

Medical Image Processing with Deep Learning

----Mammograms Classification and Automatic Tumor detection

Members: Yijie Jin (CityU), Yifan Zheng (CUHK)

Mentors: Dr. Kwai Wong, Dr. Raymond Chan

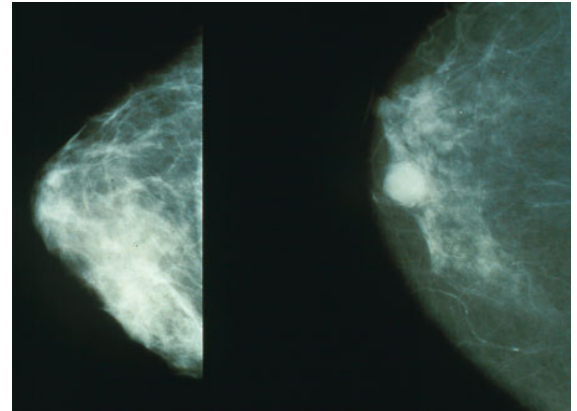


Background

- Breast cancer is the second leading cause of cancer-related death among women in the United States.
- Roughly one eighth of women in the United States will develop breast cancer during their lifetimes.
- Five-year relative survival rates can be up to 3-4 times higher for cancers detected at an early stage versus at a later stage.

Mammography

- Mammography is the most common breast screening technology. It is the process of using low-energy X-rays to examine the human breast for diagnosis and screening. It is the most reliable method for screening breast abnormalities before they become clinically palpable.
- Reading mammograms is a tedious and error-prone process, and not all radiologists achieve uniformly high levels of accuracy.



Normal

Malignant



Goals

- Classify mammograms into three classes, normal, benign and malignant (**CNNI-BCC and VGG16**)
- Automatically detect the tumor without prior information of the presence of a cancerous lesion (**IDBLL**)



Challenges

- Hard to find a database due to privacy reasons -> a public database (MIAS)
- Mammograms usually have a low contrast -> Remove black background
- “needle in a haystack” nature of mammogram classification -> Cut the images into small patches



Dataset and Data Preprocessing

Dataset: mini-MIAS database of mammograms

- 322 images in total

Data Augmentation:

- Rotation (by 90, 180, 270 degrees respectively)
- Flip (vertically)
- Equally Sampling ($1024 \times 1024 \rightarrow 128 \times 128$ and $1024 \times 1024 \rightarrow 256 \times 256$)
- Sample with overlap

Data Cleansing:

- Remove the image patches with black background



Dataset - mini-MIAS

- Labelled
- Have information about the coordinates of tumor center
- Have information about the radius of the tumor

Labelled data : G CIRC B 595 864 68

Data explanation:

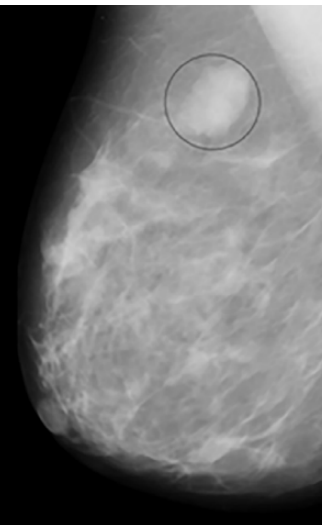
G = Fatty – Glandular

CIRC = Well-defined/
circumscribed masses

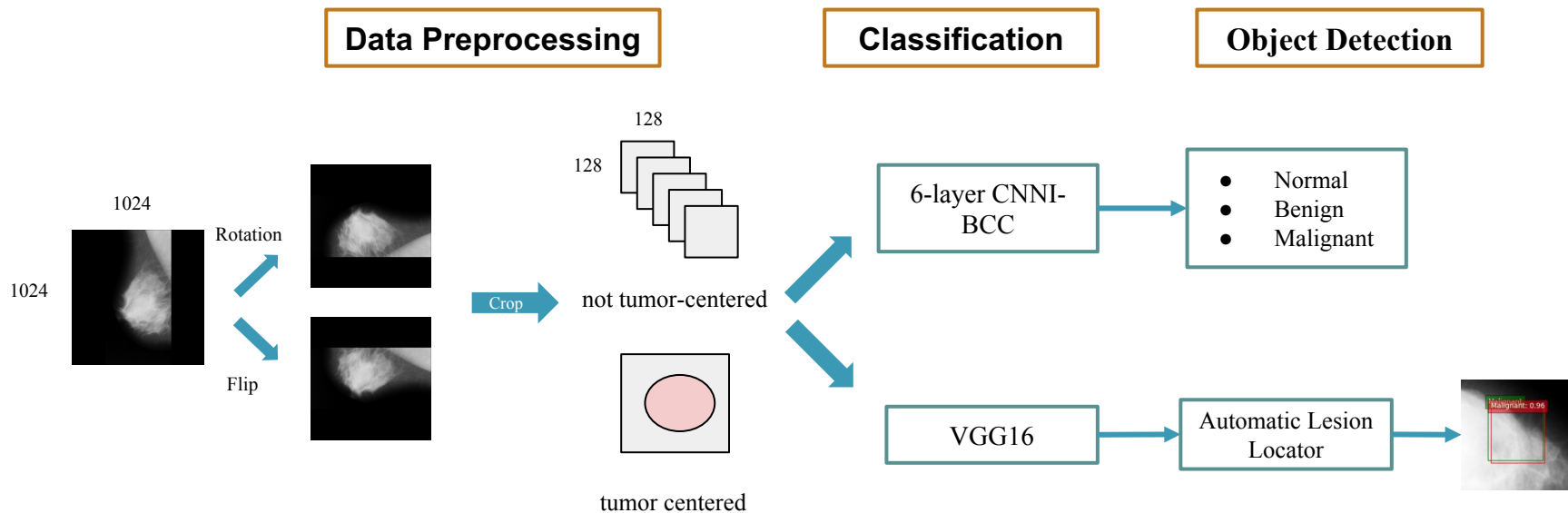
B= Benign

(595, 864) = (X, Y) coordinate

68 pixels = Radius from center



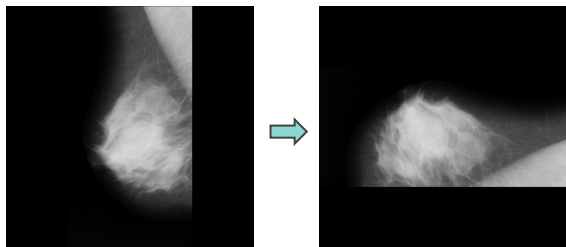
Procedure



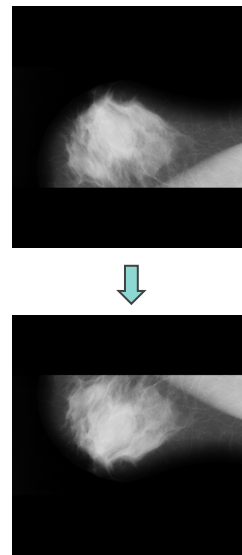


Data Augmentation

---- Flip/Rotation



Rotation clockwise by
90 degrees

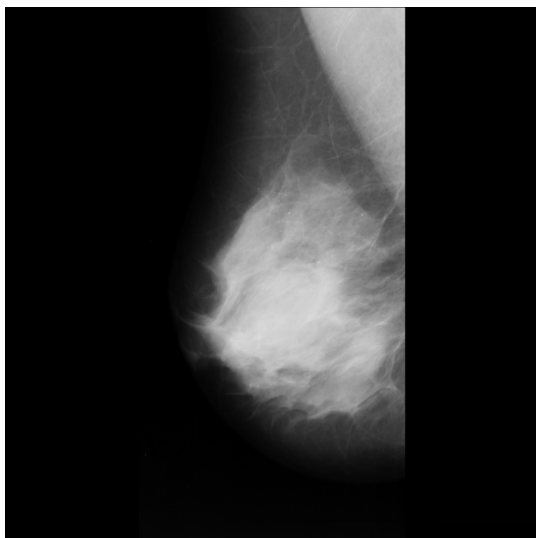


Flip
vertically

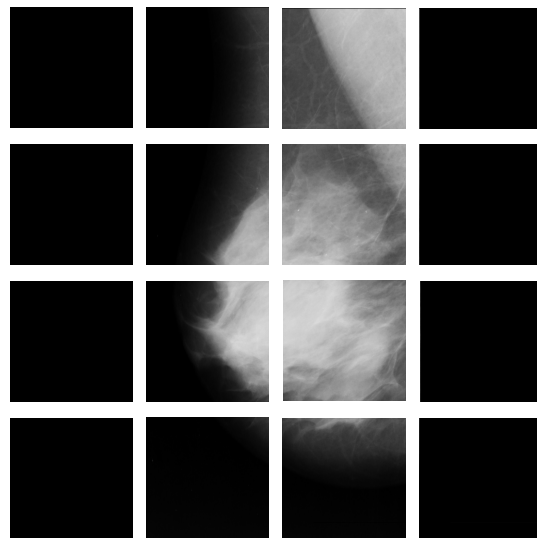


Data Augmentation

---- Equally Sampling



1024×1024

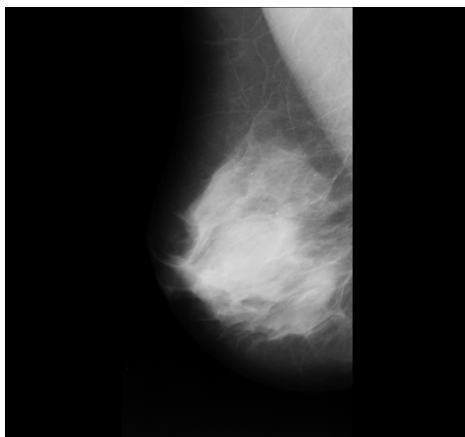


256×256

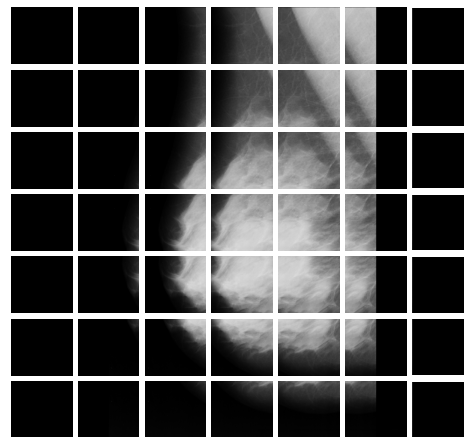


Data Augmentation

---- Sample with Overlap



1024×1024



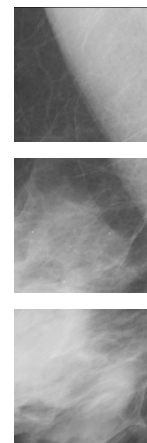
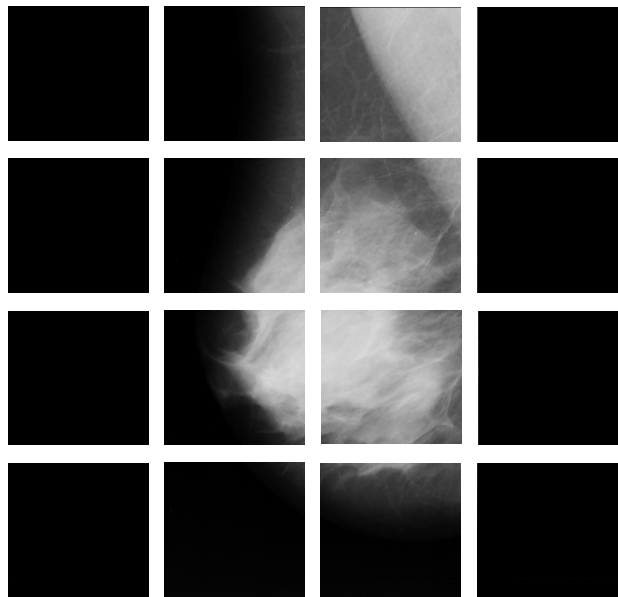
256×256

- Sample the image patches every 128 pixels.
- 1 -> 49



Data Cleansing

---- Remove images with black background





Relabel the Image Patches

B	B	B	B
B	B	B	B
B	B	B	B
B	B	B	B

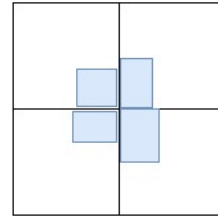
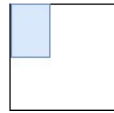
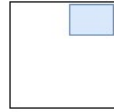
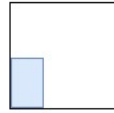
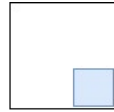
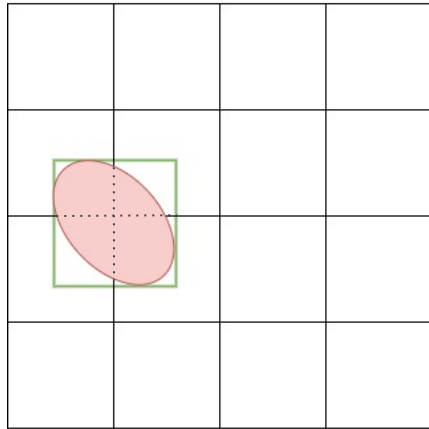
Before

