國立清華大學
National Tsing Hua University

INNOVATION ENGINEERING LAB

Comparison of original data and GAN data for the improvement of cancer cell identification, to provide a complete and reliable inspection process system.

Dr. Ming-Chuan Chiu

Presenter: 江毓翔

2020/12/24

國立清華大學
National Tsing Hua University

INNOVATION ENGINEERING LAB

Outline

1. Introduction & Research question
2. Method
3. Experiment
4. Result
5. Contribution & Reflection

國立清華大學
National Tsing Hua University

INNOVATION ENGINEERING LAB

Outline

1. Introduction & Research question
2. Method
3. Experiment
4. Result
5. Contribution & Reflection

Introduction

- 5W1H：

What

- Training high recognition rate models.

When

- When we want to improve efficiency and reduce costs.

Who

- Hospital management.

Where

- Hospital.

Why

- Reduce the burden on doctors.

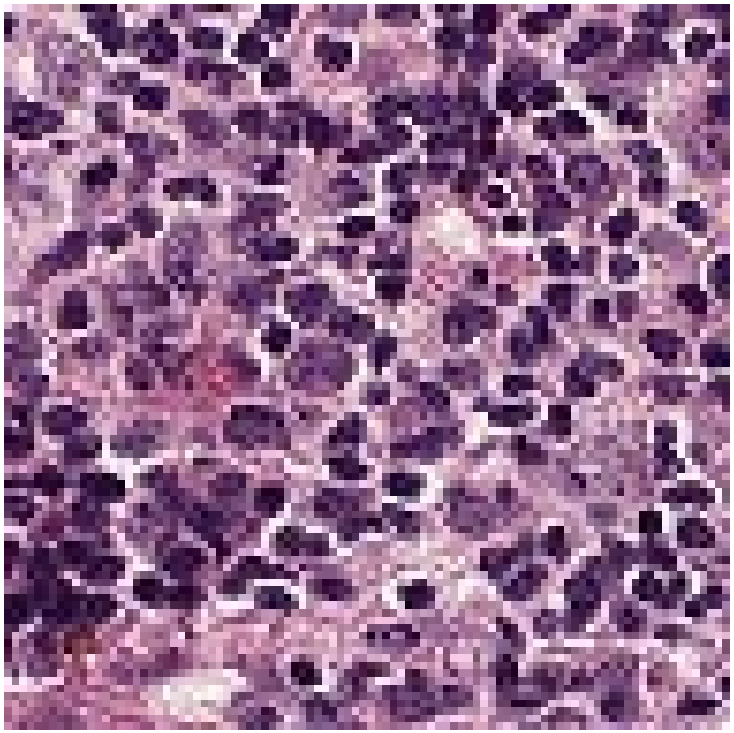
How

- Use deep learning and GAN to achieve the goal.

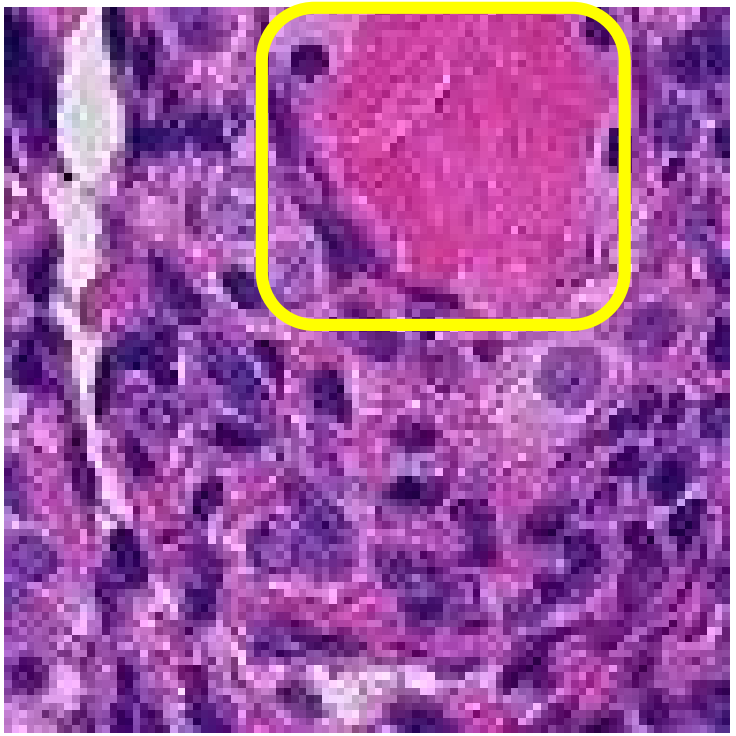
Introduction

- Data：

Normal



Cancer



Introduction

- Does this picture have cancer cells?

Introduction

- Goal：

- In view of the fact that the recognition rate of cell recognition using deep learning is low when there is no huge amount of data, and the cancer cells can only be identified by experts.
- Improve data information by GAN, and then enter the CNN network to increase the recognition rate, finally, propose a system architecture for cancer cell detection process

