

影像處理 作業2 — Fractured Scaphoid Detection

資訊所 碩一 黃佳倫 P76134862

Use Faster R-CNN and YOLOv11-OBB to detect the scaphoid fracture location.

Get started

1. Training

```
python main.py --train 1
```

2. Run System

```
python main.py
```

Model

Name	Description	path
ScaphoidDetector	Detects scaphoid bone in X-ray hand images using <u>Faster R-CNN</u>	<code>scaphoid_detector.py</code>
FractureClassifier	Classify scaphoid fractures using <u>VGG16</u> pre-trained model after detection by ScaphoidDetector	<code>fracture_classifier.py</code>
HandDetector	Detects scaphoid bones and fractures region in X-ray hand image using YOLOv11- <u>OBB</u>	<code>hand_detector.py</code>

Methods

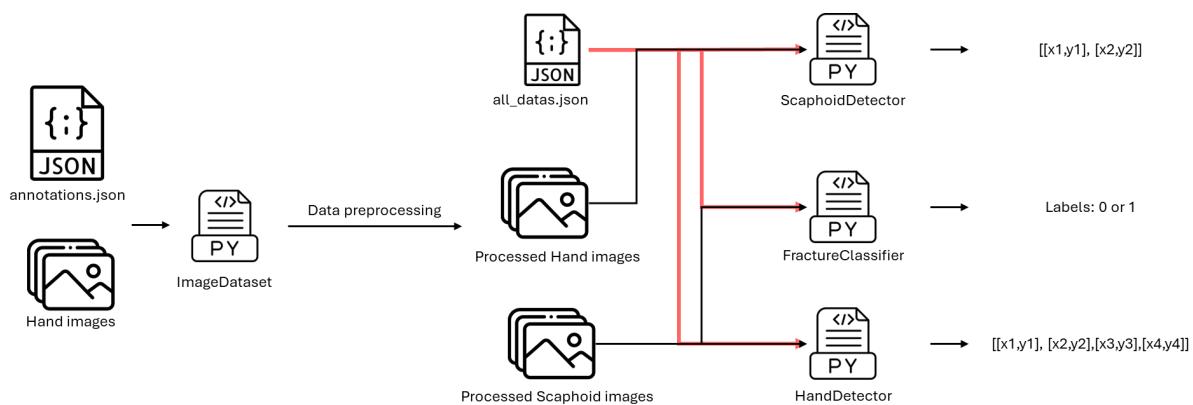
1. ScaphoidDetector + FractureClassifier + HandDetector

First, use Faster R-CNN to detect the scaphoid bone in the full X-ray hand image. Then, use VGG16 to classify whether there is a fracture. Finally, use YOLOv11-obb to detect the fracture location.

2. HandDetector

Directly use YOLOv11-obb to detect the scaphoid bone and fracture locations.

ScaphoidDetector + FractureClassifier + HandDetector



Datasets

1. File Structure:

```
ip_data
└── fracture_detection
    └── annotations // Fracture locations: [[x1, y1], [x2,
└── scaphoid_detection
    ├── annotations // Scaphoid locations: [x1, y1, x2, y2]
    └── images      // Hand X-ray images
```

2. After data preprocessing in `dataset.py` :

`all_datas.json` and new folders will be created under `fracture_detection` and `scaphoid_detection`

```
|── ip_data
|   ├── fracture_detection
|   |   ├── annotations
|   |   ├── images
|   |   └── images_rec
|   └── scaphoid_detection
|       ├── annotations
|       ├── images
|       └── images_rec
└── all_datas.json
```

- `fracture_detection/` :
 - `images/` : Contains the full scaphoid images cropped based on scaphoid locations.
 - `images_rec/` : Contains the scaphoid images with highlighted fracture locations.

```
fracture_detection
├── annotations
├── images
└── images_rec
```

- `scaphoid_detection/images_rec` : Stores hand images with the scaphoid region framed.

Training

1. Train ScaogiudDetector

```
from scahpoid_detector import ScaphoidDetector
scaphoid_detector = ScaphoidDetector(args)
scaphoid_detector.train()
```

2. Train FractureClassifier

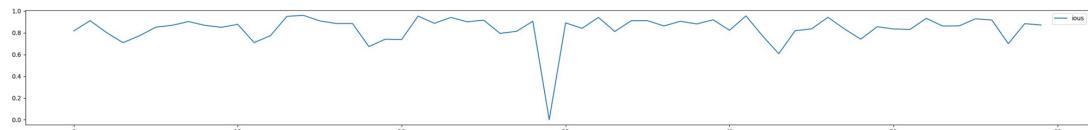
```
from fracture_classifier import FractureClassifier  
fracture_classifier = FractureClassifier(args)  
fracture_classifier.train()
```

3. Train HandDetector

```
from hand_detector import HandDetector  
hand_detector = HandDetector(args)  
hand_detector.train()
```

