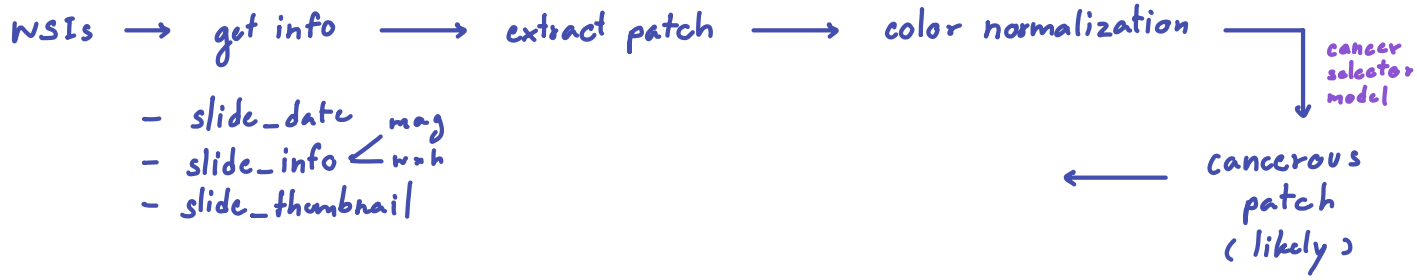
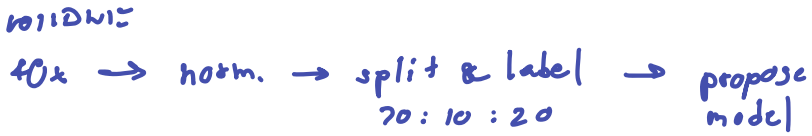
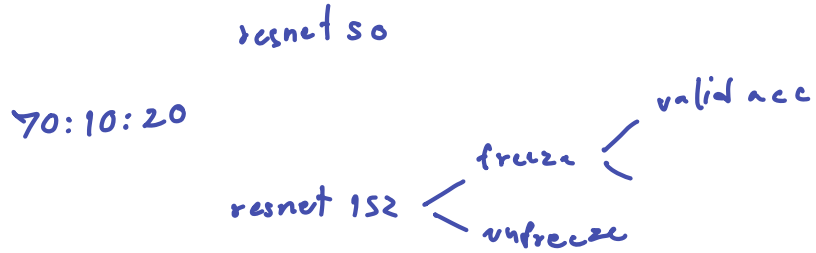
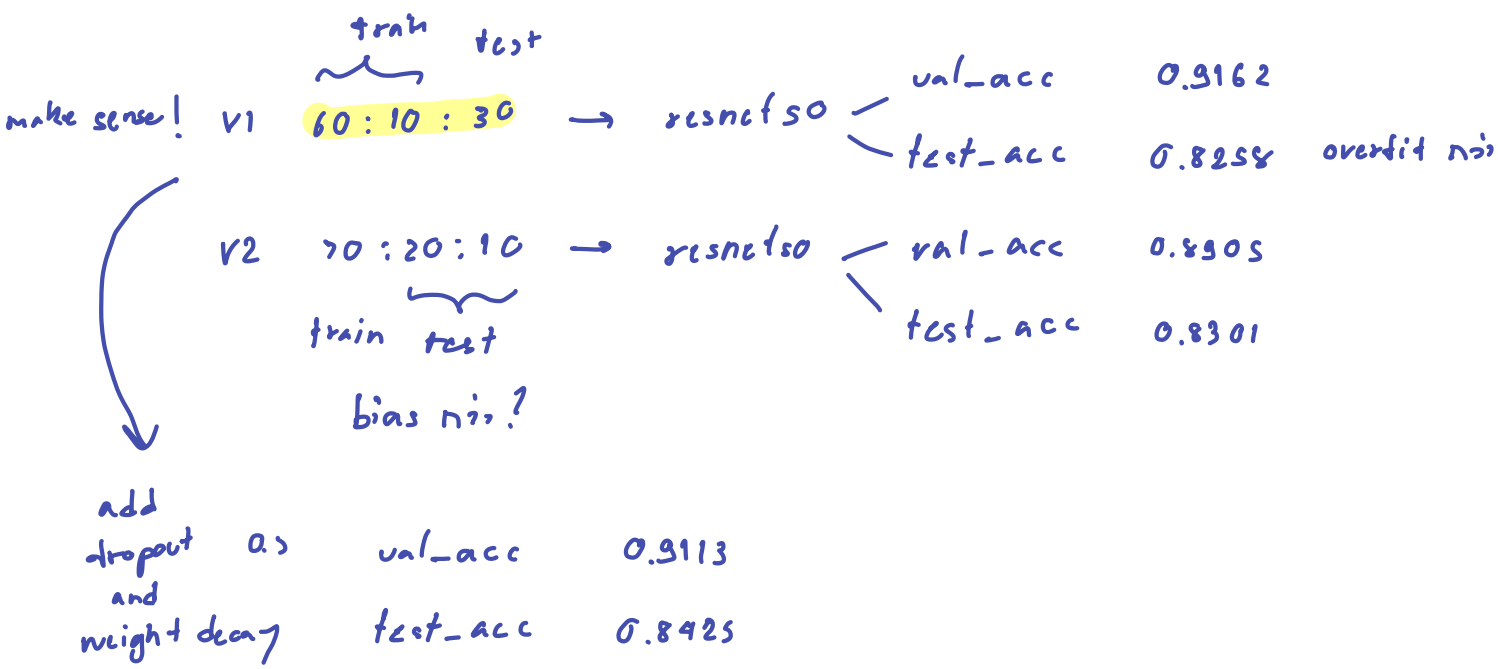
CNN_BreakHis_Logbook









