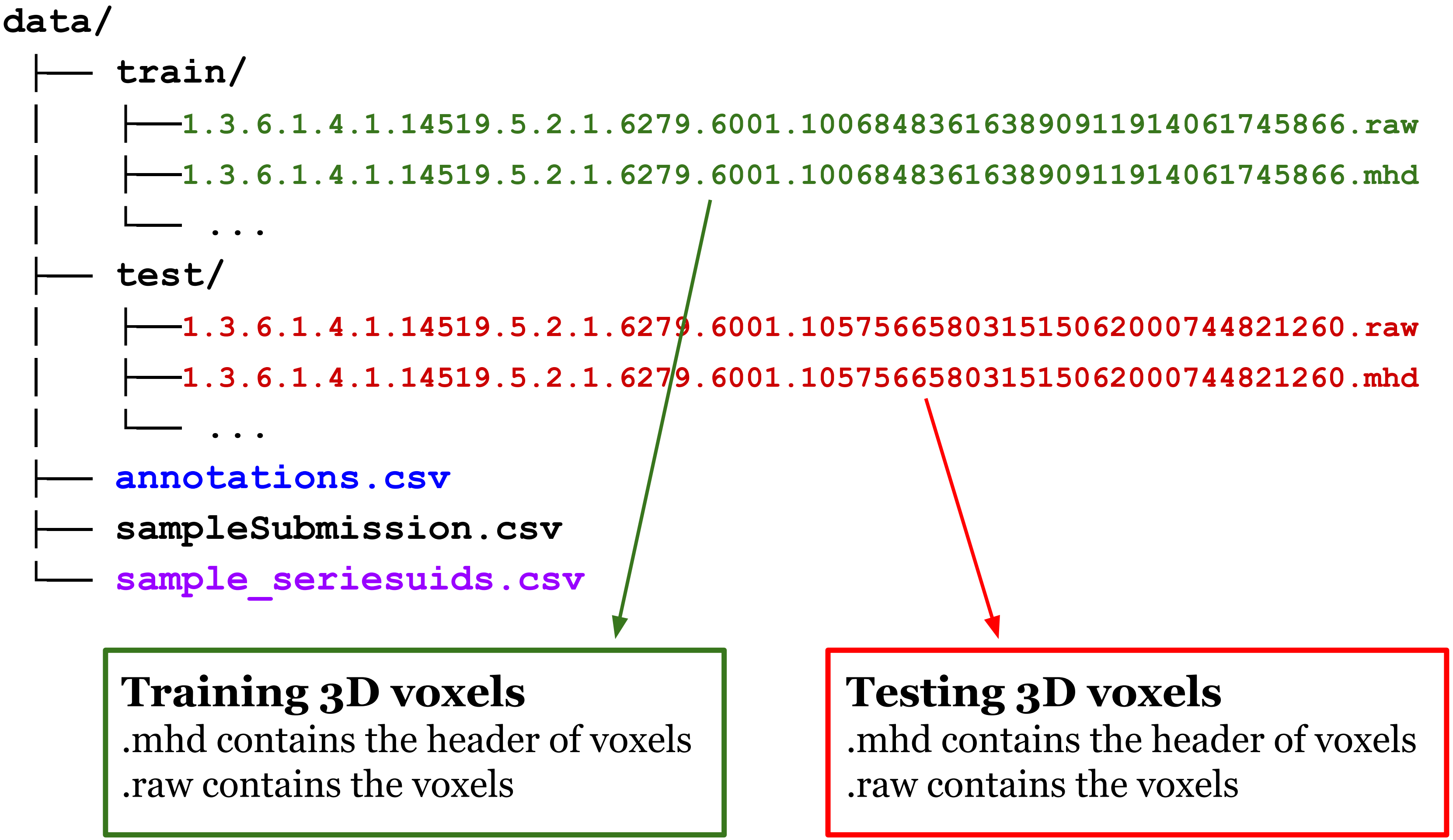


https://blog.csdn.net/itn

Dataset

- CT images of different patients with pulmonary nodules
- Training set (with label)
  - 267 patients, with 70,315 images in total
- Testing set (without label)
  - 89 patients, with 22,870 images in total
- Download link: please refer to get_dataset.sh in Github
- This dataset is large, so you may preprocess the data.

