

# Radon Transform based feature extraction

(Paper: Decision Tree Ensemble-Based Wafer Map Failure Pattern Recognition Based on Radon Transform-Based Features)

## 1. Summary

반도체 공정에서 Failure Pattern을 효과적으로 detect하는 것은 중요한 문제임. 왜냐하면 Failure Patterns을 detect하게 되면 공정내에 어느 부분에 이상이 생겼는지 파악이 가능하기 때문이다. 예를 들어, linear Scratch가 있는 경우 machine handling의 문제임을 알 수 있다. Failure Pattern을 detect하기 위해 다양한 기계학습 방법론을 이용할 수 있는데, **Raw wafer map이 High Dimension을 갖기 때문에** 정확도에 한계를 줄 수 있고 많은 Computation Cost가 발생할 수 있음. 따라서 본 논문에서는 **Radon Transform method를 이용한 Feature extraction을 통해 Raw wafer map의 Dimensionality reduction** 하여 Decision Tree Ensemble을 통한 Failure Pattern Recognition을 제안하고 있음.

## 2. Proposed method

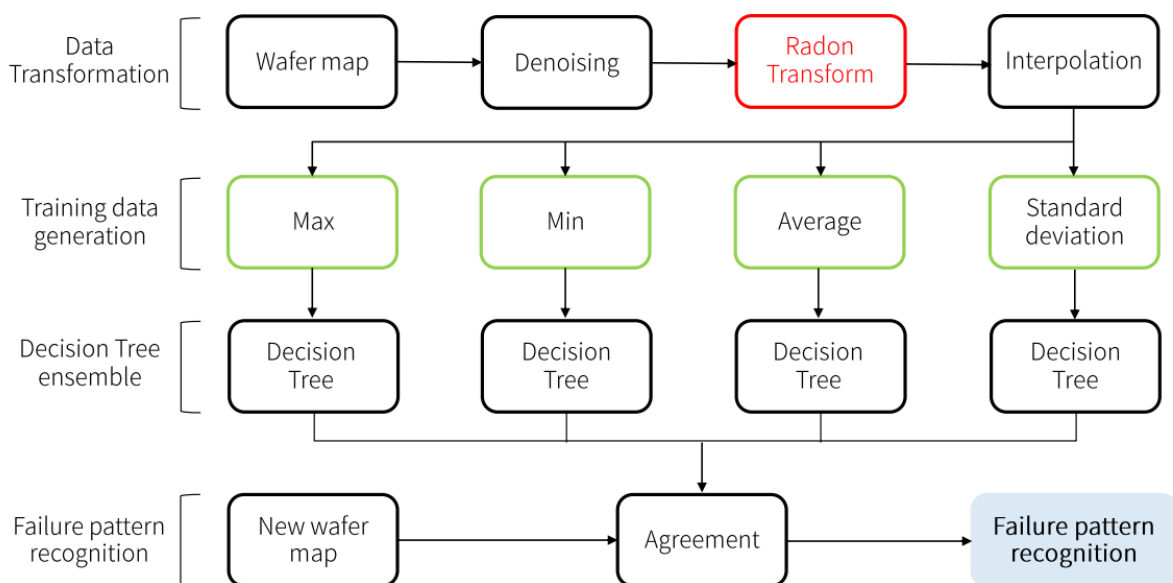


Figure 1. Flow-Chart of the Propose Method

- 위 그림은 제안하는 방법에 대한 Flow-Chart이다. Raw Wafer Map에 대하여 Median Filter를 통해서 Denoising을 한다. 그 다음 Radon Transform을 통해서 Feature extraction을 진행한다. Wafer Map의 크기가 Failure Pattern에 따라서 동일하지 않기 때문에 1D interpolation을 통해서 고정된 값으로 길이를 보정해준다. 각도에 따른 Radon Transform의 결과의 Max, Min, Average, Standard deviation을 통해서 Training data generation을 수행한다. 마지막으로 Decision Tree Ensemble을 통해서 Failure pattern recognition을 진행한다.

