

# 我們如何在醫院打造 即時醫療影像AI平台

Max Lai  
台中童綜合醫院 資訊部

# About Me

- Taichung.py 社群共同發起人
- Agile Taichung 社群共同發起人
- 台中童綜合醫院 AI Lab Senior Manager

# 台中童綜合醫院

- 位於台中市梧棲區, 1,500 病床
- 資訊部有 78位成員
- AI Lab 目前有 8位 AI工程師& 4位資料分析師
- 主要標註者為 3位 放射科醫師& 4位放射師

誠懇謙卑 感動服務  
品質為先 創新卓越



# Ethics and Privacy Announcement



All of our researches concerning ethics and privacy are under oversight by an independent committee of **IRB** (Institutional Review Boards).

All sensitive data in this presentation are **de-identified** and **de-linked** before use.

# Agenda

- Deep Learning Pipeline
- 醫學影像標註系統的設計考量
- 即時胸部X光腫瘤AI平台
- 深度學習模型封裝
- 結論

# 早期治療五年存活率>70%

## 若為第四期肺癌則<5%

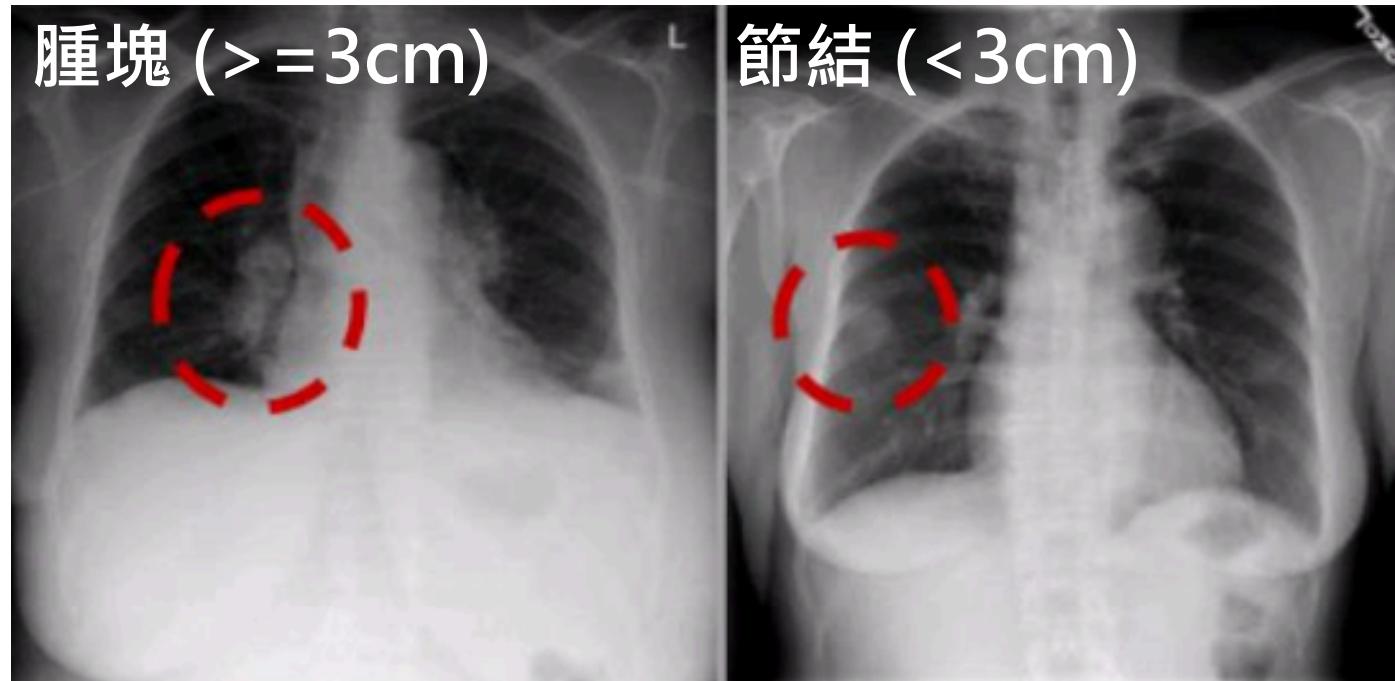
