國立清華大學
National Tsing Hua University

INNOVATION ENGINEERING LAB

IIE Final Project

Integrate CNN and Data Augmentation on Wafer Map Defect Patterns Classification

陳道明

2020/01/02

國立清華大學
National Tsing Hua University

INNOVATION ENGINEERING LAB

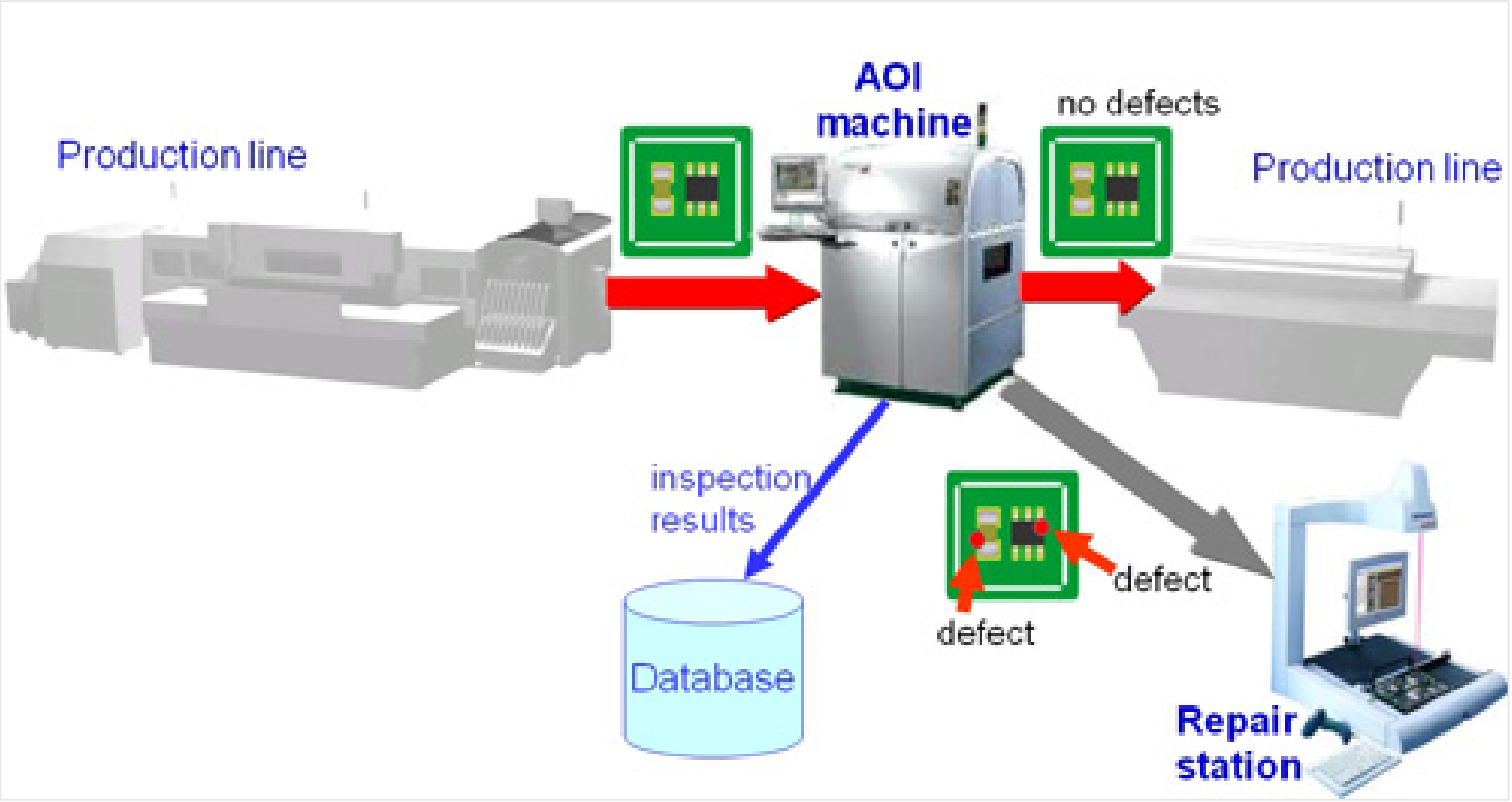
Outline

1. Introduction
2. Method
3. Result
4. Future Works

Outline

1. Introduction
2. Method
3. Result
4. Future Works

Introduction (1/3)