國立清華大學
National Tsing Hua University

INNOVATION ENGINEERING LAB

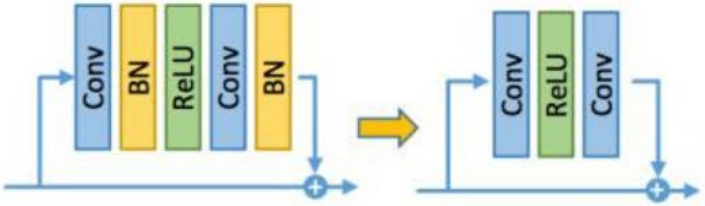
Outline

1. Introduction & Research question
2. Method
3. Experiment
4. Result
5. Contribution & Reflection

Method

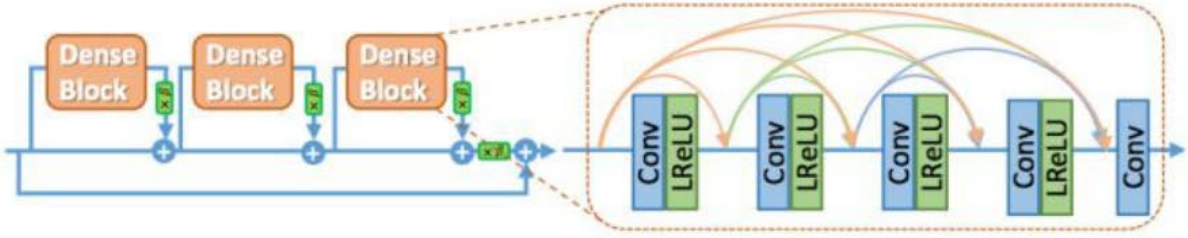
- ESRGAN：

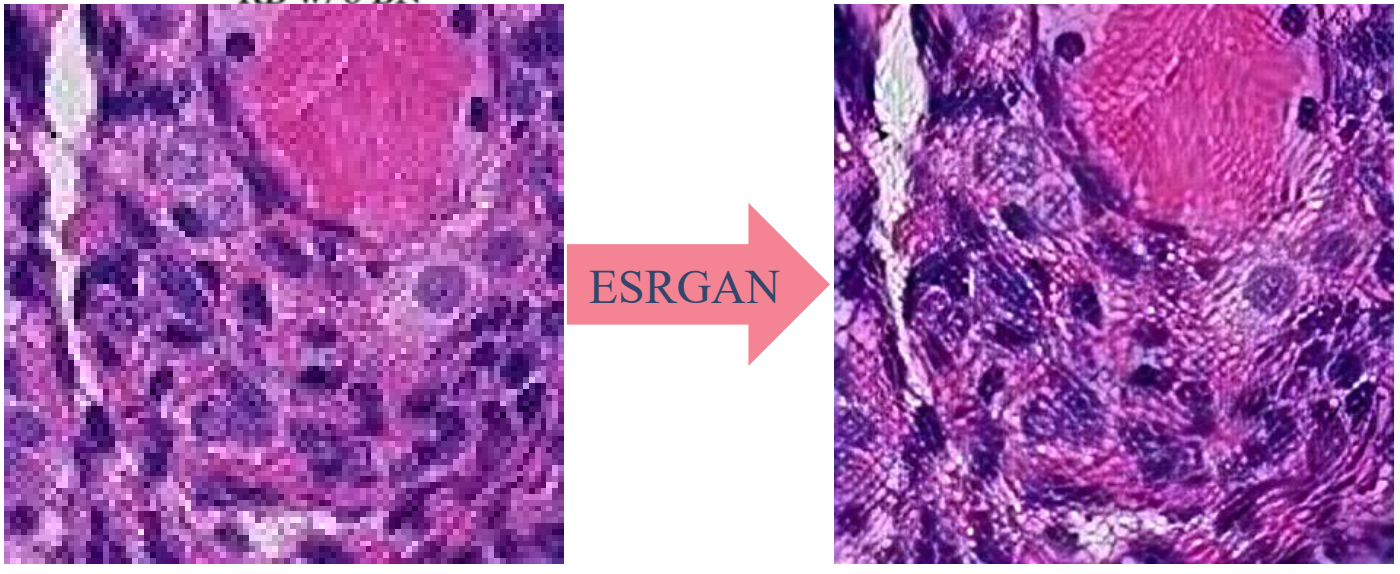
Residual Block (RB)



SRGAN　　RB w/o BN

Residual in Residual Dense Block (RRDB)





Method

- CNN：

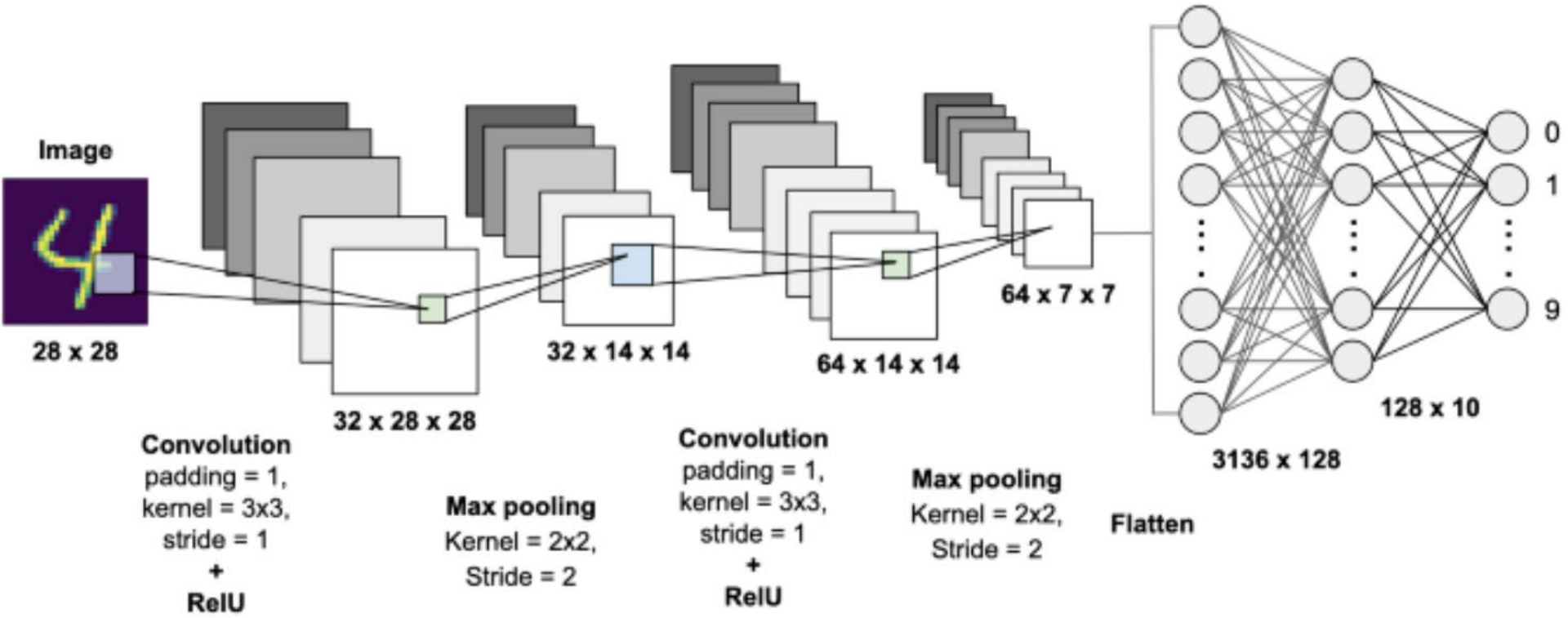


Image
28 x 28
32 x 28 x 28
Convolution
padding = 1,
kernel = 3x3,
stride = 1
+
ReIU
32 x 14 x 14
Max pooling
Kernel = 2x2,
Stride = 2
64 x 14 x 14
Convolution
padding = 1,
kernel = 3x3,
stride = 1
+
ReIU
64 x 7 x 7
Max pooling
Kernel = 2x2,
Stride = 2
3136 x 128
Flatten
128 x 10
0
1
9

國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Outline

1. Introduction & Research question
2. Method
3. Experiment
4. Result
5. Contribution & Reflection

Experiment

- Dataset：Histopathologic Cancer Detection
- Train：10000 normal cell, 10000 cancer cell
- Test：1250 normal cell, 1250 cancer cell

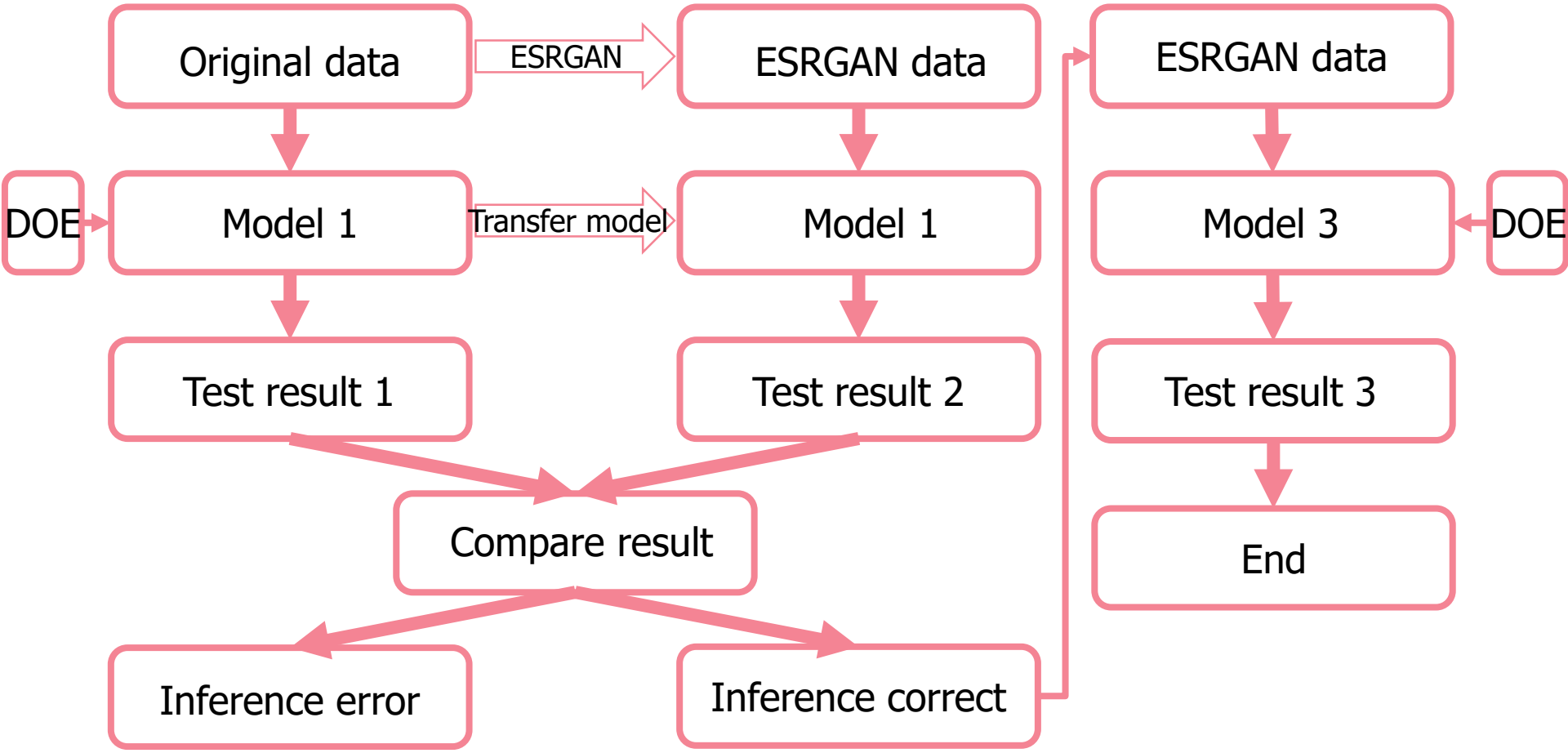
國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Experiment Architecture