### 2017年台灣男女性10大癌症標準化發生率



資料來源：國民健康署 2020/06/02 <https://bit.ly/3baZoBU>

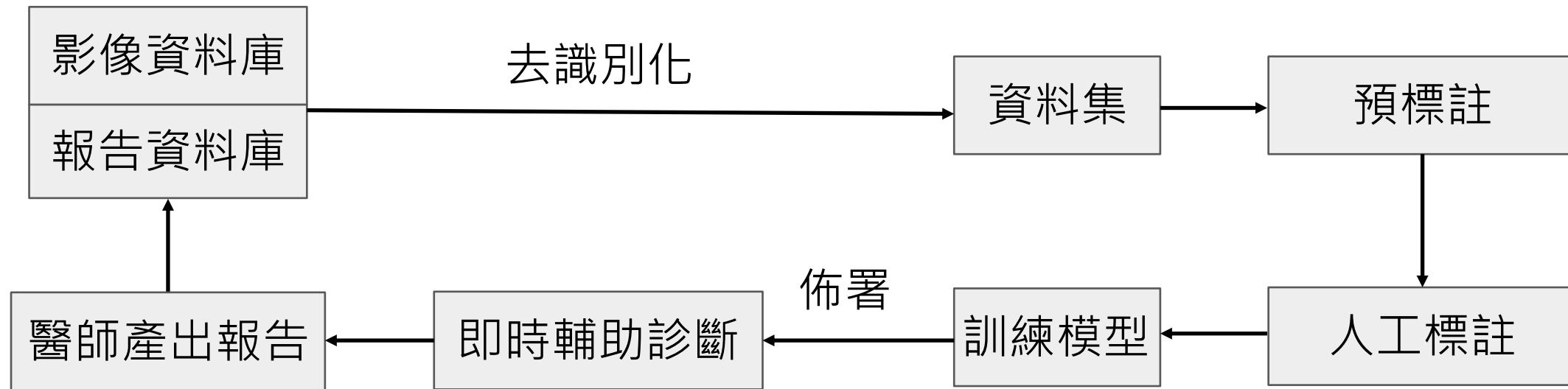
# 醫院 AI 平台目標

- 使用類神經網路在X光影像上偵測結節及腫塊
- 整合 PACS 系統，建置醫學影像標註系統
- 整合至報告系統，在醫師打報告時即時顯示病灶位置



胸部X光是放射影像中使用最多、數量最大，也是最重要的初步診斷工具。  
但小的結節有時肉眼辨識不易，因而延誤早期治療之時機。

# Medical Deep Learning Pipeline





# 醫學影像標註系統

**LABELING SYSTEM**  
**操作展示**

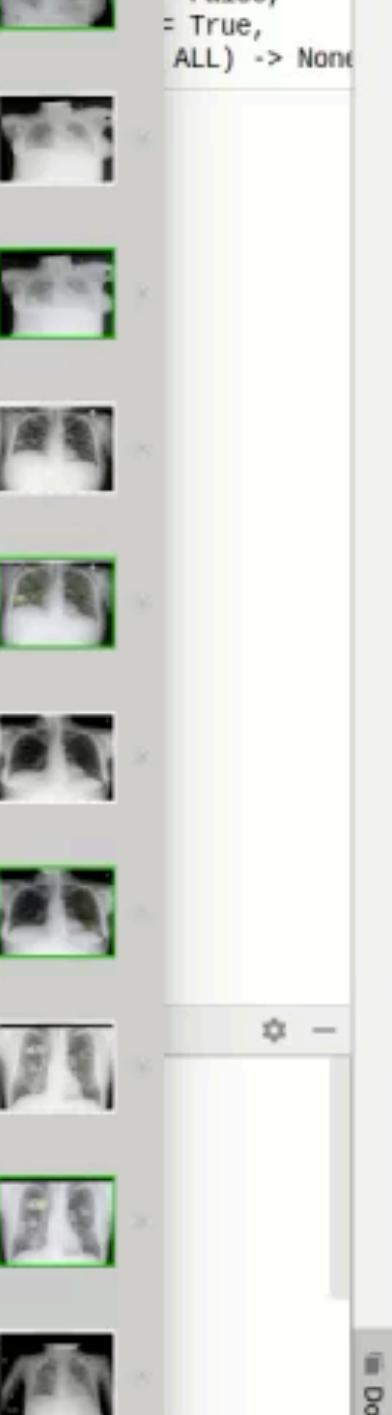
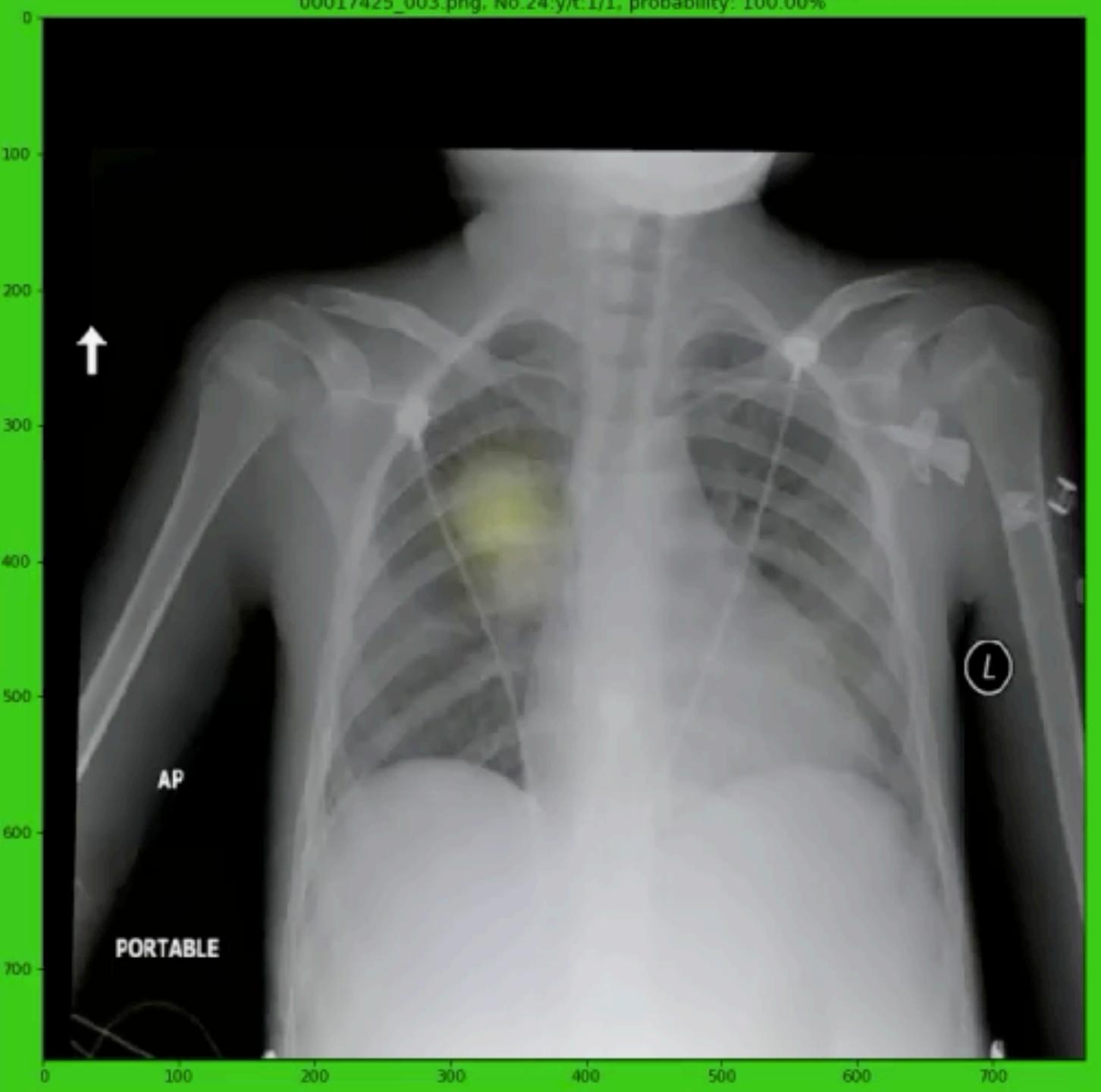
# 放射報告系統整合

```
e(my_resnet, mode  
fication_factor =  
vel = 2  
  
img_size = int(28  
g_size = int(256  
.set_pool(pool_ty  
set  
= f'./trainset_<  
= f'./testset_<or  
der, test_loader
```

```
ze_predict(my_res  
_predict(my_resne
```

```
main_
```

```
†
```