N	N	N	N
B	B	N	N
B	B	N	N
N	N	N	N

After



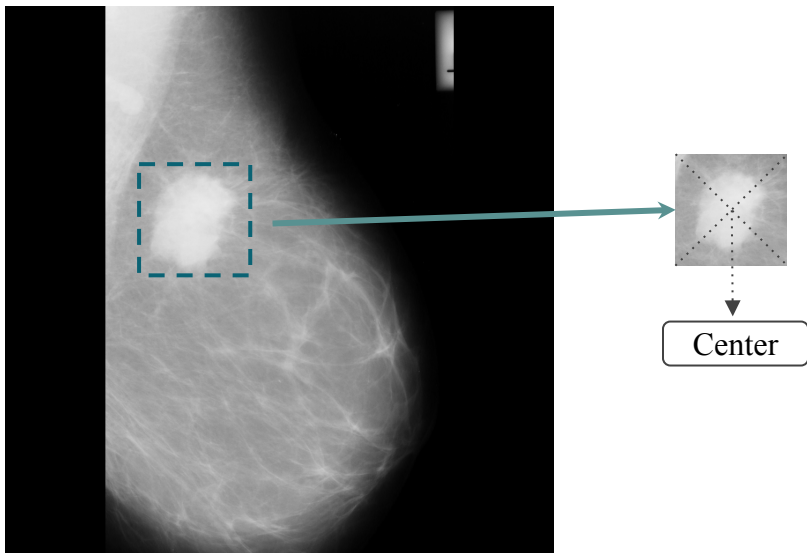
Problems



- Tumors are separated
- Hard to put the small patches together

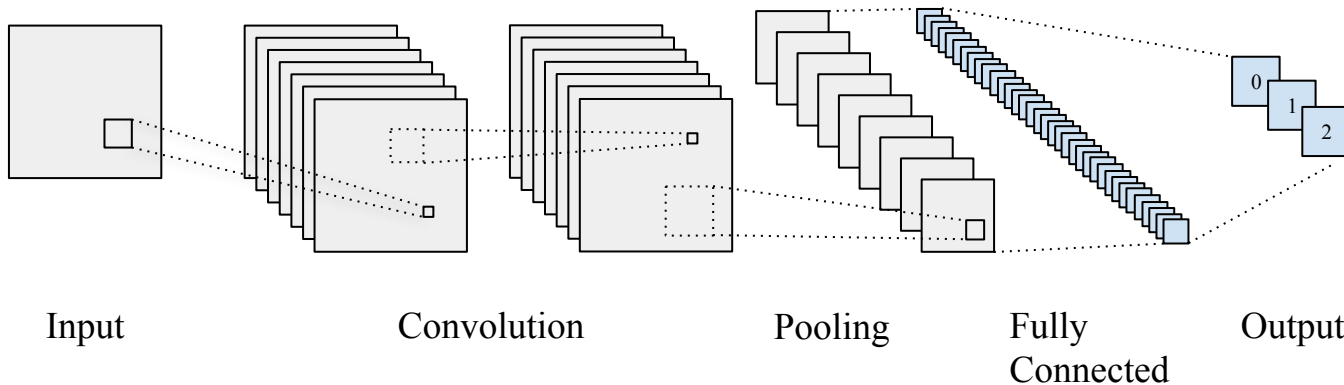
Data Augmentation

----Tumor-centered Sampling



- $1024 \times 1024 \rightarrow 256 \times 256$
- Tumor center at the image center

Convolutional Neural Network Improvement for Breast Cancer Classification (CNNI-BCC)



- 30 layers (28 convolutional layers+1 pooling layer+1 fully connected layer)
- Implemented in Keras with Tensorflow backend

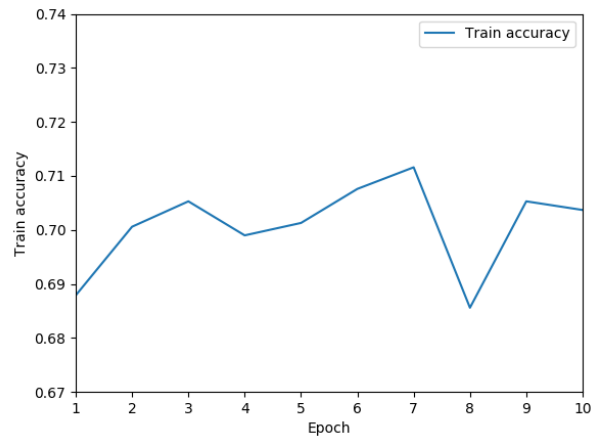
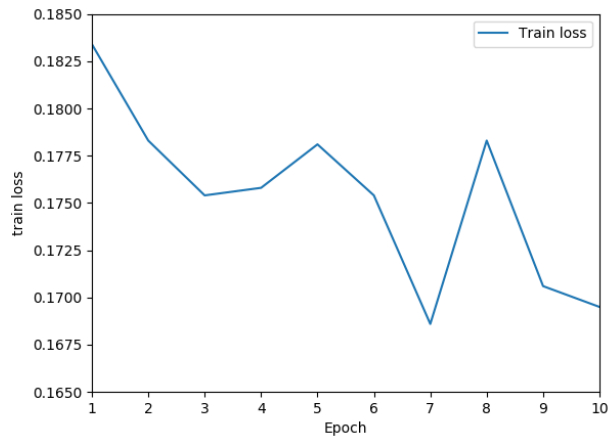


Experiments

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 64, 64, 32)	320
depthwise_conv2d_1 (DepthwiseConv2D)	(None, 64, 64, 32)	32800
conv2d_2 (Conv2D)	(None, 64, 64, 64)	2112
depthwise_conv2d_2 (DepthwiseConv2D)	(None, 32, 32, 64)	262208
average_pooling2d_1 (AveragePooling2D)	(None, 8, 8, 64)	0
flatten_1 (Flatten)	(None, 4096)	0
dense_1 (Dense)	(None, 3)	12291



Results

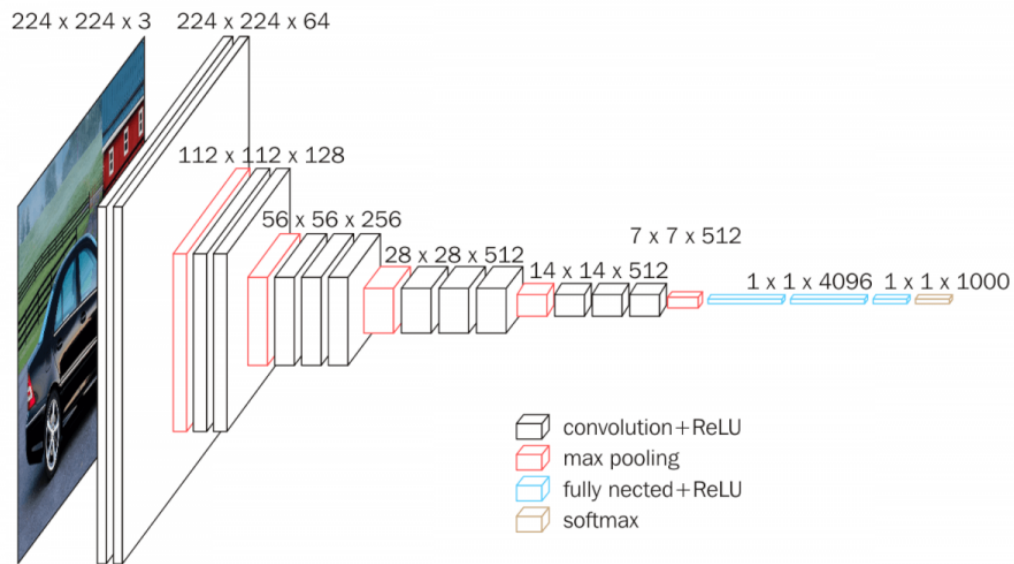


The **loss** on the test set is: 0.14544324301610326

The **accuracy** on the test set is: 0.7324840809888901

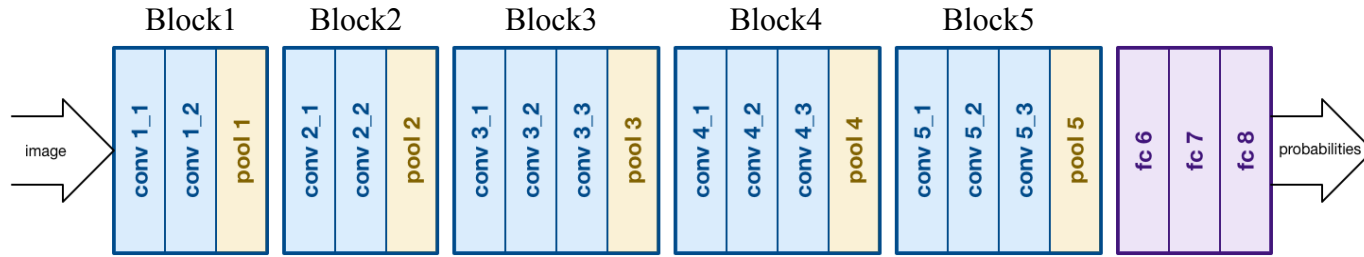


VGG16





VGG16 Architecture



- 5 blocks
- Convolutional Layer+Pooling Layer in each block
- 3 fully connected layer and softmax function



