Using Deep Learning for Breast Cancer Diagnosis

LYU1704

- Li Qi (1155062147) , Li Wei(1155062148)
- Supervisor: Prof. Michael R. Lyu
- CUHK CSE

01. Introduction

Introduction

01 Motivation

02 Background

03 Objective

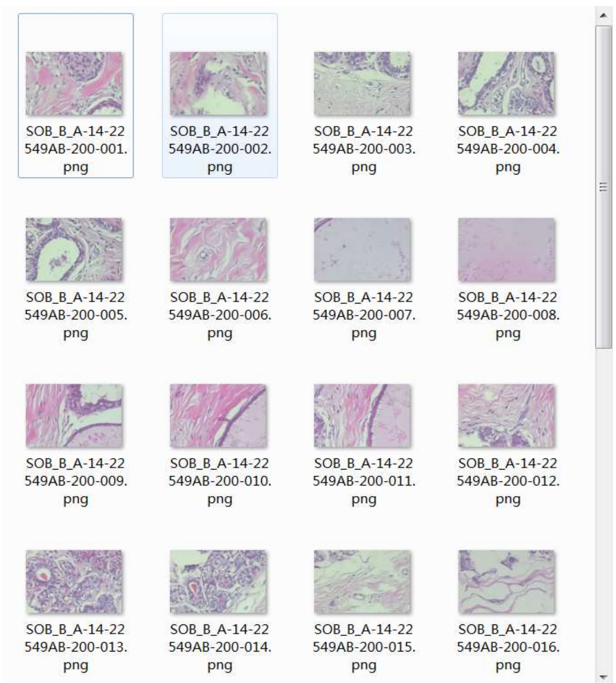
Introduction

01 Motivation

02 Background

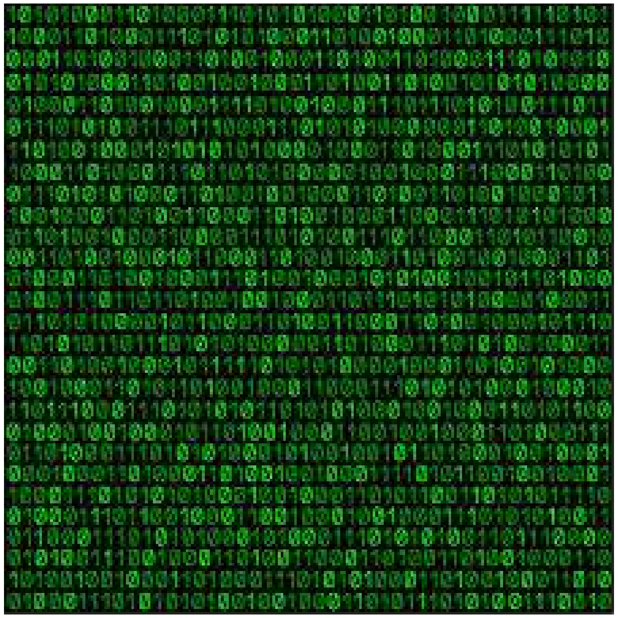
03 Objective

Introduction: Motivation

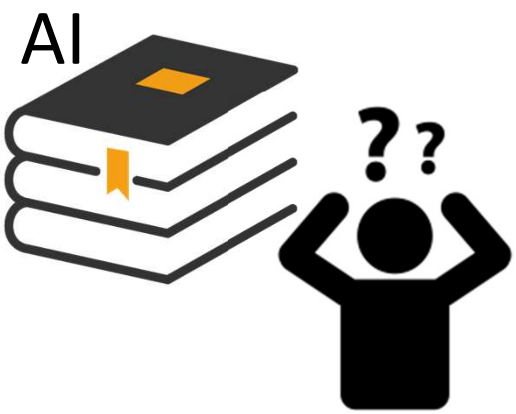


- Breast cancer diagnosis
  - **10+** gigapixels per patient
  - agreement in diagnosis < **48%**

Introduction: Motivation



- Current automatic diagnosis
  - Statistics
  - Jargons
  - Codes



Introduction

01 Motivation

02 Background

03 Objective

Introduction: Background

Naïve Bayes → SVM → Neural Network → DNN → DNN (on GPU)

Introduction: Background

- Deep Learning for Medical Images
  - structures detection
  - segmentation
  - labeling and captioning
  - computer aided detection or diagnosis

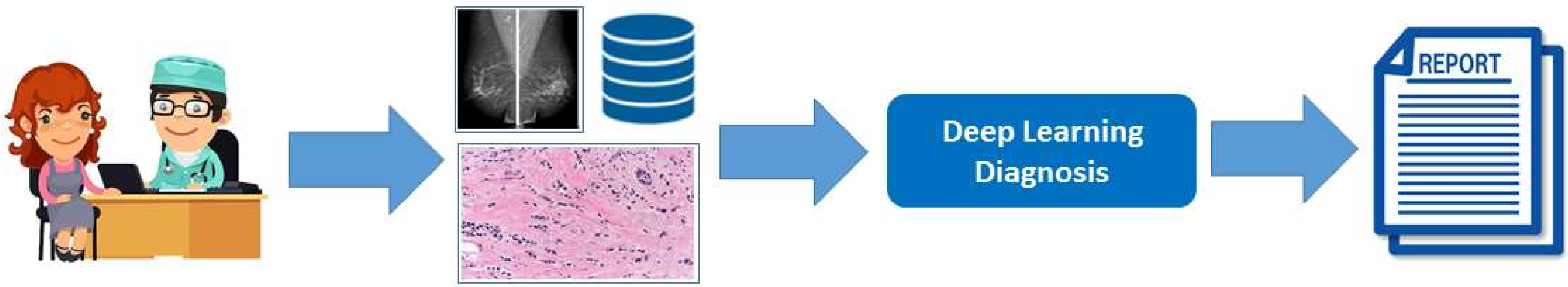
Introduction

01 Motivation

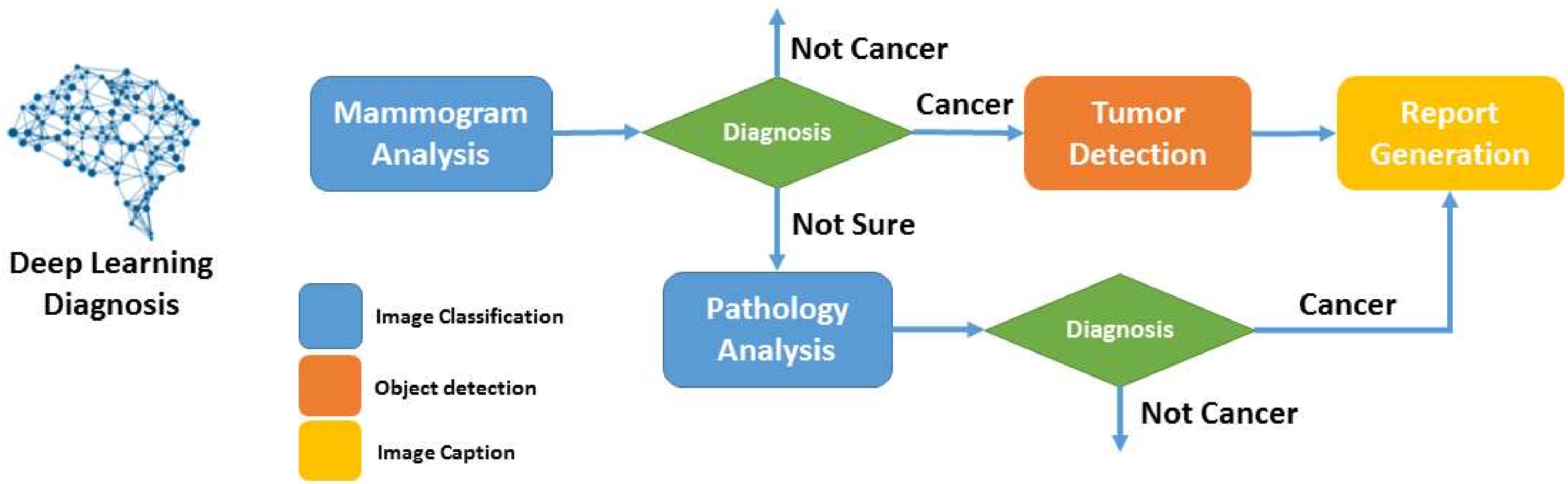
02 Background

03 Objective

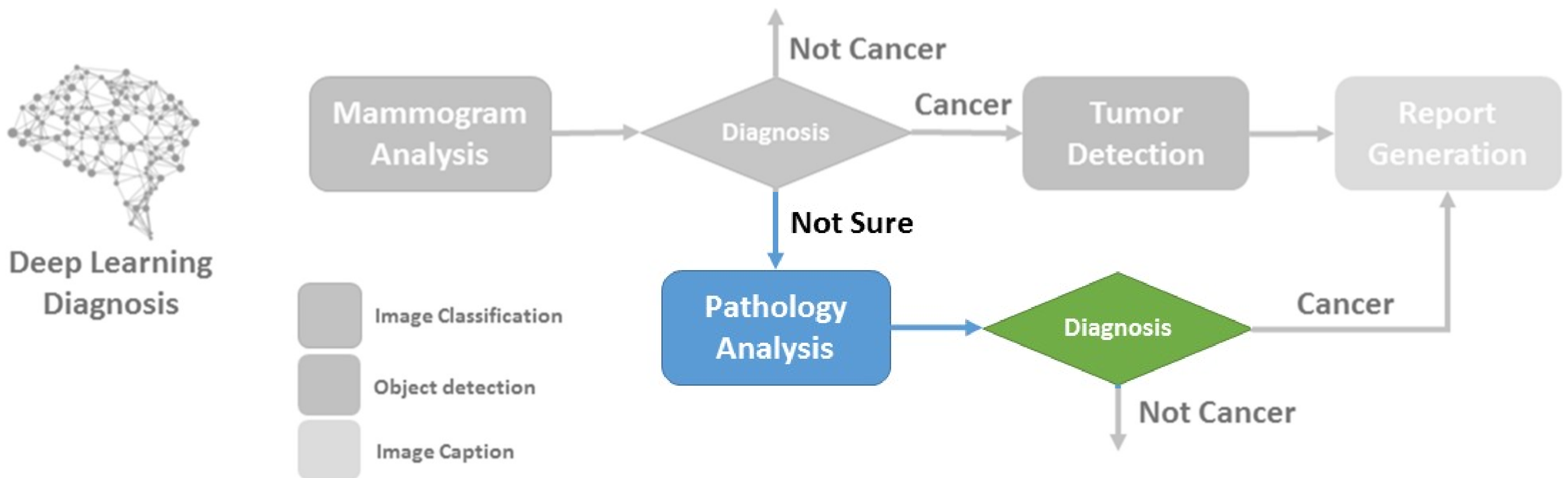
Introduction: Objective



Introduction: Objective



Introduction: Objective



02. Related Work

Related Work

01 Naïve Bayes for Breast Cancer Diagnosis

02 SVM for Remote Breast Cancer Diagnosis

03 Classification of Skin Cancer with DNN

Related Work: Naïve Bayes

01 Naïve Bayes for Breast Cancer Diagnosis

- 42 features
- Multiple models
  - Competitive neural network
  - Fuzzy C-means
  - K-means
  - Gaussian mixture model
- 500 images from 50 patients

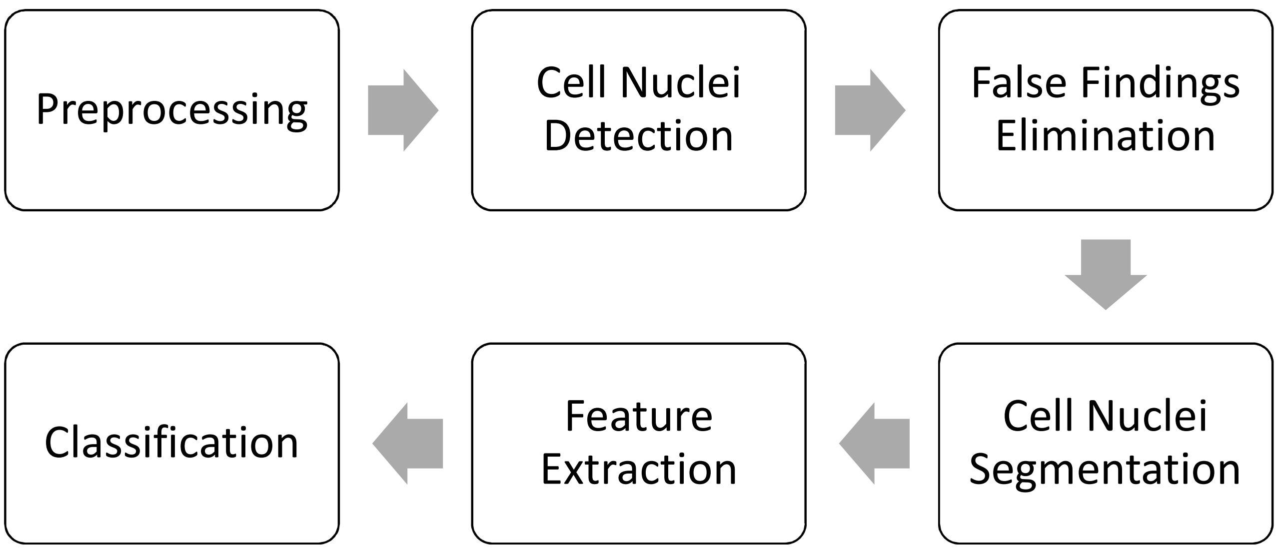
	KM	FCM	GMM	CNN
Patients Accuracy	100.00%	96.00%	100.00%	98.00%
Image Accuracy	90.22%	85.78%	88.00%	89.56%

Kowal et al.