## 2.1 Dataset

WM-811k dataset을 사용하였고, 해당 데이터는 총 811,457개의 Wafer maps을 가지고 있고, 아래 그림과 같이 총 8개의 Failure Patterns (Center, Donut, Edgr-ring, Edge-local, local, Random, Scratch, Near-full) 존재한다.

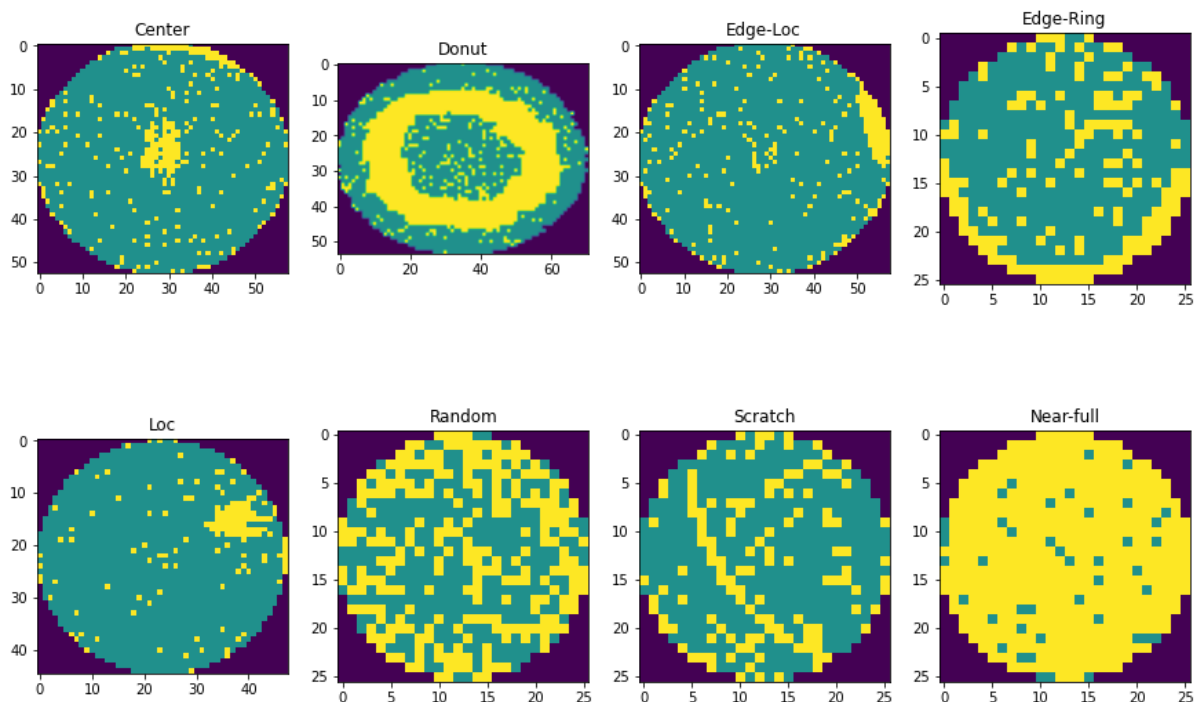


Figure 2. Failure Patterns of WM-811k

## 2.2 Denoising

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import cv2
%matplotlib inline

data = pd.read_pickle("C:/Users/KJW/Desktop/WM-811k.pkl")

data = data.drop(["waferIndex"], axis = 1)

mapping_type={'Center':0,'Donut':1,'Edge-Loc':2,'Edge-Ring':3,'Loc':4,'Random':5,'Scratch':6,'Near-full':7,'none':8}

mapping_train_test = {"Training" : 0 , "Test" : 1}

data = data.replace({"failureType" : mapping_type , "trainTestLabel" : mapping_train_test})

data_pattern = data[(data.failureType >= 0 ) & (data.failureType <=7)].reset_index()
```

- 먼저 WM-811k의 데이터를 pickle파일로 저장하여 pd.read\_pickle()을 통해 불러 온다. Failure type별로 숫자로 mapping시켜 변수를 replace한 후, Failure type 8가지 (“none” 제외)에 대해서 data\_pattern으로 따로 저장