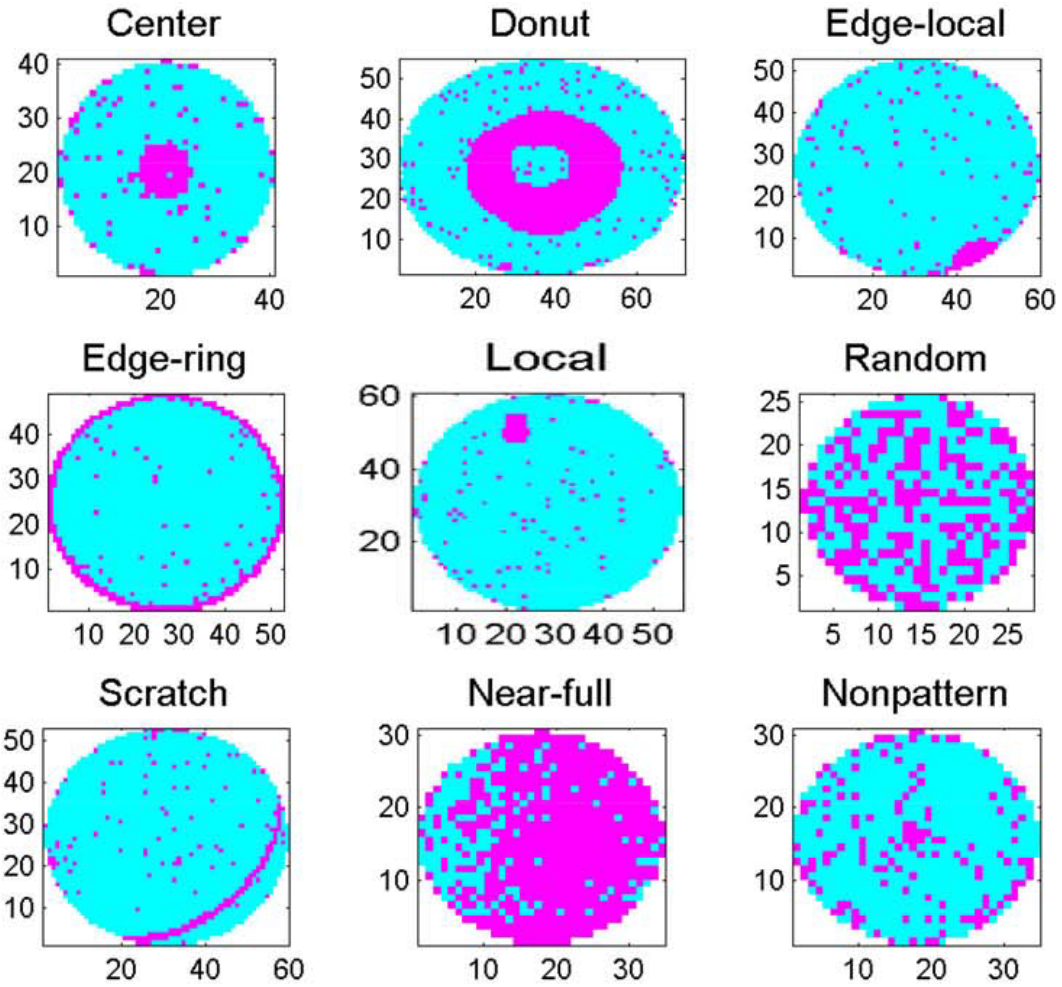
https://www.miselectronics.com/PCB-AOI-Testing.html

Introduction (2/3)

- **WM-811K** wafer map dataset
  - real data from TSMC
  - 811,457 wafer maps from 46,293 lots
  - only about 20% were labeled (172,951 / 811,457)

- **9 types** (8 defect types + 1 non defect type)

Introduction (3/3)



Outline

1. Introduction
2. Method
3. Result
4. Future Works

國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Method

- Package used:
  - Numpy
  - Pandas
  - Tensorflow 1.15
  - Keras
  - Scikit-learn
  - Matplotlib