60 : 10 : 30

resnet50 pretrained freeze

best valid accuracy : 0.9557

test accuracy : 0.8442 ← overfit!

resnet50 pretrained unfreeze Adam no scheduler

best valid accuracy : 0.9803

test accuracy : 0.8760 ← overfit

resnet50 pretrained unfreeze Adam lrscheduler ReduceLROnPlateau (mode : 'min' patience : 5, factor 0.1

best valid accuracy : 0.9754

test accuracy : 0.8811

freeze, dropout
และ unfreeze, no dropout

freeze resnet152 dropout 0.5 x 1 ← ตัวนี้
- val acc 0.921
- test acc 0.881

freeze resnet152 dropout 0.5 x 2
- val acc 0.926
- test acc 0.883

freeze resnet152 dropout 0.5 x 3
- val acc 0.926
- test acc 0.876

Adam
- val acc 0.926
- test acc 0.883

AdamW
- val acc 0.926
- test acc 0.887

SGD
- val acc 0.680
- test acc 0.719

StepLR
.step()
- val acc 0.926
- test acc 0.887

.step(val_loss)
- val acc 0.921
- test acc 0.886

ReduceOnPlateau
.step(val_loss)
- val acc 0.921
- test acc 0.889

Exponential LR
.step()
- val acc 0.936
- test acc 0.881

OnecycleLR
.step()
- val acc 0.921
- test acc 0.884

MultistepLR
.step()

		1st	2nd
20, 40, 60	val acc	0.921	0.926
	test acc	0.898	0.889
10, 30, 50	val acc	0.921	
	test acc	0.886	

StepLR
.step()
- val acc 0.926
- test acc 0.884

StepLR
.step(val_loss)
- val acc
- test acc

ReduceOnPlateau
.step(val_loss)
- val acc 0.926
- test acc 0.884

Exponential LR
.step()
- val acc 0.680
- test acc 0.719

? ดูgrain ไม่ขึ้น

OnecycleLR
.step()
- val acc 0.931
- test acc 0.863

MultistepLR
.step()

