

รายงาน

เรื่อง โมเดล Computer Vision แยกภาพ Xray วัณโรคปอด

จัดทำโดย

นาย พิสิษฐ์ วงศ์ศรีพิสันต์

6331332021

อาจารย์ที่ปรึกษา

อาจารย์ เอกพล ช่างสุวนิช

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา Individual Study 2110291

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ภาคเรียนที่ 1 ปีการศึกษา 2564

ความเป็นมาและวัตถุประสงค์

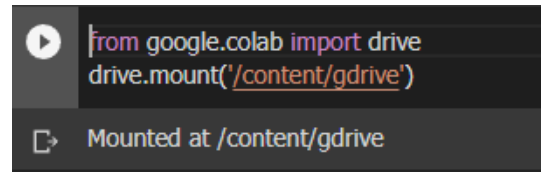
เนื่องจากเทคโนโลยีทางด้าน AI ในปัจจุบัน ในแขนง Computer Vision สามารถทำให้คอมพิวเตอร์สามารถจดจำ มองเห็น และเข้าใจกับข้อมูลรูปภาพ ที่ได้รับเข้ามา จึงทำให้เกิดการนำเทคโนโลยีนี้มาประยุกต์ใช้ในศาสตร์ต่างๆ รวมถึงด้านการแพทย์ที่นำมาใช้ประกอบการวินิจฉัยโรคต่างๆจากข้อมูลรูปภาพ Xray ให้เกิดการวินิจฉัยที่แม่นยำมากยิ่งขึ้น

วัตถุประสงค์ของโครงการนี้

1. เพื่อศึกษาหลักการทำงานของ Machine Learning เบื้องต้น
2. เพื่อศึกษาการนำ Architecture resnet มาใช้งาน
3. เพื่อศึกษาเทคนิคการ Train Model ให้ได้ Accuracy ที่ดีขึ้น
4. เพื่อศึกษาการนำความรู้ด้าน Deep Learning มาประยุกต์ใช้ในด้านต่างๆ
5. เพื่อฝึกการศึกษาหาความรู้ด้วยตนเอง

ขั้นตอนการทำงาน / สิ่งที่ทำ

1. เชื่อมต่อ Folder ใน google drive ของตัวเอง กับ google colab เพื่อให้สามารถดึงข้อมูลรูปภาพ และ export model เก็บไว้ใน google drive



2. Download dataset ใน Kaggle (tuberculosis-chest-xrays-shenzhen) ลงใน google drive จากนั้นทำการ unzip ไฟล์ออกมา

```
[ ] from google.colab import files
files.upload() #this will prompt you to update the json

!pip install -q kaggle
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!ls ~/.kaggle
!chmod 600 /root/.kaggle/kaggle.json # set permission
```

เลือกไฟล์ ไม่ได้เลือกไฟล์ใด Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving kaggle.json to kaggle.json
kaggle.json

```
[ ] !kaggle datasets download -d raddar/tuberculosis-chest-xrays-shenzhen -p /content/gdrive/MyDrive/Colab Notebooks
```

404 - Not Found

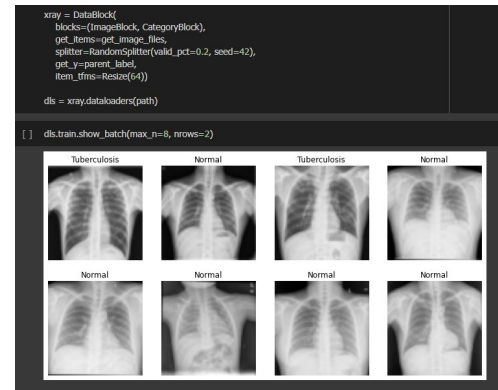
```
[ ] import os
os.chdir('/content/gdrive/MyDrive/Colab Notebooks') #change dir
!unzip -q tuberculosis-chest-xrays-shenzhen.zip
```