國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Method

Data Pre-processing

- read data
- shape
- one-hot encoder

Data Augmentation

- image auto-encoder
- generate data

CNN

- build model
- cross validate

國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Method

Data Pre-processing

- read data
- shape
- one-hot encoder

Data Augmentation

- image auto-encoder
- generate data

CNN

- build model
- cross validate

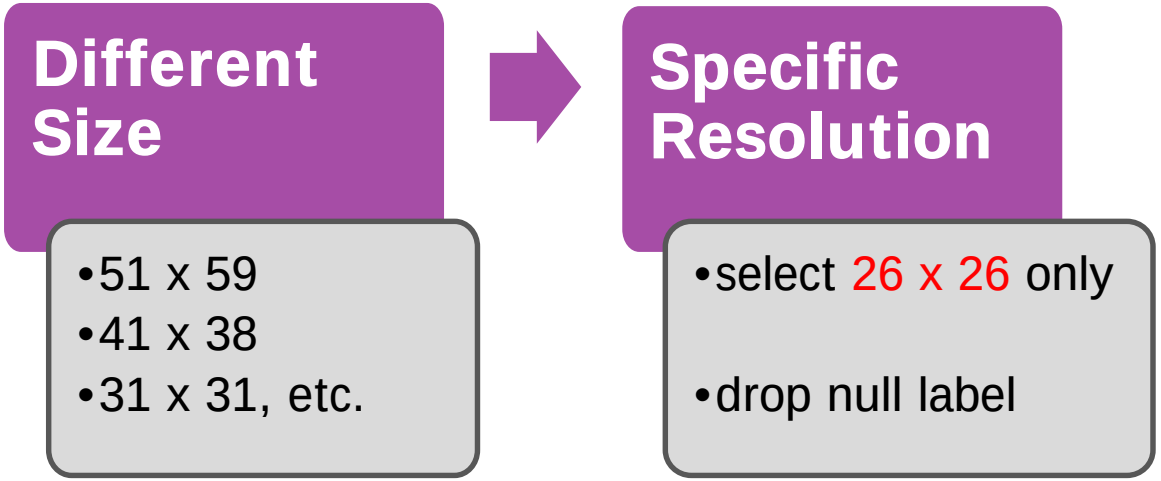
Data Preprocessing (1/3)

- Load data:

Name	Size	Type
waferMap	811,457 Non-null	object
dieSize	811,457 Non-null	float64
lotName	811,457 Non-null	object
waferIndex	811,457 Non-null	float64
trainTestLabel	811,457 Non-null	object
failureType	811,457 Non-null	object

Data Preprocessing (2/3)

- Shape:



→ **14,366** sample remain

Data Preprocessing (3/3)

- One-hot encoder (add channel):

Wafer type of the pixel	Categorical variable
Not wafer	0
Normal	1
faulty	2

Wafer type of the pixel	Not wafer	Normal	Faulty
Not wafer	1	0	0
Normal	0	1	0
faulty	0	0	1

(14366, 26, 26, **1**)

(14366, 26, 26, **3**)

國立清華大學
National Tsing Hua University

INNOVATION ENGINEERING LAB

Method

Data Pre-processing

- read data
- shape
- one-hot encoder

Data Augmentation

- image auto-encoder
- generate data

CNN

- build model
- cross validate

Data Augmentation (1/4)

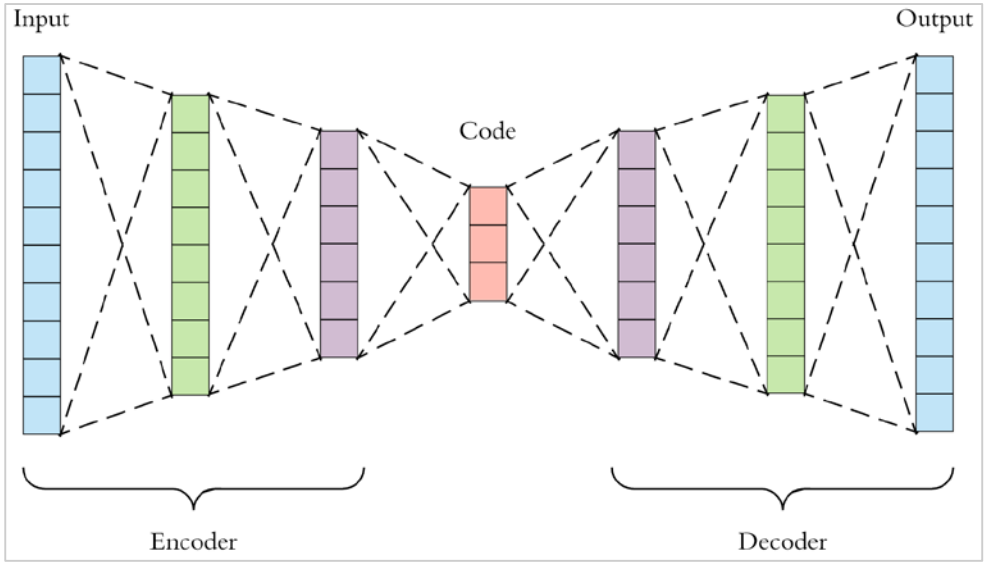
- Why data augmentation?