U132001	Chest	190
K32001	Chest	191

報告內容

AI輔助

nodule\_found

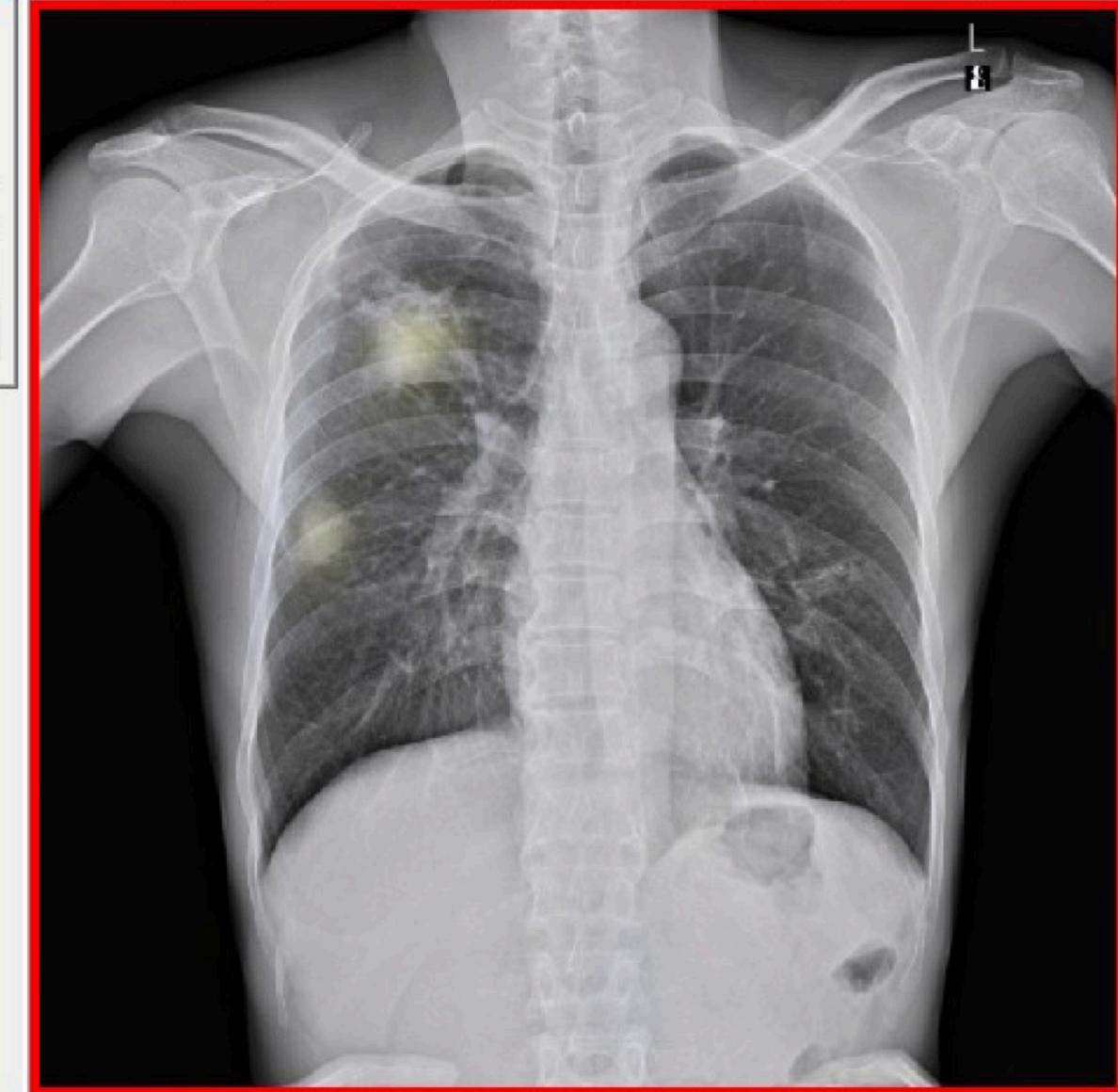
True

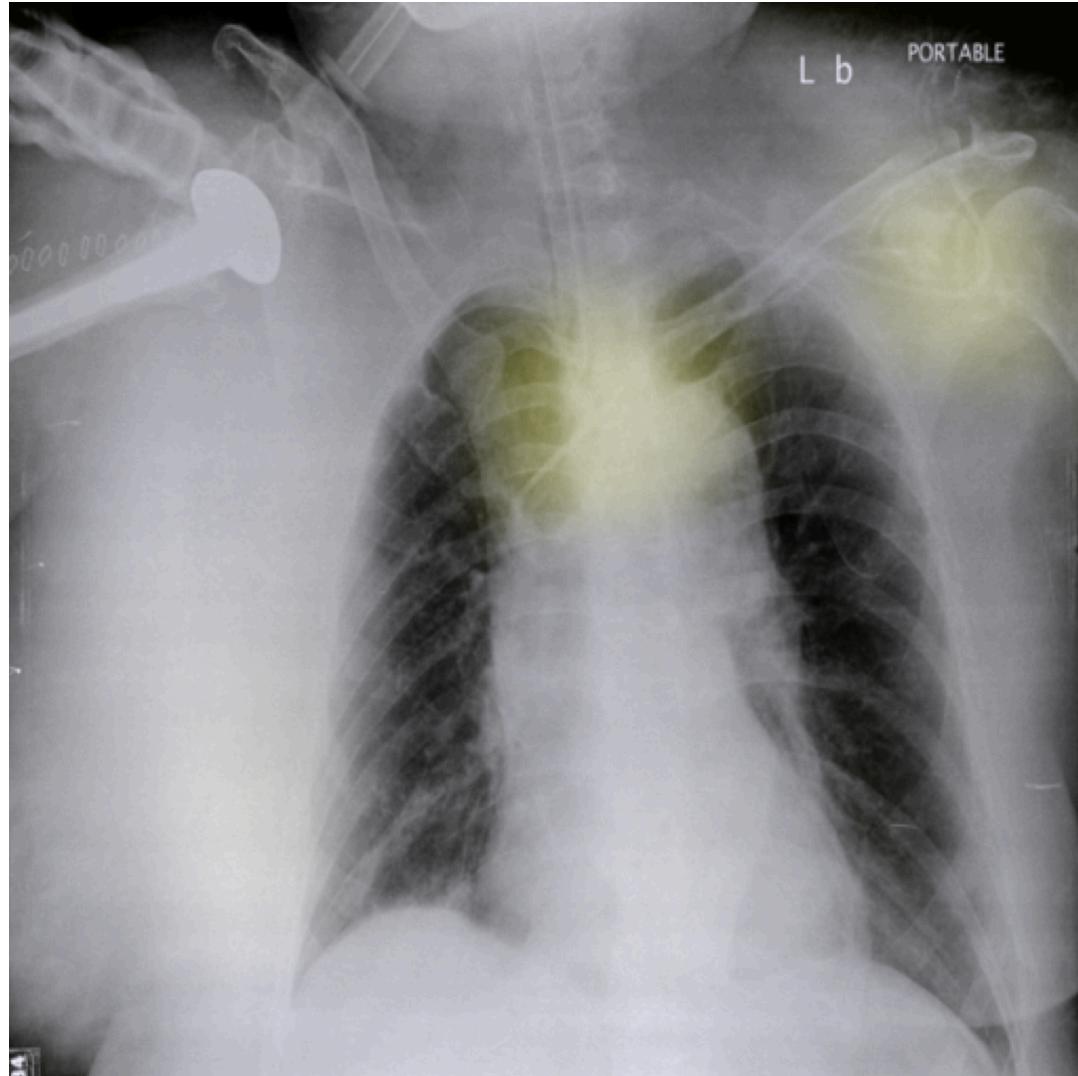
機率

1.000

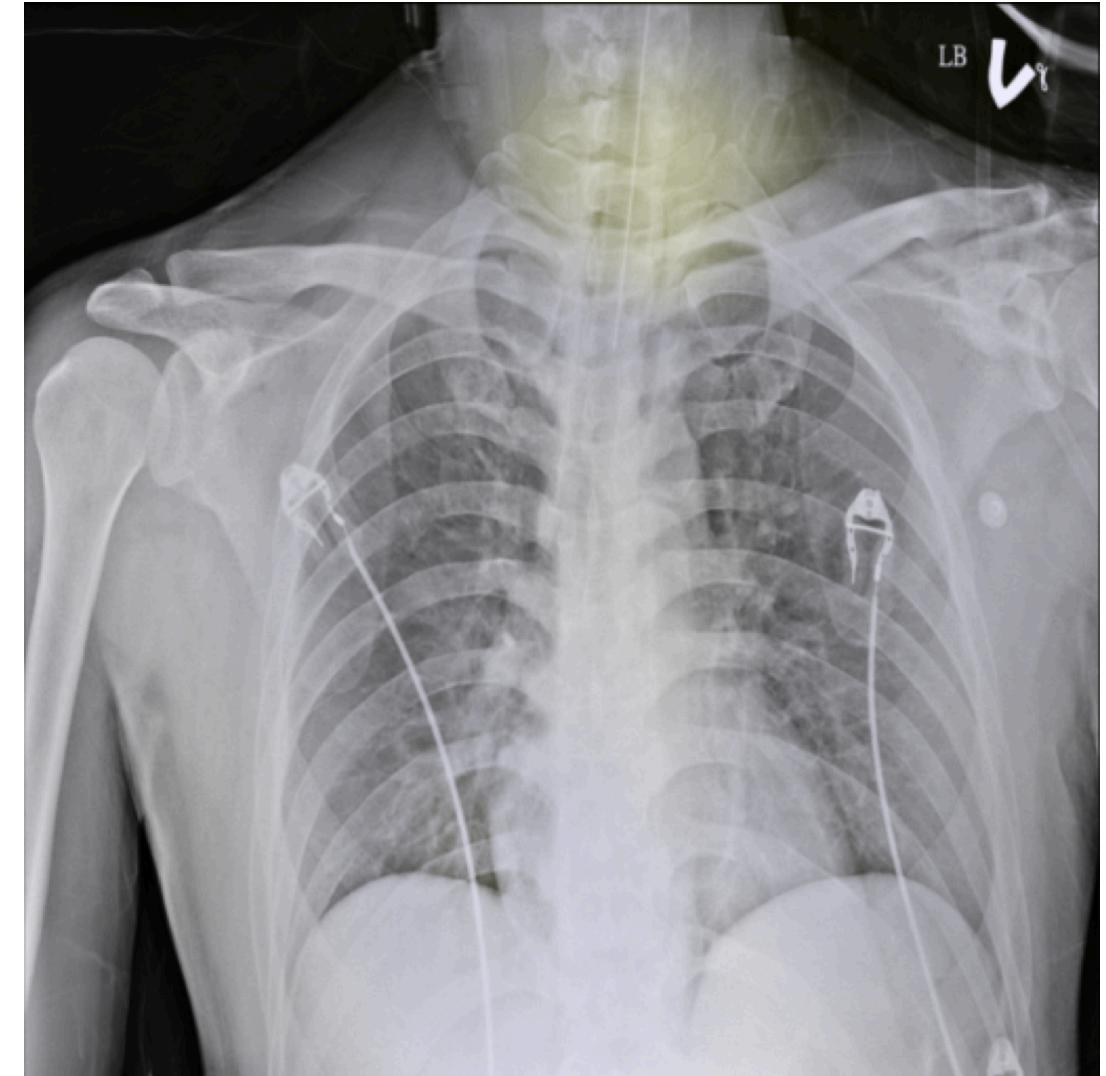
REPORT(描述)：  
最多約十行資料  
約510 個英文字  
診斷：

診斷已覆核無法再調整

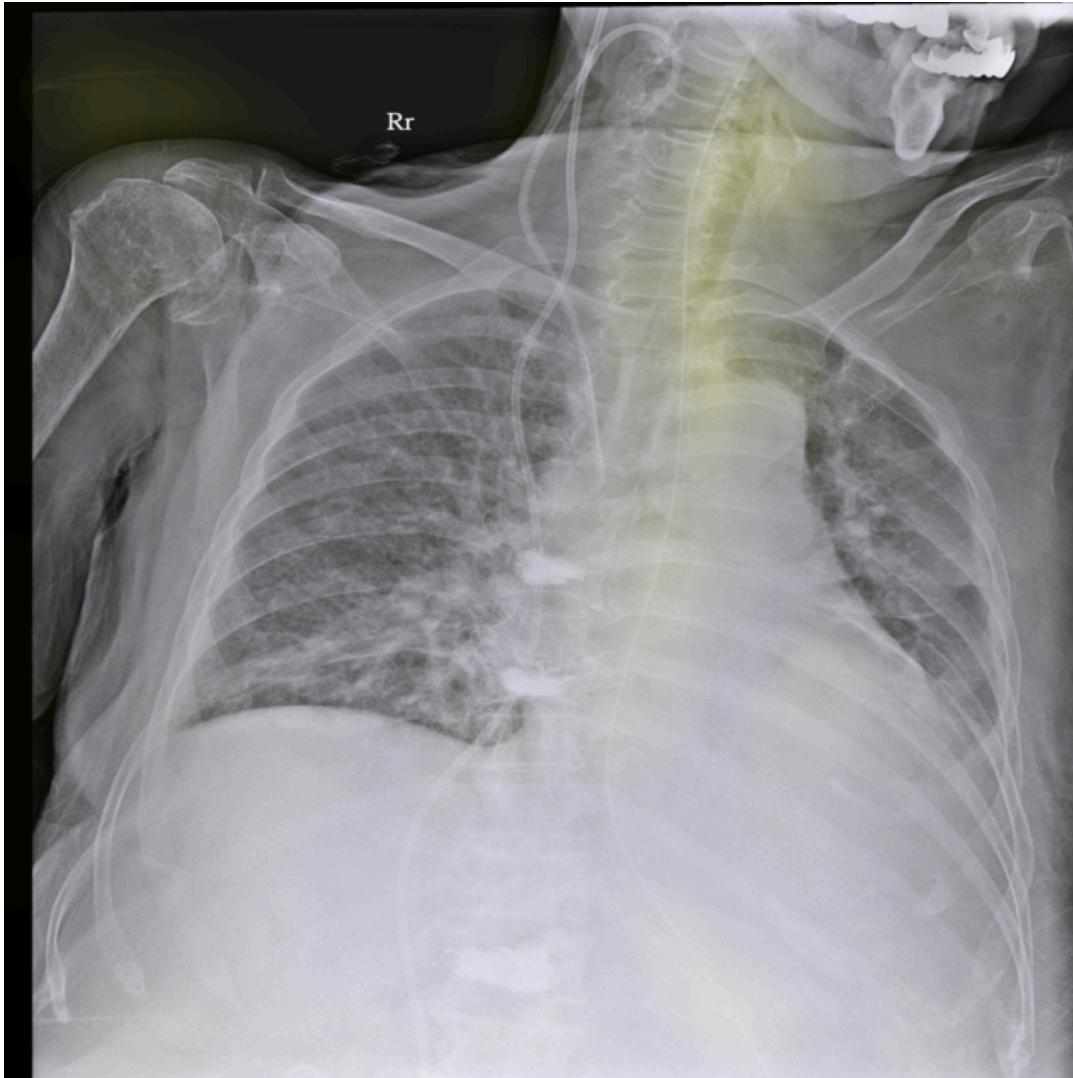




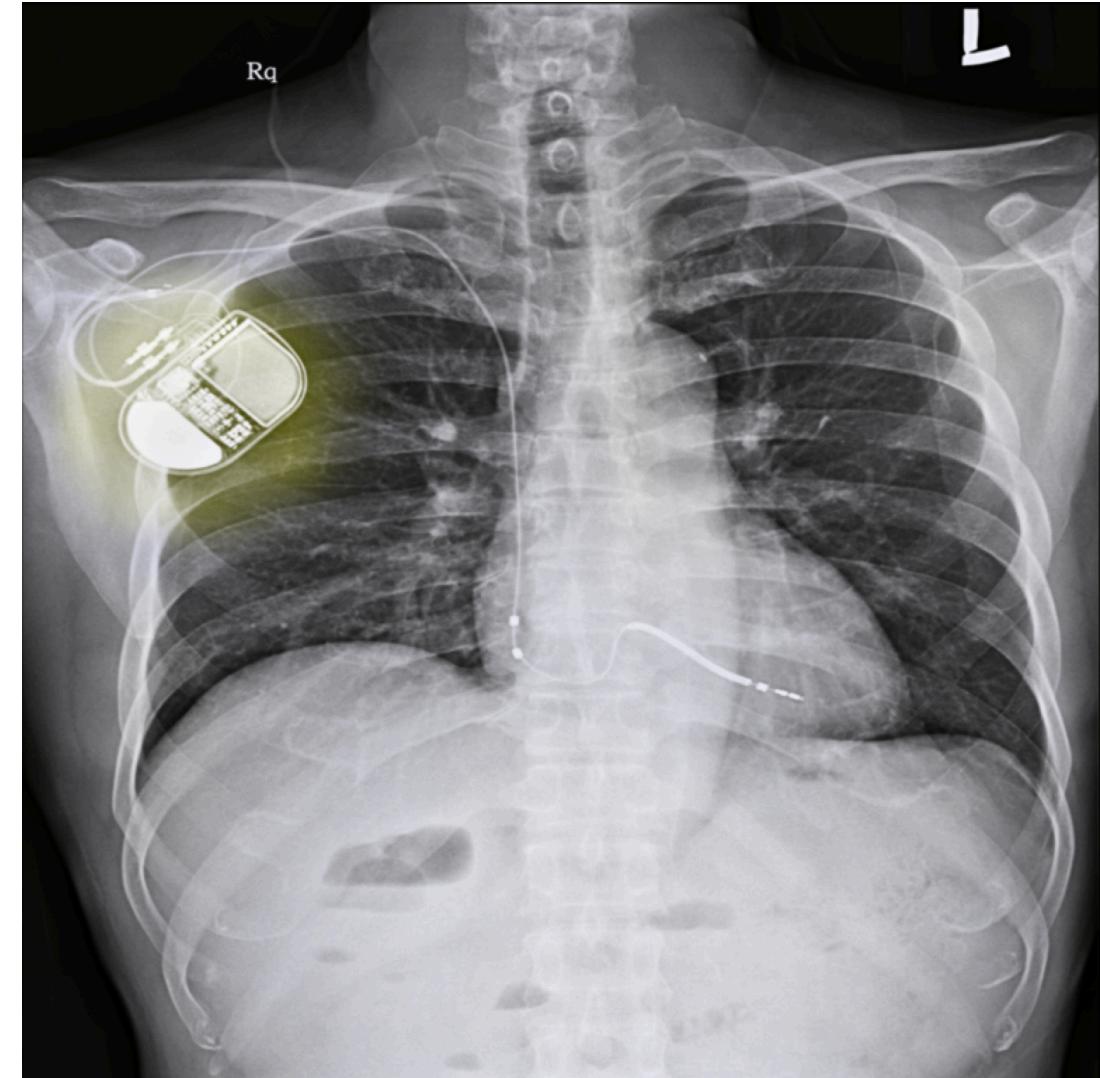
Central Venous Catheter/中心靜脈導管  
**97.98%**



Endotracheal Tube/氣管內管  
**98.18%**



Nasogastric Tube/鼻胃管  
**95.42%**



Pacemaker/心律調節器  
**100%**

# 醫學影像標註系統的設計考量

- 標註者多為領域專家，UI Flow 宜貼近原工作之習慣
- 結合醫院的報告系統，可透過關鍵字進行預標註
- 標註一致性的提升：
  - 事先溝通
  - 專案說明
  - 標註與確認
- 隱私權的考量：去識別化功能
- 訓練集(train set)、驗證集(validation set)、測試集(test set)要考量同一人不同影像的情況