Related Work: SVM

01 Naïve Bayes for Breast Cancer Diagnosis

02 SVM for Remote Breast Cancer Diagnosis



- 3260 images
- Acc=82.6%

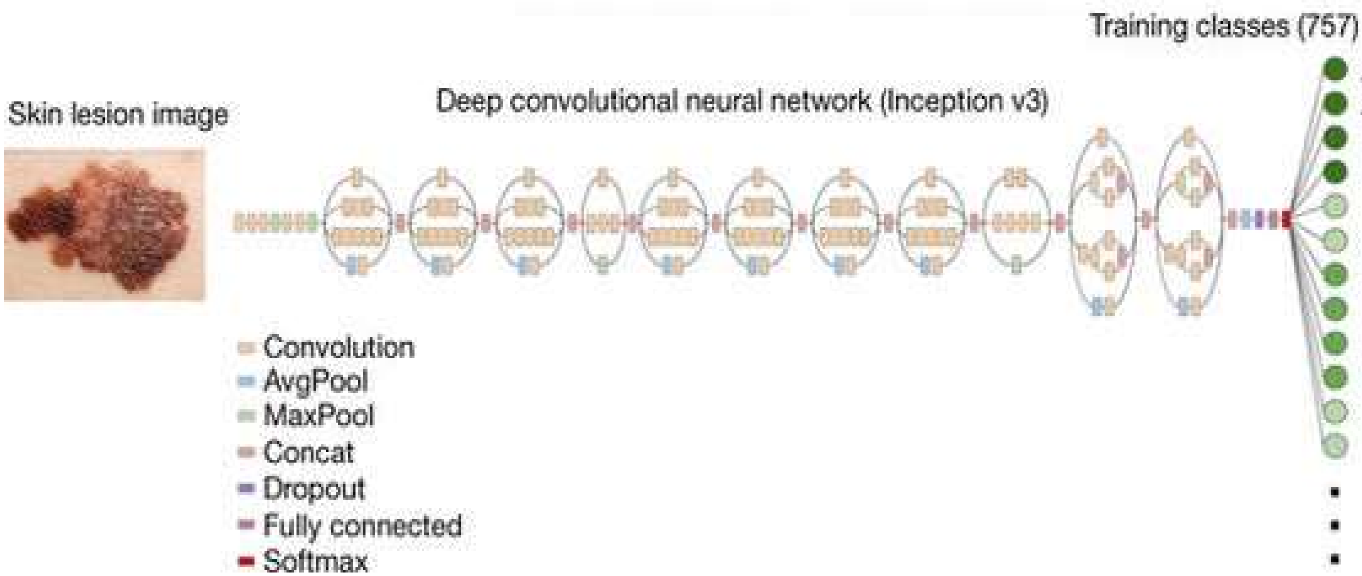
George et al.

Related Work: DNN

01 Naïve Bayes for Breast Cancer Diagnosis

02 SVM for Remote Breast Cancer Diagnosis

03 **Classification of Skin Cancer with DNN**



- **Minimum preprocessing**
- Acc=72.1±0.9%
- Human acc=66.0%

Esteva et al.

03. Methods

Method

01 Dataset

02 Preprocess

03 Model Architecture

04 Aggregation

05 Workflow

Method

01 Dataset

02 Preprocess

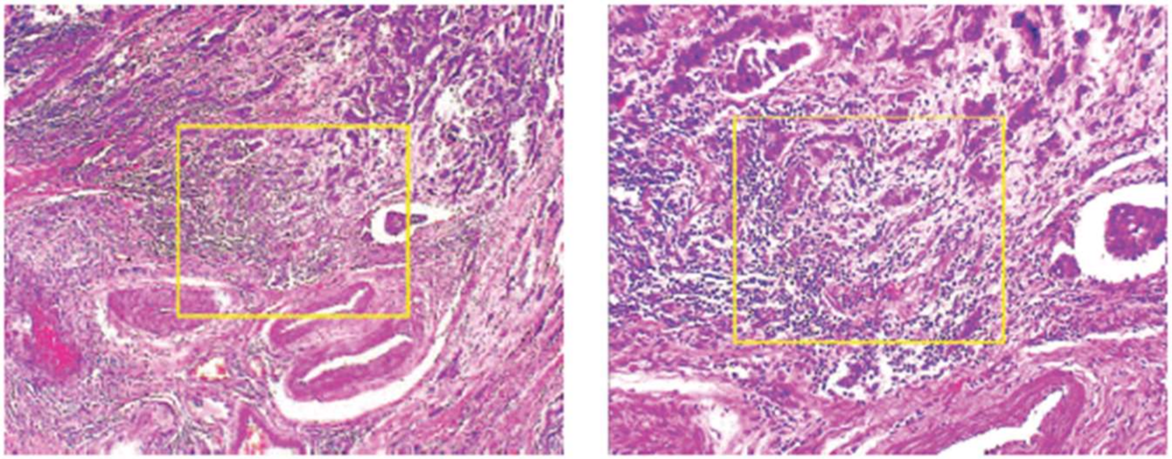
03 Model Architecture

04 Aggregation

05 Workflow

Dataset

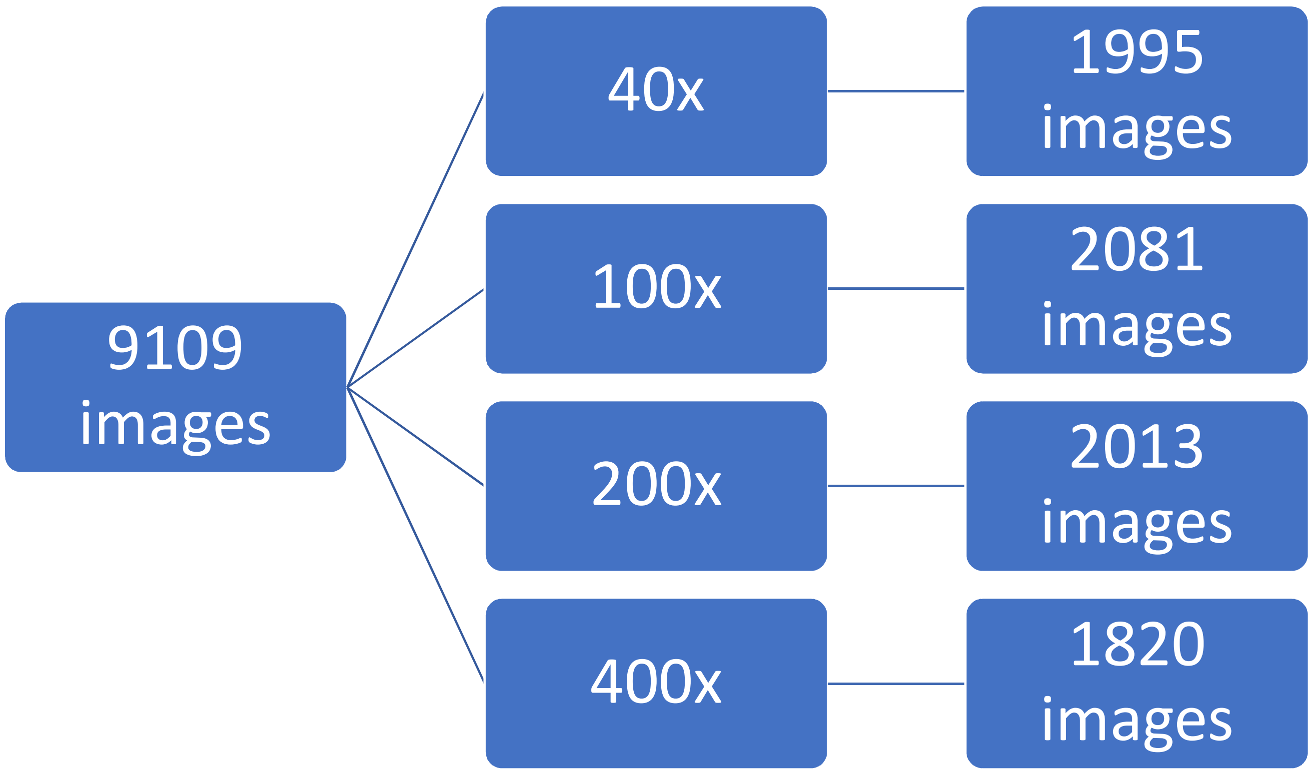
Breast Cancer Histopathological Image Classification (BreakHis)



different magnifying factors (40x, 100x, 200x, and 400x)

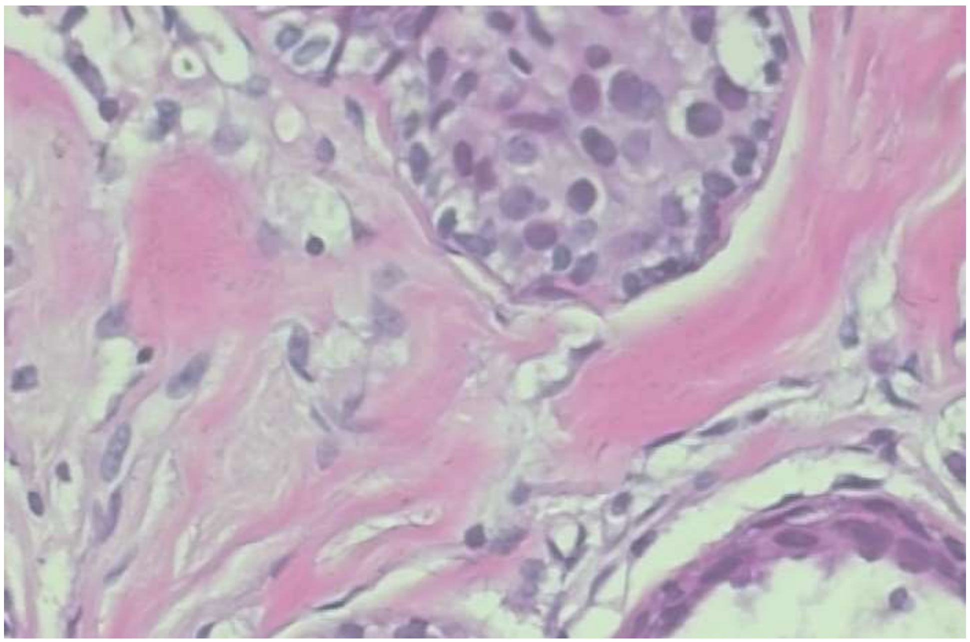
Dataset

Breast Cancer Histopathological Image Classification (BreakHis)



Dataset

Breast Cancer Histopathological Image Classification (BreakHis)



Stain:	hematoxylin and eosin
Biopsy procedure:	Surgical Open Biopsy
Format:	3-channel RGB 8-bit depth

Method

01 Dataset

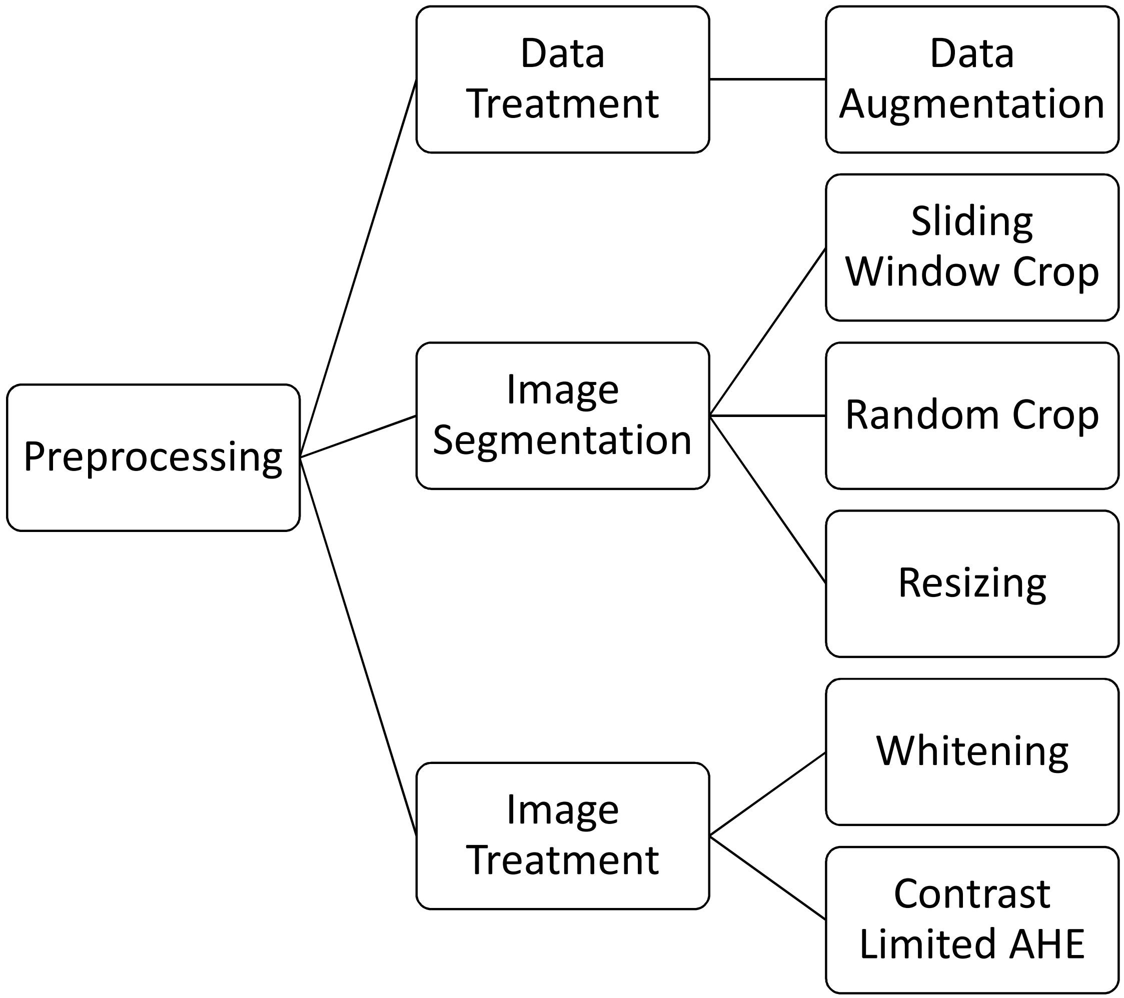
02 Preprocess

03 Model Architecture

04 Aggregation

05 Workflow

Preprocess

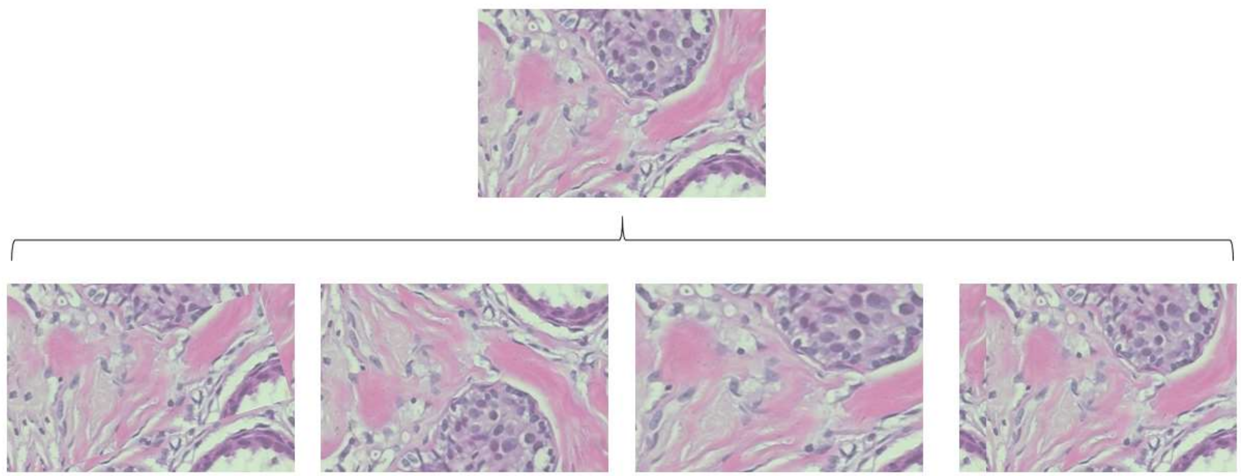


Preprocessing
Data Treatment
Data Augmentation
Image Segmentation
Sliding Window Crop
Random Crop
Resizing
Image Treatment
Whitening
Contrast Limited AHE