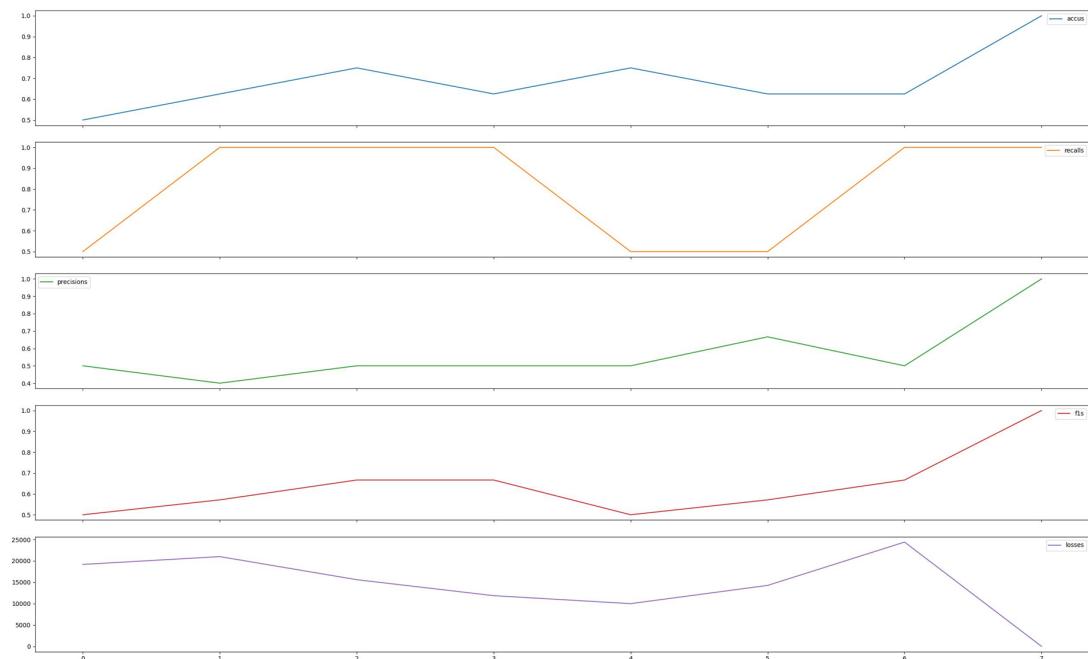
4. Analysis

- ScaphoidDetector

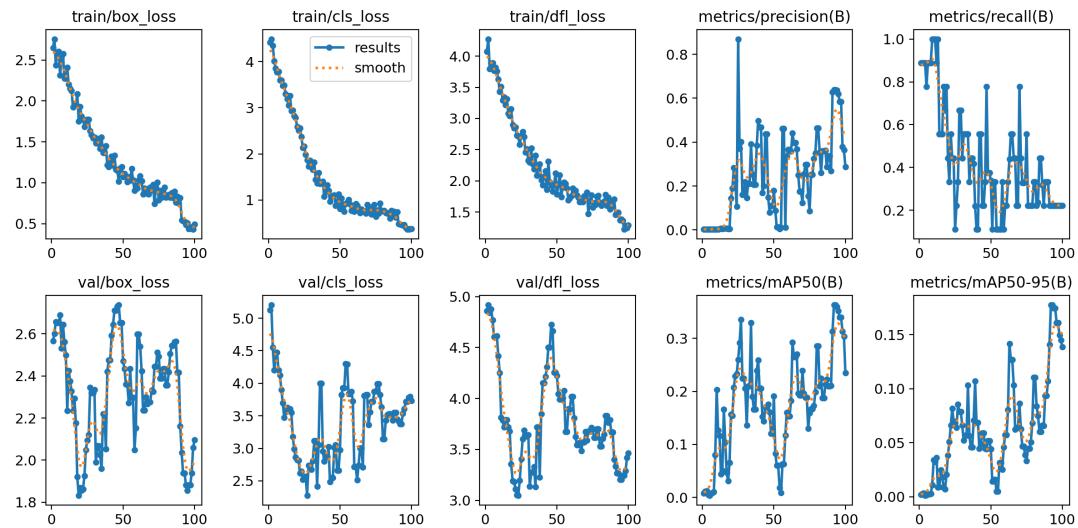


- FractureClassifier

accuracy, recalls, precision, f1, loss



- HandDetector: Curves will be saved in `performance` and `runs/` respectively



Detecting

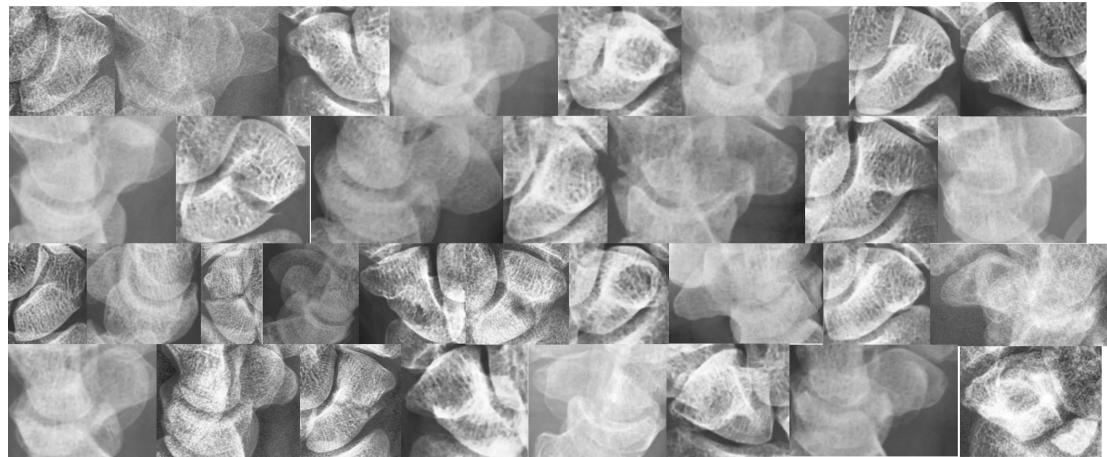
Steps 1. Detect Scaphoid

- Use `detect()` function

```
scaphoid_detector.detect(dir_path)
```

- Detected scaphoid location will be cropped and saved in

