

รายงานสรุปผลการศึกษา

วิชา 2110391 เอกซ์ศึกษาทางวิศวกรรมคอมพิวเตอร์ 3

(Individual Study in Computer Engineering III)

เรื่อง

Tuberculosis Chest X-rays

จัดทำโดย

นายพีรณัฐ ชีระวัฒน์ชัย

รหัสประจำตัวนิสิต 6231343821

เสนอ

อาจารย์ เอกพล ช่างสุวนิช

รายงานฉบับนี้เป็นส่วนหนึ่งของการศึกษารายวิชาเอกซ์ศึกษาทางวิศวกรรมคอมพิวเตอร์ 3

จุฬาลงกรณ์มหาวิทยาลัย

ภาคเรียนที่ 1 ปีการศึกษา 2564

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของการศึกษารายวิชาเอกศึกษาทางวิศวกรรมคอมพิวเตอร์ 3 (Individual Study in Computer Engineering III) จัดทำขึ้นเพื่อสรุปสิ่งที่ได้เรียนรู้จากวิชานี้ โดยได้ร่วมทำโปรเจกต์เรื่อง Machine Learning เกี่ยวกับการทำนายการเป็นวัณโรคจาก Chest X-ray image ด้วยการใช้ Deep Learning โดยในรายงานเล่มนี้ประกอบด้วยเนื้อหา ดังนี้

1. ความเป็นมาและวัตถุประสงค์
2. ขั้นตอนการทำงาน สิ่งที่ทำ
3. ผลลัพธ์ของงาน
4. อุปสรรคที่เกิดขึ้น วิธีแก้ปัญหา และสิ่งที่ได้รับจาก Individual Study นี้

ทั้งนี้ข้าพเจ้าหวังเป็นอย่างยิ่งว่ารายงานเล่มนี้จะเป็นประโยชน์ต่อผู้ที่ได้มาศึกษาเป็นอย่างดี และหากมีข้อผิดพลาดประการใดขออภัยมา ณ โอกาสนี้ด้วย

นายพีรณัฐ ธีระวัฒนชัย

ผู้จัดทำ

สารบัญ

รายละเอียด	หน้า
1. ความเป็นมาและวัตถุประสงค์	4
2. ขั้นตอนการทำงาน สิ่งที่ทำ	4
3. ผลลัพธ์ของงาน	5
4. อุปสรรคที่เกิดขึ้น วิธีแก้ปัญหา และสิ่งที่ได้รับจาก Individual Study นี้	12

1. ความเป็นมาและวัตถุประสงค์

- ความเป็นมา

ในปัจจุบันการตรวจหาความผิดปกติที่บริเวณต่างๆภายในร่างกายมนุษย์ เช่น ส่วนนอก ส่วนหัวใจ ส่วนปอด เป็นต้น อาจจะต้องใช้ค่าใช้จ่ายเป็นจำนวนมากและอาจต้องใช้เวลาานพอสมควร แต่ในบางครั้งเราก็สามารถใช้เทคโนโลยีมาประยุกต์ใช้เพื่อทำนายว่าบริเวณเหล่านั้นมีสิ่งผิดปกติอยู่หรือไม่ เพื่อนำไปใช้วิเคราะห์ทางการแพทย์ต่อไป

โดยในงานนี้ผมได้รับมอบหมายให้นำข้อมูลภาพ Chest X-ray มาใช้กระบวนการ Machine Learning ด้วยการนำ Convolutional Neural Network (CNN) เพื่อทำนายการเป็นวัณโรค โดยเป้าหมายหลัก คือ ต้องการให้ Model ที่ Train ออกมามีความแม่นยำมากที่สุดกับ Testing set (ได้ model ที่ไม่ Overfit เกินไป) โดยในระหว่างการทำได้พบว่าเมื่อเราทำการ Normalized ภาพที่ได้มาก่อนด้วยการหารด้วย 255 แล้วหาค่าเฉลี่ยของ Intensity value จากทุกรูป แล้วหารด้วย Standard deviation ของ Intensity value จากทุกรูป จะทำให้เราได้ model ที่มีความแม่นยำมากขึ้น