Preprocess: Data Augmentation

01 Data Augmentation

Task: make dataset larger

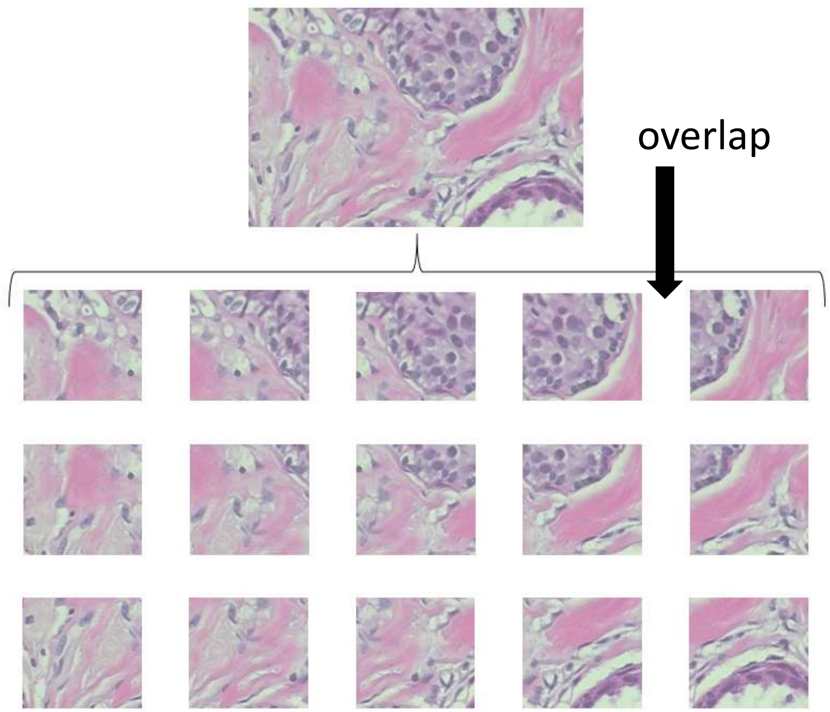


Preprocess: Sliding Window Crop

01 Data Augmentation

02 Sliding Window Crop

Idea: crop systematically



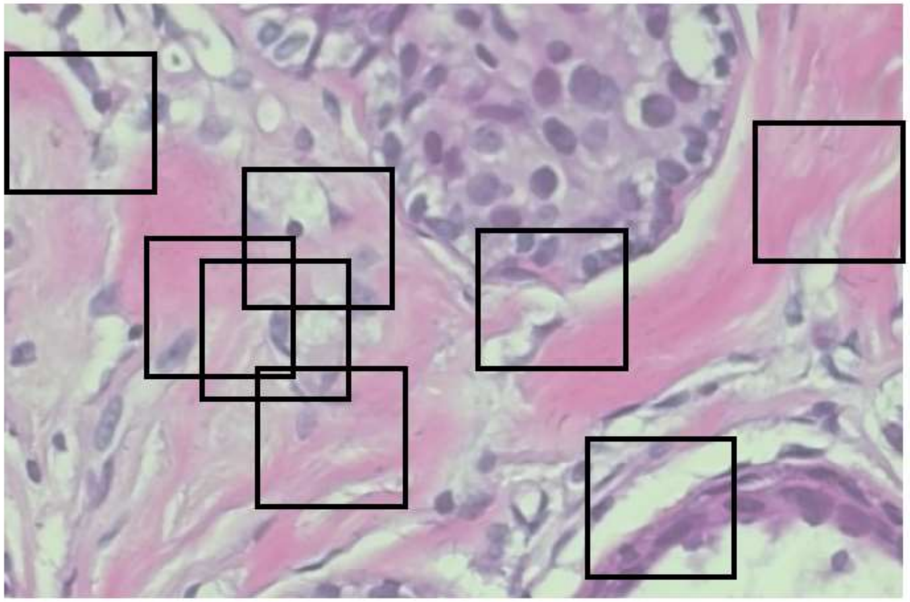
Preprocess: Random Crop

01 Data Augmentation

02 Sliding Window Crop

03 **Random Crop**

Idea: crop randomly



Preprocess: Resizing

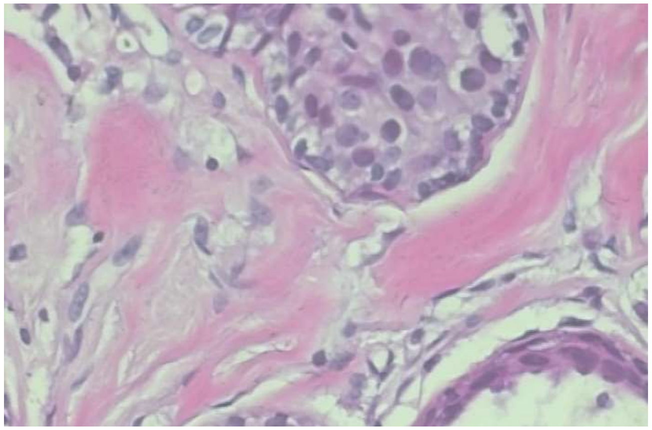
01 Data Augmentation

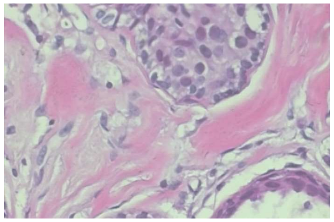
02 Sliding Window Crop

03 Random Crop

04 **Resizing**

Idea: simply shrink





Preprocess: Whitening

01 Data Augmentation

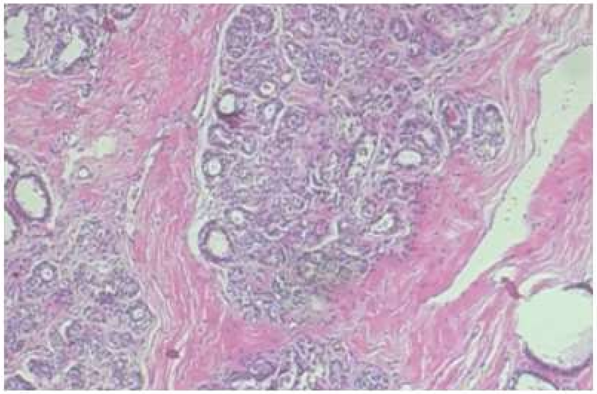
02 Sliding Window Crop

03 Random Crop

04 Resizing

05 **Whitening**

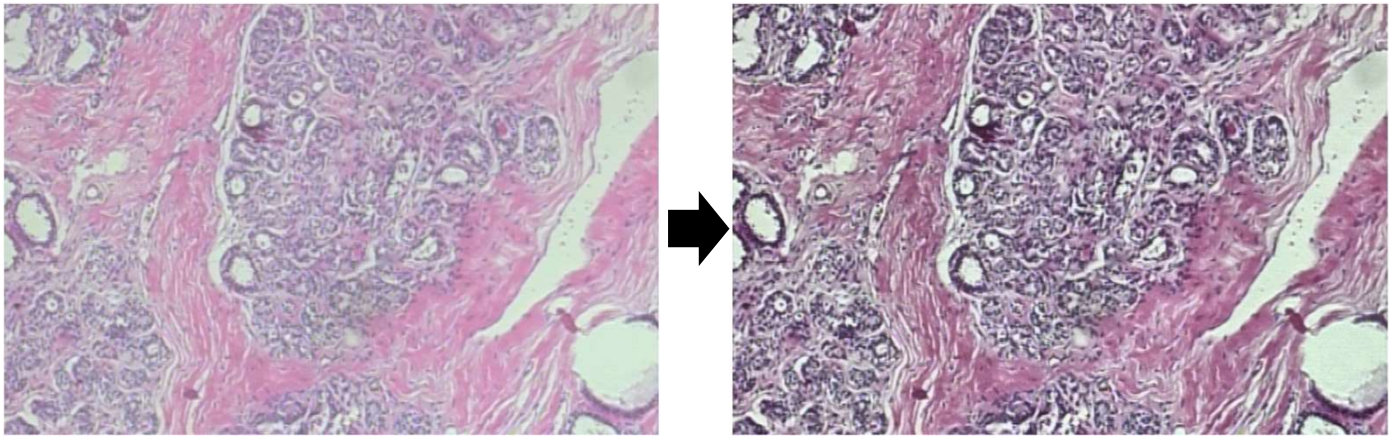
Idea: remove extra information



Preprocess: Contrast Limited AHE

- 01 Data Augmentation
- 02 Sliding Window Crop
- 03 Random Crop
- 04 Resizing
- 05 Whitening
- 06 **Contrast Limited AHE**

Idea: make image clearer



Method

01 Dataset

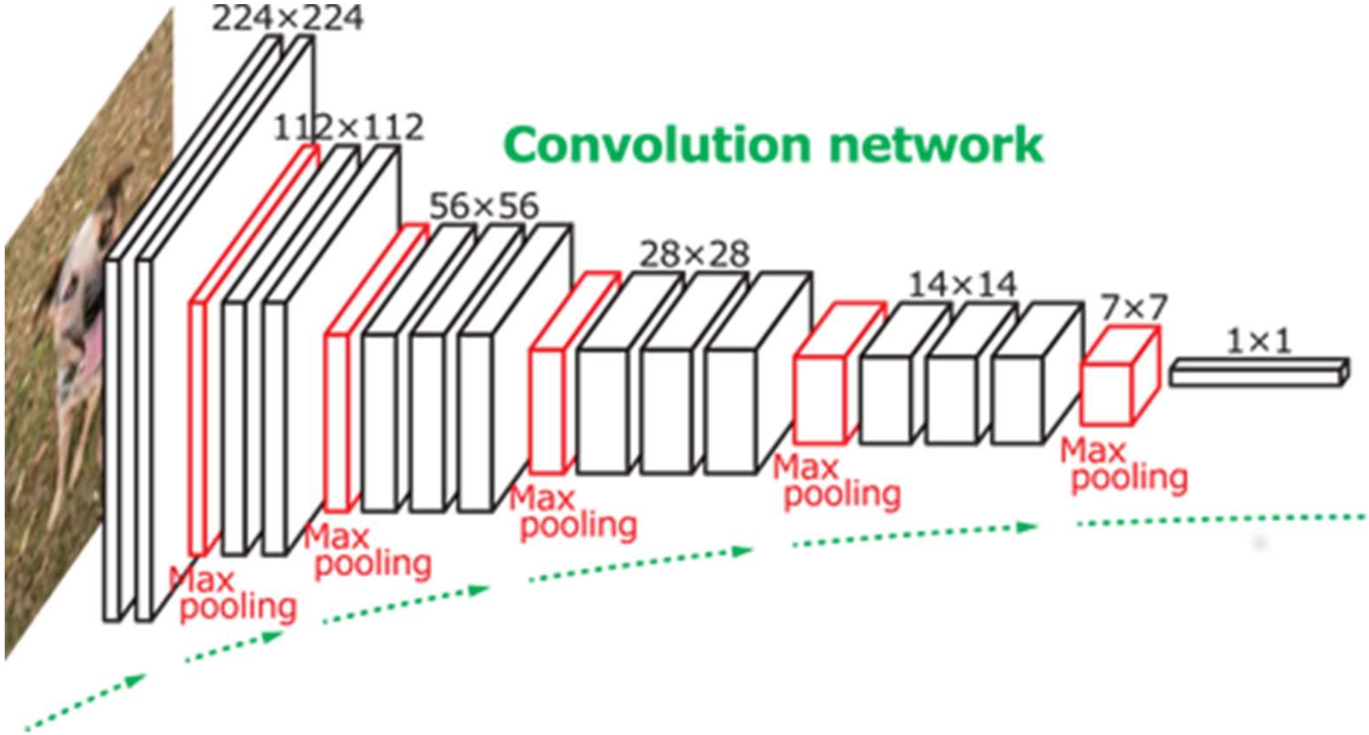
02 Preprocess

03 Model Architecture

04 Aggregation

05 Workflow

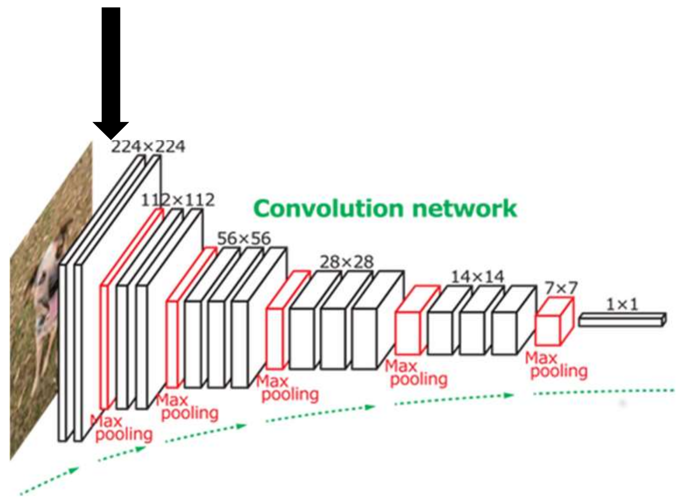
Model Architecture: CNN



Model Architecture: Input Layer

01 Input Layer

Task: read input

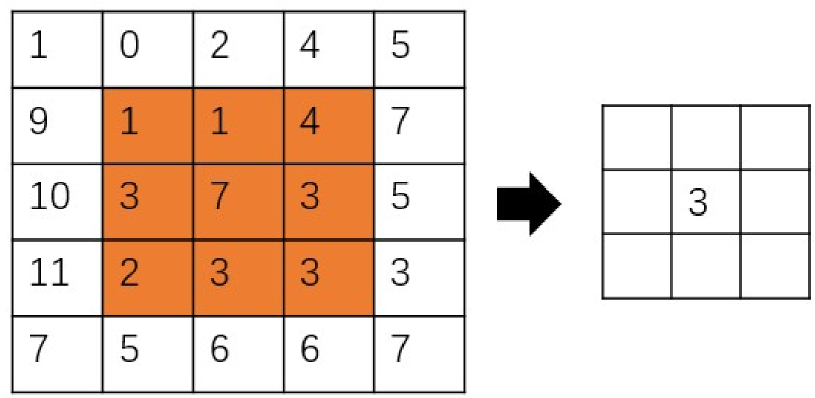


Model Architecture: Convolution Layers

01 Input Layer

02 Convolution Layers

Task: learn feature map



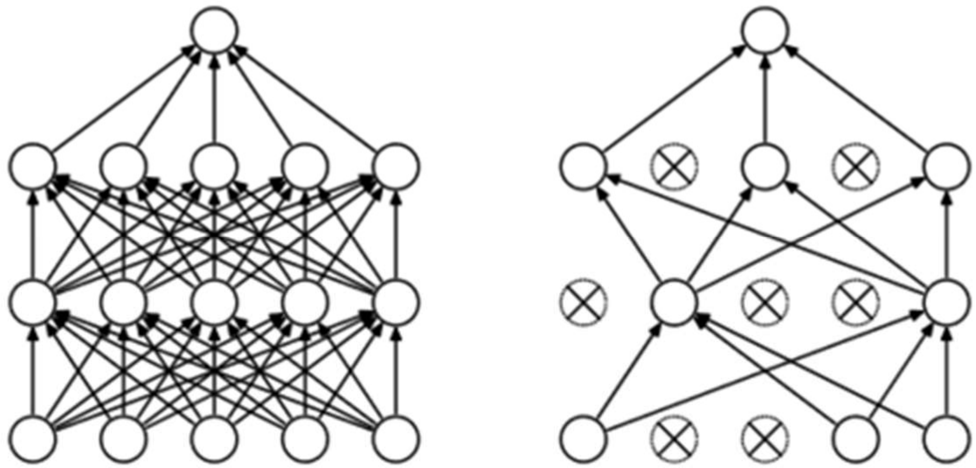
Model Architecture: Dropout

01 Input Layer

02 Convolution Layers

03 Dropout

Task: eliminate free riding



(a) Standard Neural Net (b) After applying dropout.