	Model 1	Model 1	Model 3
Data	Original data	ESRGAN data	ESRGAN data
CNN	DOE 1	Transfer from Model 1	DOE 2
Result	Test result 1	Test result 2	Test result 3
Compare	vs	vs	

Experiment Architecture



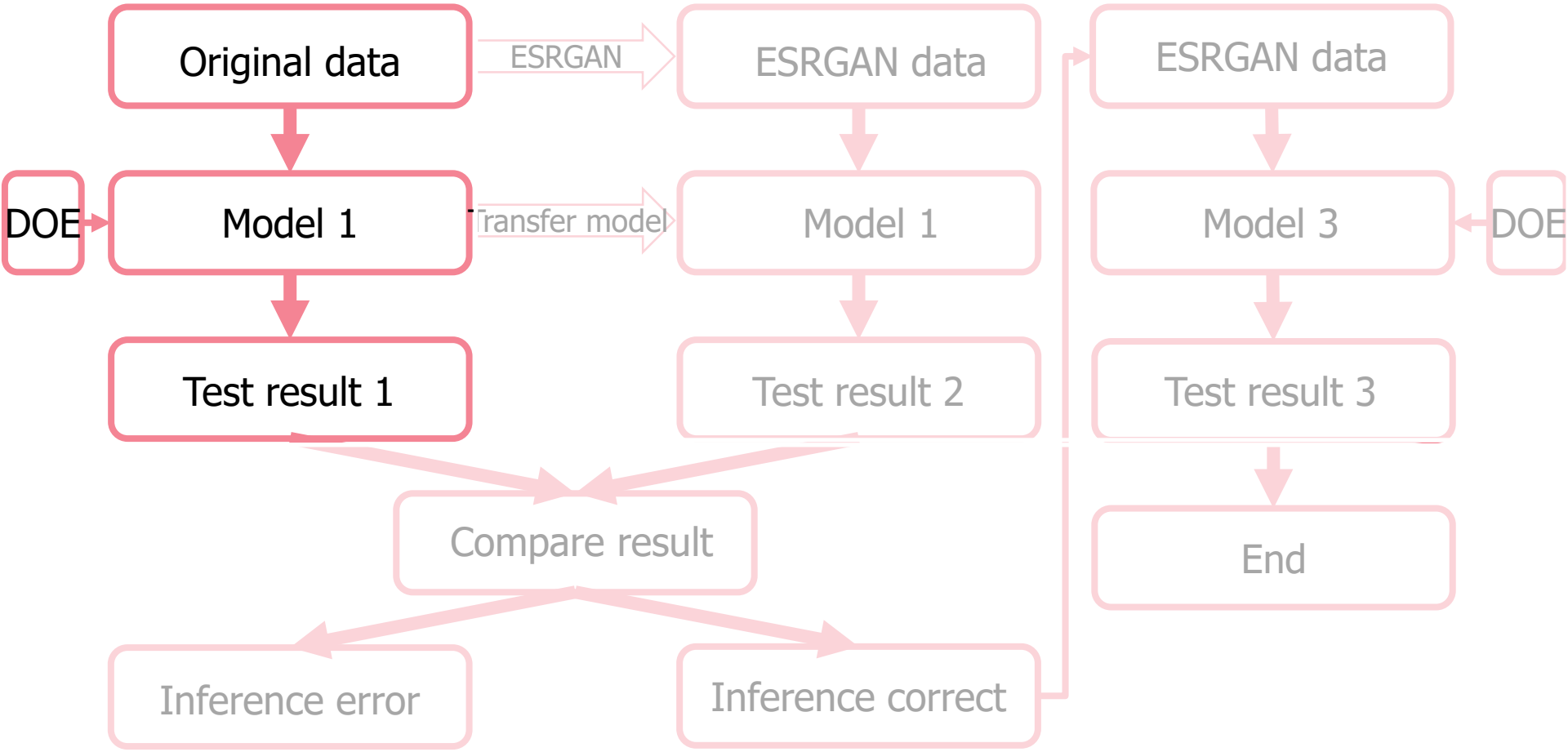
國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Outline

1. Introduction & Research question
2. Experiment
3. Method
4. Result
5. Contribution & Reflection

Step (1/5)


Original data
ESRGAN
ESRGAN data
ESRGAN data
DOE
Model 1
Transfer model
Model 1
Model 3
DOE
Test result 1
Test result 2
Test result 3
Compare result
End
Inference error
Inference correct

DOE in Model 1

- Model 1 DOE factor：

Factor \ Level	Level 1	Level 2	Level 3
Learning rate	0.0001	0.00015	0.0002
Optimizer	AdaDelta	Adam	Adagrad
Activate function	Tanh	ELU	ReLu
Dropout	0.4	0.5	0.6

- batch size 16, valid size 0.25

Chosen values from OA $L_9(3^4)$

Factor / Experiment	Learning rate	Optimizer	Activate function	Dropout	Result	Result
1	0.0001	AdaDelta	Tanh	0.4	0.7684	0.7453
2	0.0001	Adam	ELU	0.5	0.8102	0.8041
3	0.0001	Adagrad	ReLu	0.6	0.8076	0.7859
4	0.00015	AdaDelta	ELU	0.6	0.8253	0.8124
5	0.00015	Adam	ReLu	0.4	0.8484	0.8633
6	0.00015	Adagrad	Tanh	0.5	0.8247	0.8316
7	0.0002	AdaDelta	ReLu	0.5	0.8165	0.8268
8	0.0002	Adam	Tanh	0.6	0.7864	0.7944
9	0.0002	Adagrad	ELU	0.4	0.8145	0.7986