- วัตถุประสงค์

1. เพื่อทำนายการเป็นวัณโรคจากภาพ Chest X-ray
2. เพื่อศึกษาการทำ Machine Learning เรื่อง Deep learning for Computer Vision โดยเน้นที่เรื่อง Convolutional Neural Network (CNN) เป็นหลัก
3. เพื่อศึกษาการเขียนโปรแกรมภาษา Python ใน Google Colaboratory

2. ขั้นตอนการทำงาน สิ่งที่ทำ

- ขั้นตอนการทำงาน

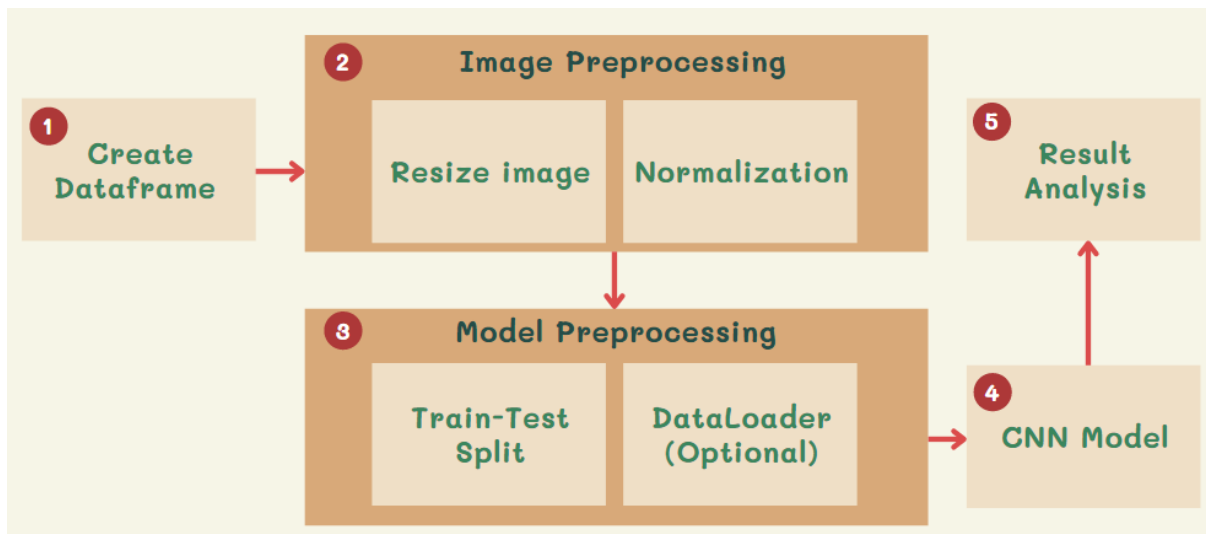
1. ศึกษาการทำ Machine Learning เรื่อง Deep learning for Computer Vision จาก <https://www.youtube.com/playlist?list=PL5-TkQAFbzxjBHtzdVCWE0Zbhomg7r>
2. วางแผน และ แนวทางในการทำงาน
3. เริ่มทำงานใน Google Colaboratory และรายงานความคืบหน้ากับอาจารย์ทุกสัปดาห์จนเสร็จ
4. ตรวจสอบความถูกต้องของงาน
5. ทำรูปเล่มสรุปการทำงาน

- สิ่งที่ทำ

1. Google Colaboratory: ภายในเป็นการเขียนโปรแกรมด้วย Python เพื่อตอบสนองวัตถุประสงค์ที่จะทำนายการเป็นวัณโรคจาก Chest X-ray image

2. รูปเล่มสรุปการทำงาน: ภายในเป็นการสรุปผลงานที่ทำ อุปสรรคที่เกิดขึ้นระหว่างการทำงาน แนวทางแก้ไข และ สิ่งที่ได้จากการศึกษาครั้งนี้