Model Architecture: Residual Blocks

01 Input Layer

02 Convolution Layers

03 Dropout

04 Residual Blocks

Task: fix degradation problem

$$\mathrm{H}(\mathrm{x}) - \mathrm{x} \rightarrow \mathrm{F}(\mathrm{x})$$

$$\mathrm{H}(\mathrm{x}) = \mathrm{F}(\mathrm{x}) + \mathrm{x}$$

Model Architecture: Residual Blocks

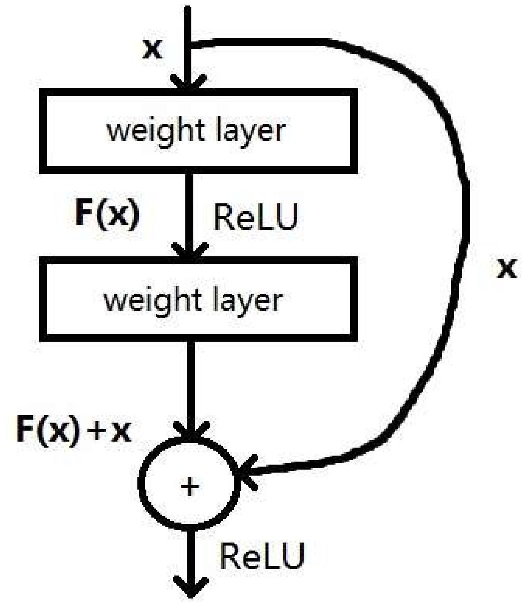
01 Input Layer

02 Convolution Layers

03 Dropout

04 Residual Blocks

Task: fix degradation problem



ImageNet Large Scale Visual Recognition Challenge 2015 winner

Model Architecture: Pooling Layers

01 Input Layer

02 Convolution Layers

03 Dropout

04 Residual Blocks

05 **Pooling Layers**

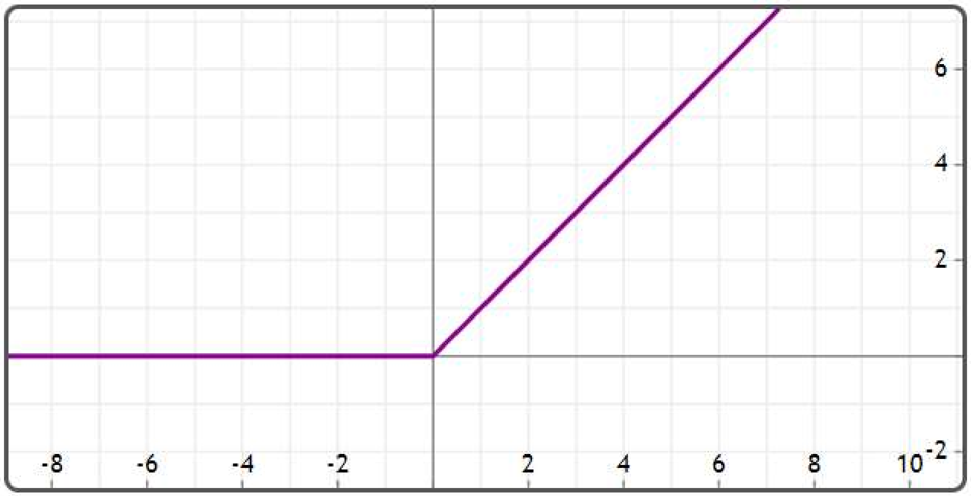
Task: subsampling

$$y = \max_{\text{local window}}(x)$$

Model Architecture: Activation Layers

Task: add non-linearity

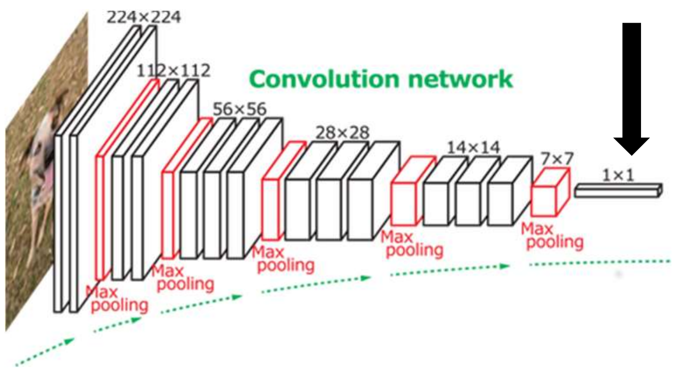
$$f(x) = \max(0, x)$$



Model Architecture: Fully Connected Layer

- 01 Input Layer
- 02 Convolution Layers
- 03 Dropout
- 04 Residual Blocks
- 05 Pooling Layers
- 06 Activation Layers
- 07 **Fully Connected Layer**

Task: make output



Method

01 Dataset

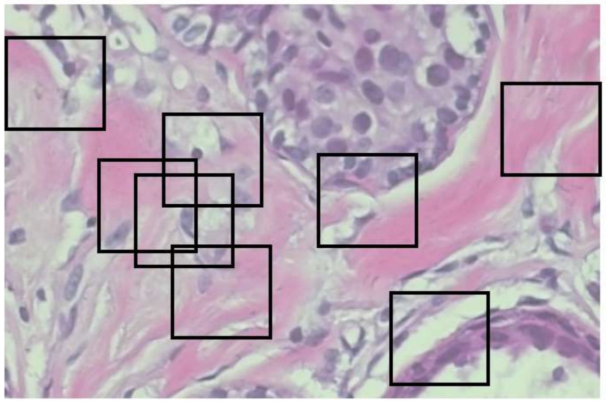
02 Preprocess

03 Model Architecture

04 **Aggregation**

05 Workflow

Aggregation



patch to image

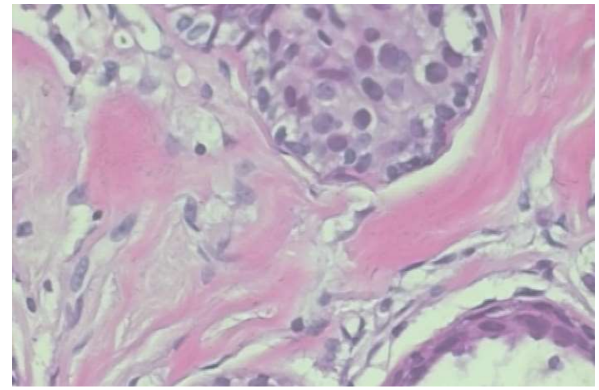


image to patient

Aggregation: Sum

01 Sum

Idea: posteriori ≈ prior

$$P(w_k|x_i) = P(w_k)(1+\delta), \delta \ll 1$$

$$\text{Prediction} = \text{argmax}[(1-R)P(w_k) + \sum P(w_k|x_i)]$$

Aggregation: Plurality Vote

01 Sum

02 Plurality Vote

Idea: wisdom of crowds

$$\text{Prediction} = \text{argmax}(\sum \Delta_i)$$

Aggregation: Average

01 Sum

02 Plurality Vote

03 **Average**

Idea: weighted voting

$$\text{Prediction} = \text{argmax}(\frac{1}{\text{R}}\sum P(w_k|x_i))$$

Aggregation: Exist

01 Sum

02 Plurality Vote

03 Average

04 **Exist**

Idea: one bad apple spoils the whole barrel

$$\text{Predition} = \begin{cases} \text{malignant}, & \sum \Delta_i > 0 \\ \text{benign}, & \sum \Delta_i = 0 \end{cases}$$

Aggregation: Exist-n

01 Sum

02 Plurality Vote

03 Average

04 Exist

05 **Exist-n**

Idea: *n* bad apple(s) spoil the whole barrel

$$\text{Predition} = \begin{cases} \text{malignant}, & \sum \Delta_i \geq \text{n} \\ \text{benign}, & \sum \Delta_i < \text{n} \end{cases}$$