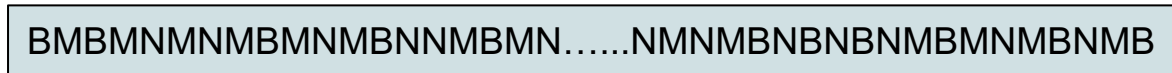
Data Shuffling



Load the data in sequence



Split training and validation set



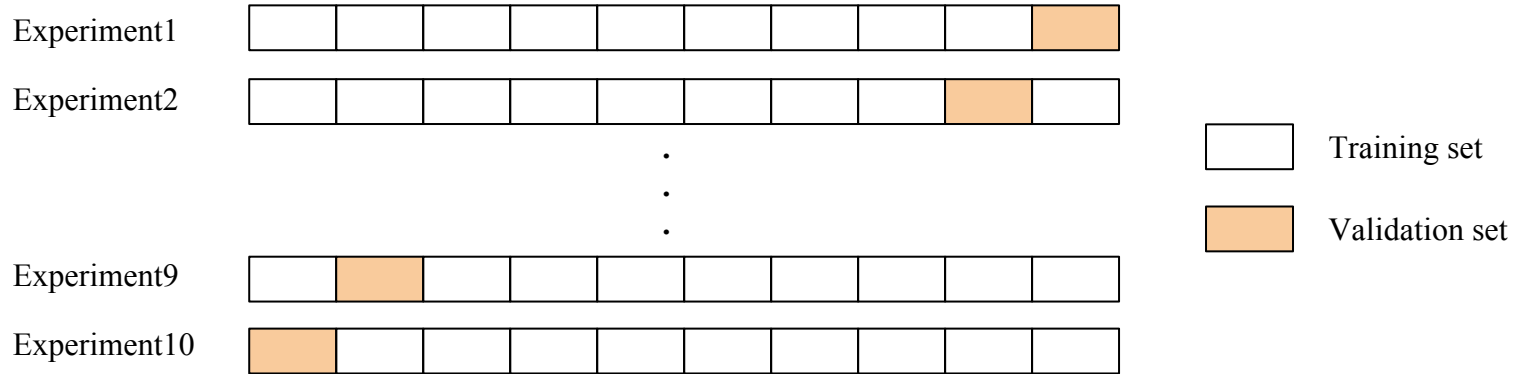
After shuffling

Problems:

- The training data is not general.
- The neural network cannot enough features of malignant cases.



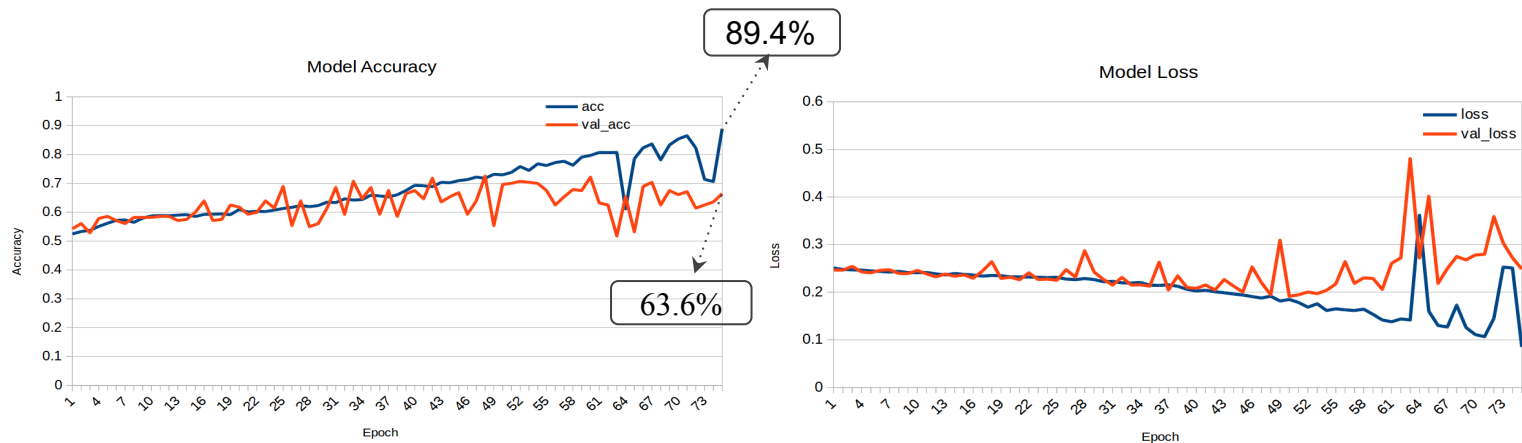
Cross-Validation



- 10-Fold Cross-validation
- Train 10 different models with different validation sets.

Experiments

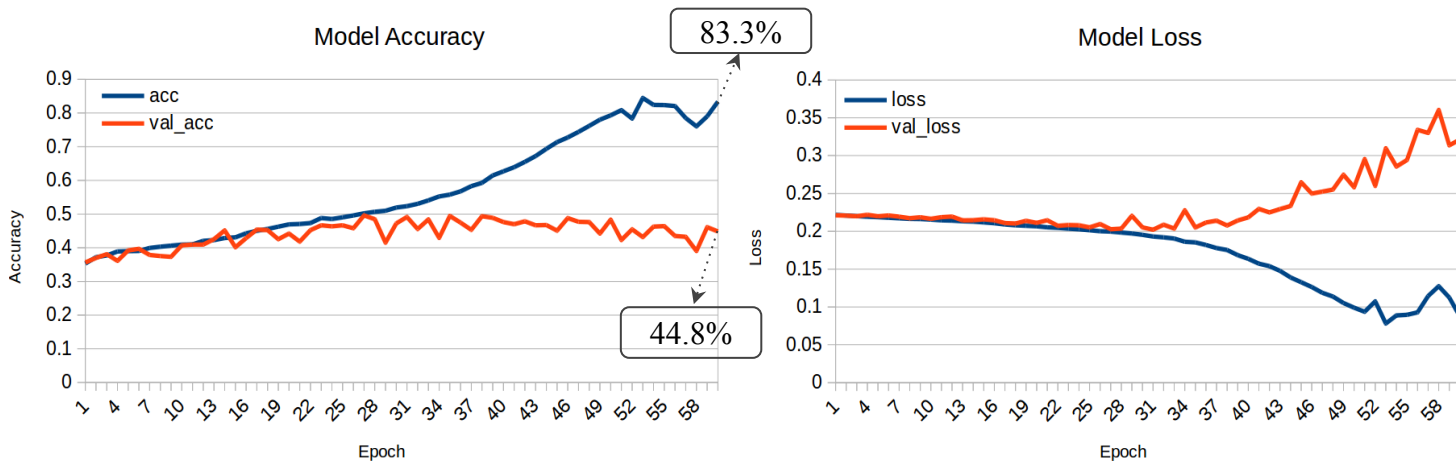
---- Classify mammograms into 2 categories