# 標註資料的品質

- NIH Chest X-ray的分類：0, 低度懷疑, 中度懷疑, 高度懷疑

00004928\_001

No Finding, 28Y, PA

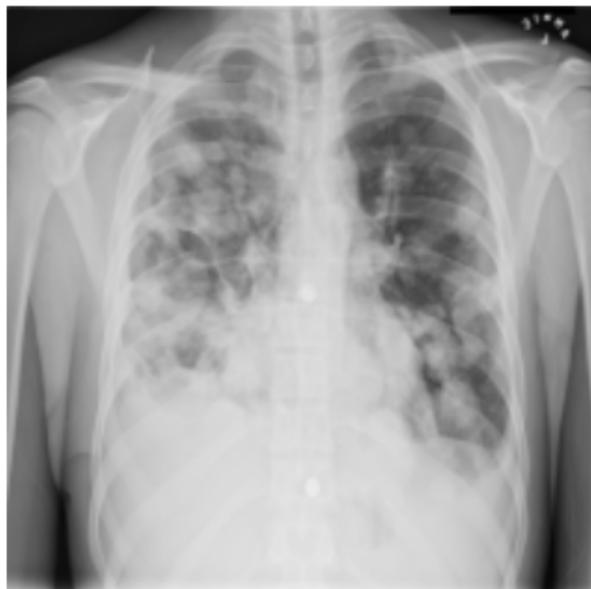
NIH註解

Nodules: 高度懷疑

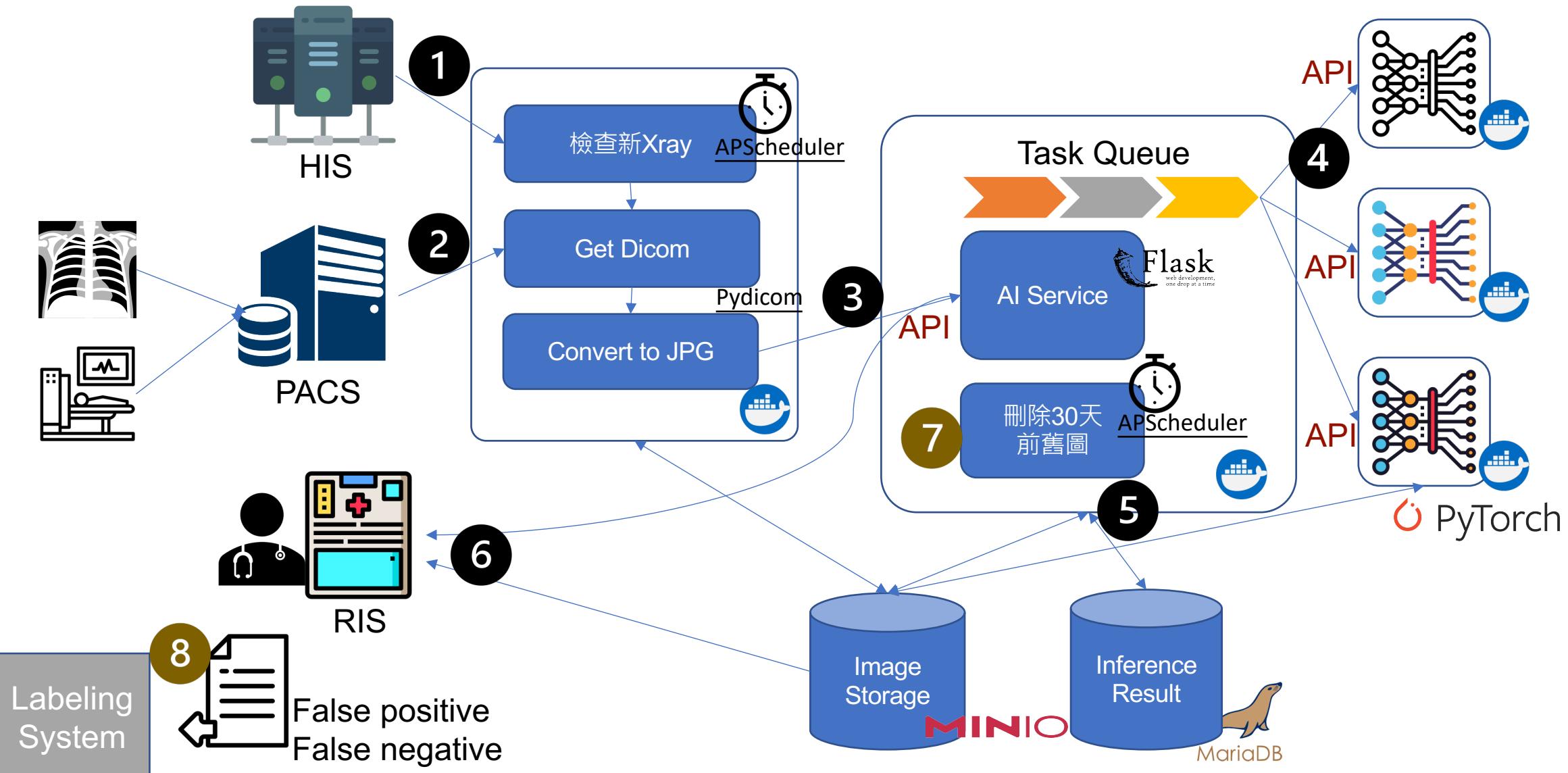
00004928\_002

Nodule, 28Y, AP

Nodules: 高度懷疑



# 即時胸部X光腫瘤AI平台

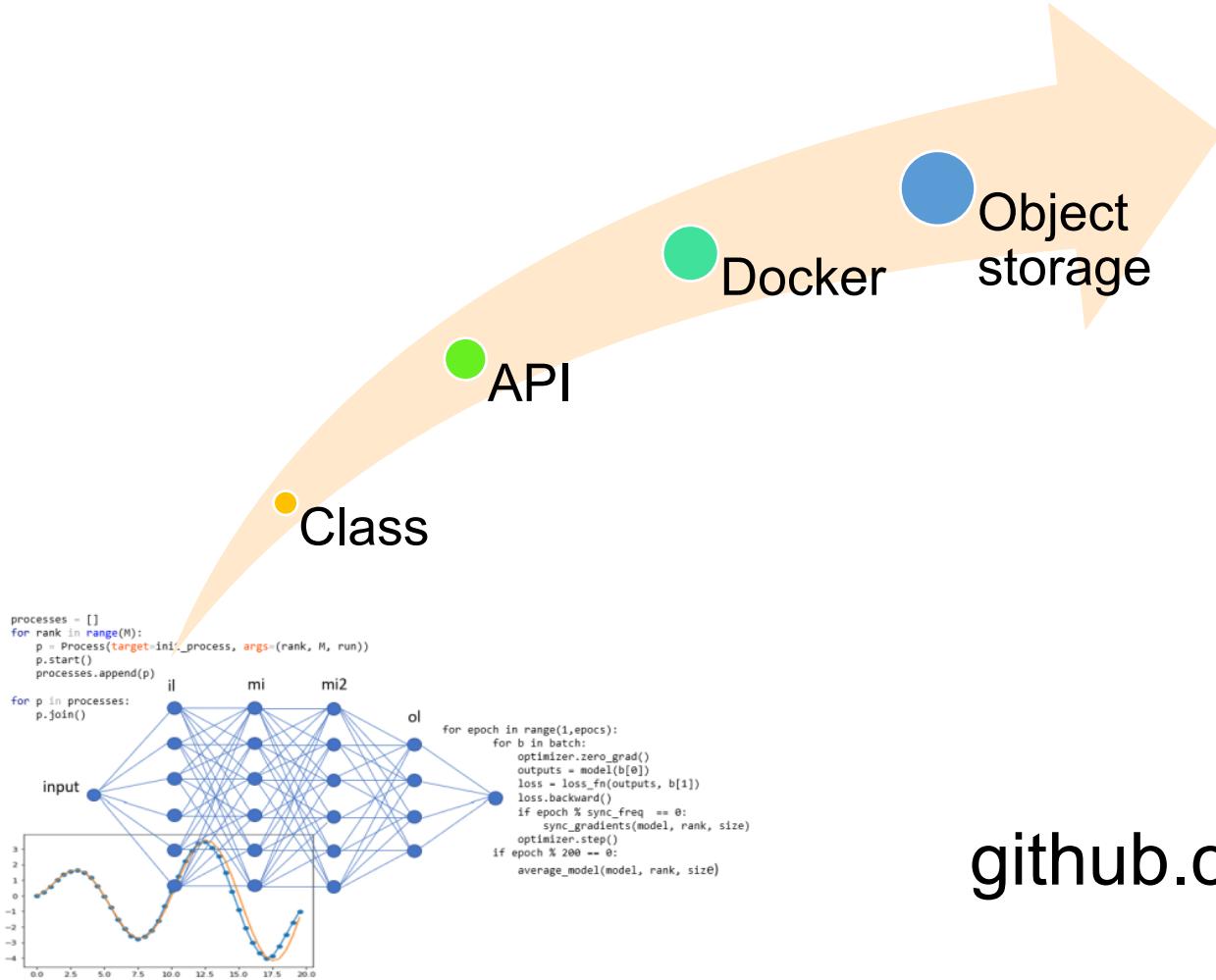


# 深度學習模型封裝

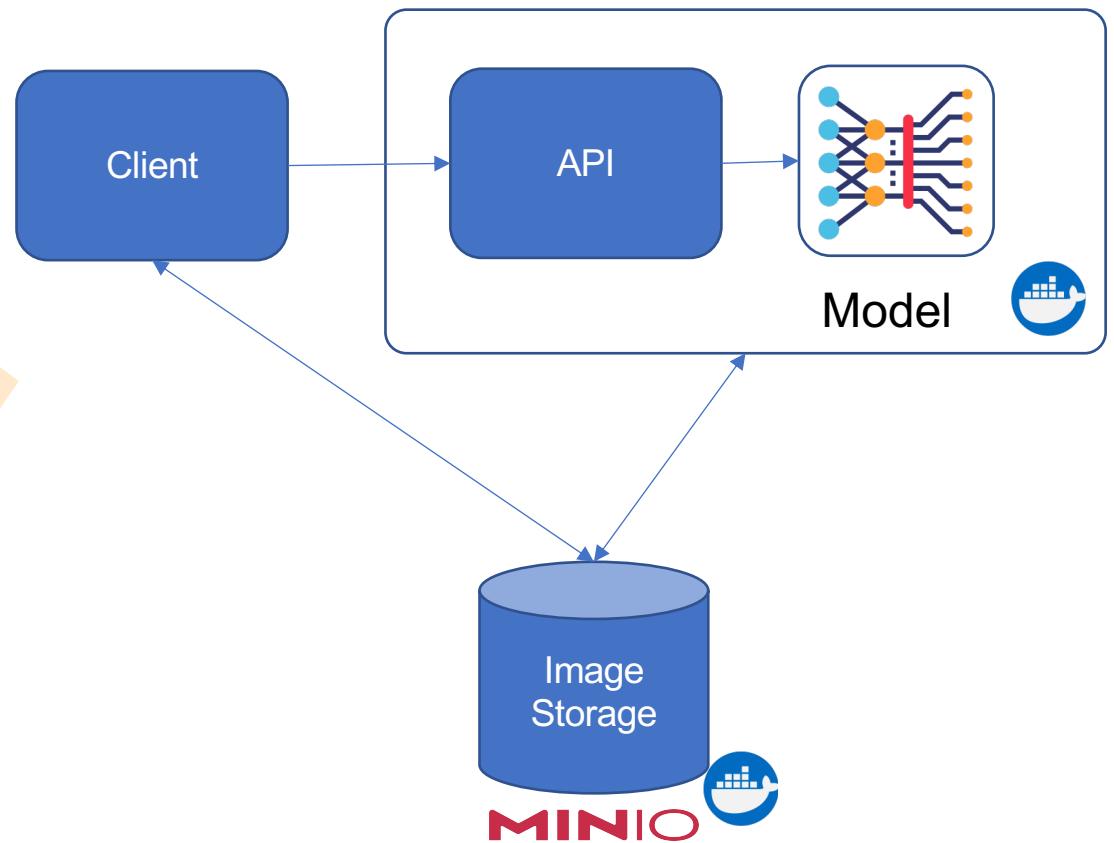
---

以 PyTorch 為例  
AI Model/ API/ Client

# AI Model/ API/ Client



[github.com/cclai999/pycontw2020-demo](https://github.com/cclai999/pycontw2020-demo)



# V1-Script

```
└── v1-script
    └── main.py
```

# V1-Script, main.py (1/2)

```
# build a transform; load image,  
transform = transforms.Compose([  
    transforms.Resize(256),  
    transforms.CenterCrop(224),  
    transforms.ToTensor(),  
    transforms.Normalize(  