Method

01 Dataset

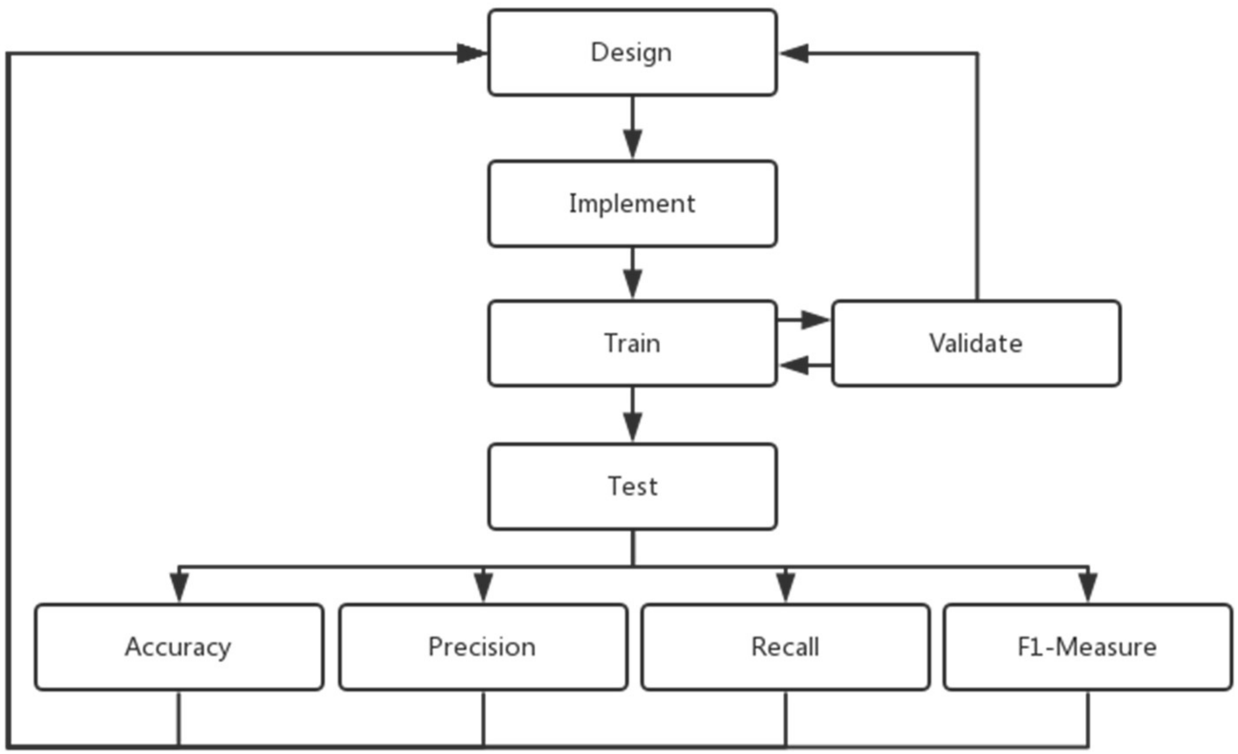
02 Preprocess

03 Model Architecture

04 Aggregation

05 Workflow

Workflow



Design
Implement
Train
Validate
Test
Accuracy
Precision
Recall
F1-Measure

04. Results

Results

01 Results of different methods

02 Results analysis

03 Comparison with past papers

04 Limitations

Results

01 Results of different methods

02 Results analysis

03 Comparison with past papers

04 Limitations

Results of different methods

1 Preprocess method

2 Model structure

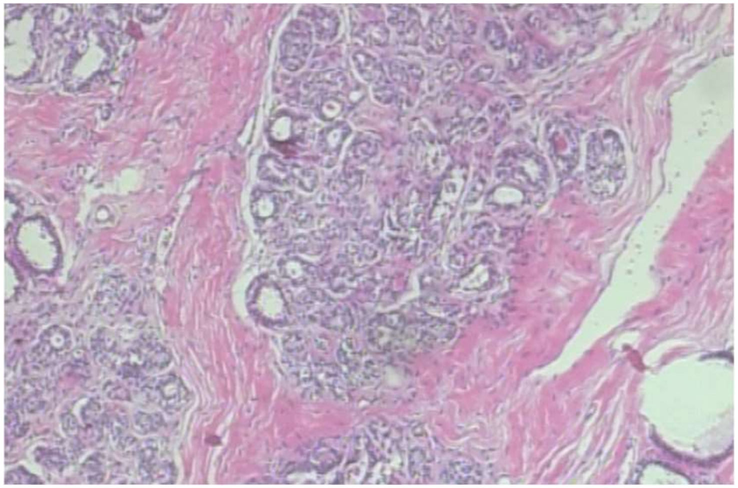
3 Segmentation method

Results of different preprocess methods

1. Preprocess method
2. Model structure
3. Segmentation method

Raw image

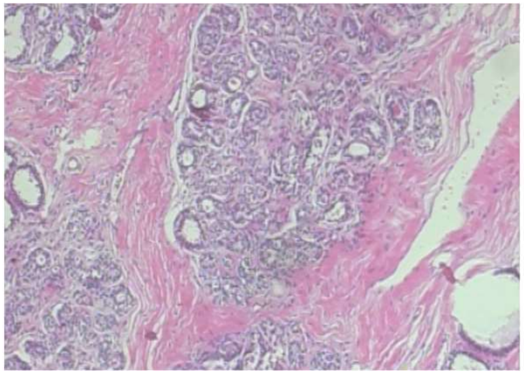
01 Raw image

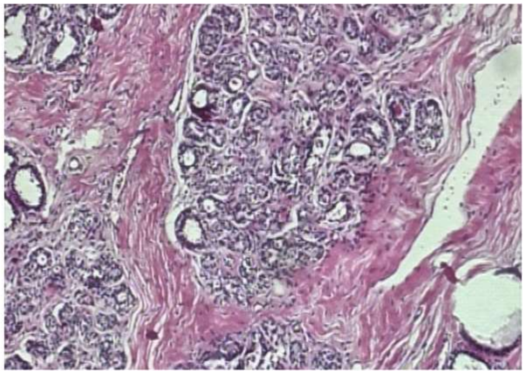


Contrast-Limited Adaptive Histogram Equalization

01 Raw image

02 Contrast-Limited Adaptive Histogram Equalization (CLAHE)



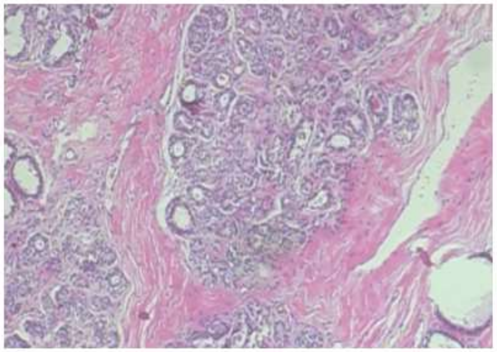


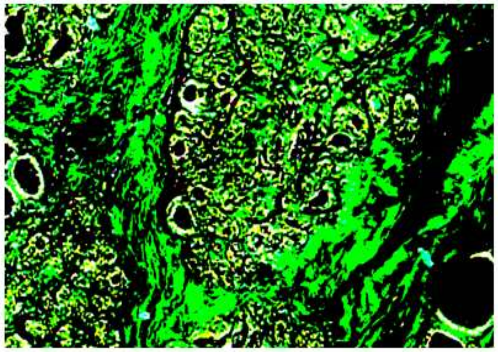
Whitening

01 Raw image

02 Contrast-Limited Adaptive Histogram Equalization (CLAHE)

03 Whitening





Demean

01 Raw image

02 Contrast-Limited Adaptive Histogram Equalization (CLAHE)

03 Whitening

04 Demean

$$DemeanImage = RawImage - mean$$

Subtract gaussian smooth image and CLAHE

01 Raw image

02 Contrast-Limited Adaptive Histogram Equalization (CLAHE)

03 Whitening

04 Demean

05 Gaussian + CLAHE

$$GuassianImage = CLAHE(RawImage - GaussianSmoothedImage)$$

Results of different preprocess methods

01 Raw image

02 Contrast-Limited Adaptive Histogram Equalization (CLAHE)

03 Whitening

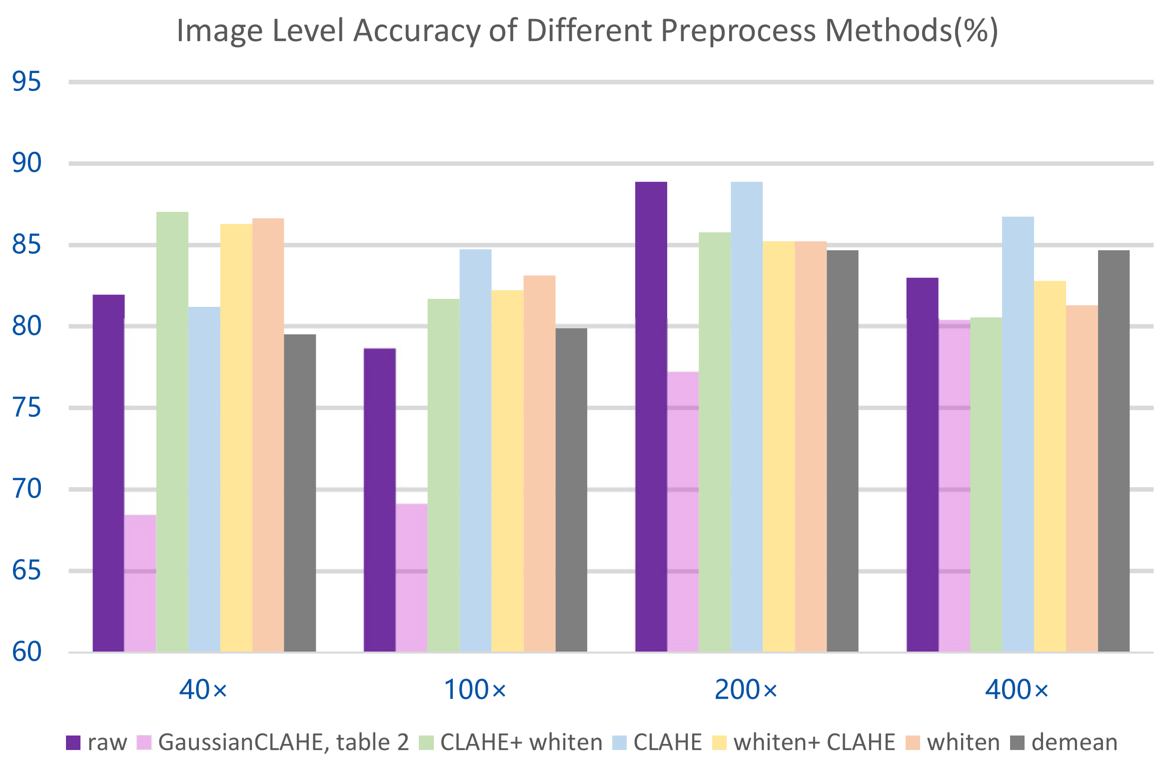
04 Demean

05 Gaussian + CLAHE

06 CLAHE + Whitening

07 Whitening + CLAHE

Results of different preprocess methods

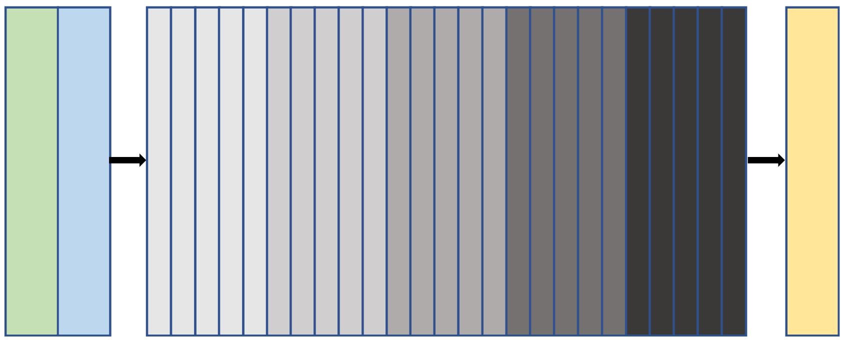


- In general, *CLAHE* is the best preprocess method
- *CLAHE* won't work when the magnification factor is 40× while *whiten* operation can help model to overcome this problem. (*CLAHE + whiten*)

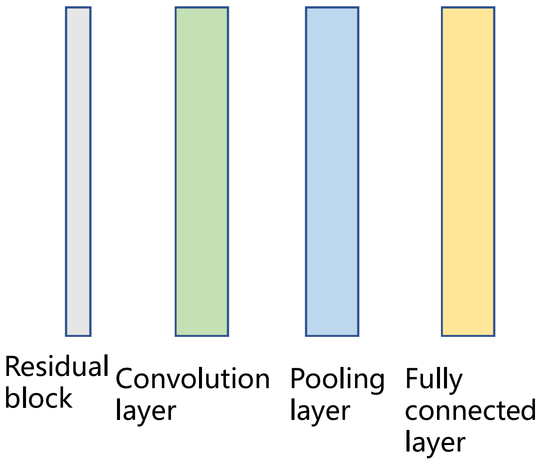
Results of different Model structures

1. Preprocess method
2. Model structure
3. Segmentation method

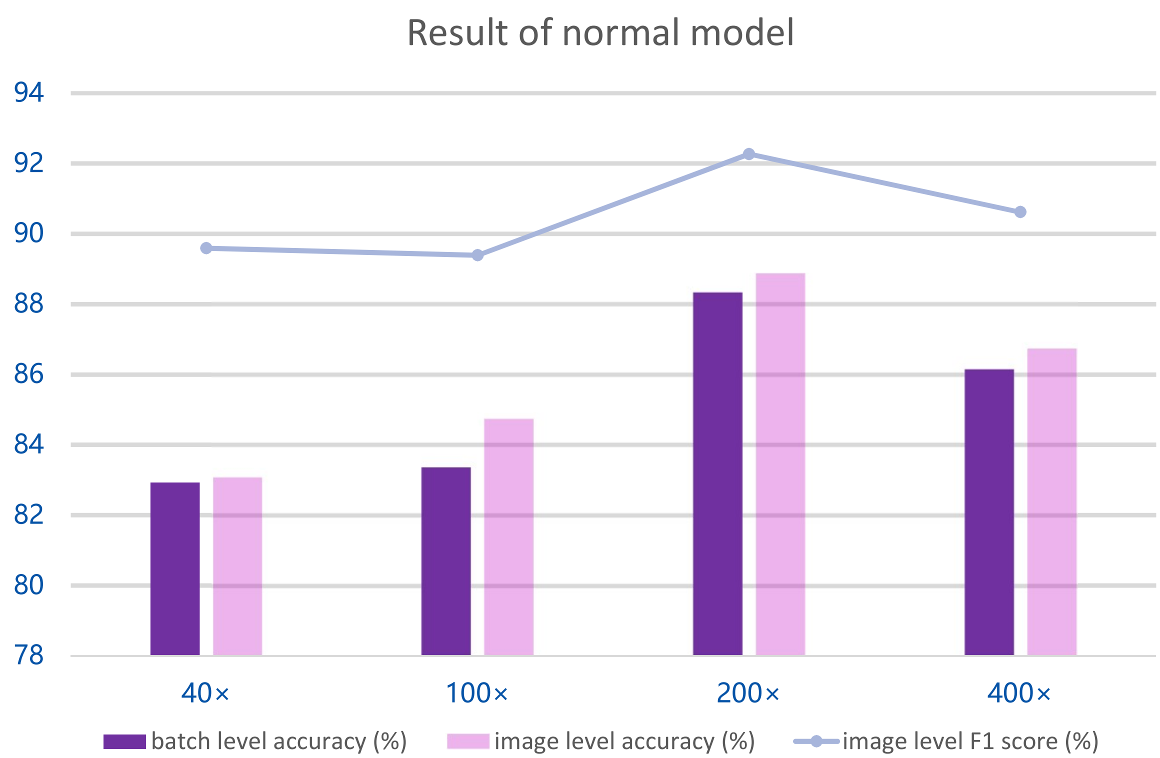
Results of different model architectures: normal model