  → **imbalanced classes**

Faulty type	Number
Center	99
Donut	1
Edge-Loc	296
Edge-Ring	31
Loc	297
Near-full	16
Random	74
Scratch	72
none	13489

Data Augmentation (2/4)

- **Convolutional Auto-encoder** for Data Augmentation
  - Encoder → Decoder
  - The input is its output.
  - **Learn how to reconstruct the input-data**.



Data Augmentation (3/4)

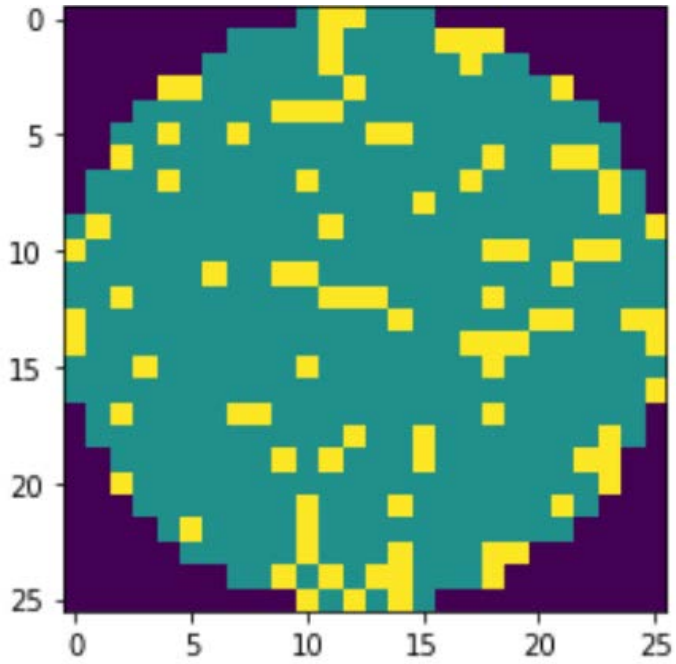
- **Convolutional Auto-encoder** for Data Augmentation

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 26, 26, 3)]	0
conv2d (Conv2D)	(None, 26, 26, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 13, 13, 64)	0
conv2d_transpose (Conv2DTran	(None, 13, 13, 64)	36928
up_sampling2d (UpSampling2D)	(None, 26, 26, 64)	0
conv2d_transpose_1 (Conv2DTr	(None, 26, 26, 3)	1731

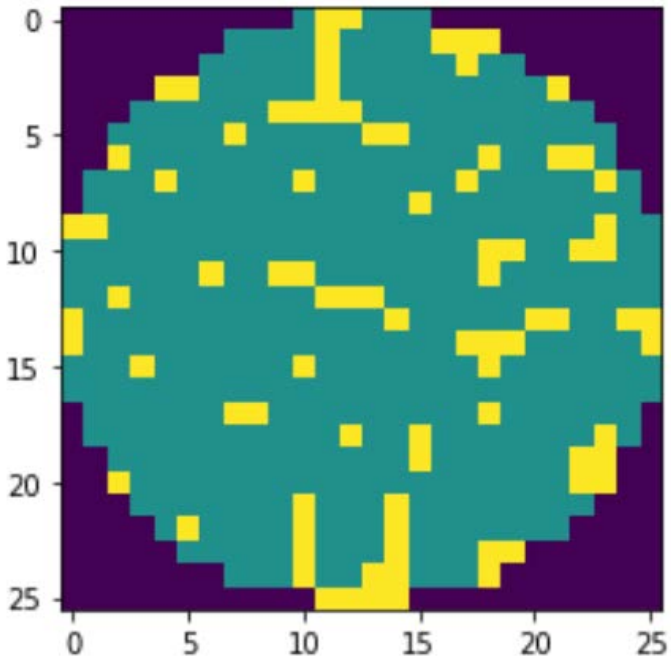
Total params: 40,451
Trainable params: 40,451
Non-trainable params: 0

Data Augmentation (4/4)

- Generated data = encoded data + random noise.