Dataset - File Structure



```
data/
  ├── train/
  │    ├──1.3.6.1.4.1.14519.5.2.1.6279.6001.100684836163890911914061745866.raw
  │    ├──1.3.6.1.4.1.14519.5.2.1.6279.6001.100684836163890911914061745866.mhd
  │    └──  ...
  ├── test/
  │    ├──1.3.6.1.4.1.14519.5.2.1.6279.6001.105756658031515062000744821260.raw
  │    ├──1.3.6.1.4.1.14519.5.2.1.6279.6001.105756658031515062000744821260.mhd
  │    └──  ...
  ├── annotations.csv
  ├── sampleSubmission.csv
  └── sample_seriesuids.csv
```

Training 3D voxels
.mhd contains the header of voxels
.raw contains the voxels

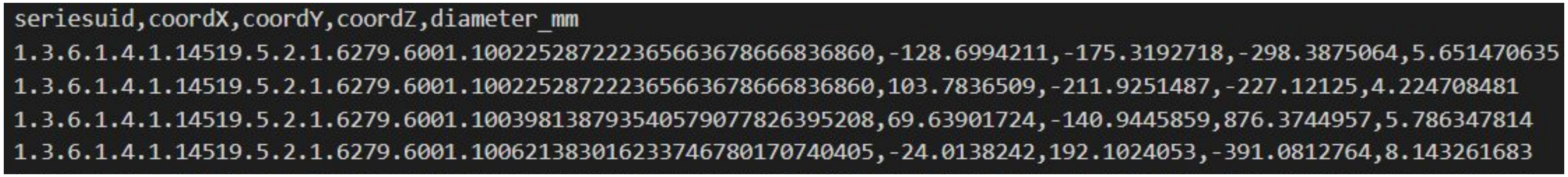
Testing 3D voxels
.mhd contains the header of voxels
.raw contains the voxels

Dataset - File Structure

```
data/
 ├── train/
 │     ├──1.3.6.1.4.1.14519.5.2.1.6279.6001.100684836163890911914061745866.raw
 │     ├──1.3.6.1.4.1.14519.5.2.1.6279.6001.100684836163890911914061745866.mhd
 │     └──  ...
 ├── test/
 │     ├──1.3.6.1.4.1.14519.5.2.1.6279.6001.105756658031515062000744821260.raw
 │     ├──1.3.6.1.4.1.14519.5.2.1.6279.6001.105756658031515062000744821260.mhd
 │     └──  ...
 ├── annotations.csv
 ├── sampleSubmission.csv
 └── sample_seriesuids.csv
```

Seriesuids of training set
This csv file is for the evaluation on training set.

Annotations of training set
Each line holds the SeriesInstanceUID of the scan, the x, y, z position of each finding in world coordinates, and the corresponding diameter in mm.



```
seriesuid,coordX,coordY,coordZ,diameter_mm
1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860,-128.6994211,-175.3192718,-298.3875064,5.651470635
1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860,103.7836509,-211.9251487,-227.12125,4.224708481
1.3.6.1.4.1.14519.5.2.1.6279.6001.100398138793540579077826395208,69.63901724,-140.9445859,876.3744957,5.786347814
1.3.6.1.4.1.14519.5.2.1.6279.6001.100621383016233746780170740405,-24.0138242,192.1024053,-391.0812764,8.143261683
```