`prediction/scaphoid/`



Steps 2. Classify fracture

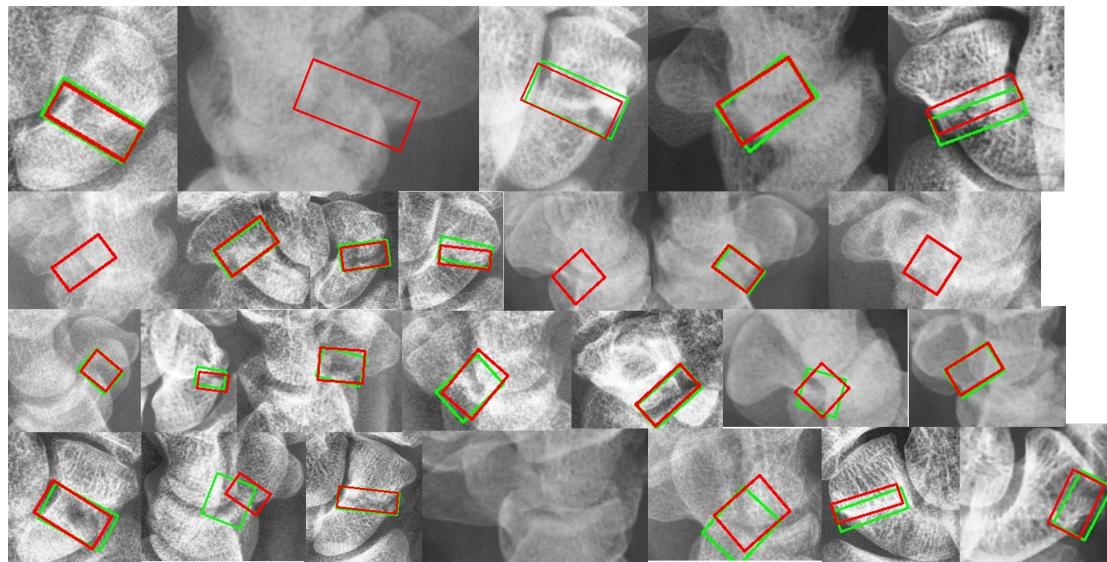
- Use `classify()` function

```
fracture_classifier.classify(dir_path)
```

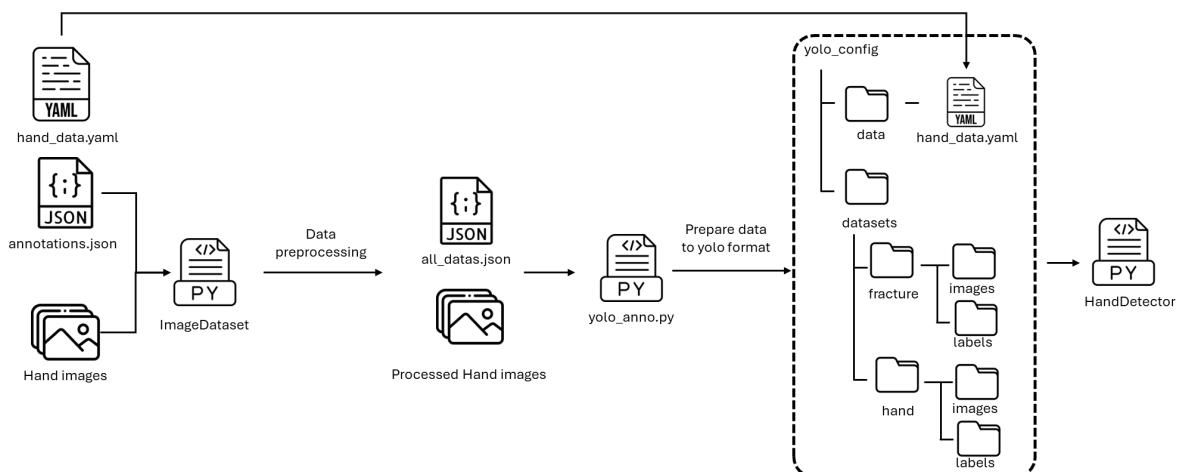
- Fracture scaphoid will be saved in `prediction/classifier/`

Steps 3. Detect fracture location

- Use `detect_fracture()` function
- The images with marked fracture locations will be saved in `prediction/fracture/`