```
x = [9,340, 3, 16, 0, 25, 84, 37]

labels_example = ['Center','Donut','Edge-Loc','Edge-Ring','Loc','Random','Scratch','Near-full']

plt.figure(figsize = (15 , int((9//4)+1)*5))

for i in range(len(x)):

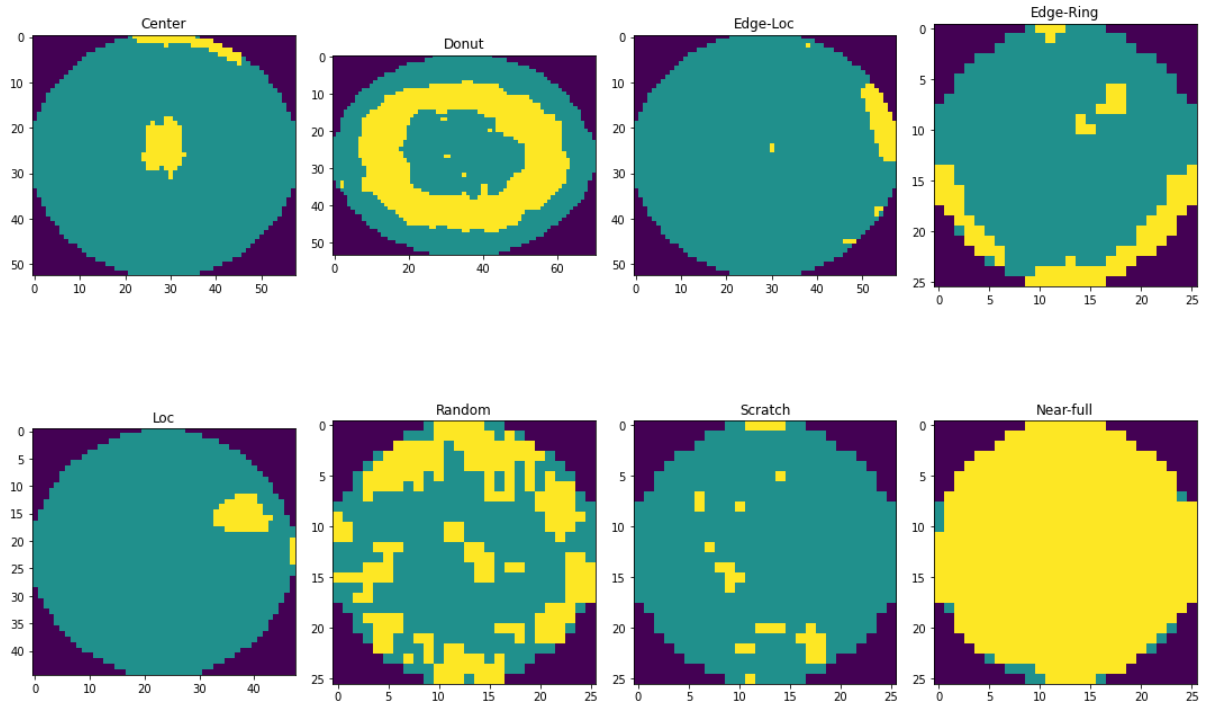
    img = data_pattern.waferMap[x[i]]
    blur = cv2.medianBlur(img , 3)
    plt.subplot(int((9//4)+1) ,4 , i+1)

    plt.title(labels_example[i])

    plt.imshow(blur)