Original



Reconstructed

國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Method

Data Pre-processing

- read data
- shape
- one-hot encoder

Data Augmentation

- image auto-encoder
- generate data

CNN

- build model
- cross validate

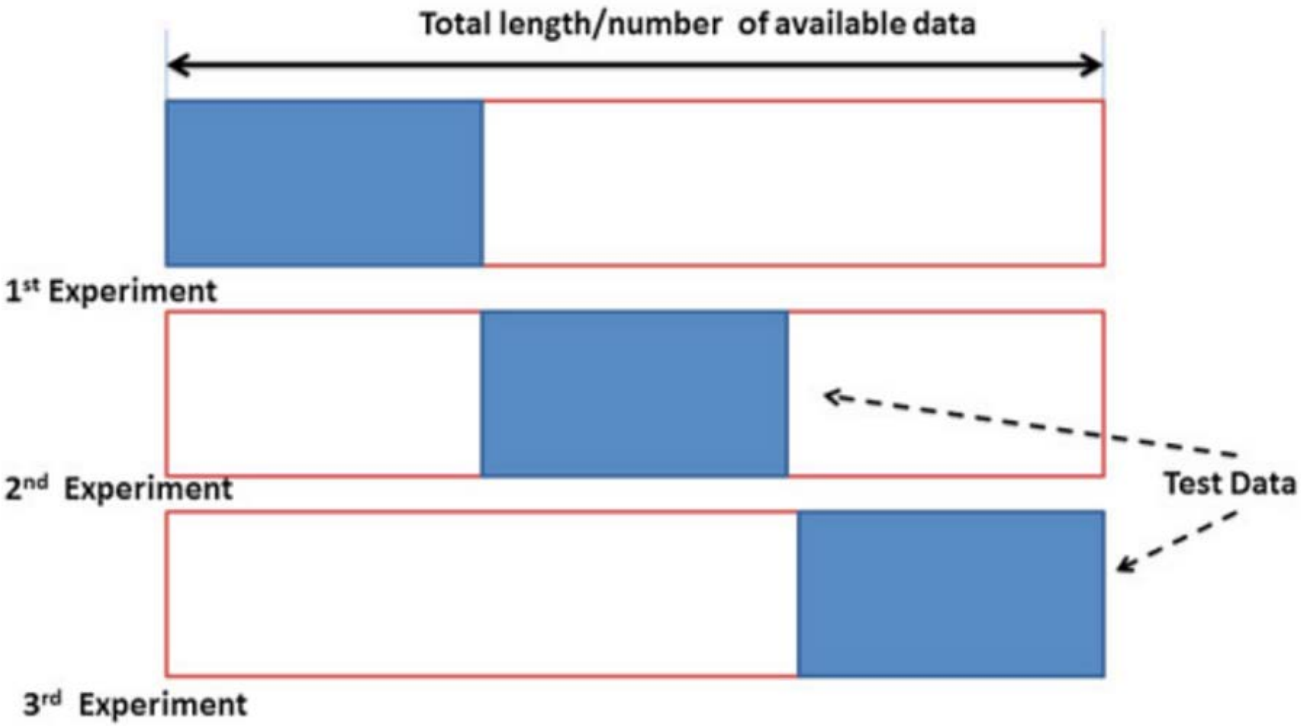
CNN (1/2)