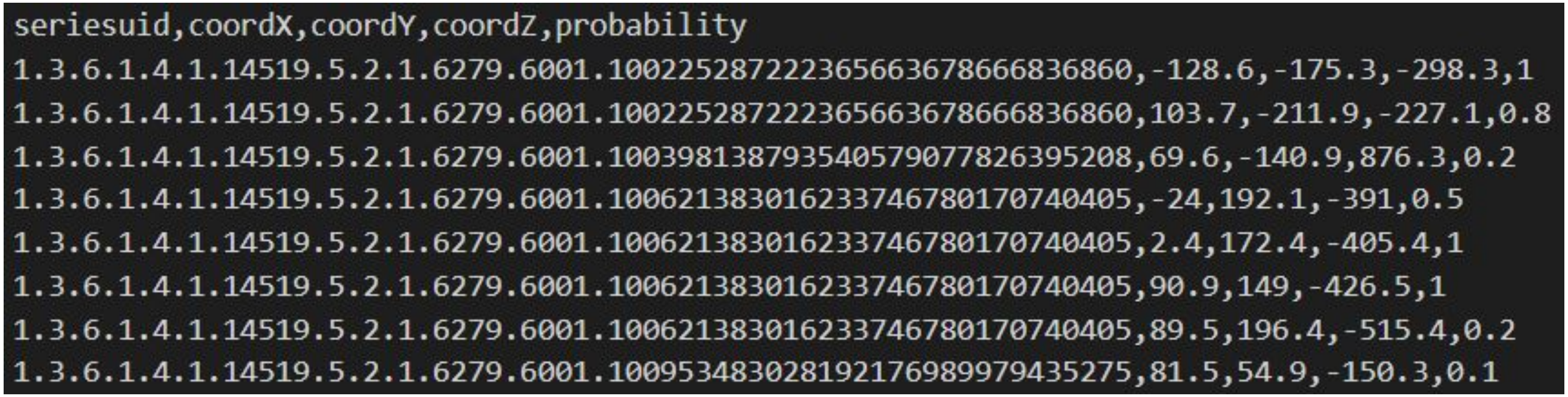
Evaluation

- A detected candidate is considered to be a true positive if the candidate is located within a distance R of the nodule center, where R is set to the diameter of the nodule divided by 2.
- Metrics: free receiver operating characteristic (FROC) analysis.
  - The final score is defined as the average sensitivity at 7 predefined false positive rates: 1/8, 1/4, 1/2, 1, 2, 4, and 8 FPs per scan.
  - We provide evaluation code to compute the final score.
  - Usage:

```
python3 evaluation.py {annotations.csv} {submission.csv} {seriesuids.csv}
```

Evaluation - Sample Submission

- We provide sampleSubmission.csv as an example of a submission file in the correct format.
- Each line holds the SeriesInstanceUID of the scan, the x, y, z position of each finding in world coordinates, and the corresponding probability.



```
seriesuid,coordX,coordY,coordZ,probability
1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860,-128.6,-175.3,-298.3,1
1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860,103.7,-211.9,-227.1,0.8
1.3.6.1.4.1.14519.5.2.1.6279.6001.100398138793540579077826395208,69.6,-140.9,876.3,0.2
1.3.6.1.4.1.14519.5.2.1.6279.6001.100621383016233746780170740405,-24,192.1,-391,0.5
1.3.6.1.4.1.14519.5.2.1.6279.6001.100621383016233746780170740405,2.4,172.4,-405.4,1
1.3.6.1.4.1.14519.5.2.1.6279.6001.100621383016233746780170740405,90.9,149,-426.5,1
1.3.6.1.4.1.14519.5.2.1.6279.6001.100621383016233746780170740405,89.5,196.4,-515.4,0.2
1.3.6.1.4.1.14519.5.2.1.6279.6001.100953483028192176989979435275,81.5,54.9,-150.3,0.1
```

Grading

- **Final 35%** (Bonus up to 3%)
  - **Code / Eval.ai 10%**
    Eval.ai for references, final accuracy evaluated by TAs
    - Simple baseline 4% (Public 2% / Private 2%)
    - Strong baseline 1% (Public 0.5% / Private 0.5%)
    - Relative ranking in the class 5%
      (Public 2.5% / Private 2.5%)
  - **Approach & Presentation 25%**
    - Novelty and technical contribution 10%
    - Completeness of experiments 10%
      (e.g., comparisons to baseline and recent models, ablation studies, visualization, etc.)
    - Presentation (Oral + Poster) 5% + bonus up to 3% (top 3 teams voted by class)

Relative ranking (Public / Private)

Score	Points	Topic 2
highest	2.5%	3
	2.0%	3
↓	1.5%	2
	1.0%	2
lowest	0.5%	2

(number of teams)

Challenge Awards

- Top three teams
  - 1st: NTD $10k
  - 2nd: NTD $5k
  - 3rd: NTD $3k
- (Optional) Special challenge/award [TBD]

EvalAI & Github links

- EvalAI Competition Link:
    - LINK
- Github Link:
    - LINK
- The team leader should create a team on EvalAI **with the same team name as in the GitHub Classroom**.
- It is **strictly prohibited** for members in the same team to create multiple teams on EvalAI.

(Super Important) Rules

- You are **NOT** allowed to use any pretrained models or data related to **computer tomography**, but other types of pretrained models or datasets (e.g., MRI, ImageNet, etc.) are OK.
- Your models and scores need to be reproducible.
- Any violation would result in 0 score for your final project!!

Challenge III

Fine-grained long-tailed food image classification

2021 ICCV OCTOBER 11-17 VIRTUAL

CVF

ICCV’21 Workshop

Outline

- Introduction
- Challenges
- Dataset
- Evaluation
- Grading
- Kaggle / Github links
- Challenge Award
- Rules

Introduction

- Dataset

  Large Scale Visual Food Recognition dataset (LSVFR)

- Goal

  Train a neural network to **classify the given food images**.

- Task **Classification**

Challenges - Things are not that easy!