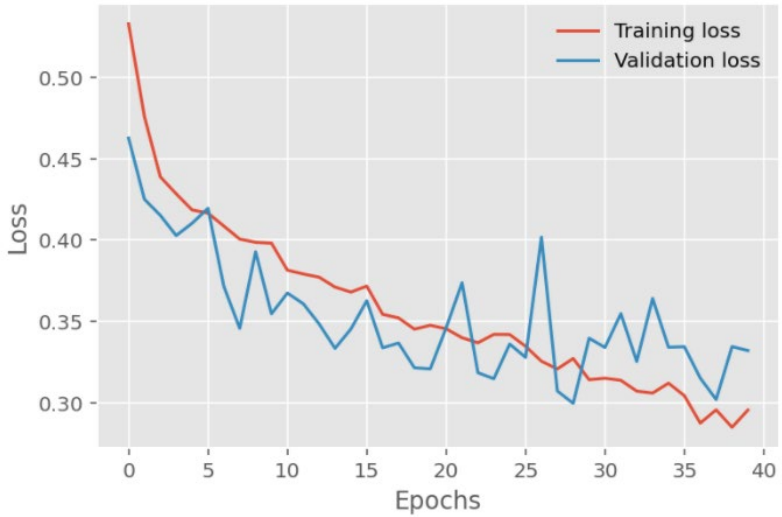
Summary

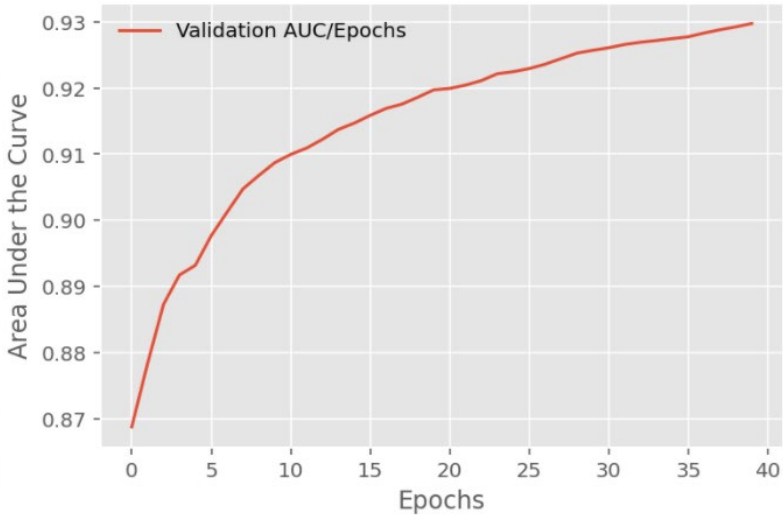
- Model 1 summary：

Summary	Kernel size	Stride	Padding	Activate function	Maxpool	Dropout
Layer 1	3x3	1	0	ReLu	2x2	
Layer 2	2x2	1	1	ReLu	2x2	
Layer 3	3x3	1	1	ReLu	2x2	
Layer 4	3x3	1	1	ReLu	2x2	
Layer 5	3x3	1	1	ReLu	2x2	
Fully C				ReLu+ Sigmoid		0.4 +0.4