HandDetector



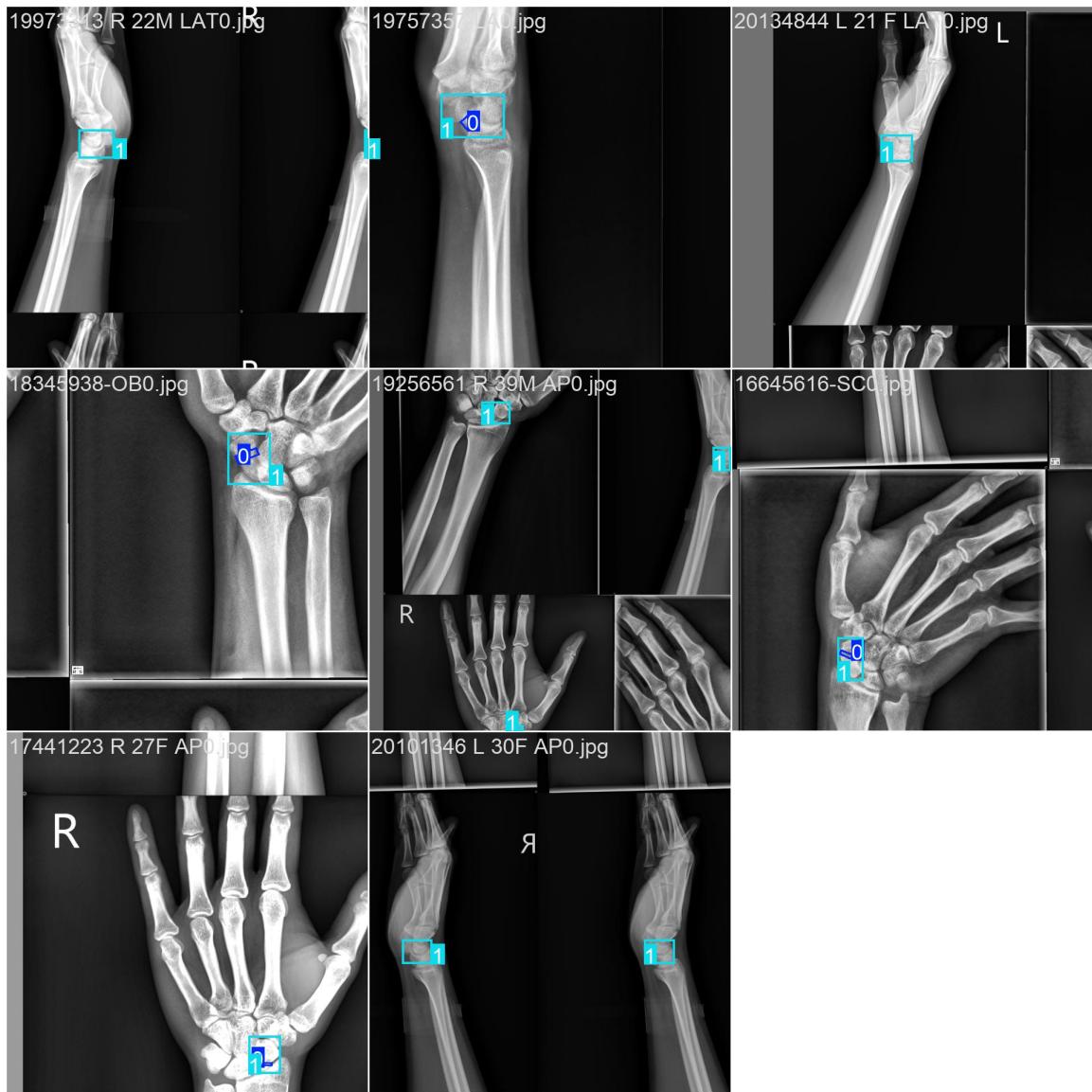
Training Datasets

Use functions from `yolo_anno.py` to construct data for YOLOv11-OBB

1. File Structure

```
yolo_config
├── data
└── datasets
    ├── fracture
    │   ├── images
    │   │   ├── train
    │   │   └── val
    │   └── labels
    │       ├── train
    │       └── val
    └── hand
        ├── images
        │   ├── train
        │   └── val
        └── labels
            ├── train
            └── val
└── weights
```

2. During Training: YOLO 會自動將所有圖片拼在一起，最後再裁成設定得大小 (以下範例為1024)，圖片就會前處理成以下，一個batch的圖片數量會根據 `batch_size` (以下範例為 8)

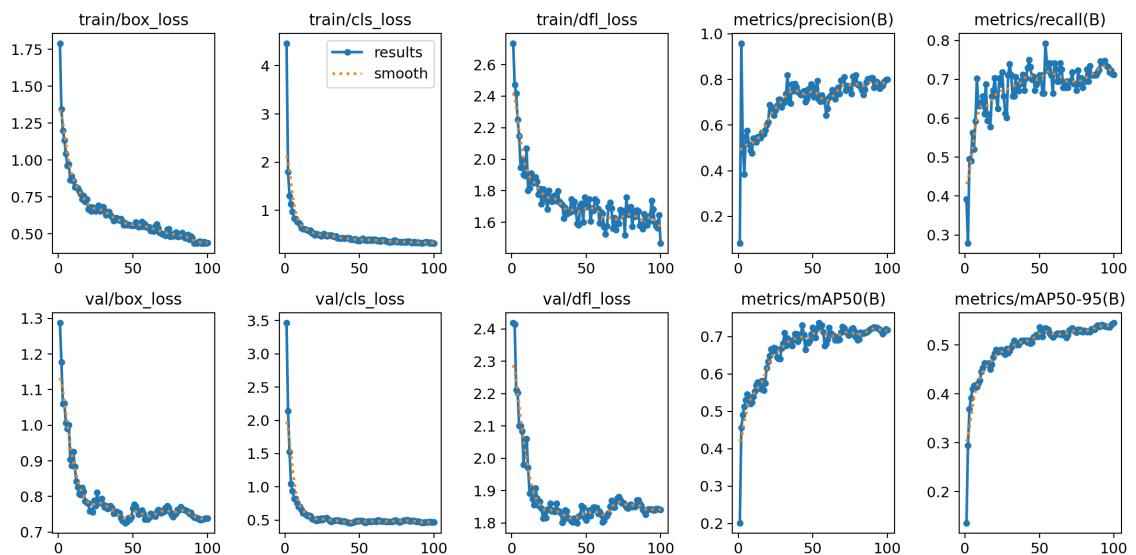


Training

1. Train HandDetector

```
from hand_detector import HandDetector  
hand_detector = HandDetector(args)  
hand_detector.train()
```