- Training set: 2520 image patches (128×128)
- Epoch number: 80
- Batch size: 32

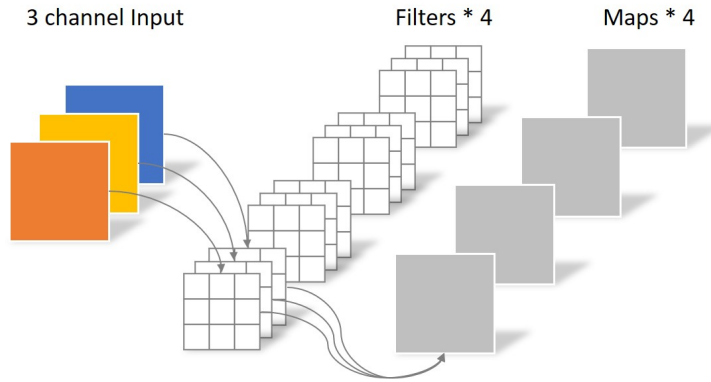
Experiments

---- Classify mammograms into 3 categories



- Training set: 11340 image patches
- Epoch number: 60
- Batch size: 32

Normal Convolution and Separable Convolution

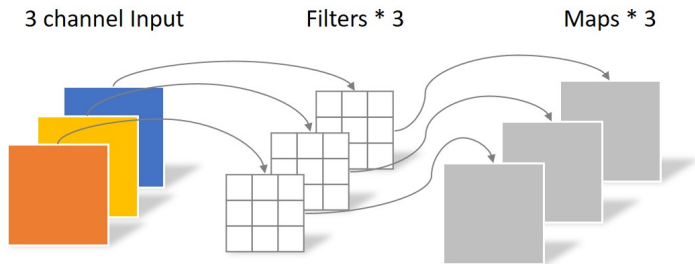


Normal Convolution

- Filter number: 4
- Filter size: 3×3
- Channel number: 3
- Total number of parameters:
 $4 \times 3 \times 3 \times 3 = 108$

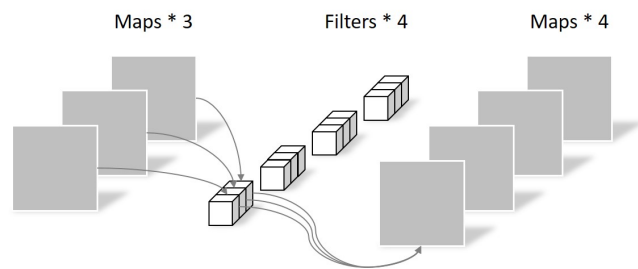
Normal Convolution and Separable Convolution

Depthwise Convolution



- Filter number: 3
- Filter size: 3×3
- Channel number: 3
- Number of parameters: $3 \times 3 \times 3 = 27$

Pointwise Convolution



- Filter number: 4
- Filter size: 1×1
- Channel number: 3
- Number of parameters: $1 \times 1 \times 3 \times 4 = 12$

Total number of parameters: $27 + 12 = 39$



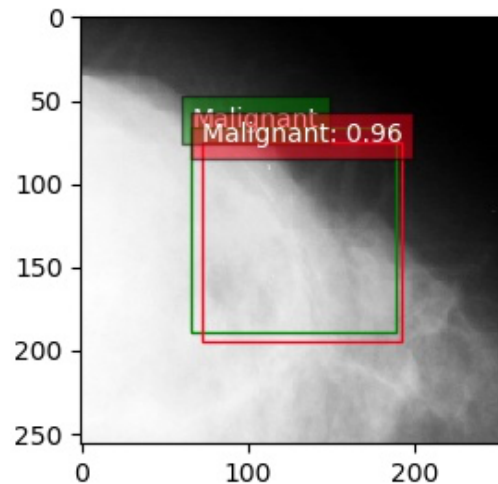
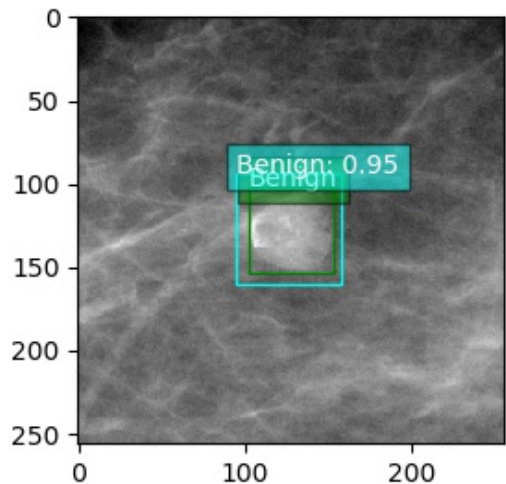
Comparison between Normal Convolution and Separable Convolution

	Number of Parameters in 1 Layer	Total Parameters
Conv2D	1,792	23,715,730
SeparableConv2D	283	4,820,525

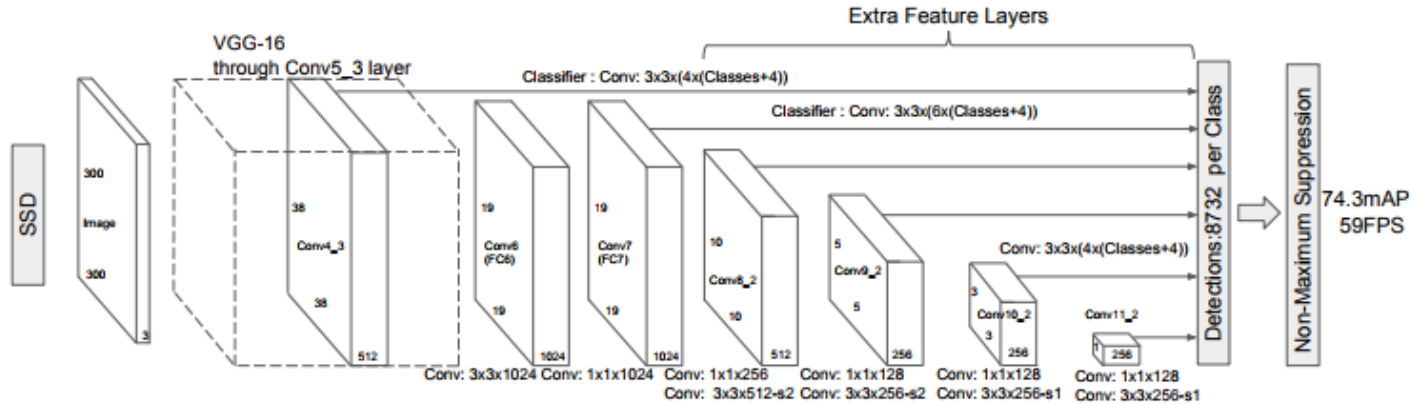
- Keras implementation: SeparableConv2D



Interactive Detection Based Lesion Locator (IDBLL)