- Fine-Grained Image Classification!
- Imbalanced data distribution is a common problem in the food analysis (or many real-world applications).
- Large-scale dataset might contains 100,000+ images with 200+ different categories!

Dataset

- Training set (with label)
  - 140,708 images in total, with 1,000 categories
  - As shown in Table 1, the categories are divided into three groups (frequent, common, and rare categories) based on the amount of data of each category.
  - # of frequent/common/rare categories: 290 / 561 / 149
- Val / Testing set
  - 31,263 / 38,425 images in total
- Download link
- Download link (backup)

Categories	Frequent	Common	Rare
# of data	100+	10 <= & < 100	< 10

Table 1. Definition of the frequent, common, and rare categories.

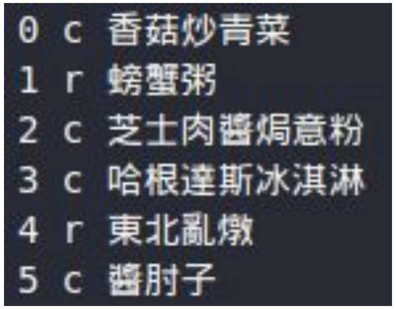
Dataset -- File Structure

```
food_data/
 ├── train/
 |    ├── 0/
 |    |    ├── 16330004.jpg
 |    |    └── ...
 |    └── .../
 ├── val/
 |    ├── 0/
 |    |    ├── 16330009.jpg
 |    |    └── ...
 |    └── .../
 ├── test/
 |    ├── 000000.jpg
 |    └── ...
 ├── testcase/
 |    ├── sample_submission_main_track.csv
 |    └── …
 |
 └── label2name.txt
```

Training Images
`class_id/{imageID}.jpg`

Testing Images
`{imageID}.jpg`

```
class_id, freq/comm/rare, name
...
```



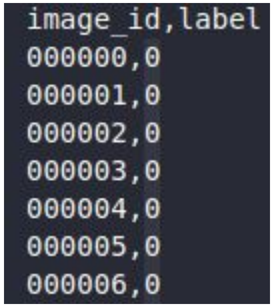
```
0 c 香菇炒青菜
1 r 螃蟹粥
2 c 芝士肉醬焗意粉
3 c 哈根達斯冰淇淋
4 r 東北亂燉
5 c 醬肘子
```

Dataset -- File Structure

```
food_data/
  ├── train/
  │     ├── 0/
  │     │     ├── 16330004.jpg
  │     │     └── ...
  │     └── .../
  ├── val/
  │     ├── 0/
  │     │     ├── 16330009.jpg
  │     │     └── ...
  │     └── .../
  ├── test/
  │     ├── 000000.jpg
  │     └── ...
  ├── testcase/
  │     ├── sample_submission_main_track.csv
  │     └── …
  │
  └── label2name.txt
```

These csv files identify the images for each testing track. The image file can be founded in test/ with the name of "image_id".jpg, and the labels are all dummy terms (i.e., "0"). Once you replace the labels with the prediction results of your model, you can upload this file to kaggle directly.

e.g., sample_submission_main_track.csv



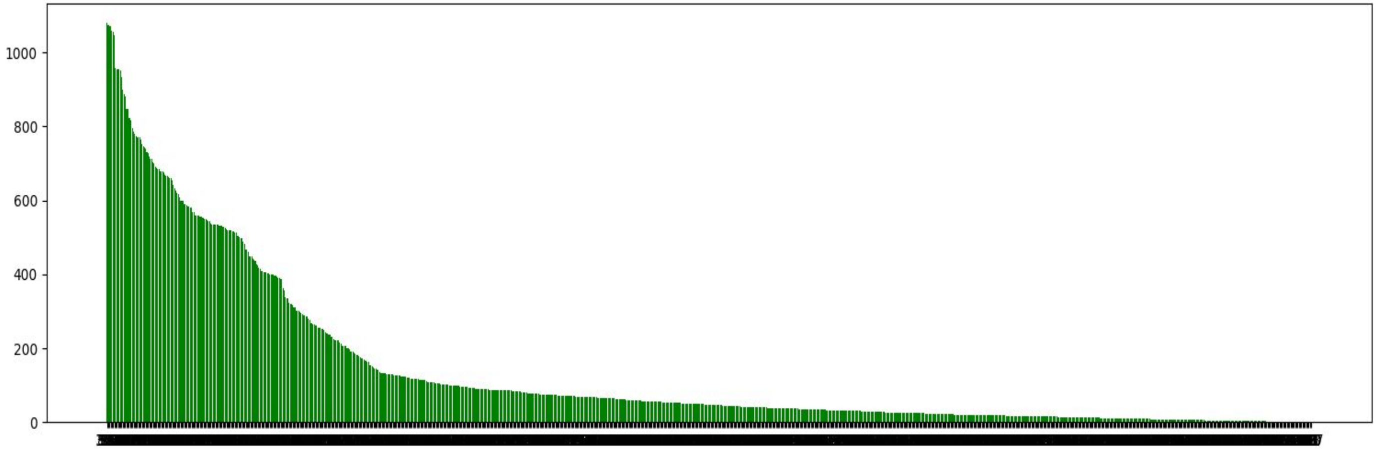
```
image_id,label
000000,0
000001,0
000002,0
000003,0
000004,0
000005,0
000006,0
```