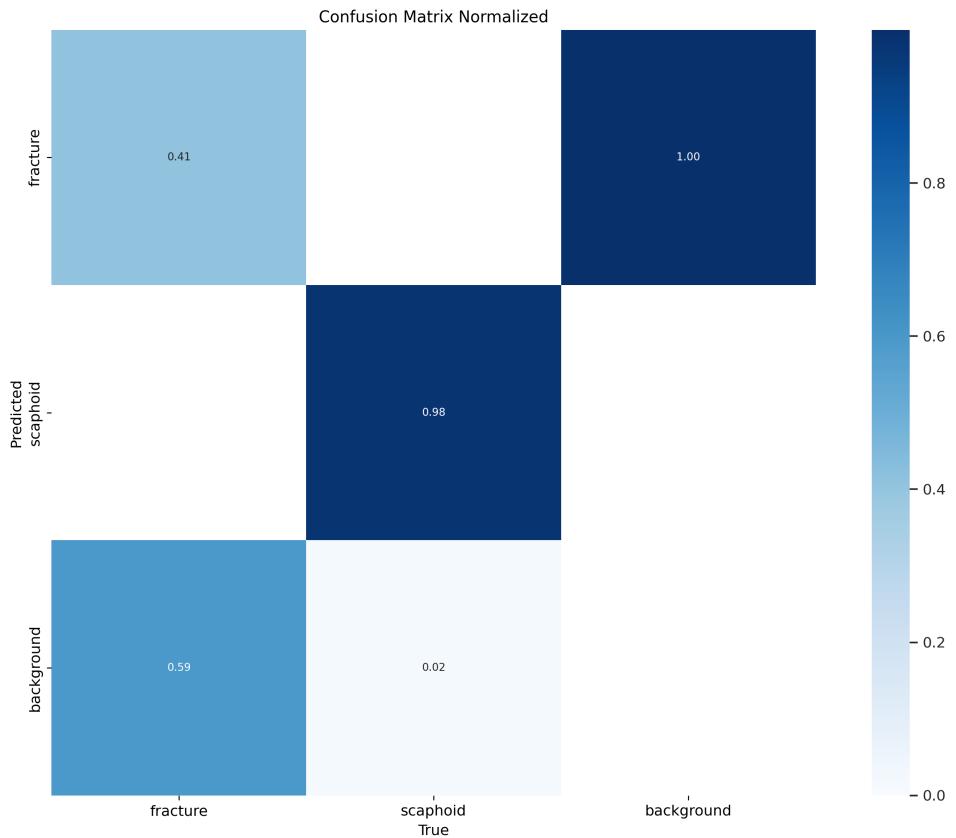
2. Results will be saved in `runs/`



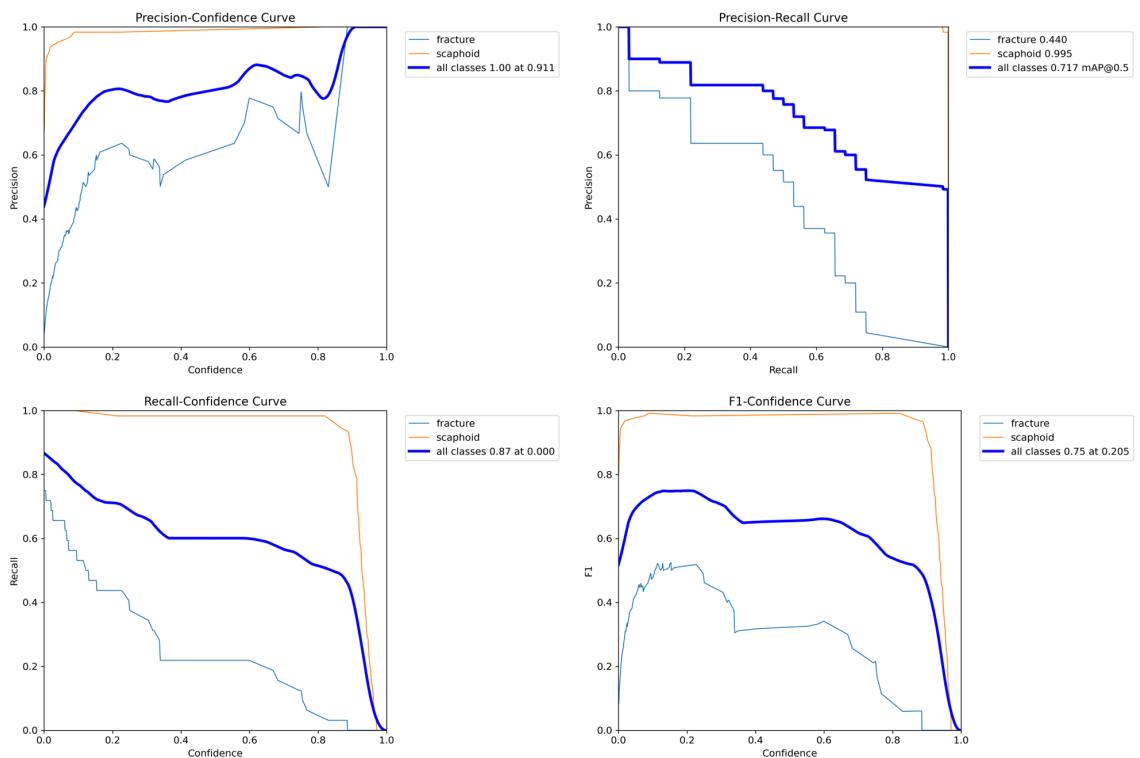
Results

1. Confusion Matrix:

- **Scaphoid:** Using YOLOv11-OBB to detect the position of the scaphoid performed exceptionally well, with an accuracy of up to 98% in predictions.
- **Fracture:** YOLOv11-OBB correctly predicted 41% of fracture locations in full-hand X-ray images, slightly outperforming the two-stage detection method.