3. ผลลัพธ์ของงาน



3.1 Download data from Kaggle

- Download dataset จาก Kaggle (<https://www.kaggle.com/raddar/tuberculosis-chest-xrays-shenzhen>) มาเก็บไว้ใน Google Drive

3.2 Create dataframe

- สร้าง dataframe จากไฟล์ shenzhen_metadata.csv โดยการเพิ่ม column label ที่เป็นการดูตาม column finding โดย 0:normal, 1:abnormal (เป็นวัณโรค)

3.3 Preprocessing data

3.3.1 Image preprocessing

- Resize Chest X-ray image ให้เป็นขนาด 256*256 กับ 224*224 pixels เพื่อนำไปใช้ Train Machine Learning model
- Normalize image ทุกรูปด้วยการหารค่าในทุก pixels ด้วย 255 จากนั้นลบค่า mean และ หารด้วยค่า standard deviance ของรูปภาพทุกรูป เพื่อให้รูปภาพแต่ละรูปมีการกระจายตัวของ intensity value ที่ดีขึ้น

3.3.2 Model Preprocessing

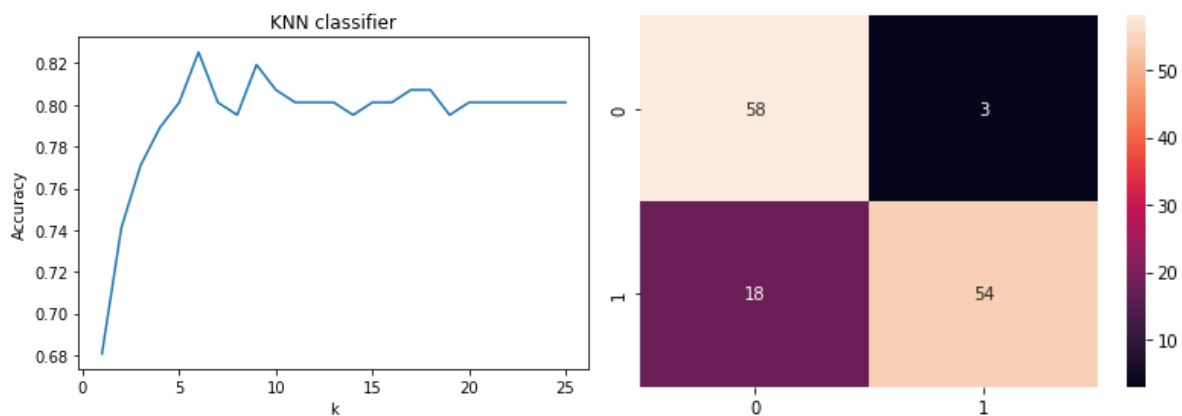
- ใช้ train-test split โดยมี Split size = 15% เพื่อแบ่งรูปภาพไว้ใช้ train กับ validate สุดท้ายจะได้รูปที่ใช้ train 562 รูป และใช้ validate 100 รูป
- สร้าง Class XRayDataset ใช้เก็บ image intensity label กับ labels
- ใช้ Dataloader ในการแบ่ง batch เพื่อใช้ในการ train และ validate โดยจะใช้ batch size = 16 กับ training set และ batch size = 32 กับ validating set