Normal model architecture we used

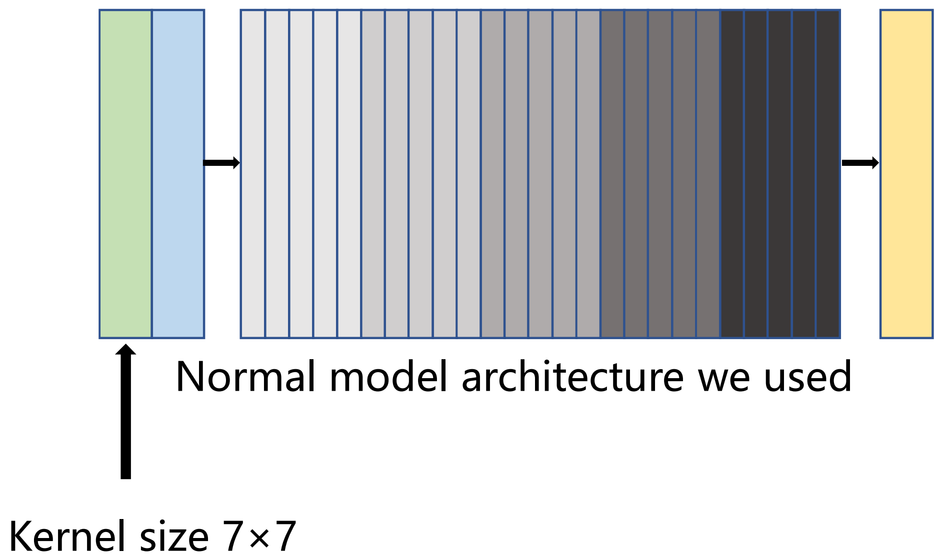


Results of different model architectures



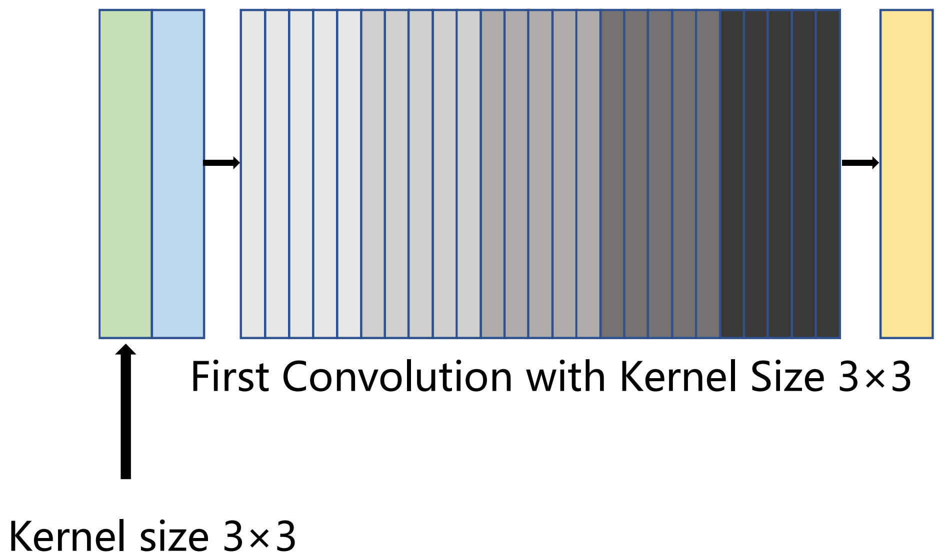
First Convolution with Kernel Size 3×3

01 3×3 convolution



First Convolution with Kernel Size 3×3

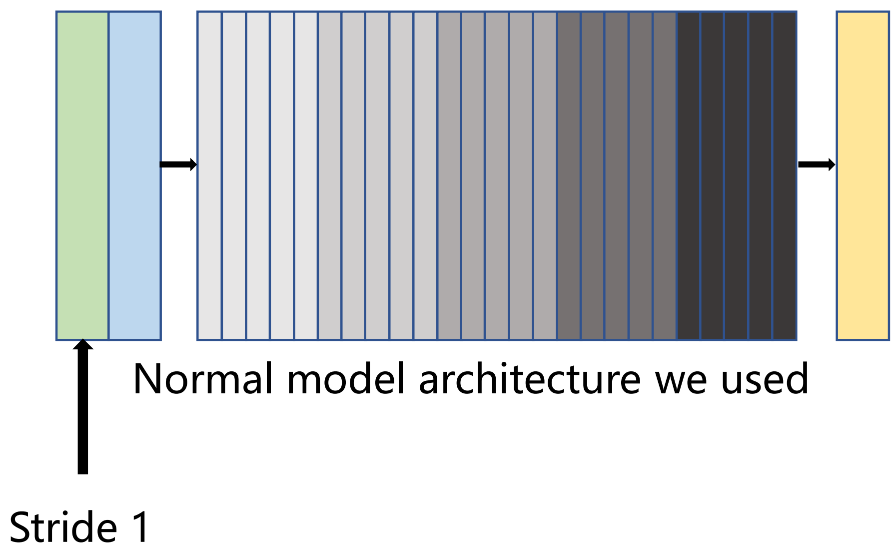
01 3×3 convolution



First Convolution with Stride 2

01 3×3 convolution

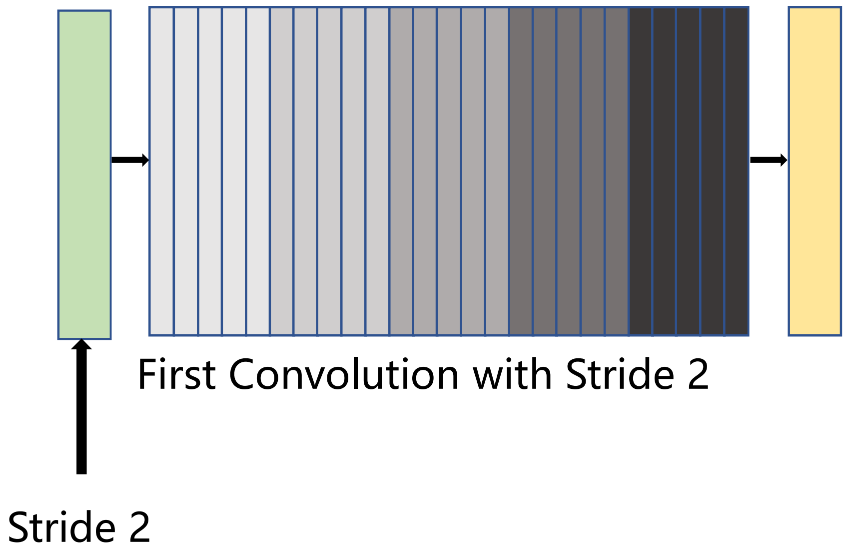
02 Stride 2



First Convolution with Stride 2

01 3×3 convolution

02 Stride 2



Model with Feature Maps Doubled

01 3×3 convolution

02 Stride 2

03 Feature maps doubled

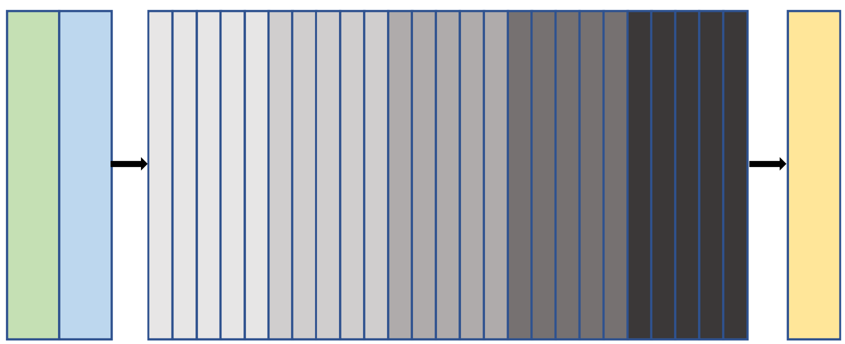
Model with Two Pooling Layers Before ResNet

01 3×3 convolution

02 Stride 2

03 Feature maps doubled

04 Two pooling layers



Normal model architecture we used

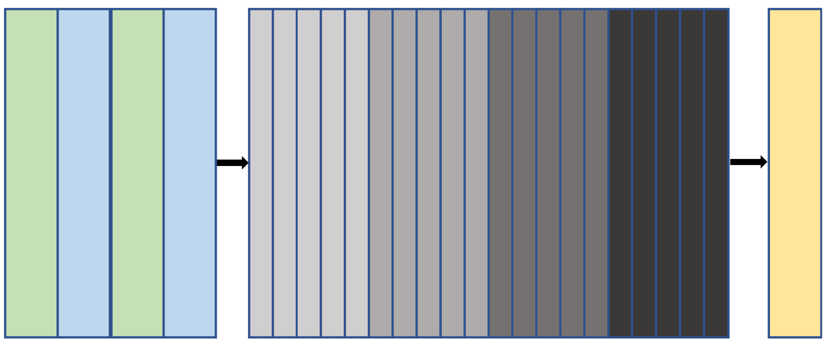
Model with Two Pooling Layers Before ResNet

01 3×3 convolution

02 Stride 2

03 Feature maps doubled

04 Two pooling layers



Model with Two Pooling Layers Before ResNet

Model with Dropout

01 3×3 convolution

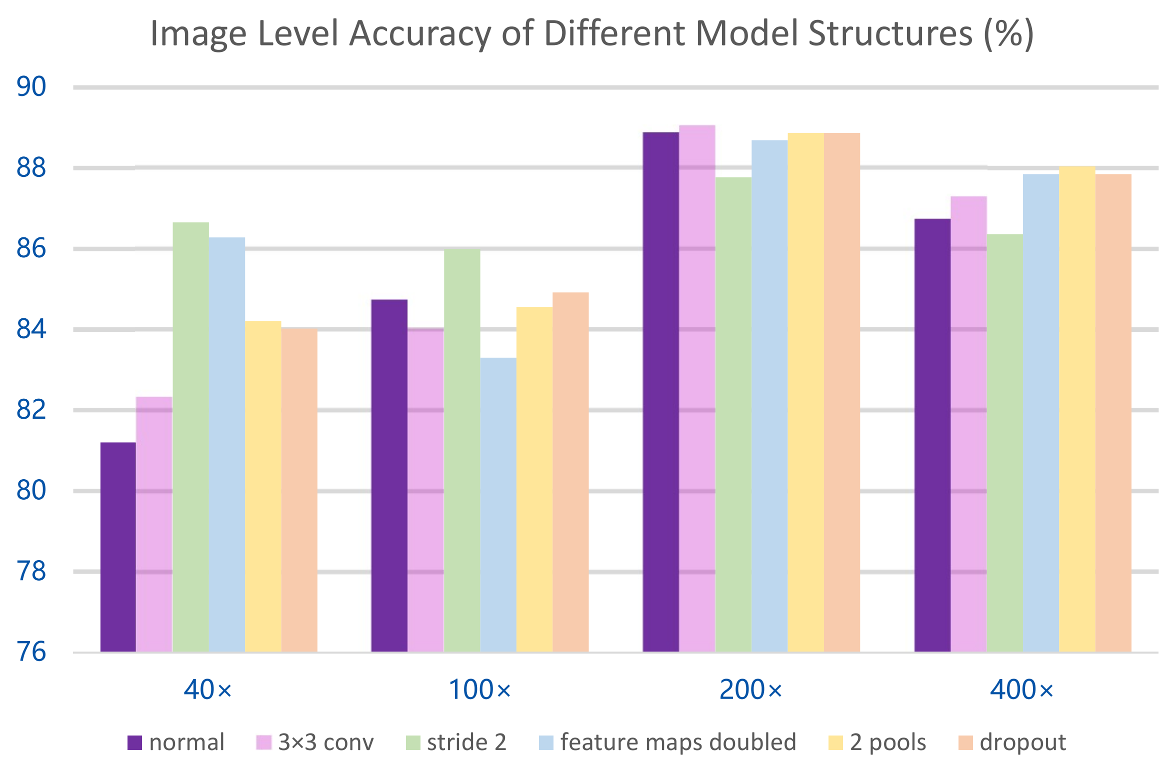
02 Stride 2

03 Feature maps doubled

04 Two pooling layers

05 Dropout

Results of Different Model Structures



- In general, *stride 2* is the best model architecture
- ***Feature maps doubled*** also makes sense, which means that the results can be better with the increase of model structure's complexity

Results of preprocess methods

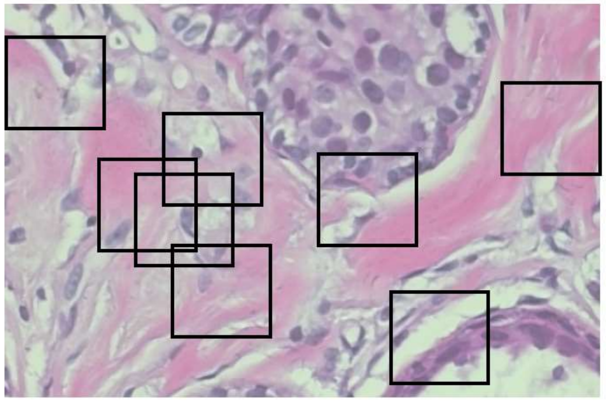
1 Preprocess method

2 Model structure

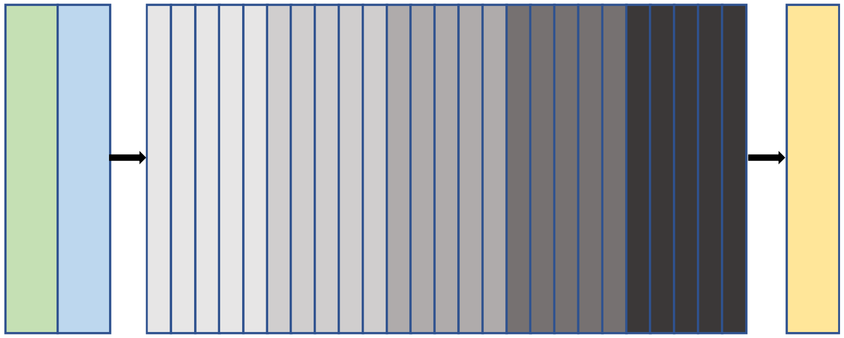
3 Segmentation method

Random crop with input size 256×256

01 Random, 256×256



Random crop

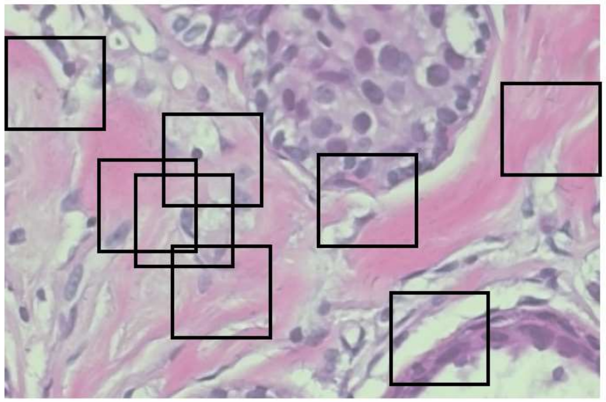


Structure of the model we used with input size 256×256

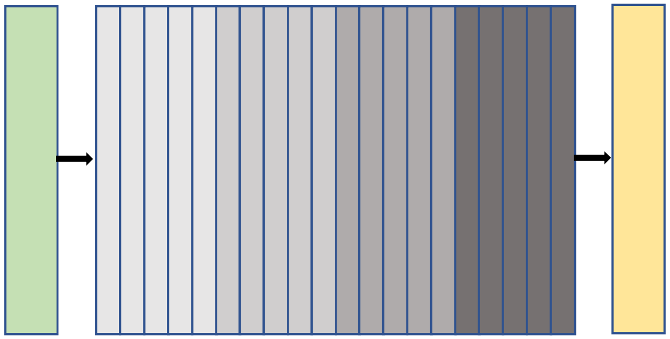
Random crop with input size 64×64

01 Random, 256×256

02 Random, 64×64



Random crop



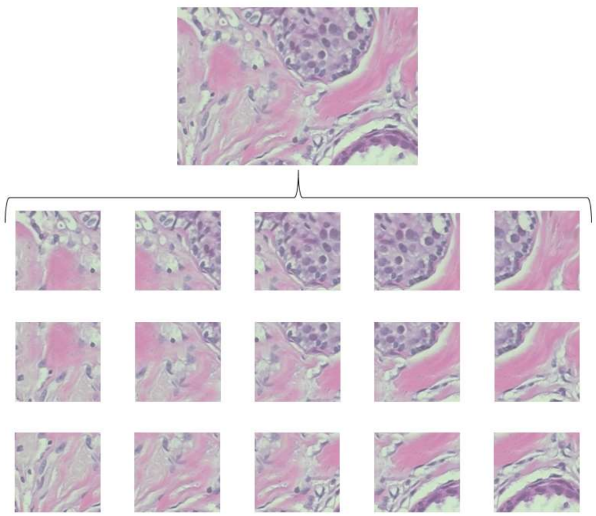
Structure of the model we used with input size 64×64

Sliding window crop with input size 128×128

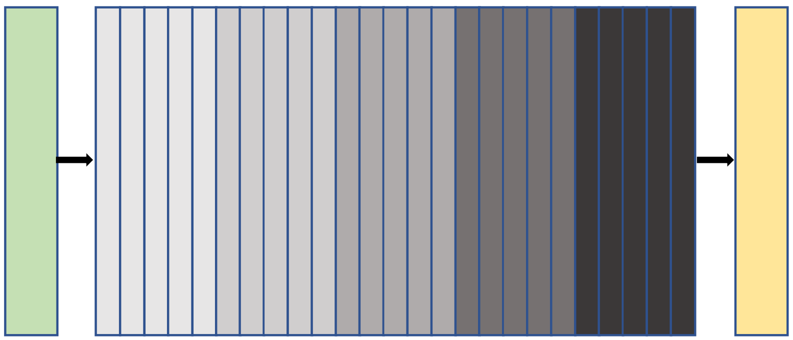
01 Random, 256×256

02 Random, 64×64

03 Sliding window, 128×128



Sliding window crop



Structure of the model we used with input size 128×128

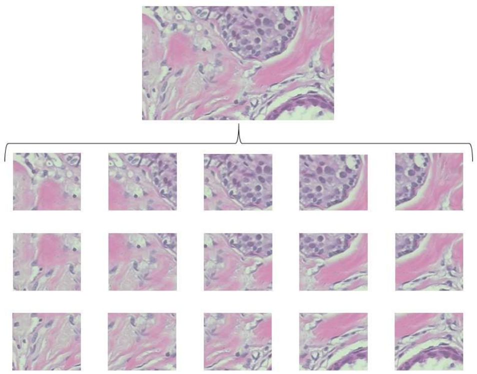
Sliding window crop with input size 64×64