Test Result 1

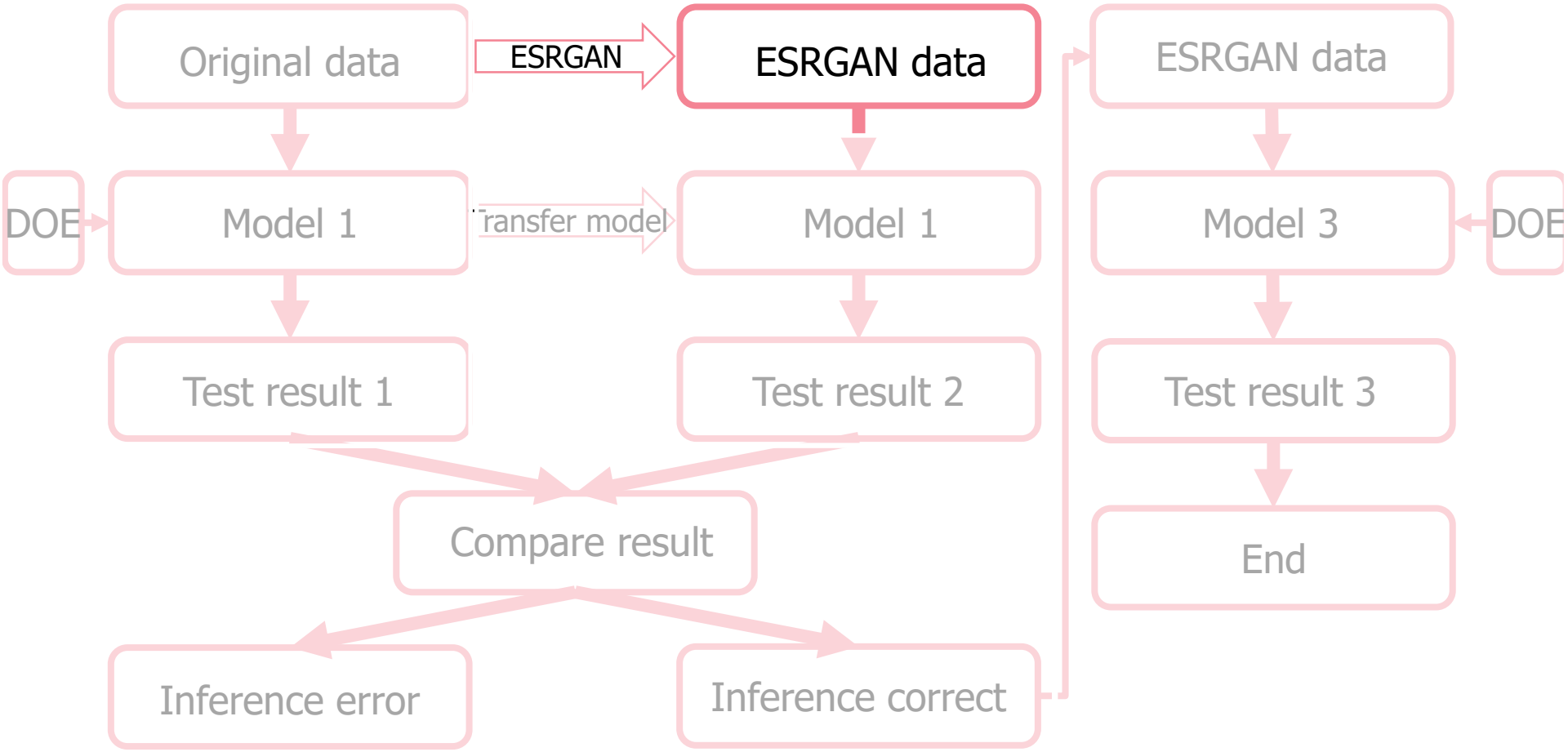
- Test result 1：





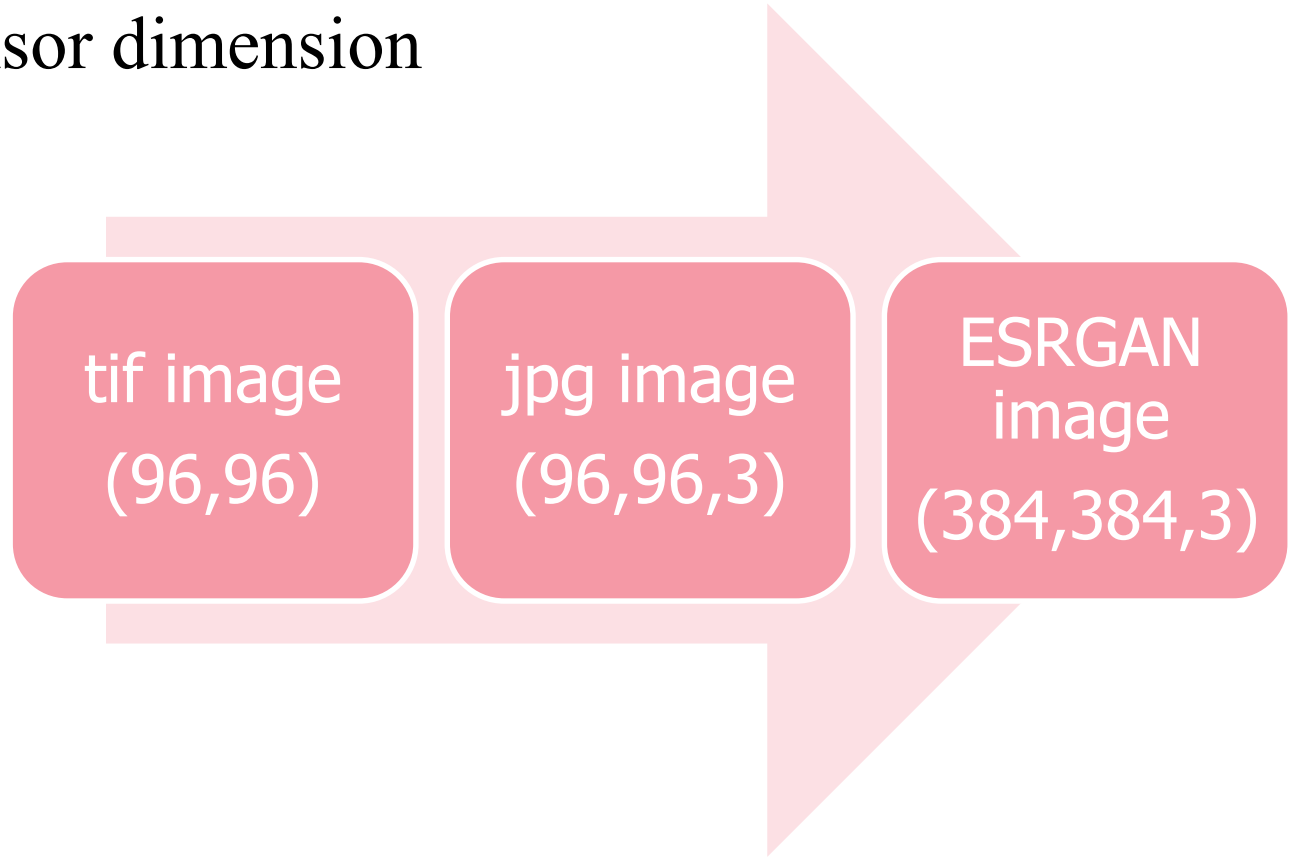
Accuracy : 0.8544
Precision : 0.8569
Recall : 0.8720
Specificity : 0.8544

Step (2/5)


Original data
ESRGAN
ESRGAN data
ESRGAN data
DOE
Model 1
Transfer model
Model 1
Model 3
DOE
Test result 1
Test result 2
Test result 3
Compare result
End
Inference error
Inference correct

ESRGAN

- Tensor dimension



Shape of tif to jpg

- ESRGAN pre-process：

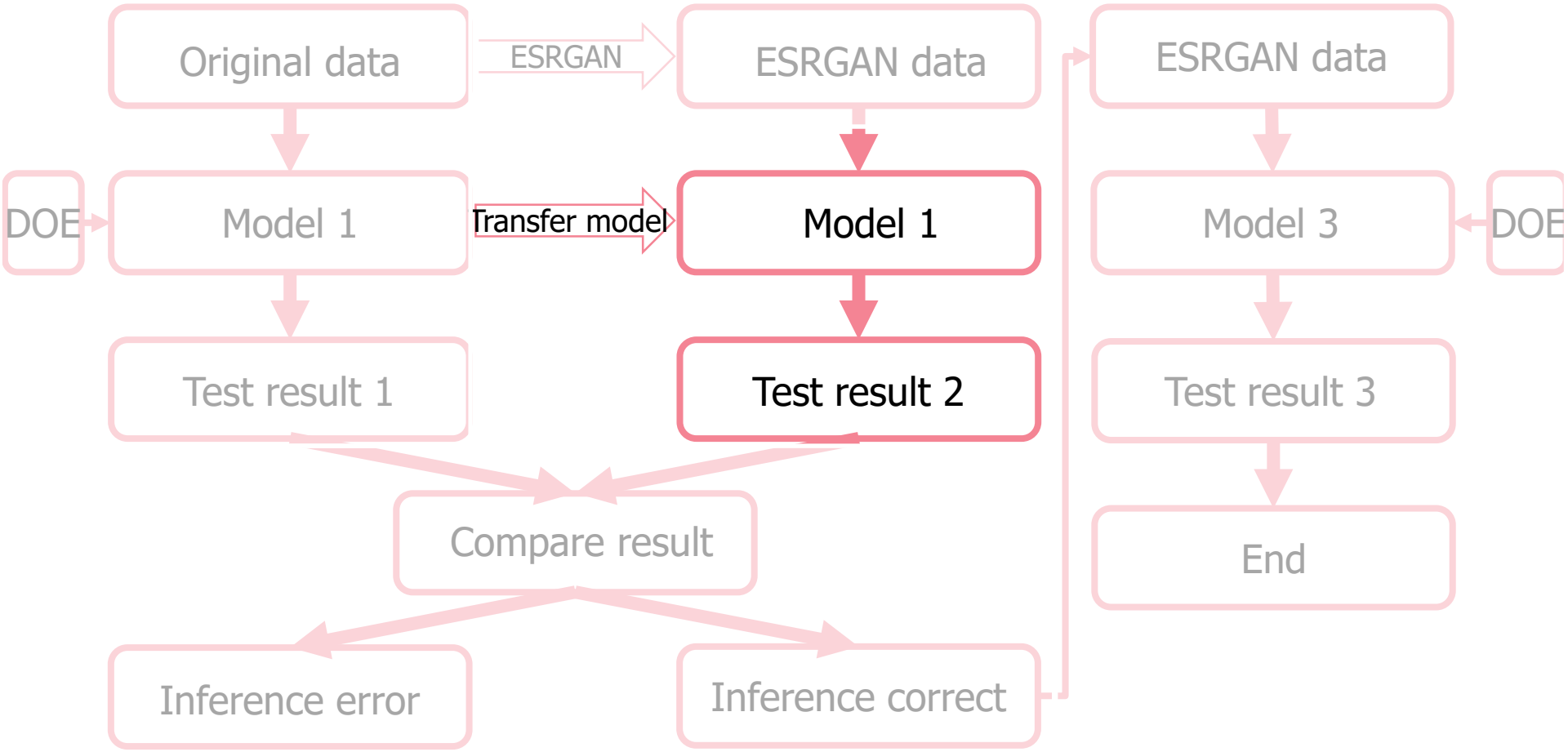
```
[[202 171 232 ...  91 193 230]
 [227 248 247 ... 249 208 238]
 [ 50  64 148 ...  74 112  94]
 ...
 [108 117 152 ... 112 136 106]
 [161 114 111 ... 156  61  50]
 [153 187 143 ... 123  26  78]]
```

tif.shape

```
[[[ 60  41  96]
  [ 71  50 107]
  [ 73  47 110]
  ...
  [166 111 179]
  [161 108 176]
  [145  94 160]]

 [[ 71  50 107]
  [ 70  49 108]
  [ 64  35 101]
  ...
  [166 109 178]
  [175 120 188]
  [170 118 184]]

 [[ 92  68 128]
  [ 84  60 122]
  [ 70  39 107]
  ...
  [162 101 171]
  [162 105 174]
  [158 103 170]]

 ...

 [[214 151 208]
  [221 158 215]
  [216 152 212]
  ...
  [211 141 193]
  [242 172 222]
  [228 158 208]]

 [[195 134 191]
  [210 149 206]
  [210 146 206]
  ...
  [187 117 169]
  [207 137 187]
  [199 130 177]]

 [[197 136 193]
  [217 156 213]
  [218 154 214]
  ...
  [147  77 127]
  [171 101 151]
  [184 115 162]]]
```