3. ลองใช้ library pandas ในการเปิดไฟล์ csv ที่ download มาจาก Kaggle ซึ่งบอกข้อมูลว่าชื่อไฟล์รูปไหนเป็นภาพ Xray ปอดที่ปกติ หรือเป็นวัณโรค

```
[ ] import pandas as pd
import os
os.chdir('/content/gdrive/MyDrive/Colab Notebooks') #change dir
df=pd.read_csv('./shenzhen_metadata.csv')
df
```

	study_id	sex	age	findings
0	CHNCXR_0001_0.png	Male	45	normal
1	CHNCXR_0002_0.png	Male	63	normal
2	CHNCXR_0003_0.png	Female	48	normal
3	CHNCXR_0004_0.png	Male	58	normal
4	CHNCXR_0005_0.png	Male	28	normal
...
657	CHNCXR_0658_1.png	Male	41	bilateral secondary PTB with right pneumothorax
658	CHNCXR_0659_1.png	Male	33	secondary PTB in the left upper field
659	CHNCXR_0660_1.png	Male	50	1.bilateral secondary PTB with right upper atelectasis;2.right pleural adhesions;3.left compensatory emphysema
660	CHNCXR_0661_1.png	Male	26	bilateral secondary PTB with right pleural thickening
661	CHNCXR_0662_1.png	Male	32	secondary PTB in the right upper field

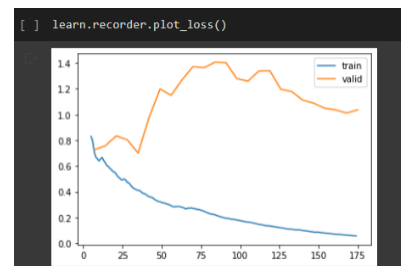
4. สร้าง data loader ซึ่งข้อมูลรูปภาพทั้งหมด 662 รูป แบ่งเป็น ปอดปกติ 326 รูป และวัณโรคปอด 336 รูป ใช้วิธีแบ่ง data โดยแบ่ง test set ออกมาก่อน 10% คือรูปปอดปกติ 32 รูป และ รูปวัณโรคปอดอีก 33รูป รวมทั้งสิ้น 65รูป จากนั้นรูปที่เหลือแบ่งเป็น training set 80% และ validation set 20%



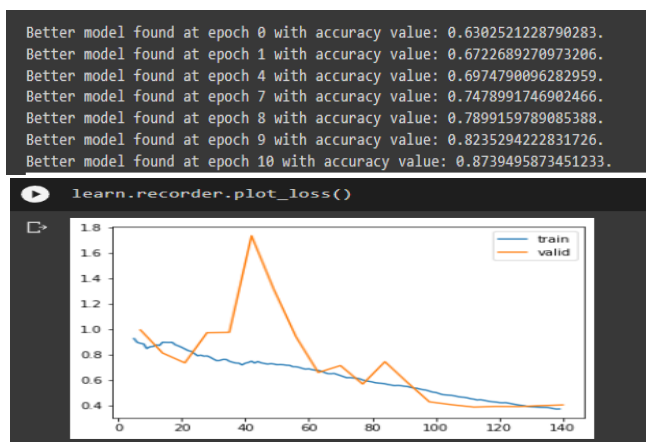
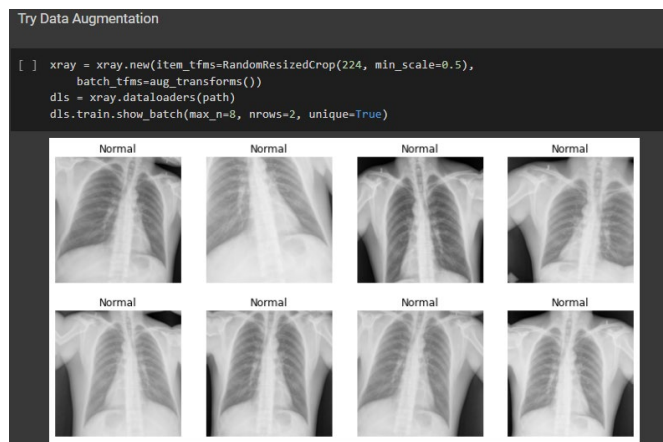
5. Model ที่1 ลอง Train model ทันที โดยใช้ resnet50 architecture และใช้มาตรวัดเป็น Accuracy จะได้ค่า Accuracy จะอยู่ราว ๆ 78% – 80%

```
# Train with resnet50
learn = cnn_learner(dls, resnet50, metrics=accuracy)
learn.fine_tune(25, cbs=[SaveModelCallback(monitor='accuracy')])

Better model found at epoch 0 with accuracy value: 0.630251228790283.
Better model found at epoch 1 with accuracy value: 0.6722689270973206.
Better model found at epoch 4 with accuracy value: 0.6974790096282959.
Better model found at epoch 7 with accuracy value: 0.7478991746902466.
Better model found at epoch 8 with accuracy value: 0.7899159789085388.
Better model found at epoch 9 with accuracy value: 0.8235294222831726.
Better model found at epoch 10 with accuracy value: 0.8739495873451233.
```