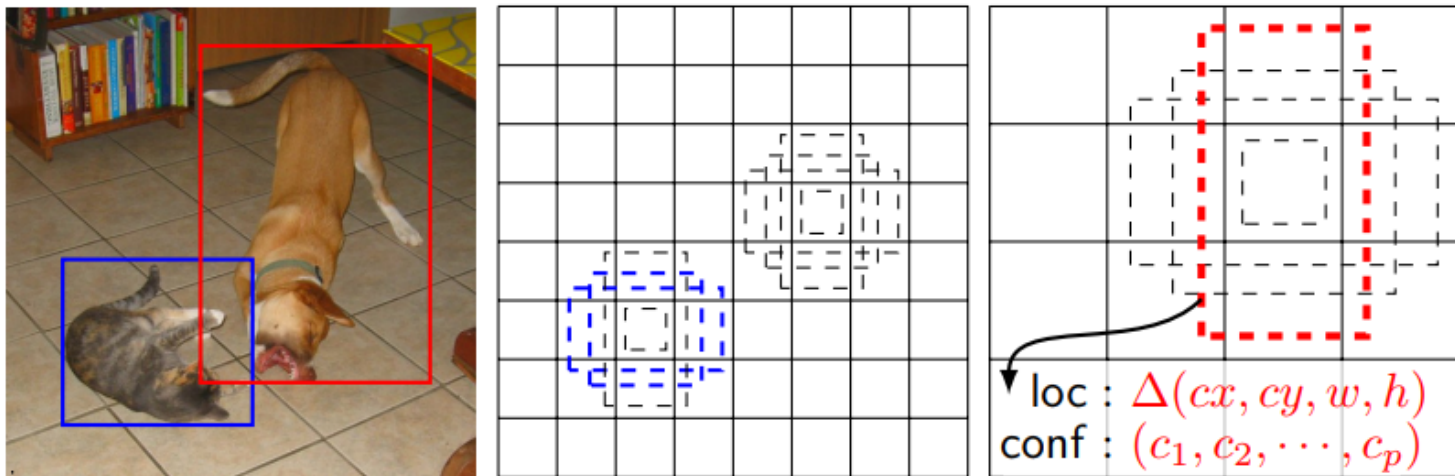
Structure of Single Shot MultiBox Detector (SSD)



Base Network for classification

- + Multi-scale feature maps for detection
- + Convolutional predictors for detection: multiple classes confidences
- + Default boxes and aspect ratios: localization

Prior Box & Objective Loss Function

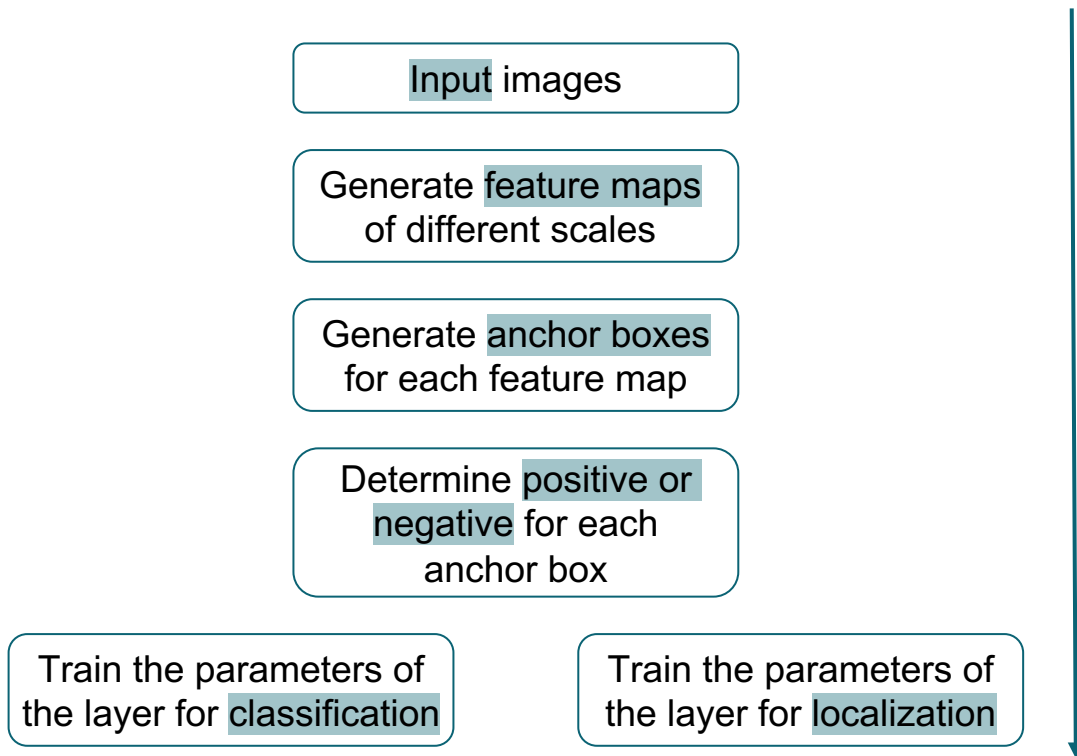


(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$



Training Process





CPU & GPU We Used

CPU

- 128GB
- 28 cores
- 8TB on-node storage

GPU

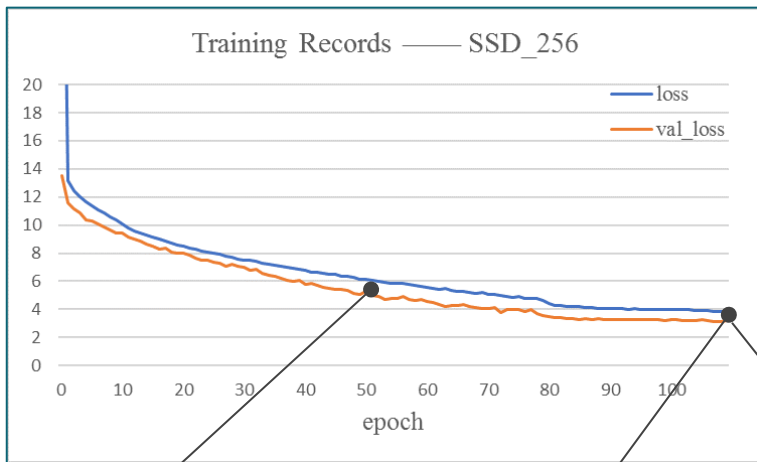
- 2 NVIDIA Tesla K80
Kepler architecture
- 12GB/GPU
- 48GB total/node

GPU

- 8 NVIDIA Volta V100
- 16GB/GPU
- 128GB total/node

Experiment A

---- Best Result



50 epochs: 5.06561

110 epochs: 3.14813

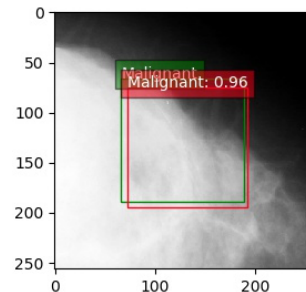
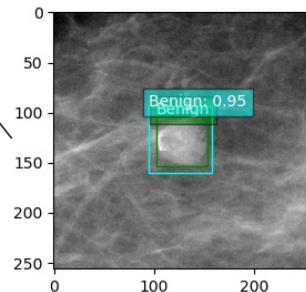
converges slow