- Simple 2D CNN

```
input_shape = (26, 26, 3)
input_tensor = Input(input_shape)

conv_1 = layers.Conv2D(16, (3,3), activation='relu', padding='same')(input_tensor)
conv_2 = layers.Conv2D(64, (3,3), activation='relu', padding='same')(conv_1)
conv_3 = layers.Conv2D(128, (3,3), activation='relu', padding='same')(conv_2)

flat = layers.Flatten()(conv_3)

dense_1 = layers.Dense(512, activation='relu')(flat)
dense_2 = layers.Dense(128, activation='relu')(dense_1)
output_tensor = layers.Dense(9, activation='softmax')(dense_2)

model = models.Model(input_tensor, output_tensor)
model.compile(optimizer='Adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

CNN (2/2)

- Cross Validation



https://stats.stackexchange.com/questions/51416/k-fold-vs-monte-carlo-cross-validation

國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Outline

1. Introduction
2. Method
3. Result
4. Future Works

Result (1/2)

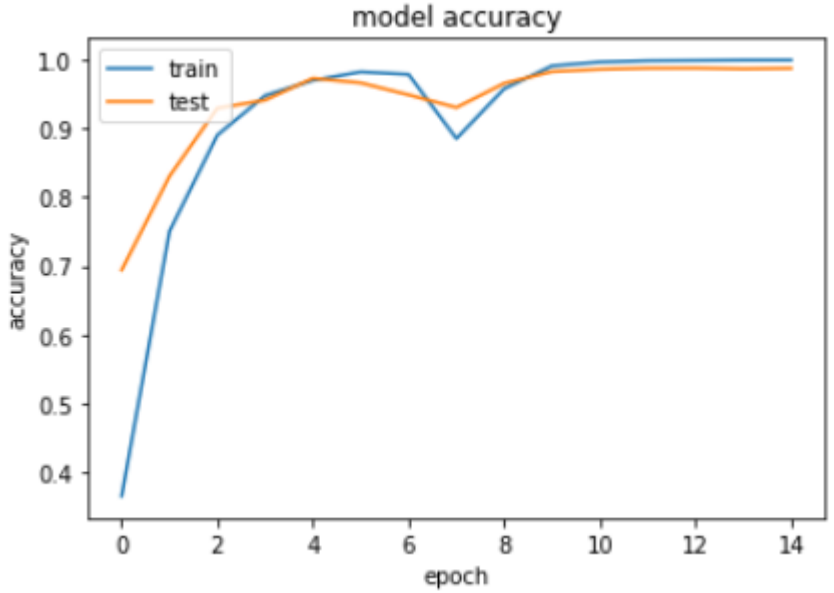
- Data Augmentation

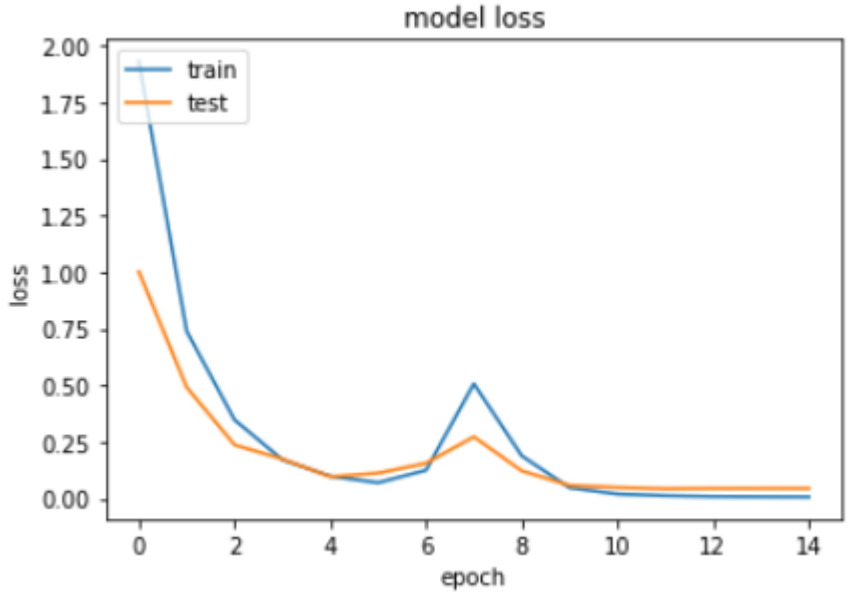
Faulty type	Number
Center	99
Donut	1
Edge-Loc	296
Edge-Ring	31
Loc	297
Near-full	16
Random	74
Scratch	72
none	13489

Faulty type	Number
Center	2160
Donut	2002
Edge-Loc	2368
Edge-Ring	2046
Loc	2376
Near-full	2032
Random	2146
Scratch	2088
none	2489

Result (2/2)

- Cross Validation score: **0.9802**
  - splits = 3
  - 0.9861 / 0.9796 / 0.9748





國立清華大學 National Tsing Hua University

INNOVATION ENGINEERING LAB

Outline

1. Introduction
2. Method
3. Result
4. Future Works

Future Works

- Wafer Maps of different size.
- Auto-encoder for noise-reduction

國立清華大學
National Tsing Hua University

INNOVATION
ENGINEERING
LAB

Thank You for Your Listening