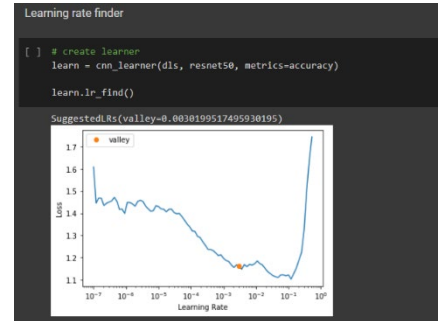


6. Model ที่2 train model โดยใช้เทคนิค data augmentation ผ่านฟังก์ชัน `aug_transforms()` ซึ่งจะทำการ flip / rotate / zoom / warp / lighting transforms ภาพ จากนั้น train โดยใช้ resnet50 architecture และใช้มาตรวัดเป็น Accuracy ซึ่งจะได้ Accuracy อยู่ราว ๆ 87%



7. Model ที่ 3 train model โดยใช้เทคนิค data augmentation และ discriminative learning rate ได้ Accuracy อยู่ที่ 90%

- เริ่มจากการทำ Learning rate finder เราจะเลือก learning rate ที่ค่า loss ลดลงด้วยความชันมากที่สุด



- จากนั้น train model เฉพาะ layer ของ classifier และ freeze weight ของ layer ก่อนหน้าไว้ทั้งหมด , train ด้วย จำนวน epoch น้อยๆ ประมาณ 1-3 epoch เพื่อปรับจูนค่า weight ใน layer ของ classifier กับข้อมูลชุดใหม่ก่อน

```
[ ] learn = cnn_learner(dls, resnet50, metrics=accuracy)
learn.fit_one_cycle(3, 0.0030199517495930195)
```

epoch	train_loss	valid_loss	accuracy	time
0	1.193958	2.919961	0.529412	01:11
1	1.274510	1.226995	0.680672	01:11
2	1.153265	1.076876	0.663866	01:11

- Unfreeze weight ใน layer ก่อนหน้า classifier จากนั้นทำ learning rate finder อีกรอบ เนื่องจากตอนนี้เรามี layer ที่ต้อง train มากขึ้น และ weights ใน layer classifier ถูก train มาบ้างแล้ว แปลว่าค่า learning rate ก่อนหน้านี้ที่หามาได้นั้น อาจจะไม่ใช่ว่าที่เหมาะสมแล้ว



- จากนั้นใช้เทคนิค discriminative learning rate โดยกำหนดให้ learning rate ใน layer ต่ำกว่า learning rate ใน layer ทั่วๆ ผ่าน function slice เริ่มจาก learning rate ที่ค่อยๆเพิ่มสูงขึ้น แล้วจึงค่อยๆลดลงต่ำกว่า learning rate เริ่มต้นในตอนแรก เพื่อเป็นการ warm up ช่วงแรก แล้วจึงค่อยๆเข้าสู่จุดที่ weight ของ model เริ่มเสถียรด้วยการลด learning rate ลง

Discriminative learning rate

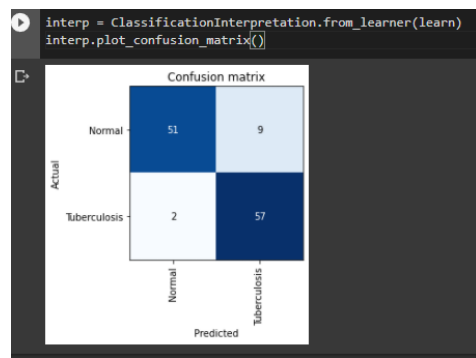
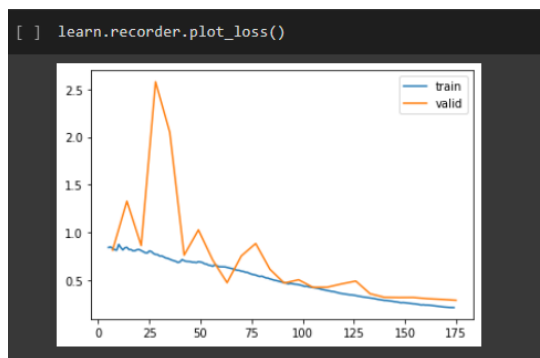
```
learn = cnn_learner(dls, resnet50, metrics=accuracy)
learn.fit_one_cycle(3, 0.0014454397605732083)
learn.unfreeze()
learn.fit_one_cycle(25, lr_max=slice(0.0002290867705596611, 6e-4), cbs=[SaveModelCallback(monitor='accuracy')])
```