3.4 Machine learning model and Result analysis

3.4.1 KNN Classifier

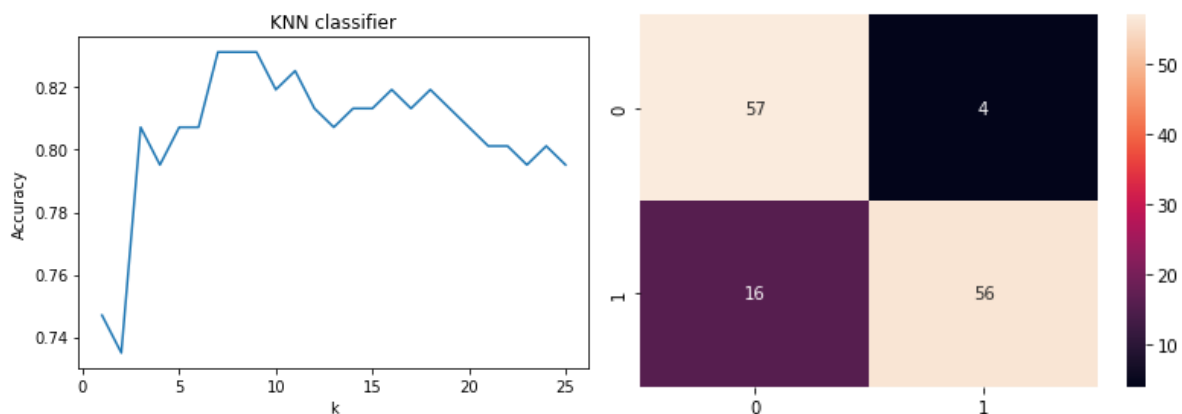
3.4.1.1 ใช้ Minkowski distance (l_p)

- จะพบว่าค่า k ที่ให้ความแม่นยำมากที่สุดคือ $k=6$ ให้ Accuracy: 0.8253012048192772 และ F1-score: 0.84



3.4.1.2 ใช้ Manhattan distance (l_1) และ Euclidean distance (l_2)

- จะพบว่าค่า k ที่ให้ความแม่นยำมากที่สุดคือ $k=7$ ให้ Accuracy: 0.8313253012048193 และ F1-score: 0.85



3.4.2 CNN

3.4.2.1 Model 1

```
Net(  
  (cnn_layers): Sequential(  
    (0): Conv2d(1, 4, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (1): BatchNorm2d(4, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (2): ReLU(inplace=True)  
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (4): Conv2d(4, 4, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (5): BatchNorm2d(4, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (6): ReLU(inplace=True)  
    (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
  )  
  (linear_layers): Sequential(  
    (0): Linear(in_features=16384, out_features=2, bias=True)  
  )  
)
```

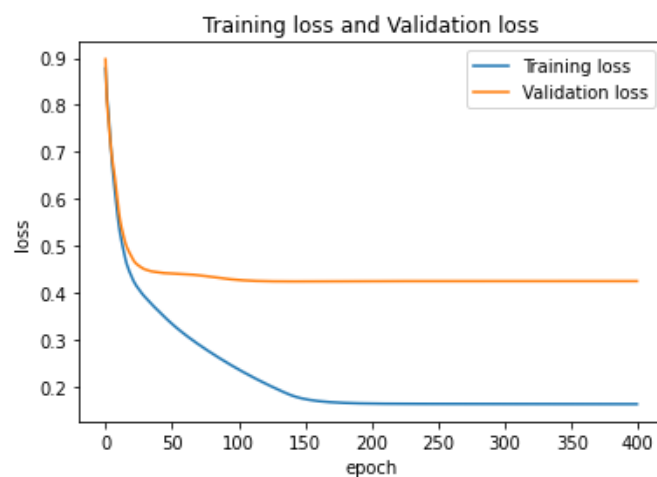
- Training 1 (Overfit)

Parameter:

- optimizer = Adam
- Learning rate = 0.00004
- criterion = CrossEntropyLoss
- scheduler = ReduceLROnPlateau
- number of epochs = 400

Accuracy:

- Training accuracy: 0.9608540925266904
- Validate accuracy: 0.84



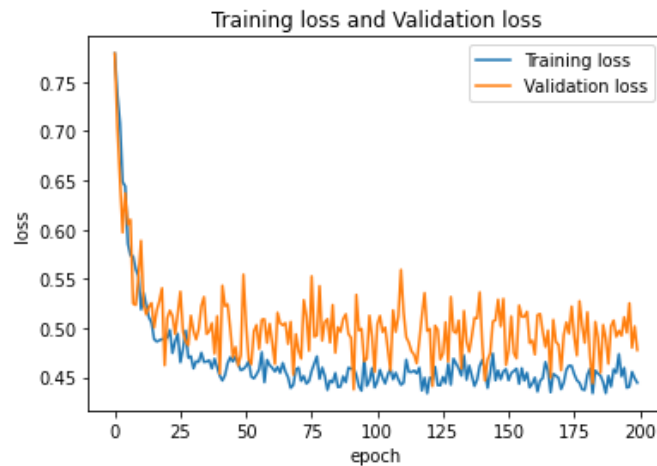
- **Training 2:** Dropout = 0.2

Parameter:

- optimizer = Adam
- Learning rate = 0.00004
- criterion = CrossEntropyLoss
- scheduler = ReduceLROnPlateau
- number of epochs = 200

Accuracy:

- Training accuracy: 0.8096085409252669
- Validate accuracy: 0.79



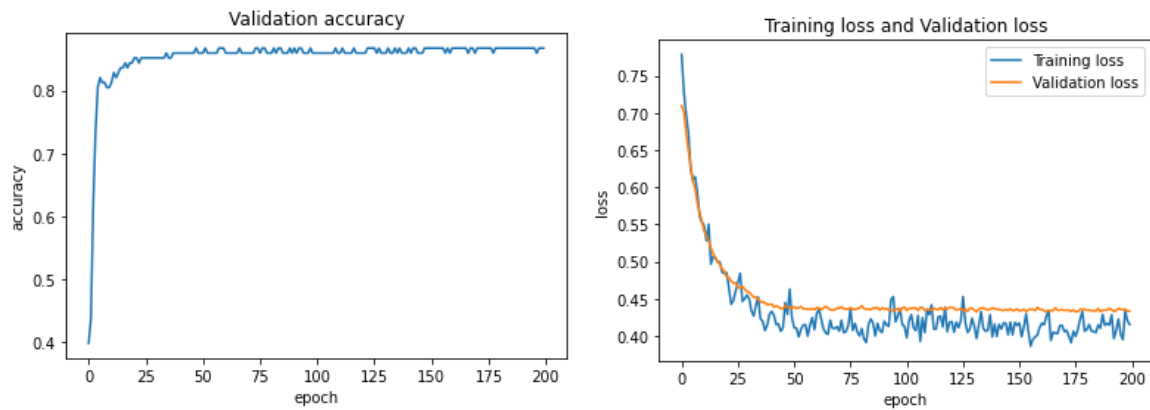
- **Training 3:** Train-validate with batch

Parameter:

- optimizer = Adam
- Learning rate = 0.00004
- criterion = CrossEntropyLoss
- scheduler = ReduceLROnPlateau
- number of epochs = 200

Accuracy:

- Training accuracy: 0.8454861044883728
- Validate accuracy: 0.8671875



3.4.2.2 Model 2

```
XrayClassification(
  (network): Sequential(
    (0): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU()
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU()
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU()
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU()
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU()
    (14): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (15): Flatten(start_dim=1, end_dim=-1)
    (16): Linear(in_features=262144, out_features=1024, bias=True)
    (17): ReLU()
    (18): Linear(in_features=1024, out_features=512, bias=True)
    (19): ReLU()
    (20): Linear(in_features=512, out_features=2, bias=True)
  )
)
```

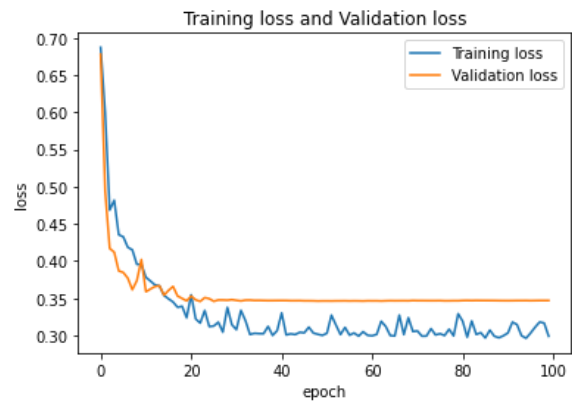
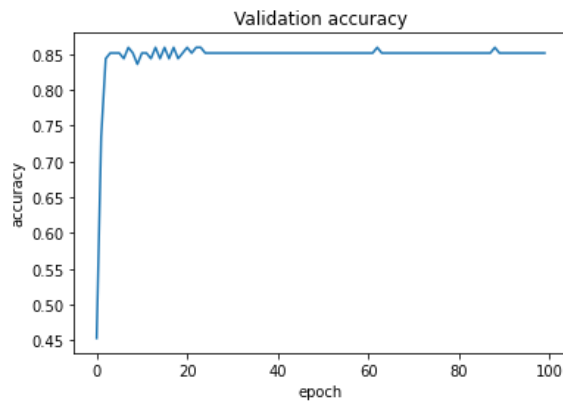
- Training 1