        mean=[0.485, 0.456, 0.406],  
        std=[0.229, 0.224, 0.225]  
    )])  
  
img = Image.open("cat.jpg")  
img_t = transform(img)  
batch_t = torch.unsqueeze(img_t, 0)
```

# V1-Script , main.py (2/2)

```
# load model, and do inference
```

## **# First, load the model**

```
resnet = models.resnet101(pretrained=False)
state_dict = torch.load("resnet101-5d3b4d8f.pth")
resnet.load_state_dict(state_dict)
```

## **# Second, put the network in eval mode**

```
resnet.eval()
```

## **# Third, carry out model inference**

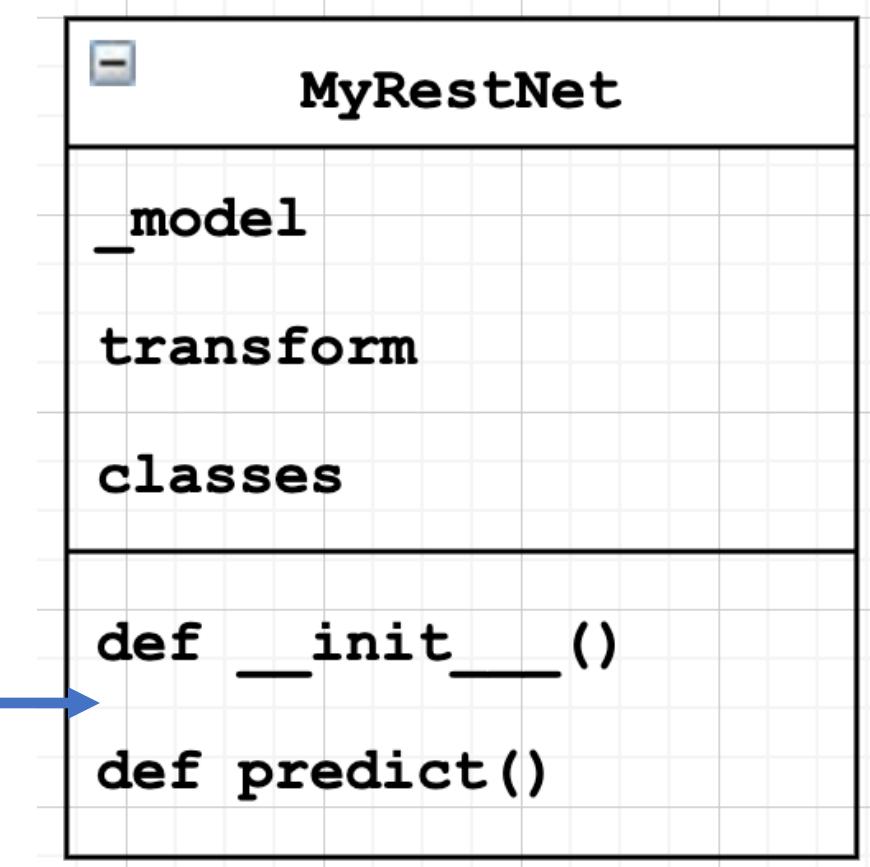
```
out = resnet(batch_t)
```

```
_ , index = torch.max(out, 1)
```

```
percentage = torch.nn.functional.softmax(out, dim=1) [0] * 100
print(classes[index[0]], percentage[index[0]].item())
```