01 Random, 256×256

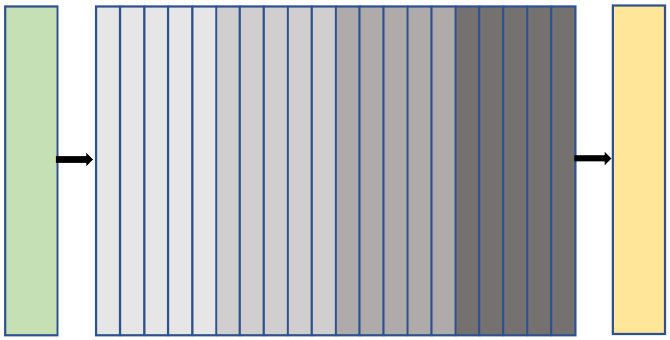
02 Random, 64×64

03 Sliding window, 128×128

04 Sliding window, 64×64

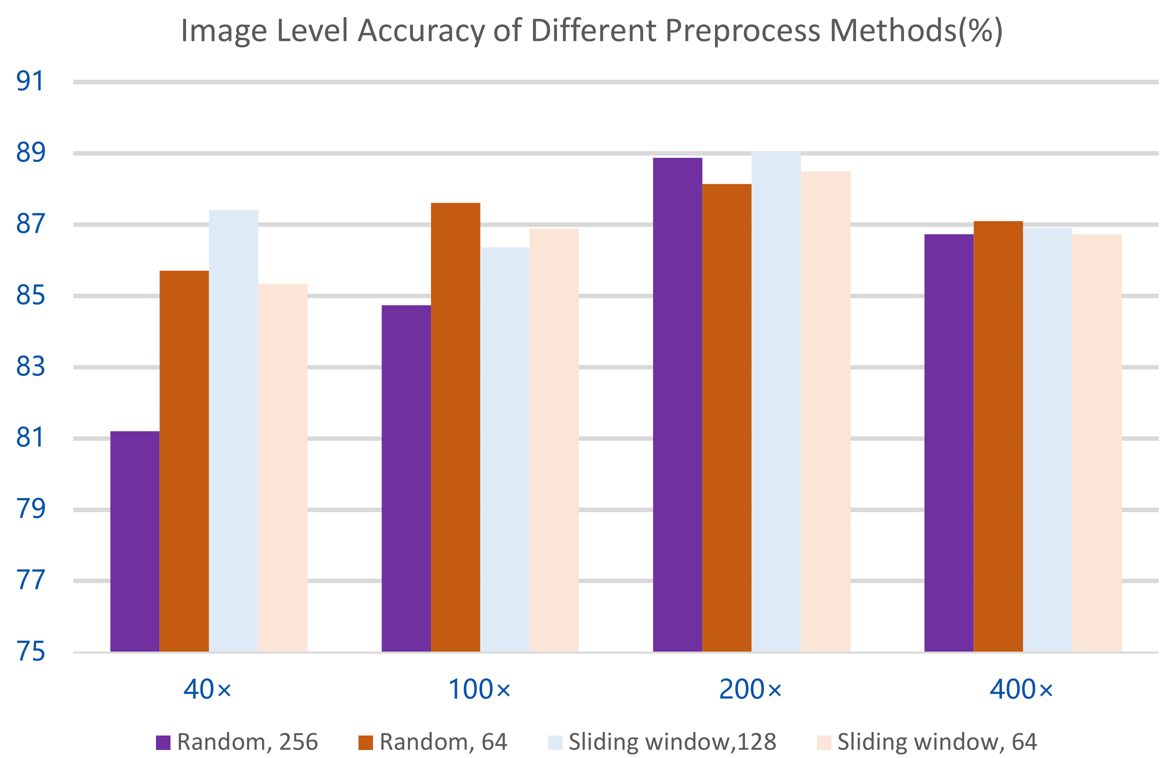


Sliding window crop



Structure of the model we used with input size 64×64

Results of different segmentation methods



- In general, ***sliding window crop with input size 128×128*** is the best preprocess method
- ***random*** segmentation method, which increases the variance of train dataset, is a little better than ***sliding window*** method.

Results analysis

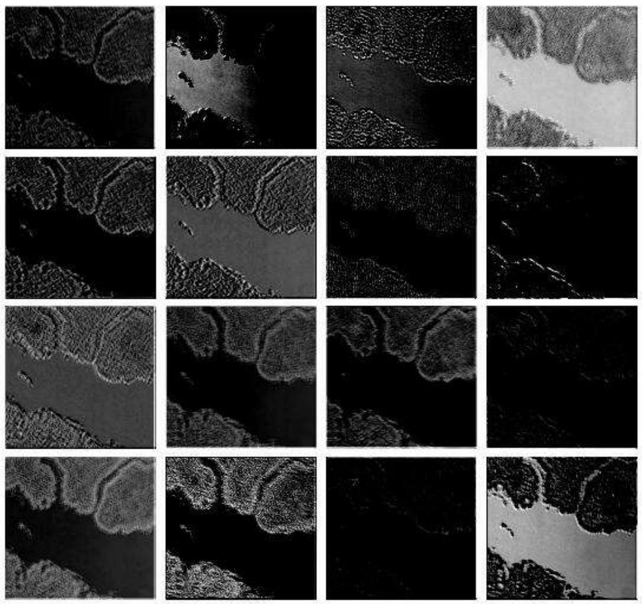
01 Results of different methods

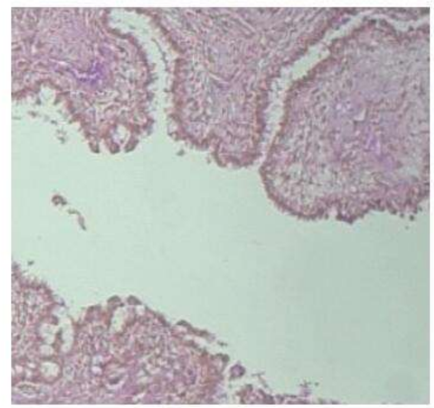
02 Results analysis

03 Comparison with past papers

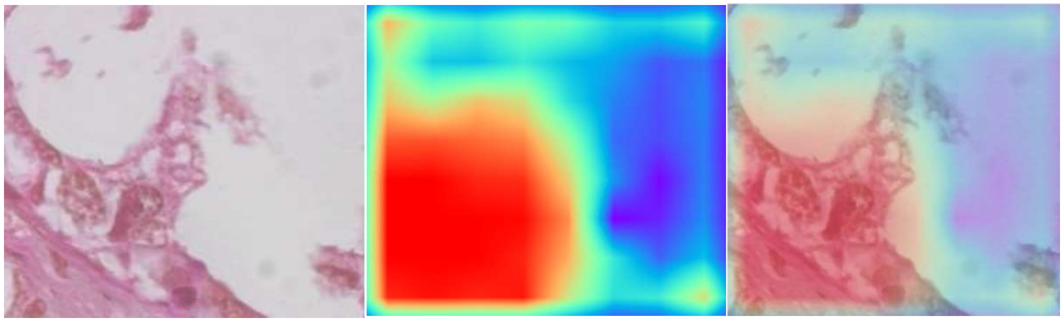
04 Limitations

Feature maps learned by first convolution layer



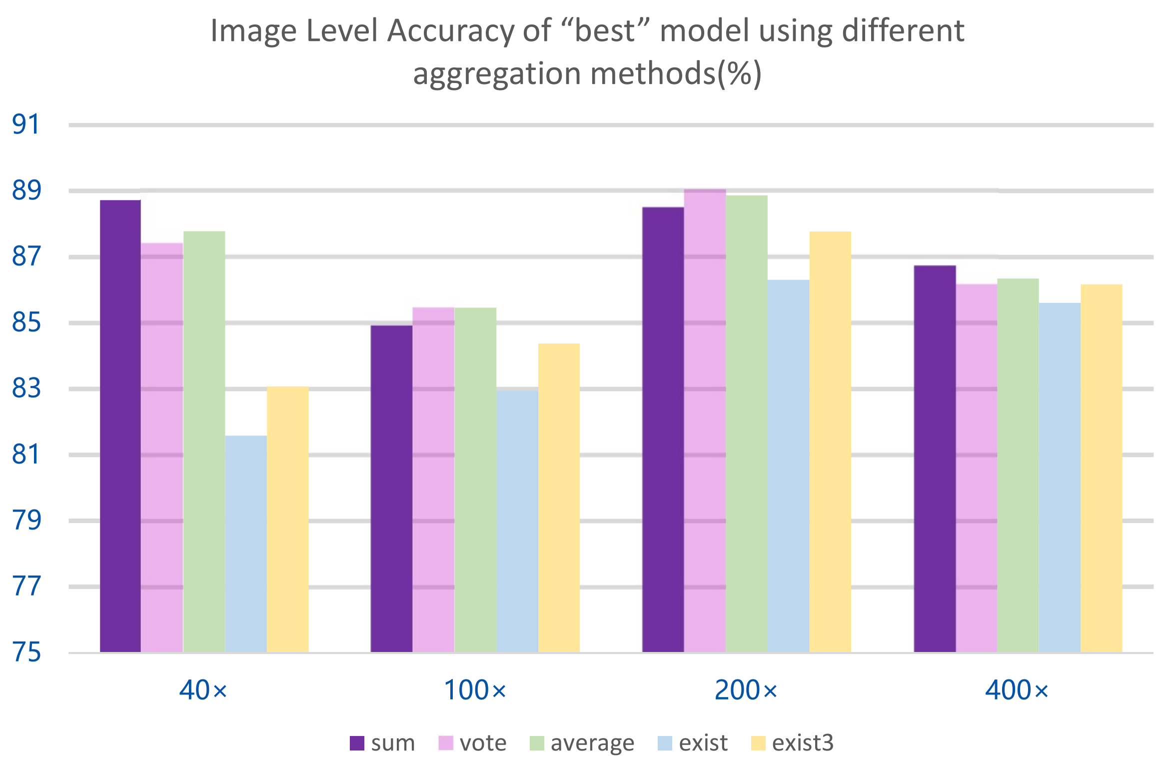


Localized prediction



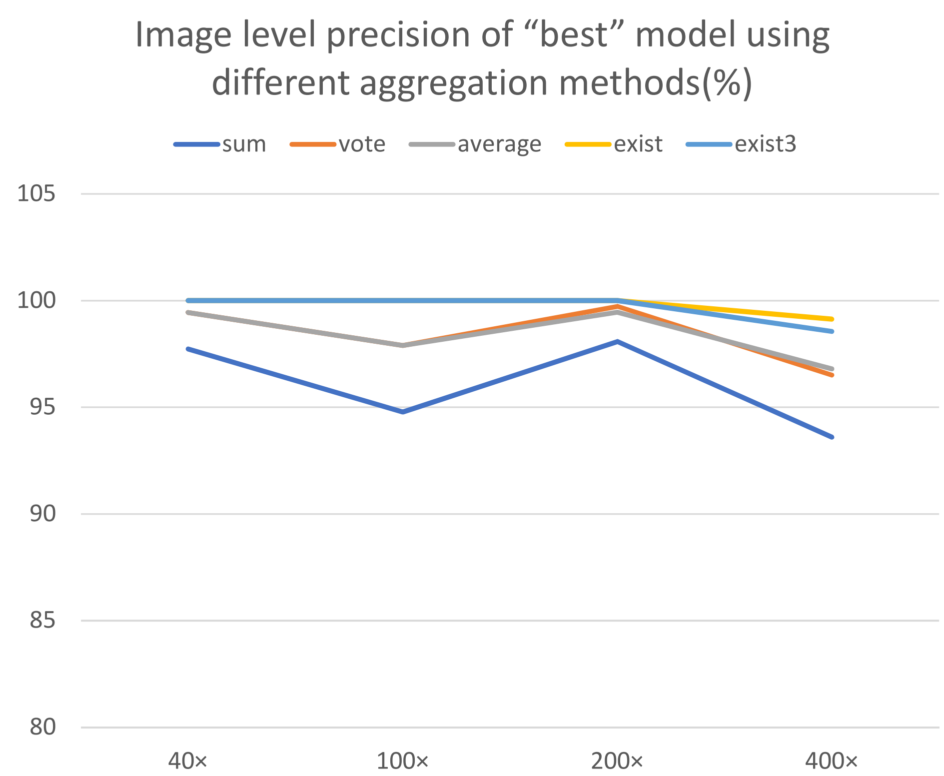
Red color means more likely, blue color means less likely.

Results of “best” model



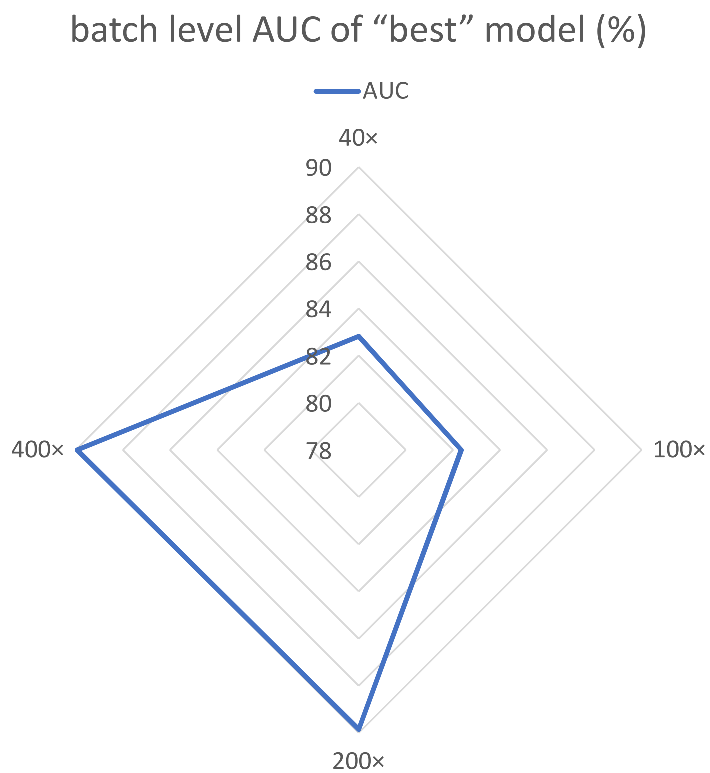
- Five aggregation methods we apply have slightly different influence on accuracy, in general, ***sum/vote/average*** *are* better than others.

Results of “best” model



- Our model achieves **high precision** on image level, which is very practical because almost all malignant patients can be predicted as malignant.