760s/epoch

time consuming

Training Setting

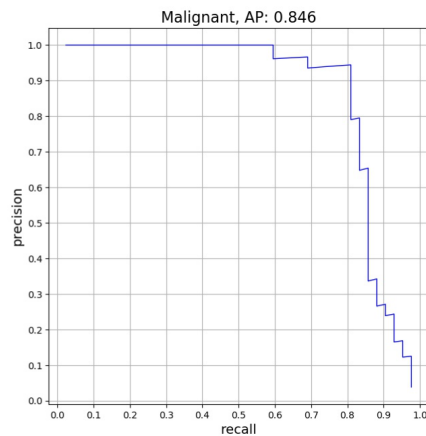
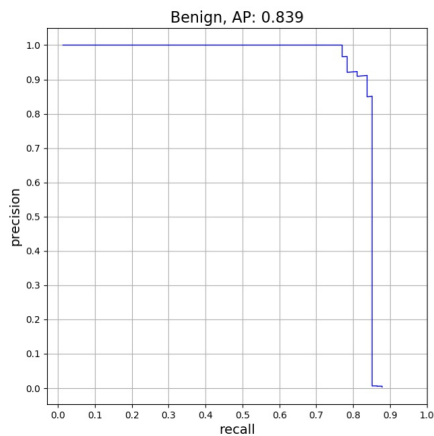
- 110 epochs
- batch size 100
- steps per epoch 60





Experiment A

---- Best Result



Benign AP	0.839
Malignant AP	0.846
mAP	0.842

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 600
Fast YOLO	52.7	155	1	98	448 × 448
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8732	300 × 300
SSD512	76.8	19	1	24564	512 × 512
SSD300	74.3	59	8	8732	300 × 300
SSD512	76.8	22	8	24564	512 × 512

Table 7: **Results on Pascal VOC2007 test.** SSD300 is the only real-time detection method that can achieve above 70% mAP. By using a larger input image, SSD512 outperforms all methods on accuracy while maintaining a close to real-time speed.

Precision-Recall Curve & mAP calculation

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

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

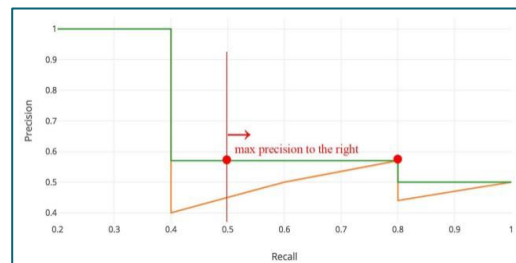
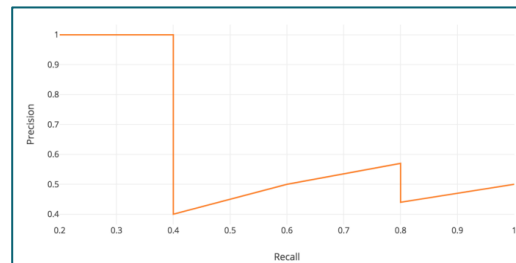
where TP = True Positive, FP = False Positive, FN = False Negative.

In our case of testing for cancer:

$$\text{Precision} = \frac{TP}{\text{Total Positive Results Shown by Model}}$$

$$\text{Recall} = \frac{TP}{\text{Total Cancer Cases in Ground Truth}}$$

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0



$$AP = \frac{1}{20} \sum_{r \in \{0.0, \dots, 1.0\}} p_{interp}(r)$$

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$$

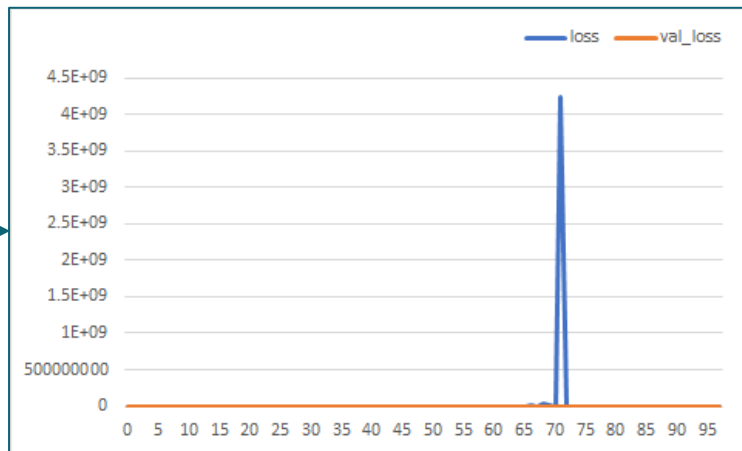


Experiment B

---- Increasing Learning Rate

```
def lr_schedule(epoch):  
    if epoch < 80:  
        return 0.001  
    elif epoch < 90:  
        return 0.0001  
    else:  
        return 0.00001
```

```
def lr_schedule(epoch):  
    if epoch < 80:  
        return 0.002  
    elif epoch < 90:  
        return 0.0002  
    else:  
        return 0.00002
```

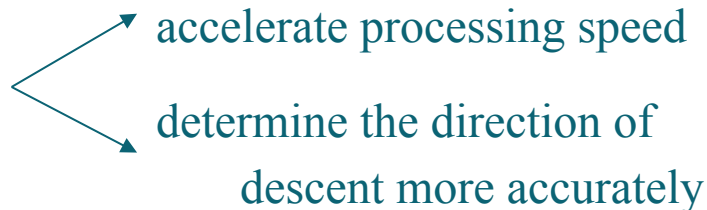




Experiment C

---- Enlarging Batch Size

Usually, enlarging batch size:



However, our test shows:

Batch Size	Steps per Epoch	Epoch	Validation Loss	Speed
300	2	300	9.24585	87s/epoch
200	3	300	8.64310	80s/epoch
100	60	30	7.024	760s/epoch



Experiment D

---- Using Separable Convolution

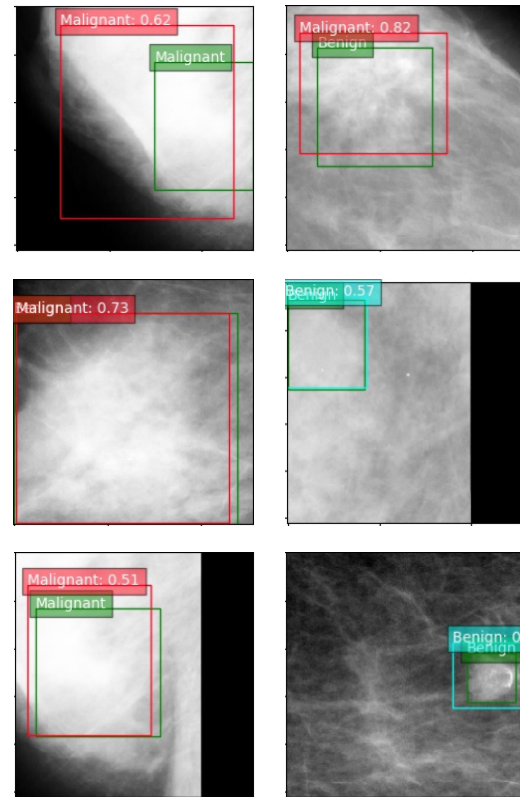
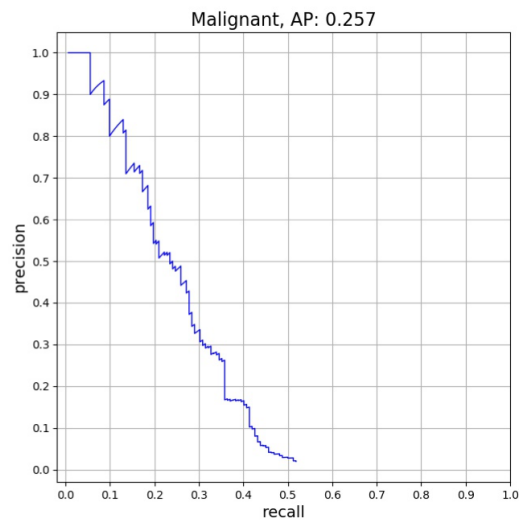
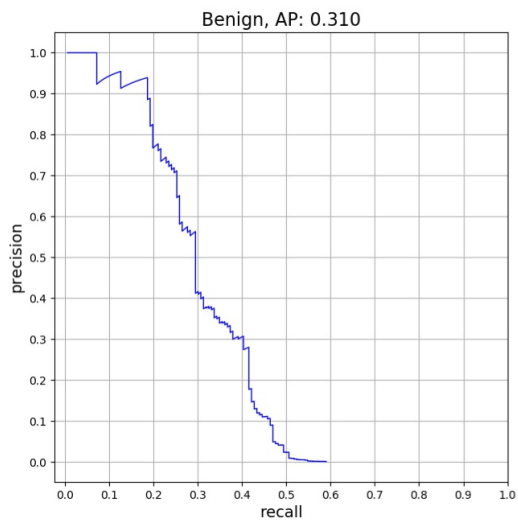
Batch Size	Steps per Epoch	Epoch	Validation Loss	Speed
100	6	150	4.36554	245s/epoch
200	3	150	4.80492	245s/epoch
300	2	150	5.03458	245s/epoch