20, 40, 60
- val acc 0.921
- test acc 0.889

10, 30, 50
- val acc 0.911
- test acc 0.899

No scheduler
- val acc 0.926
- test acc 0.881

Cross Entropy Loss

class weight loss

0.5	0.5	val acc	0.911	recall	0.9347
		test acc	0.899		
0.5	0.55	val acc	0.921	recall	0.9347
		test acc	0.899		
0.5	0.6	val acc	0.911	recall	0.9347
		test acc	0.894		
0.5	0.7	val acc	0.916	recall	0.9417
		test acc	0.884		
0.5	0.75	val acc	0.911	recall	0.9487
		test acc	0.894		
0.5	0.8	val acc	0.906	recall	0.9557
		test acc	0.898		
0.5	0.85	val acc	0.891	recall	0.9557
		test acc	0.811		

recheck dropout layer

1 dropout
- val acc 0.911
- test acc 0.884

recall 0.9580

✓ 2 dropout
- val acc 0.906
- test acc 0.894

recall 0.9987 why drop?

3 dropout
- val acc 0.897
- test acc 0.891

resnet 2048 ~~→~~ 1000 →

1000 → 512 → 128 → 2 → sigmoid

vgg alexnet densenet

efficientnet

try efficientnet (multisteplr
- val acc 0.877
- test acc 0.809

CrossEntropyLoss ([0.5, 0.8])

AdamW

efficientnet freeze → val acc 0.9310, test acc 0.7940 → overfit!

no scheduler

CrossEntropyLoss ([0.5, 0.5])

AdamW

efficientnet unfreeze → val acc 0.9709, test acc 0.8860 → overfit, recall 0.9254

no scheduler

CrossEntropyLoss ([0.5, 0.5])

AdamW

ปกติเขาตัด layer 2048 → 1000 แล้วไป 2048 → 2 เลย

Noscheduler, Adam, Crossentropyloss ([0.5, 0.5])

resnet 2048 ~~→~~ 1000 → 2 → val acc ~0.84, test acc ~0.84

MultistepLR, AdamW, Crossentropy ([0.5, 0.5])

resnet 2048 ~~→~~ 1000 → add 1000 → 512 → 128 → 2 → sigmoid

val acc 0.906, test acc 0.894, recall 0.9987

MultistepLR , AdamW, Crossentropy ([0.5, 0.8])

resnet 2048 → 1000 → 512 → 128 → 2 → sigmoid

(get pretrained)

freeze
- val acc 0.9014
- test acc 0.8927

recall 0.9394

unfreeze
- val acc 0.9802
- test acc 0.9078

recall 0.9091 ?

Noscheduler, Adam, Crossentropyloss ([0.5, 0.5])

unfreeze
- val acc 0.9704
- test acc 0.8777

recall 0.9393

∴ เดินหน้าต่อดีกว่า

resnet 152

dropout 2 อัน layer 2048 และ 1000

dropout_rate_1, dropout_rate_2 *

AdamW optimizer – weight decay *

learning rate *

lr.scheduler ใช้ multistep – milestone [10, 30, 50] *

gamma 0.1 *

CrossEntropyLoss ใช้ weight 0.5, 0.8 *

AdamW optimizer : lr 0.0001

weight decay 0.01

```
** Model Evaluation **
Best Valid Accuracy: 0.9064039408866995
Test Accuracy: 0.897822445561139
Precision: 0.9070796460176991
Recall: 0.9557109557109557
F1-score: 0.9307604994324631
AUROC: 0.9523809523809523
AUCPR: 0.9805531734516826
```

AdamW optimizer : lr 0.0001

weight decay 0.01

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

ทดลองเพิ่มเติม โมเดลที่ขนาด ≈ resnet152 (~60M parameter)

efficientnet b7

AdamW

Reduce on Plateau lr scheduler

CrossEntropyLoss (0.5, 0.5)

freeze < val acc 0.8768 / test acc 0.7940 — recall 0.7940

unfreeze < val acc 0.9557 / test acc 0.8693 — recall 0.8601

alexnet

AdamW

Reduce on Plateau lr scheduler

CrossEntropyLoss ((0.5, 0.5))

freeze
- val acc 0.8424
- test acc 0.7739
- recall 0.8275

unfreeze
- val acc 0.6798
- test acc 0.7186
- recall 1.000