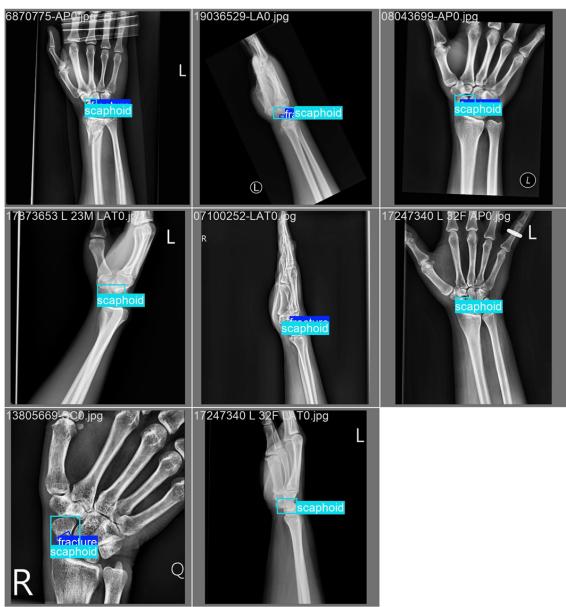


2. Precision, Recall, F1

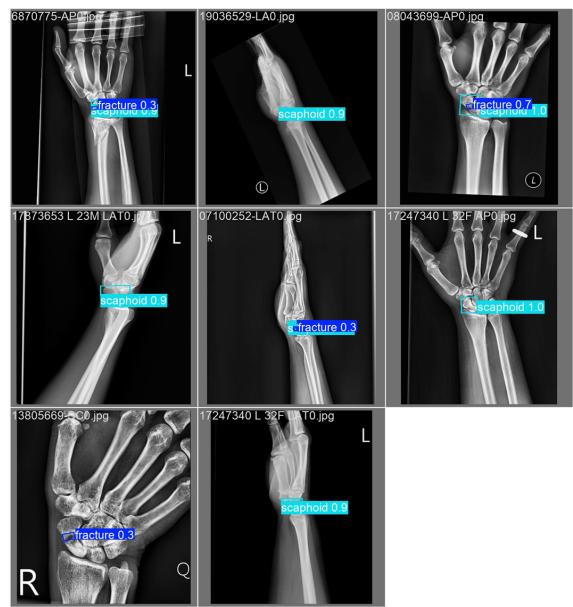


3. During Testing

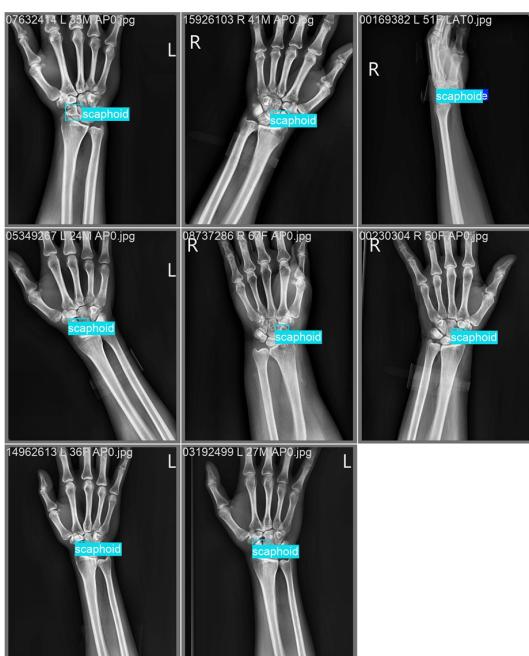
Labels



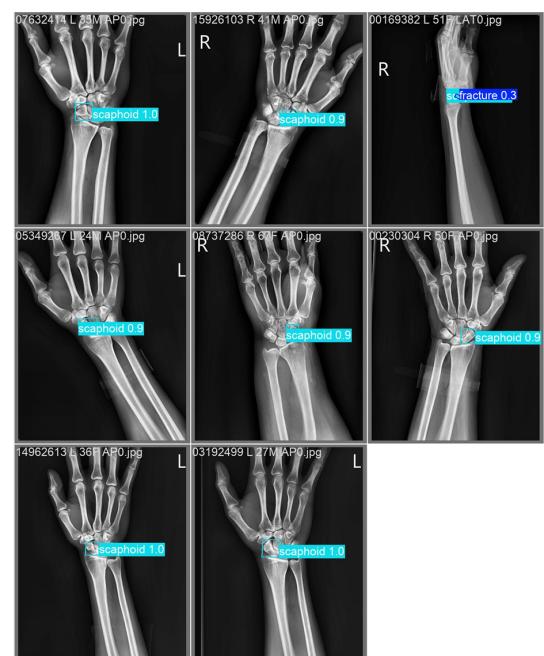
Predicted

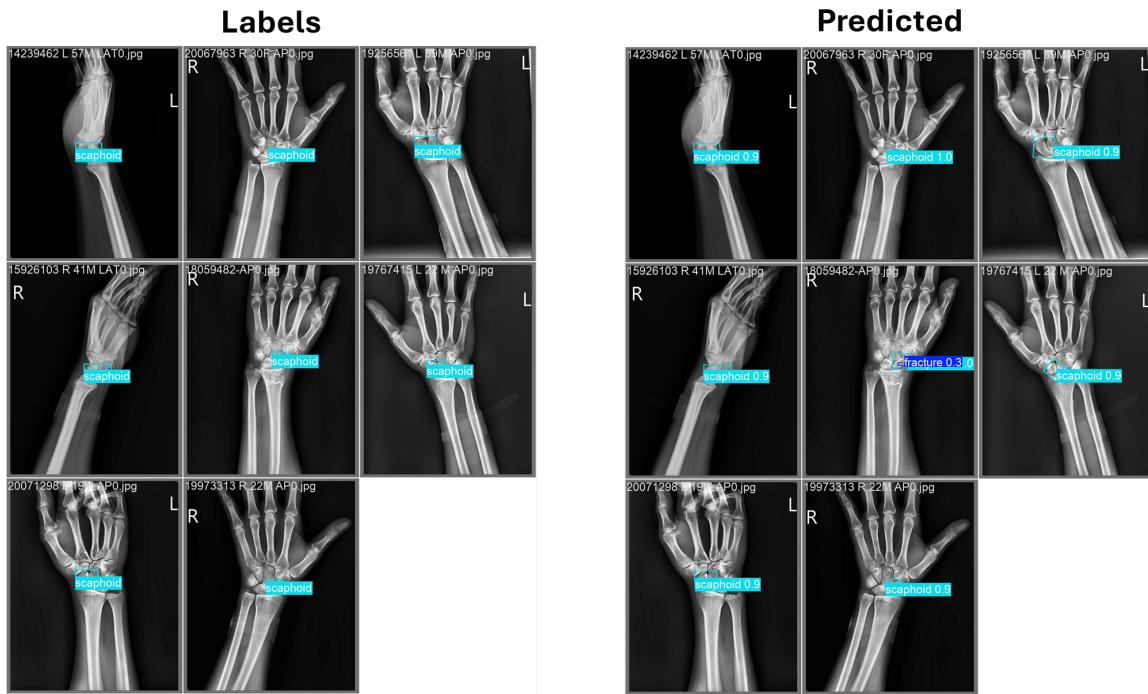


Labels



Predicted





Detecting

1. Detect scaphoid

- Detect images in folder