# V2-model-class

```
|- v2-model-class
  |- main.py
  |- model
    |- __init__.py
    |- myresnet.py
```



# V2-model-class, myresnet.py (1/2)

```
# wrap AI model into a class

class MyResnet():
    def __init__(self, model_state_path: str,
                 classes_names_path: str):
        # load the model
        self._model = models.resnet101(pretrained=False)
        state_dict = torch.load(model_state_path)
        self._model.load_state_dict(state_dict)
        # put the network in eval mode
        self._model.eval()
        # create an image transform
        .....
        # load names of all classes
        .....
```

## V2-model-class, myresnet.py (2/2)

```
# wrap AI model into a class

class MyResnet():

    def predict(self, img: Image):
        img_t = self.transform(img)
        batch_t = torch.unsqueeze(img_t, 0)

        # carry out model inference
        out = self._model(batch_t)

        _, index = torch.max(out, 1)
        percentage = torch.nn.functional.softmax(out,
dim=1)[0] * 100

        return self.classes[index[0]],
               percentage[index[0]].item()
```

# V2-model-class, main.py

```
# pass an image to MyResnet and do predict
from PIL import Image
from model import MyResnet

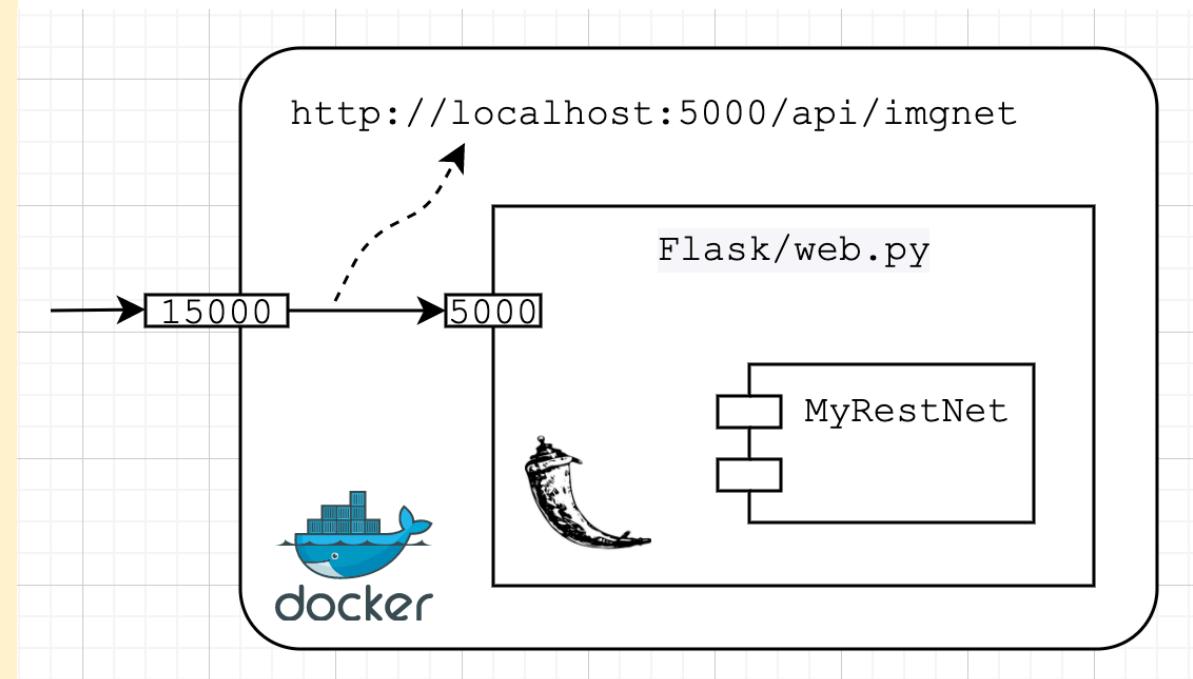
imgnet = MyResnet("resnet101-5d3b4d8f.pth",
                  "imagenet_classes.txt")

img = Image.open("../test-data/cat.jpg")

class_name, percentage = imgnet.predict(img)
```

# V3-docker

```
v3-docker
├── Dockerfile
├── ...
├── docker-compose.yml
└── model
    ├── __init__.py
    ├── myresnet.py
    ├── ...
    └── requirements.txt
        ...
    └── web.py
```



# V3-docker, web.py

```
# apply Flask to wrap model class into an API

@app.route('/api/imgnet', methods=['POST'])
def inference():
    req_data = request.get_json()
    if req_data.get('img_url', None):
        img = Image.open('....')
        class_name, percentage = imgnet.predict(img)
    result = {
        "status": "success",
        "status_code": 200,
        "img_url": req_data['img_url'],
        "class_name": class_name,
        "percentage": percentage,
    }
    return jsonify(result), 200
```

# V3-docker, Dockerfile

```
# 使用 Docker 封裝 Restful API
```

```
FROM aibase:0.1.01beta
```

```
....
```

```
WORKDIR /root/aimodel
```

## **# add app**

```
COPY model model
```

```
COPY web.py .
```

```
....
```

## **# start web**

```
CMD gunicorn --bind 0.0.0.0:5000 web:app
```

# V3-docker, docker-compose.yml

```
# 使用 docker-compose 來執行 docker container
```

```
version: '3.7'
```

```
services:
```

```
aimodel:
```

```
    image: aimodel:pycontw2020
```

```
    build:
```

```
        context: .
```

```
    ports:
```

```
        - "15000:5000"
```

# V4-docker-w-minio

```
└── v4-docker-w-minio
    ├── Dockerfile
    ├── ...
    ├── docker-compose.yml
    └── model
        ├── __init__.py
        └── myresnet.py
    └── model-state
        ├── config.yaml
        ├── ...
    └── requirements.txt
    └── web.py
```