Parameter:

- optimizer = Adam
- Learning rate = 0.00001
- criterion = CrossEntropyLoss
- scheduler = ReduceLROnPlateau
- number of epochs = 100

Accuracy:

- Training accuracy: 0.8784722089767456
- Validate accuracy: 0.8515625



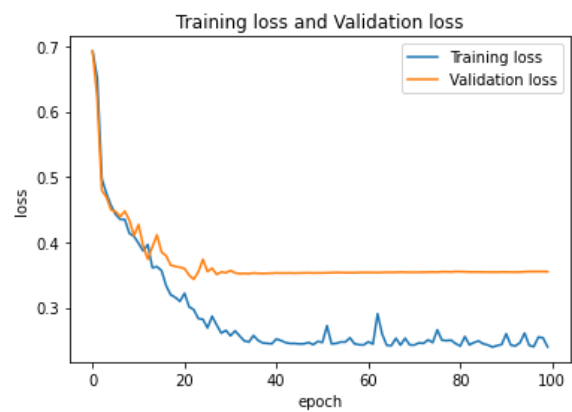
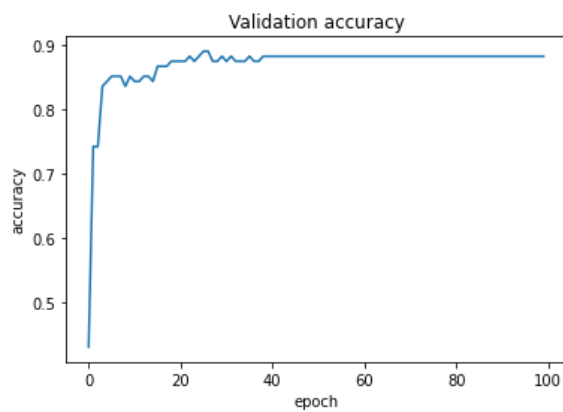
- **Training 2:** Dropout = 0.3

Parameter:

- optimizer = Adam
- Learning rate = 0.00004
- criterion = CrossEntropyLoss
- scheduler = ReduceLROnPlateau
- number of epochs = 100

Accuracy:

- Training accuracy: 0.8888888955116272
- Validate accuracy: 0.88



3.4.2.3 Model 3 (ResNet18)

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	$7 \times 7, 64$, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

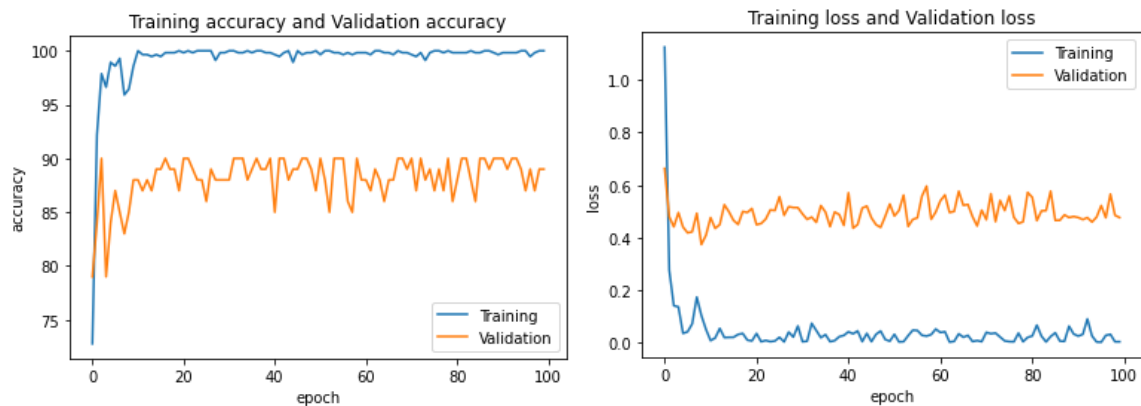
- Training 1 (Best model)