```
hand_detector.detect_scaphoid(dir_name)
```

- Detect one image

```
hand_detector._detect_scaphoid(img_name, img_path)
```

2. Detect fracture

- Detect images in folder

```
hand_detector.detect_fracture(dir_name)
```

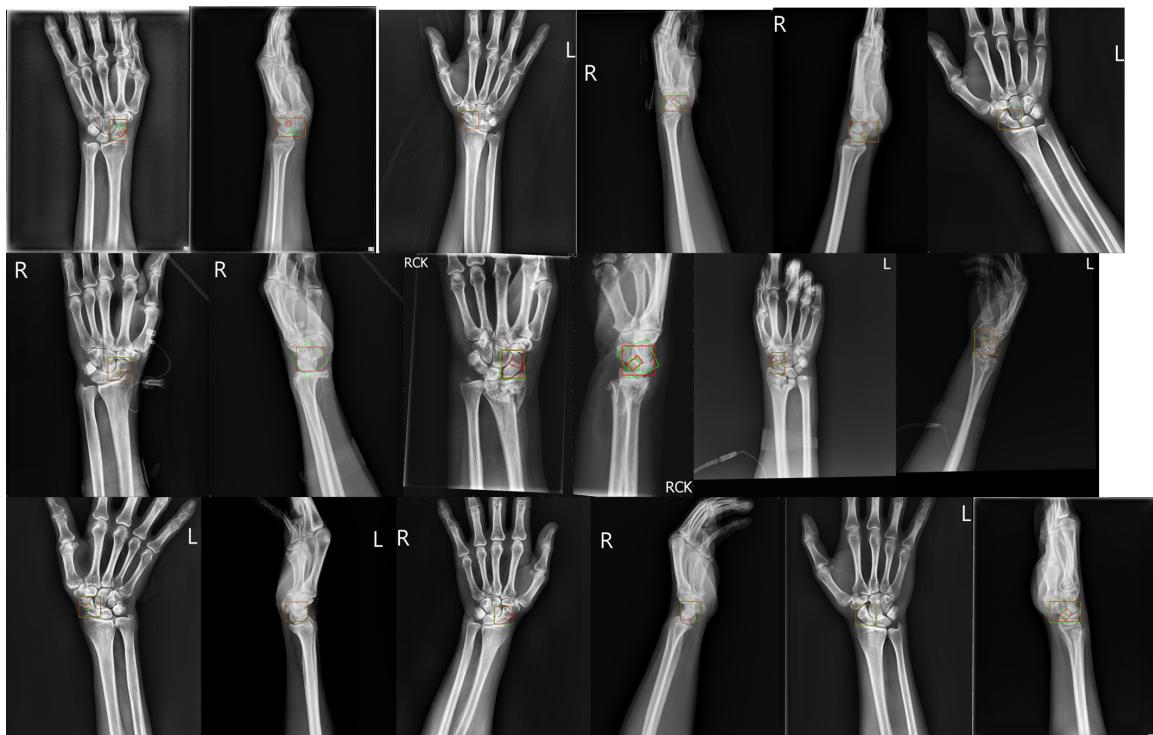
- Detect one image

```
hand_detector._detect_fracture(img_name, img_path)
```