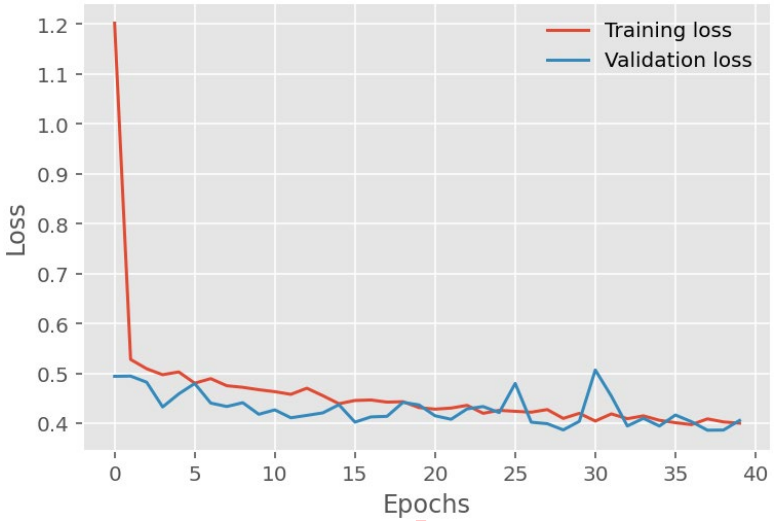
jpg.shape

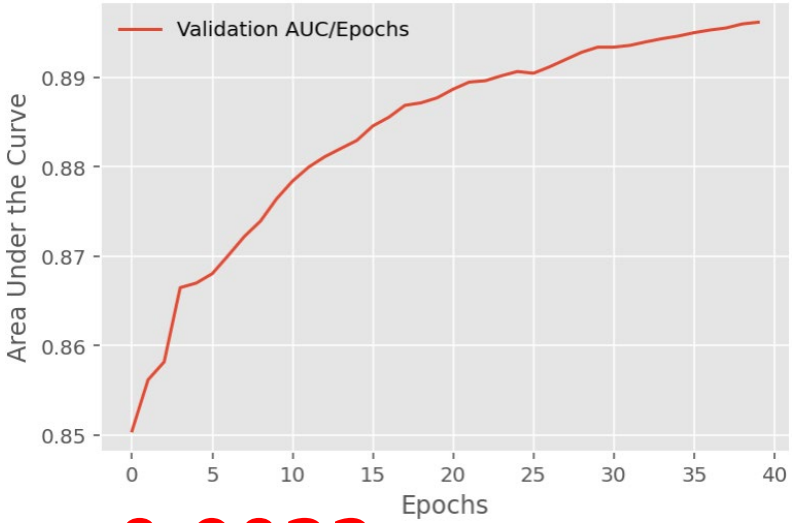
Step (3/5)


Original data
ESRGAN
ESRGAN data
ESRGAN data
DOE
Model 1
Transfer model
Model 1
Model 3
DOE
Test result 1
Test result 2
Test result 3
Compare result
End
Inference error
Inference correct

Result

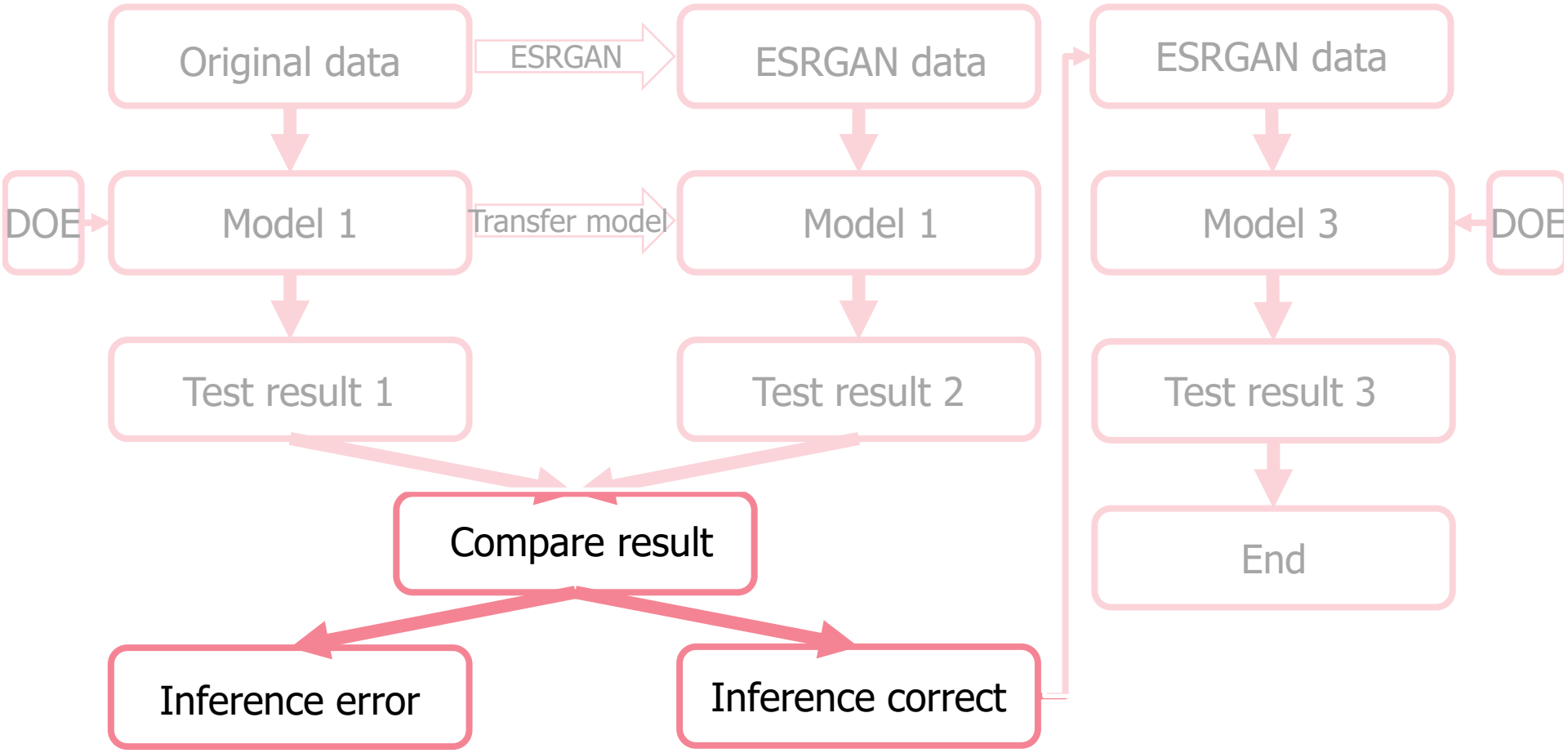
- Test result 2：





Accuracy : 0.9032
Precision : 0.9047
Recall : 0.9192
Specificity : 0.9032

Step(4/5)


Original data
ESRGAN
ESRGAN data
ESRGAN data
DOE
Model 1
Transfer model
Model 1
Model 3
DOE
Test result 1
Test result 2
Test result 3
Compare result
End
Inference error
Inference correct