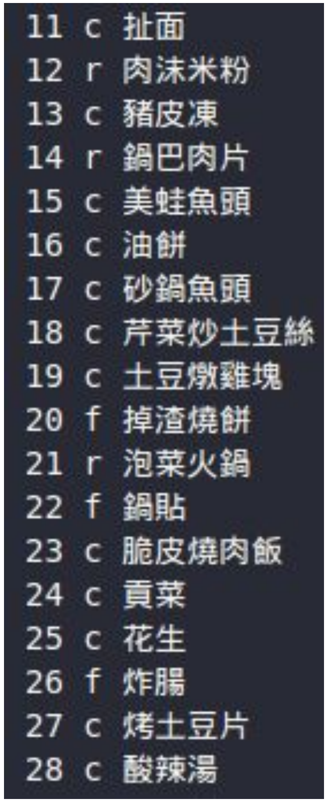
Note that these csv files are for the testing only, and **you are not allow** to pre-classify the testing images into frequent / common / rare categories by scraping these files.

Evaluation

- Top-1 classification accuracy is used for evaluation.
- Report classification accuracy on the frequent/common/rare categories separately.
- **Baseline Accuracy (test set):**

	All	Freq.	Common	Rare
Accuracy	**0.6617**	**0.8457**	**0.6336**	**0.2439**





```
11 c 扯面
12 r 肉沫米粉
13 c 豬皮凍
14 r 鍋巴肉片
15 c 美蛙魚頭
16 c 油餅
17 c 砂鍋魚頭
18 c 芹菜炒土豆絲
19 c 土豆燉雞塊
20 f 掉渣燒餅
21 r 泡菜火鍋
22 f 鍋貼
23 c 脆皮燒肉飯
24 c 貢菜
25 c 花生
26 f 炸腸
27 c 烤土豆片
28 c 酸辣湯
```

Grading

- **Final 35%** (Bonus up to 3%)
  - **Code / Kaggle 10%**
    Kaggle / Eval.ai for references, final accuracy evaluated by TAs
    - Main track
      - Baseline 2% (Public 1% / Private 1%)
      - Relative ranking in the class 2% (Public 1% / Private 1%)
    - **Each** Freq./Comm./Rare track
      - Baseline 1% (Public 0.5% / Private 0.5%)
      - Relative ranking in the class 1% (Public 0.5% / Private 0.5%)
  - **Approach & Presentation 25%**
    - Novelty and technical contribution 10%
    - Completeness of experiments 10%
      (e.g., comparisons to baseline and recent models, ablation studies, visualization, etc.)
    - Presentation (Oral + Poster) 5% + bonus up to 3% (top 3 teams voted by class)

Relative ranking (Public / Private)

Score	Points	Topic 3
highest	100%	2
	75%	2
↓	50%	2
	25%	1
lowest	0%	1

(number of teams)

Kaggle & Github links

- Kaggle Competition Link:
  - Main track: LINK
  - Track for frequent categories: LINK
  - Track for common categories: LINK
  - Track for rare categories: LINK
- Github Link: LINK
- The team leader should create a team on Kaggle **with the same team name as in the GitHub Classroom**.
- It is **strictly prohibited** for members in the same team to create multiple teams on Kaggle.

Challenge Awards (tentative)

- Top three teams for overall performance (accuracy, presentation, novelty…)
  - 1st: NTD $10k
  - 2nd: NTD $5k
  - 3rd: NTD $3k

(Super Important) Rules

- **Do not disclose the dataset! Dataset 請勿外流！**
- You are **NOT** allowed to use any pretrained models or data related to **Food analysis workshop**, but other types of pretrained models or datasets (e.g., ImageNet, etc.) are OK.
- Your models and scores need to be reproducible. **(Including both train & inference phases)**
- Any violation would result in 0 score for your final project!!

Have fun !