- Take more time to train per epoch
- Converge faster than Conv2D-model in terms of epochs
- Converge slower than Conv2D-model in terms of real time

Experiment E

----Testing on More General Cases

Test on uniformly cut set



Experiment E (Cont.)

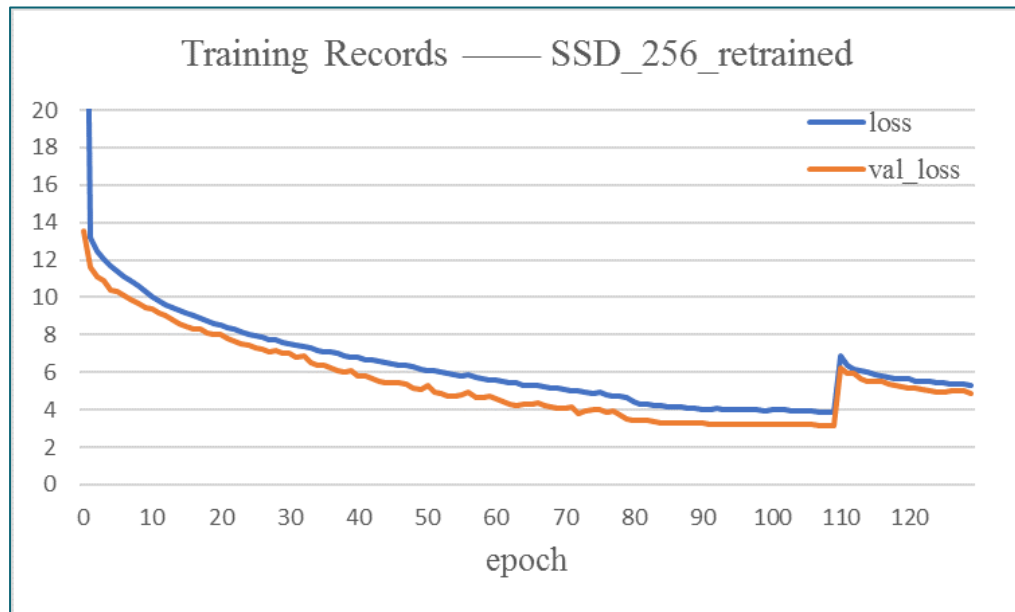
----Retraining on More General Cases

Uniformly cut data set

- 3500 in training set
- 500 in validation set
- 329 in test set

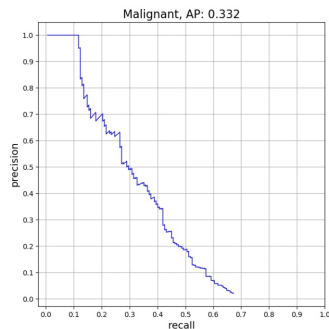
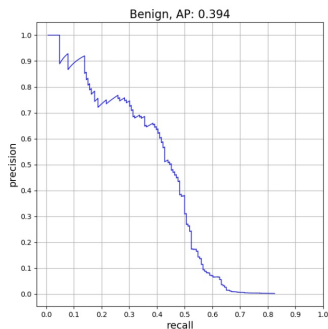
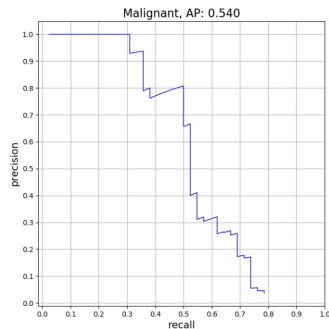
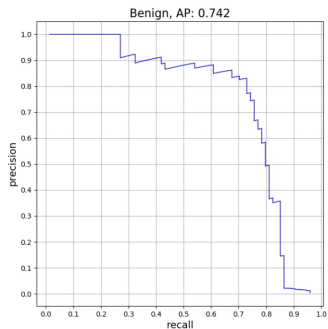
Training Setting

- 20 epochs
- batch size 100
- steps per epoch 35



Experiment E (Cont.)

----Retraining on More General Cases



Tumorcenter test set		Uniformly cut test set	
Benign AP	0.742	Benign AP	0.394
Malignant AP	0.54	Malignant AP	0.332
mAP	0.641	mAP	0.363

Before retraining:

Tumorcenter test set		Uniformly cut test set	
Benign AP	0.839	Benign AP	0.310
Malignant AP	0.846	Malignant AP	0.257
mAP	0.842	mAP	0.284



Related Work

- **P Xi. et.al. IEEE MeMeA 2018:** VGGNet for classification and ResNet for localizing abnormalities
- **N Wu. arXiv:1903.08297:** ResNet-22 for lesion detection and hybrid model for validation
- **TG Debelee. LNICST, volume 244:** PCA for feature dimensionality reduction, KNN for classification, Particle Swarm Optimized Wavelet Neural Network (PSOWNN)



Future Works

- Enhance the contrast of mammograms
- Use ResNet to reduce the training time.
- Use more traditional machine learning method, such as k-nearest neighbors algorithm (KNN), support vector machine for classification
- Use other public database, e.g.: Digital Database for Screening Mammography (DDSM)



References

1. Ferlay, J., Héry, C., Autier, P. & Sankaranarayanan, R. (2010). Global burden of breast cancer. *Breast cancer epidemiology*, 1–19, Springer.
2. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. *European conference on computer vision* (pp. 21-37). Springer, Cham.
3. Ting, F. F., Tan, Y. J., & Sim, K. S. (2019). Convolutional neural network improvement for breast cancer classification. *Expert Systems with Applications*, 120, 103-115.
4. Separable Convolution. <https://yinguobing.com/separable-convolution/>
5. Breast cancer: Prevention and Control. Retrieved from: <https://www.who.int/cancer/detection/breastcancer/en/>



Q&A

Thank you for listening!