Results of “best” model



- Lower magnification results have a lower AUC value, which means that more batches are labeled with not solid predictions. (Prediction of probabilities are closer to [0.5,0.5]). Therefore, we can conclude that **the model learns less information of low magnification dataset.**

Comparison with past papers

Comparison with past papers

01 SVM

02 Traditional CNN 1

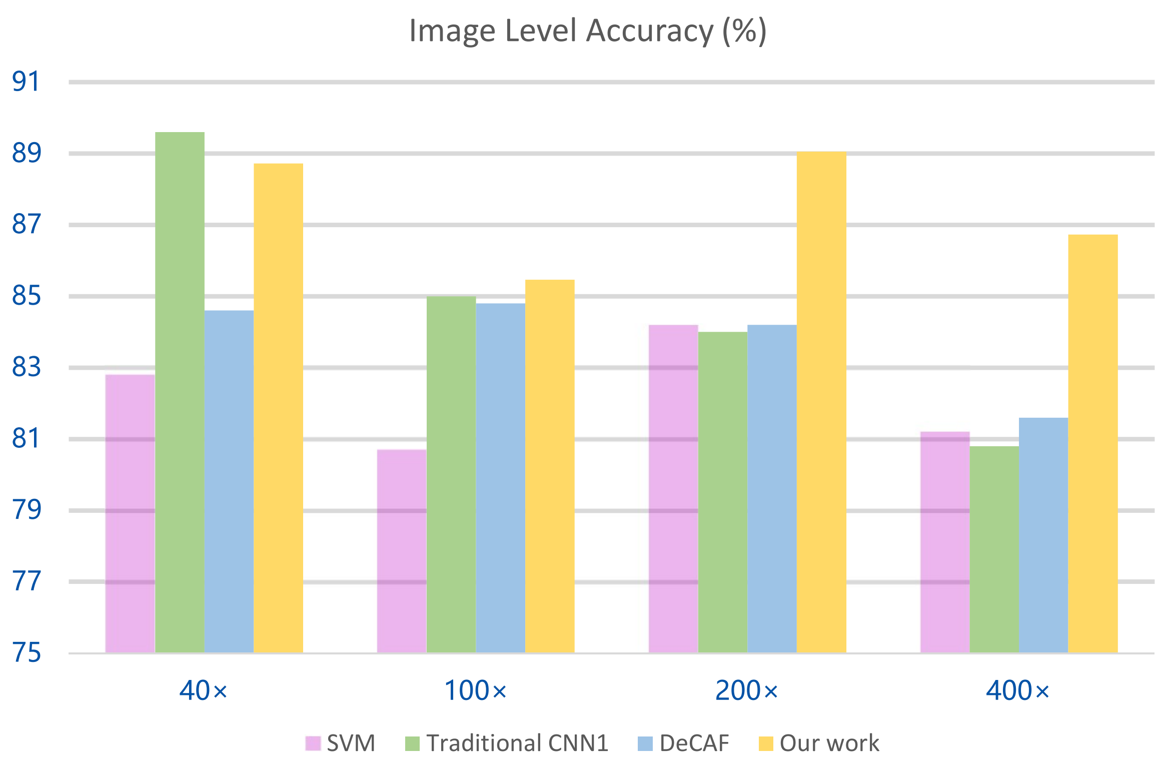
03 Traditional CNN 2

04 DeCAF

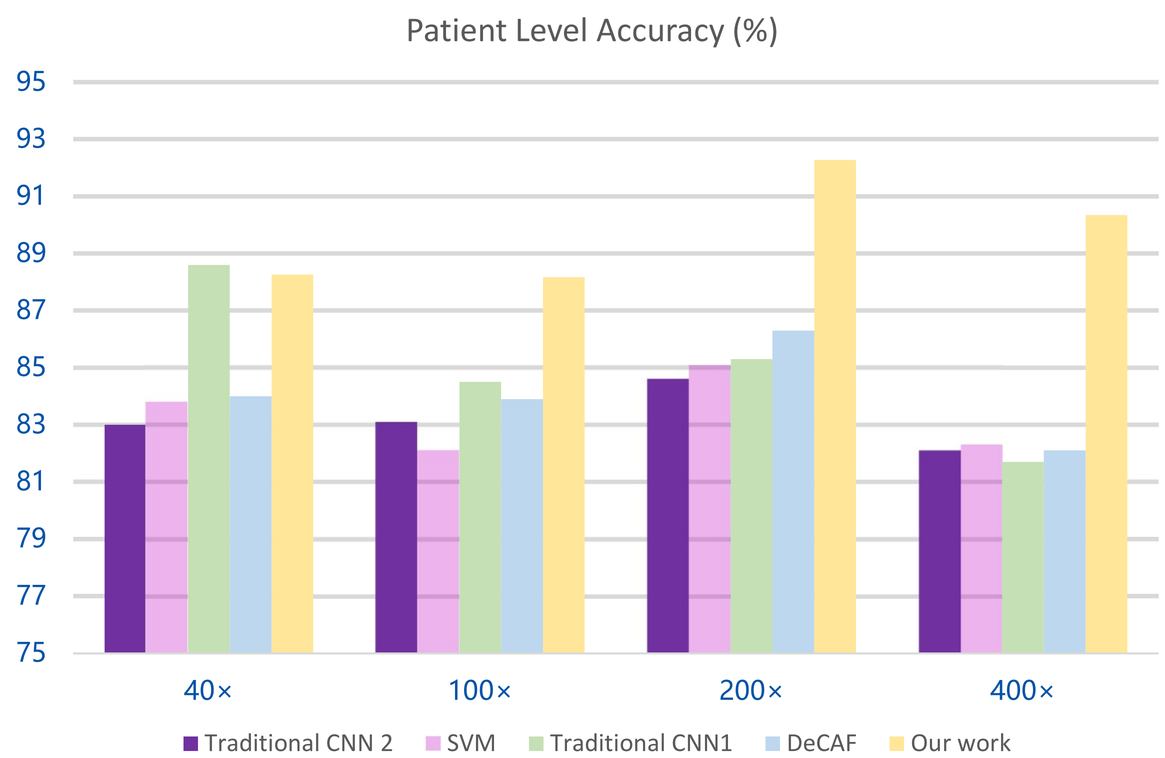
reuse a previously trained CNN only as feature vectors, which is then used as input for a classifier

Comparison with past papers

Comparison with past papers



Comparison with past papers

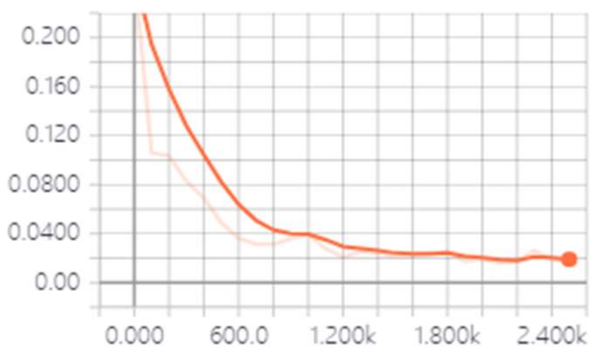


- Our work is **better** than other research using same dataset in **almost all of cases**
- The difference can be as large as **5%** in most cases.
- low magnification factors, such as 40× and 100×, has a **fewer information** and features for model to catch and learn

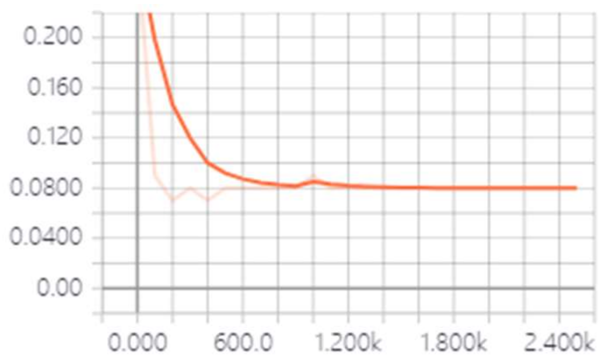
Limitations

01	Results of different methods
02	Results analysis
03	Comparison with past papers
04	Limitations

Overfitting



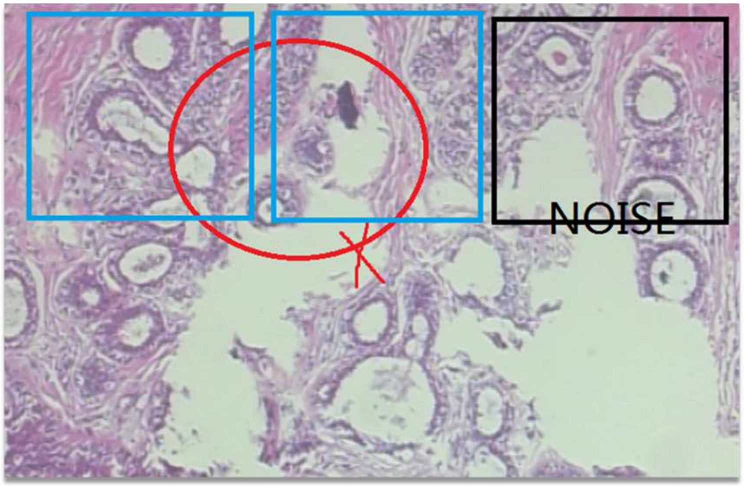
Train error



Validation error

- We have tried different technical to solve the problem, early stop, L2 regularization and dropout, none of them make a huge improvement
- The result can be better with the increase of model structure’s complexity
- We think the reason may be the **poor dataset**, the dataset we use contains only 82 patients

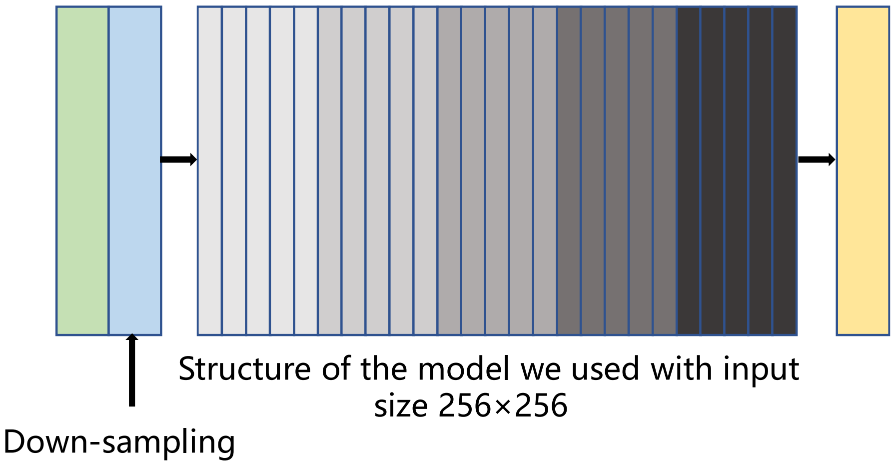
GPU memory limitation



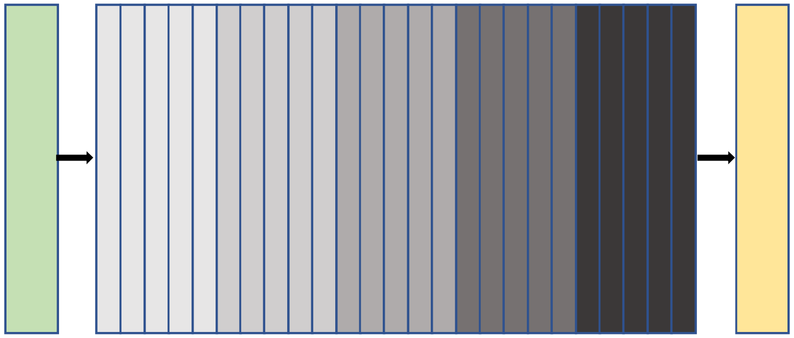
If red circle indicates a malignant tumor, then blue rectangle can be labeled as malignant correctly while black rectangle will become noise because there is no malignant tumor in it.

- ResNet consumes a high GPU memory
- And larger input size means a less possibility to generate noise input. Therefore we may need a larger input size, which also consumes a higher GPU memory

GPU memory limitation



Structure of the model we used with input size 256×256

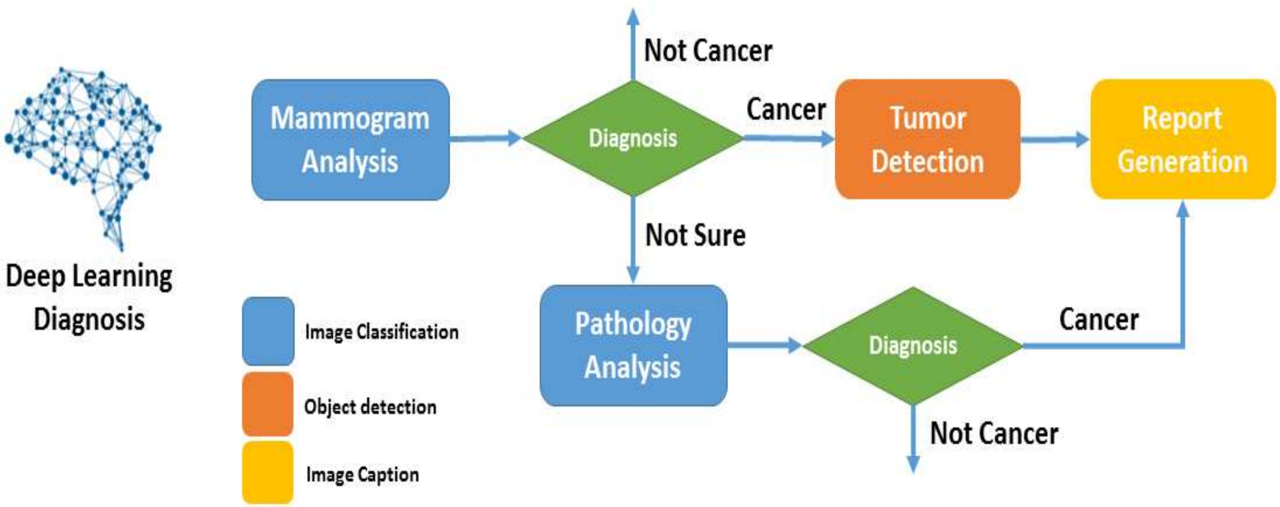


Structure of the model we used with input size 128×128

- But 128×128 is the maximal size to use a pure ResNet model, otherwise we need a down-sampling operation to reduce the input size of ResNet.
- In our current work, we uses pooling layer/ stride with 2 to do down-sampling, which causes a information loss definitely.

05. Future works

Future works



- Diagnosis using histopathological image
- Diagnosis using mammogram
- Tumor detection using mammogram
- Build a automated web-system to help breast cancer diagnosis

Thank you