Result

- Compare result：

Compare	Input data	Learning rate	Batch size	Valid size	Dropout	Result	Result
Model 1	Original data	0.00015	16	0.25	0.4	0.8484	0.8633
Model 1	ESRGAN data	0.00015	16	0.25	0.4	0.9052	0.9085

Original data 0.8559 → improve → ESRGAN data 0.9069

Inference correct

Problem of ESRGAN data in model 1

- Fully connected layer input：

Original data in Model 1

```
self.fc=nn.Sequential(
                nn.Linear(4608,1024),
                nn.ReLU(inplace=True),
                nn.Dropout(0.4),
                nn.Linear(1024,512),
                nn.Dropout(0.4),
                nn.Linear(512,   1),
                nn.Sigmoid())
```

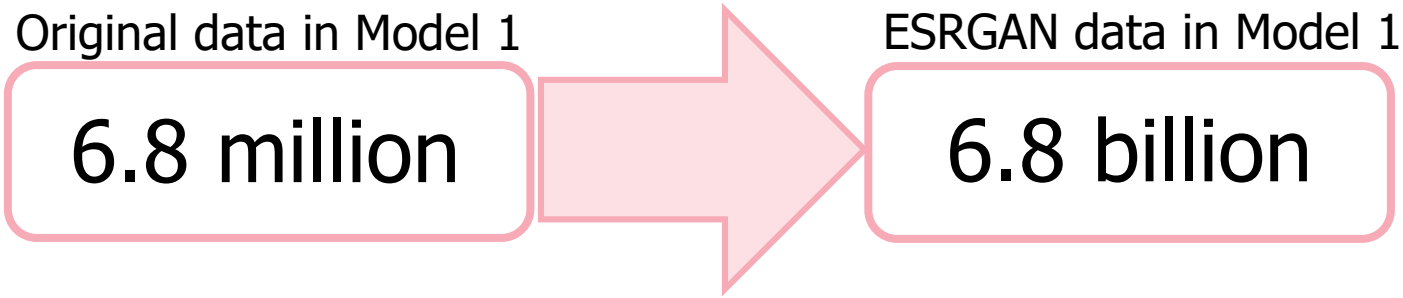
ESRGAN data in Model 1

```
self.fc=nn.Sequential(
                nn.Linear(73728,1024),
                nn.ReLU(inplace=True),
                nn.Dropout(0.4),
                nn.Linear(1024,512),
                nn.Dropout(0.4),
                nn.Linear(512,   1),
                nn.Sigmoid())
```

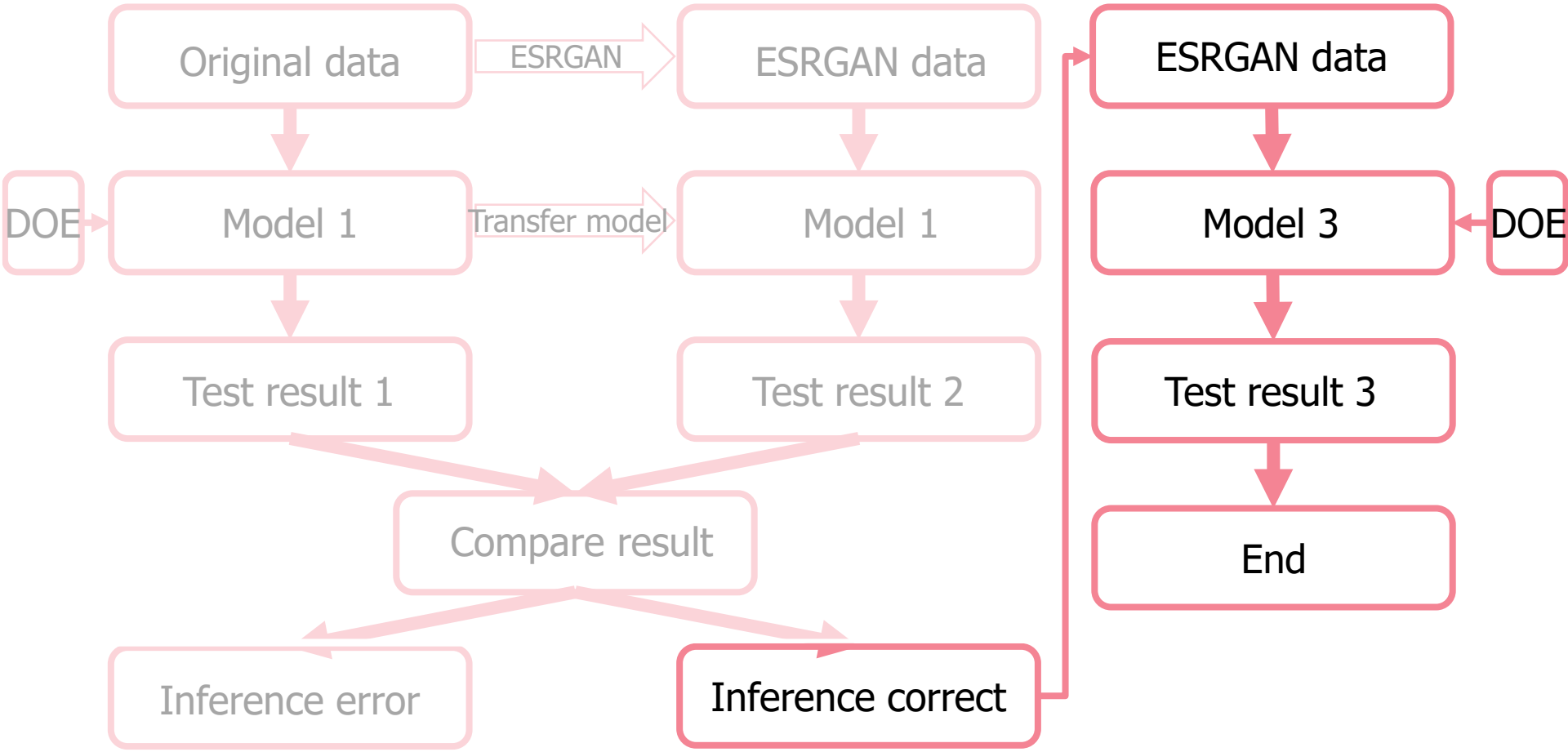
4608 → FC input explosion → 73728

Problem of ESRGAN data in model 1

- Estimated parameters explosion：



Step(5/5)


Original data
ESRGAN
ESRGAN data
ESRGAN data
DOE
Model 1
Transfer model
Model 1
Model 3
DOE
Test result 1
Test result 2
Test result 3
Compare result
End
Inference error
Inference correct

Deal with FC input explosion

- Increase layers

Original data in Model 1

```
self.fc=nn.Sequential(
                nn.Linear(4608,1024),
                nn.ReLU(inplace=True),
                nn.Dropout(0.4),
                nn.Linear(1024,512),
                nn.Dropout(0.4),
                nn.Linear(512,   1),
                nn.Sigmoid())
```

ESRGAN data in Model 1

```
self.fc=nn.Sequential(
                nn.Linear(73728,1024),
                nn.ReLU(inplace=True),
                nn.Dropout(0.4),
                nn.Linear(1024,512),
                nn.Dropout(0.4),
                nn.Linear(512    1),
                nn.Sigmoid())
```