Parameter:

- optimizer = Adam
- Learning rate = 0.0001
- criterion = CrossEntropyLoss
- scheduler = ReduceLROnPlateau
- number of epochs = 100

Accuracy:

- Training accuracy: 1.0
- Validate accuracy: 0.89



4. อุปสรรคที่เกิดขึ้น วิธีแก้ปัญหา และสิ่งที่ได้จาก Individual Study นี้

- อุปสรรคที่เกิดขึ้น และ วิธีแก้ปัญหา

1. **อุปสรรค:** ไม่ทราบโครงสร้างของโค้ด และ model บางส่วนที่อยู่นอกเหนือจากการไปศึกษามา และ จำเป็นต้องใช้ในงานนี้

วิธีแก้: ไปค้นหาเพิ่มเติมในอินเทอร์เน็ต ปรึกษาอาจารย์ในการรายงานประจำสัปดาห์ หรือ ถามเพื่อนที่ทำงานด้วยกัน

2. **อุปสรรค:** ภาพ Chest X-ray ที่ได้รับมามีขนาดไม่เท่ากันและใหญ่เกินกว่าที่จะใช้ใน CNN model

วิธีแก้: ใช้ library ใน python ช่วย resize รูปให้เป็นขนาด 256*256 กับ 224*224 pixels

3. **อุปสรรค:** การใช้ GPU บน Google Colaboratory สามารถใช้ได้อย่างจำกัด

วิธีแก้: Save model ไว้ แล้ว reset runtime แล้วค่อย train model ต่อ หรือ มีการลดขนาด batch size เพื่อให้สามารถ train ได้หลาย epoch มากขึ้น

4. **อุปสรรค:** เวลา Mount Google Drive บางครั้งขึ้นว่า Mount เกิน Quota

วิธีแก้: ลองกด Mount ในวันถัดไปเมื่อ Quota reset แล้ว

- สิ่งที่ได้รับจาก Individual Study นี้

1. สรุปผลได้ว่าเมื่อมีการใช้ Machine Learning มาช่วยทำนายการเป็นวัณโรคจาก Chest X-ray image จะพบว่าการใช้ KNN Classifier จะมีความแม่นยำกับ Testing set ที่ 82-83% แต่ถ้าใช้ Convolutional Neural Network (CNN) ด้วย model ที่เหมาะสม จะมีความแม่นยำที่มากที่สุดกับ Testing set ที่ 89%

2. ได้เรียนรู้เกี่ยวกับการทำ Machine Learning ในเรื่อง Deep Learning for Computer Vision

3. ได้เรียนรู้เกี่ยวกับการใช้ Convolutional Neural Network (CNN) มาทำนายข้อมูล

4. ได้ทบทวนการเขียนโปรแกรมด้วย Python และ การใช้ Google Colaboratory

Source Code: https://github.com/Zeno-PT/Resume/blob/master/Chest%20X-ray/X_ray_image.ipynb

Reference

- <https://www.youtube.com/playlist?list=PL5-TkOAfAZFbzxjBHtzdVCWE0Zbhomg7r>
 - <https://www.kaggle.com/pranjalsoni17/natural-scene-classification>
 - <https://www.analyticsvidhya.com/blog/2021/09/convolutional-neural-network-pytorch-implementation-on-cifar10-dataset/>
 - https://pytorch.org/tutorials/beginner/basics/buildmodel_tutorial.html
 - <https://www.kaggle.com/raddar/tuberculosis-chest-xrays-shenzhen>
 - <https://www.pluralsight.com/guides/introduction-to-resnet>
-