ผลลัพธ์ของงาน

จากการที่ได้ลอง Train Model ทั้ง 3 รูปแบบ ได้ผลลัพธ์ว่า model ที่ 3 ได้ Accuracy ที่ทดสอบใน validation set สูงที่สุด อยู่ที่ 90% ซึ่งเป็น model ที่ใช้เทคนิค data augmentation และ discriminative learning rate

รูปด้านล่าง แสดงผลลัพธ์จากการ Train model ที่ 3 บน validation set

```
Better model found at epoch 0 with accuracy value: 0.7226890921592712.  
Better model found at epoch 5 with accuracy value: 0.7899159789085388.  
Better model found at epoch 6 with accuracy value: 0.8403361439704895.  
Better model found at epoch 15 with accuracy value: 0.8739495873451233.  
Better model found at epoch 18 with accuracy value: 0.8823529481887817.  
Better model found at epoch 19 with accuracy value: 0.9075630307197571.
```



เนื่องจาก model ที่ 3 ได้ค่า Accuracy สูงสุด ในการทดสอบบน validation set เราจึงเลือก model นี้มาใช้ทดสอบกับข้อมูล test set ของเรต่อไป ซึ่งผลลัพธ์ของการ Predict รูปบน test set ได้ Accuracy อยู่ที่ 90.769%

```
Use third model to predict the test set  
  
[ ] learner = load_learner("/content/gdrive/MyDrive/Colab Notebooks/models/thirdModel.pkl")  
  
[ ] path = Path("/content/gdrive/MyDrive/Colab Notebooks/test set")  
all_images = get_image_files(path)  
  
correct_pred = 0  
wrong_pred = 0  
  
for im in all_images:  
    result = learner.predict(im)[0][:]  
    if(result == "Normal" and im.name[-5] == "0"):  
        correct_pred += 1  
    elif(result == "Tuberculosis" and im.name[-5] == "1"):  
        correct_pred += 1  
    else:  
        wrong_pred += 1  
  
print("correct_predict = ", correct_pred)  
print("wrong_predict = ", wrong_pred)  
print("accuracy = " + str( (correct_pred/(correct_pred+wrong_pred))*100 ) + " %")  
  
correct_predict = 59  
wrong_predict = 6  
accuracy = 90.76923076923077 %
```

อุปสรรคที่เกิดขึ้น และวิธีแก้ปัญหา

อุปสรรค1 : การเรียนคลิปวิดีโอ Fast AI เป็นภาษาอังกฤษ ซึ่งบางทีเกิดปัญหาการแปลความหมาย

วิธีแก้ปัญหา : Search Google เพื่อหาความหมายของศัพท์นั้นๆ และหาแหล่งเรียนรู้ภาษาไทยเพิ่มเติม

อุปสรรค2 : ไม่เคยใช้ Google Colab ในการทำงาน และไม่เคยดึงข้อมูลลง Google Colab จาก Kaggle

วิธีแก้ปัญหา : สอบถามอาจารย์ และลองค้นคว้าด้วยตนเอง

สิ่งที่ได้รับจาก Individual Study ในครั้งนี้

1. ได้เรียนรู้การทำ Machine Learning เบื้องต้น
2. ทราบเทคนิคการ Train Model ให้ได้ Accuracy ที่ดีขึ้น เช่น Data augmentation / Discriminative learning rate
3. ทราบปัญหาที่อาจพบจากการ Train model เช่น overfitting / underfitting / การเลือกค่า learning rate ที่ไม่เหมาะสม / การทำ data augmentation มากไปจนทำให้รูปภาพในการ Train ไม่มีคุณภาพ
4. ได้เรียนรู้การใช้งาน Google colab เชื่อมกับ Google drive
5. ได้เรียนรู้การดึงข้อมูลจากเว็บ Kaggle
6. ได้เรียนรู้การใช้ library pandas ในการอ่านข้อมูลไฟล์ CSV