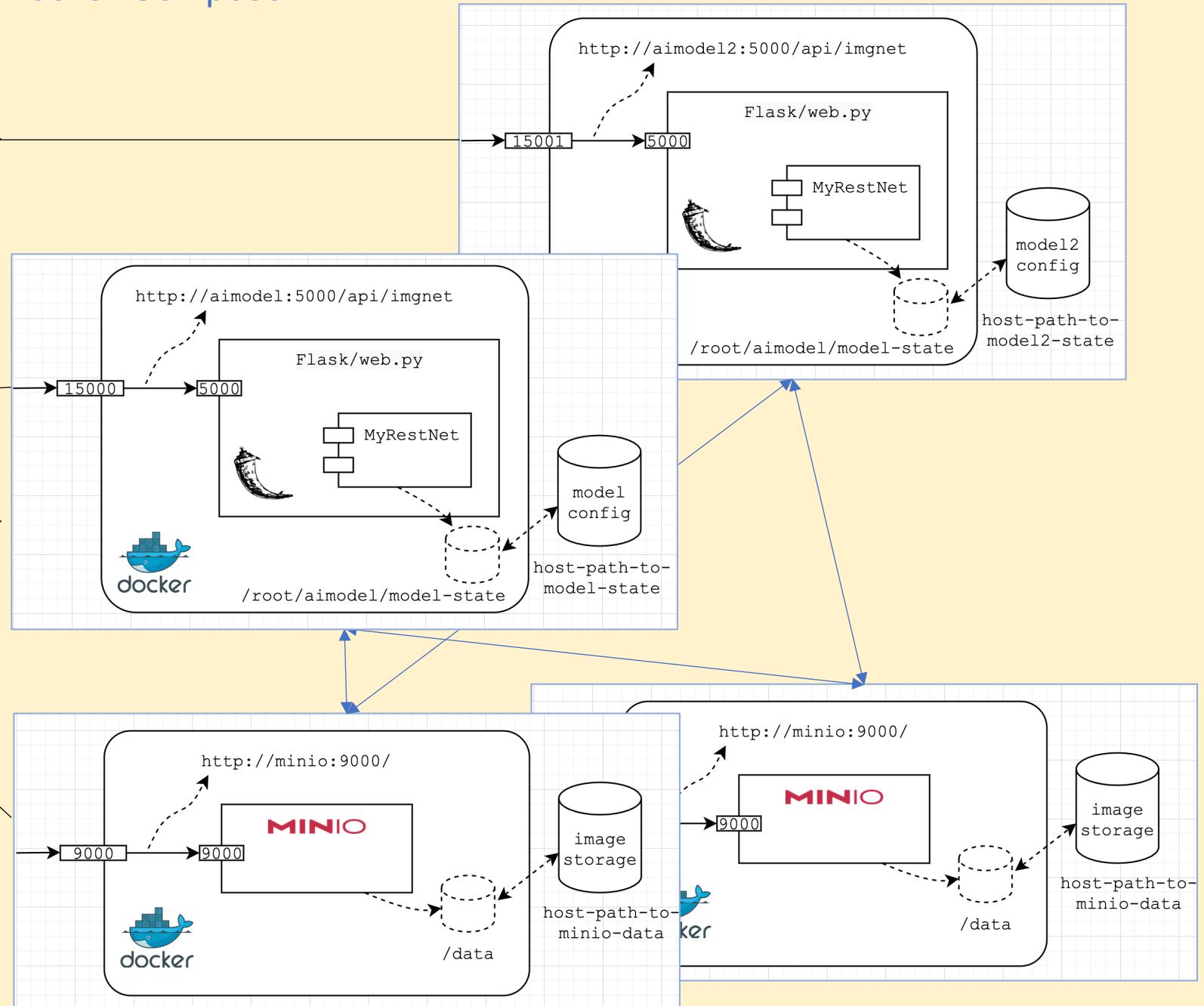
## Docker-Compose

http://127.0.0.1:15001/api/imgnet

ai-client.pay

http://127.0.0.1:15000/api/imgnet

http://127.0.0.1:9000



# V4-docker-w-minio, docker-compose.yml(1/2)

```
# model-state 目錄提出到 host machine

version: '3.7'

services:
  aimodel:
    image: aimodel2:pycontw2020
    build:
      context: .
    ports:
      - "15000:5000"
  volumes: # 更改 AI model 不必重新再 build docker image
    - "./model-state:/root/aimodel/model-state"
```

# V4-docker-w-minio, docker-compose.yml(2/2)

```
# 新增一個 service: minio (container)
```

```
minio:  
  image: minio/minio  
  container_name: minio  
  ports:  
    - "9000:9000"  
  environment:  
    MINIO_ACCESS_KEY: minio  
    MINIO_SECRET_KEY: minio123  
  command: server /data  
volumes: # image storage bind 到 host machine  
  - "./minio-data:/data"
```

# V4-docker-w-minio, ai-client.py

```
# client 先上傳一張影像到 minio, 再 call AI API

minioClient = Minio('127.0.0.1:9000', access_key='minio',
                     secret_key='minio123', secure=False)

try:
    s1 = minioClient.fput_object('mybucket', 'violin.jpg', 'violin.jpg')
except ResponseError as err:
    ...

r = requests.post('http://127.0.0.1:15000/api/imgnet', json={"img_url":
"violin.jpg"})
if r.status_code == 200:
    print(r.text)
else:
    ...
```

# V4-docker-w-minio, web.py(1/2)

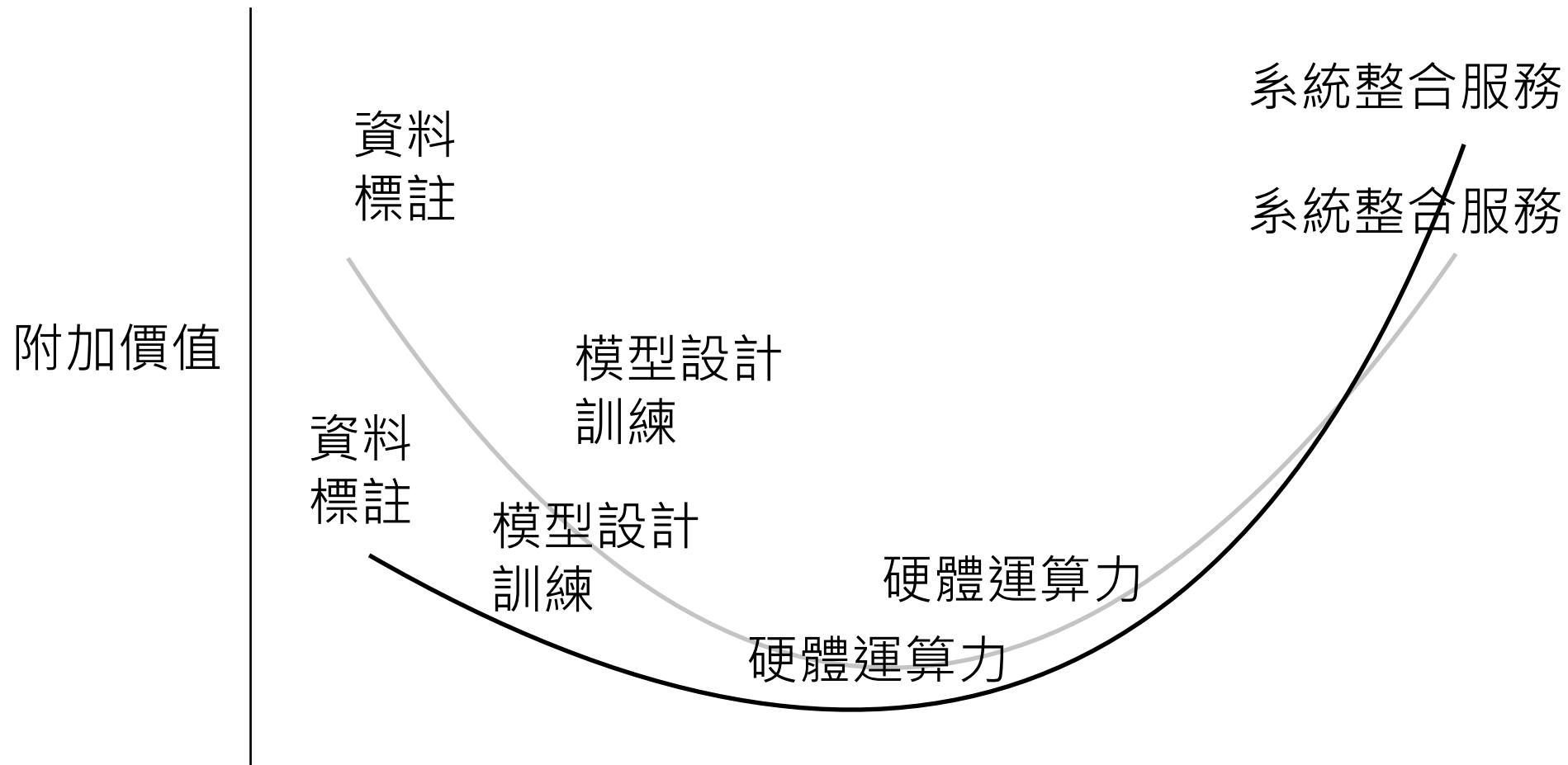
```
# 讀取 host machine 之 model-state 目錄的 model config
```

# V4-docker-w-minio, web.py(2/2)

```
def get_minio_img(img_url):
    resp = minioClient.get_object("mybucket", img_url)
    return Image.open(io.BytesIO(resp.data))

@app.route('/api/imgnet', methods=['POST'])
def inference():
    req_data = request.get_json()
    if req_data.get('img_url', None):
        img = get_minio_img(req_data["img_url"])
        class_name, percentage = imgnet.predict(img)
        result = {
            ...
            "class_name": class_name,
            "percentage": percentage,
            "model_name": model_config['AIMODEL_NAME'],
            "model_version": model_config['AIMODEL_VERSION'],
        }
        return jsonify(result), 200
```

# 結論：AI服務的價值





童 純 合  
醫療社團法人 童綜合醫院  
Tungs' Taichung MetroHarbor Hospital

- 資料標註生產線
- 高品質的資料
- 應用場域

誠徵前端及後端工程師

歡迎加入我們AI團隊，一起來探索醫療新應用

謝謝您的聆聽