3. Plot the rectangle

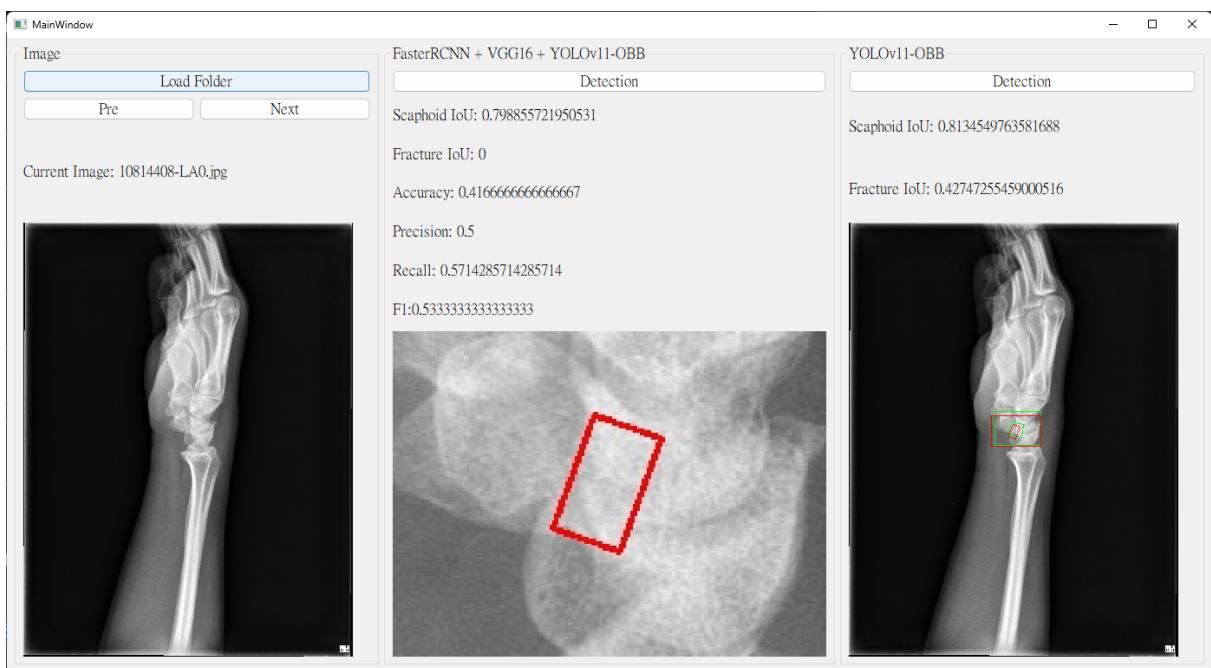
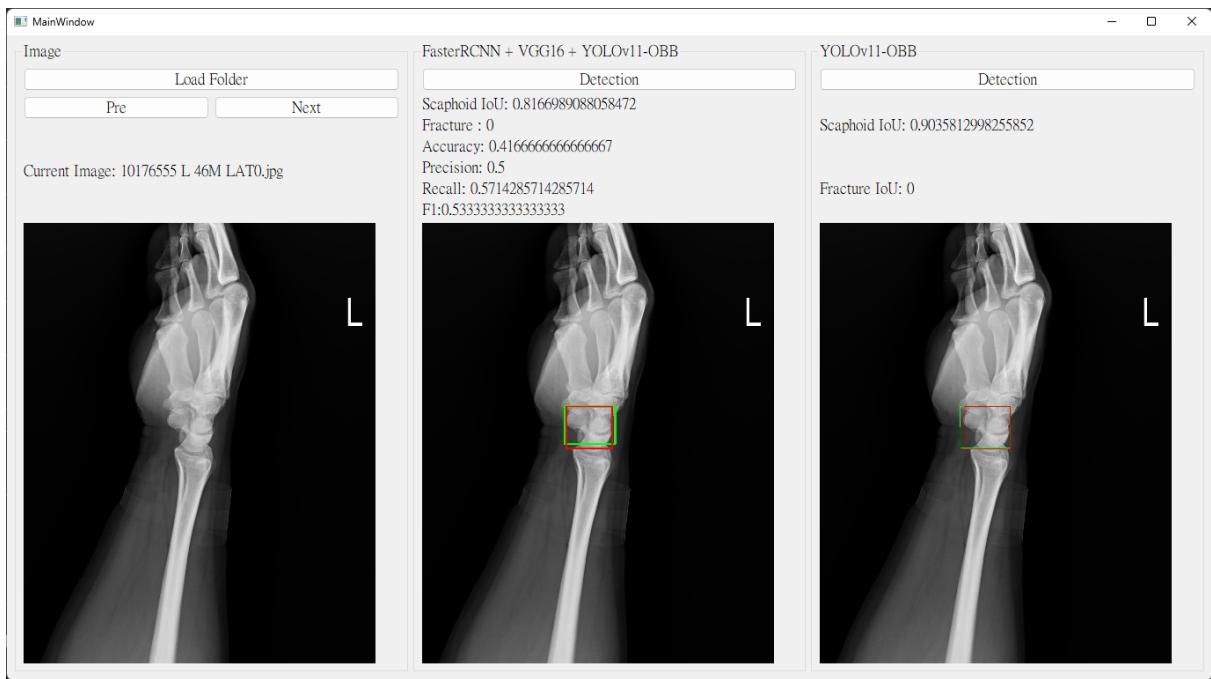
The `detect_*` function performs two key operations:

- Predicts the location of the scaphoid or fracture
- Uses `plot_xyxyxy()` to visualize the results with
 - Red rectangles showing the target (ground truth) locations
 - Green rectangles showing the predicted locations
 - Pictures will be saved in `prediction/hand/`



System

Load a folder containing the dataset file structure. The system will then begin predicting and save the images with the scaphoid and fracture locations highlighted.



Code Availability

<https://github.com/Hlunlun/Fractured-Scaphoid-Detection>

Datasets Availability

From [NCKU CSIE Visual System Lab](#)

Reference

- [FastRCNNPredictor doesn't return prediction in evaluation](#)
- [Oriented Bounding Box \(OBB\) Datasets Overview](#)
- [一篇文章快速认识YOLO11 | 旋转目标检测 | 原理分析 | 模型训练 | 模型推理](#)
- [Understanding and Implementing Faster R-CNN](#)