ESRGAN data in Model 3

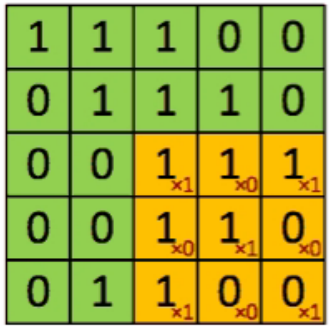
```
self.conv8=nn.Sequential(
                nn.Conv2d(in_channels=4096,   out_channels=8192,   kernel_size=4,   stride=1,   padding=1),
                nn.BatchNorm2d(8192),
                nn.ReLU(inplace=True),
                nn.Dropout(0.1),
                nn.MaxPool2d(2,2))

self.dropout2d   =   nn.Dropout2d()


self.fc=nn.Sequential(
                nn.Linear(8192,4096),
                nn.ReLU(inplace=True),
                nn.Dropout(0.2),
                nn.Linear(4096,2048),
                nn.Dropout(0.2),
                nn.Linear(2048,1024),
                nn.Dropout(0.2),
                nn.Linear(1024,512),
                nn.Dropout(0.2),
                nn.Linear(512,   1),
                nn.Sigmoid())
```

4608 → FC input explosion → 73728 → FC input decrease → 8192

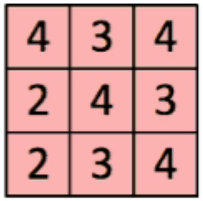
Deal with parameter explosion

- Adjustment kernel size

Image shape	Kernel size	Feature map shape
(15,15)	11x11 7x7 3x3	(5,5) (9,9) (13,13)



Image



Convolved Feature

DOE

- Model 3 DOE factor：

Factor \ Level	Level 1	Level 2	Level 3
Learning rate	0.0001	0.00015	0.0002
Optimizer	AdaDelta	Adam	Adagrad
Activate function	Tanh	ELU	ReLu
kernel size	3x3	7x7	11x11

- batch size 8, valid size 0.1

Chosen values from OA $L_9(3^4)$

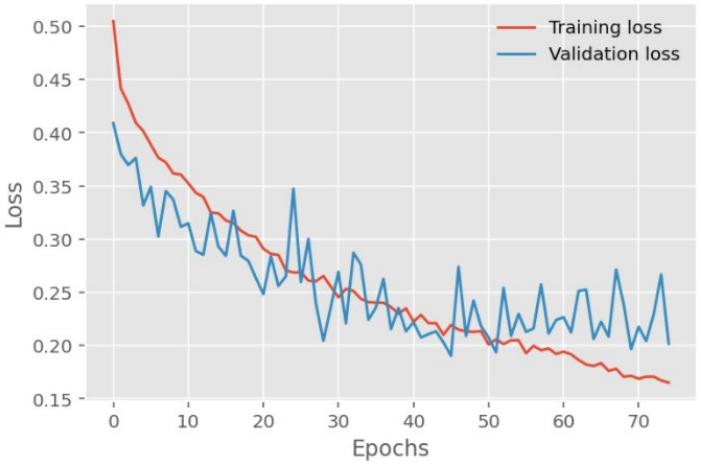
Factor / Experiment	Learning rate	Optimizer	Activate function	Kernel size	Result	Result
1	0.00015	Adam	ReLu	3x3	0.9523	0.9612
2	0.00015	AdaDelta	ELU	7x7	0.8534	0.8324
3	0.00015	Adagrad	Tanh	11x11	0.8164	0.8064
4	0.0002	Adam	ELU	11x11	0.8764	0.8643
5	0.0002	AdaDelta	Tanh	3x3	0.7834	0.7954
6	0.0002	Adagrad	ReLu	7x7	0.8095	0.8134
7	0.00025	Adam	Tanh	7x7	0.7985	0.8135
8	0.00025	AdaDelta	ReLu	11x11	0.8634	0.8587
9	0.00025	Adagrad	ELU	3x3	0.8065	0.8234

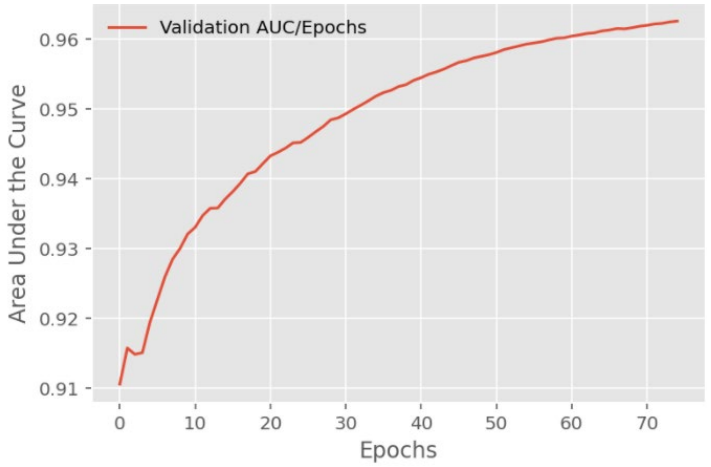
Model 3 summary

Summary	Kernel size	Stride	Padding	Activate function	Maxpool	Dropout
Layer 1	3x3	1	0	ReLu	2x2	
Layer 2	2x2	1	1	ReLu	2x2	
Layer 3	3x3	1	1	ReLu	2x2	
Layer 4	3x3	1	1	ReLu	2x2	
Layer 5	3x3	1	1	ReLu	2x2	
Layer 6	3x3	1	1	ReLu	2x2	0.1
Layer 7	3x3	1	1	ReLu	2x2	0.1
Layer 8	4x4	1	1	ReLu	2x2	0.1
Fully C				ReLu+ Sigmoid		0.2+0.2+ 0.2+0.2

Result

- Test result 3：



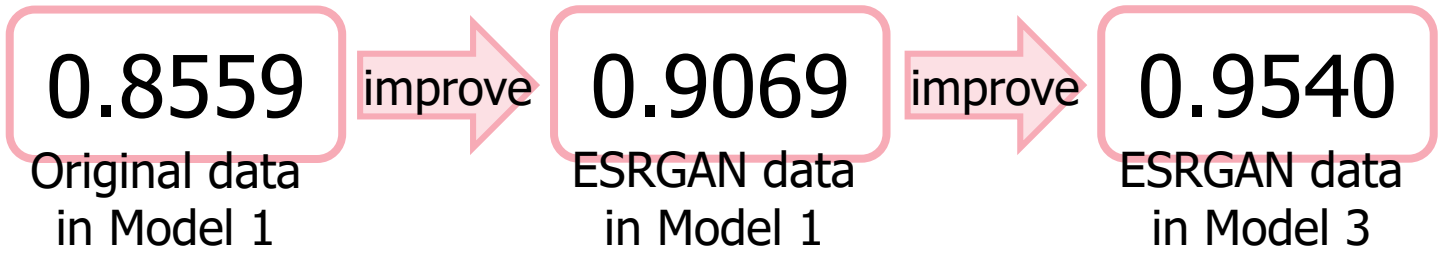


Accuracy : 0.9456
Precision : 0.9464
Recall : 0.9624
Specificity : 0.9456

Result

- Compare result：

Compare	Input data	Learning rate	Batch size	Valid size	Dropout	Result	Result
Model 1	Original data	0.00015	16	0.25	0.4	0.8484	0.8633
Model 1	ESRGAN data	0.00015	16	0.25	0.4	0.9052	0.9085
Model 3	ESRGAN data	0.00015	8	0.1	0.1, 0.2	0.9588	0.9492



國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Outline

1. Introduction & Research question
2. Experiment
3. Method
4. Result
5. Contribution & Reflection

Contribution

- Provide a ESRGAN+CNN system architecture, such a system can be provided to doctors for reverification judgment.

- This topic proposes to solve the problem of parameter explosion when ESRGAN data imported in CNN models, and the input explosion of fully connected layers.

Future direction

- The combination of AI model and microscope can also improve the accuracy of doctors in judging cancer.

- The proposed system architecture can be imported into other small and difficult pictures problems, such as wafer inspection.

國立清華大學
National Tsing Hua University

INNOVATION
ENGINEERING
LAB

Thank You for Your Listening