plt.tight_layout()
plt.show()
```

- Denoising을 진행하기 위해서 본 논문에서는 Median Filter를 이용하였다. Cv2 library에서 원본 이미지(img)에 median Blur를 이용하여 Denoising 할 수 있다.



**Figure 3. Results of Denoising**

- Figure 3은 Figure 2의 원본 이미지에 대해서 Median Filter Denoising을 진행한 결과이다. 이제 해당 이미지를 통해서 Radon Transform을 통해 feature extraction을 진행한다.

## 2.3 Radon Transform

- 먼저 Radon Transform은 CT 진단법을 만들 때 중요하게 사용한 수학 이론이다. 2차원 물체를 예로 들면, Figure 4와 같이 파란색으로 표시한 것과 같은 물건이 존재할 때 동그라미로 둘러싼 영역 내부는 밖에서 보이지 않는 상황이고, X선 사진을 각도에 따라서 여러 장 찍어 보이지 않는 물체의 위치 및 모양을 알아내는 것이 목표이다. 바깥에서 내부를 향해 일정 방향(각도)로 X선을 쬔는데, 파란색 부분에서는 일정 비율로 흡수가 일어나고, 나머지 부분은 온전히 통과한다.

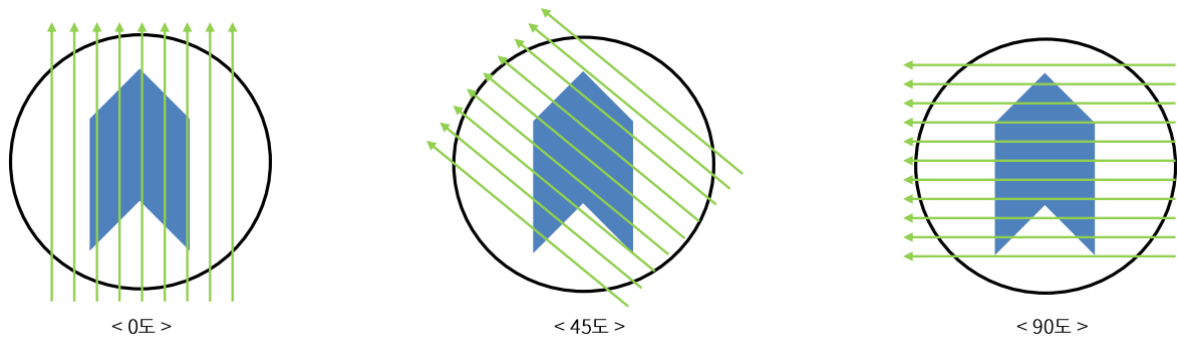


Figure 4. Examples of Radon Transform

- Wafer map에 존재하는 좌표( $x, y$ )를  $x$ 축과의  $\theta$  각도와  $\rho$  거리에 있는 직선으로 Projection. 이때 이 직선은  $x\cos\theta + y\sin\theta = \rho$ 로 표현할 수 있다.

$$- R(\rho, \theta) = \sum_{x=1}^m \sum_{y=1}^n f(x, y) \delta(x\cos\theta + y\sin\theta - \rho)$$

- Wafer map size  $m \times n$

-  $f(x, y)$  is set at 1 to indicate a defective die, and 0 otherwise,  $\delta$ 는 **Dirac delta function**으로 다음과 같이 나타냄

$$\delta(k) = \begin{cases} 1, & \text{if } k = 0 \\ 0, & \text{otherwise} \end{cases}$$

```
import skimage
from skimage import measure
from skimage.transform import radon

plt.figure(figsize = (15 , int((9//4)+1)*5))

for i in range(len(x)):

    img = data_pattern_copy.waferMap[x[i]]
    blur = cv2.medianBlur(img , 3)
    theta = np.arange(0., 180., 2)
    radon_projection = radon(blur , theta = theta)

    plt.subplot(int((9//4)+1) , 4 , i+1)

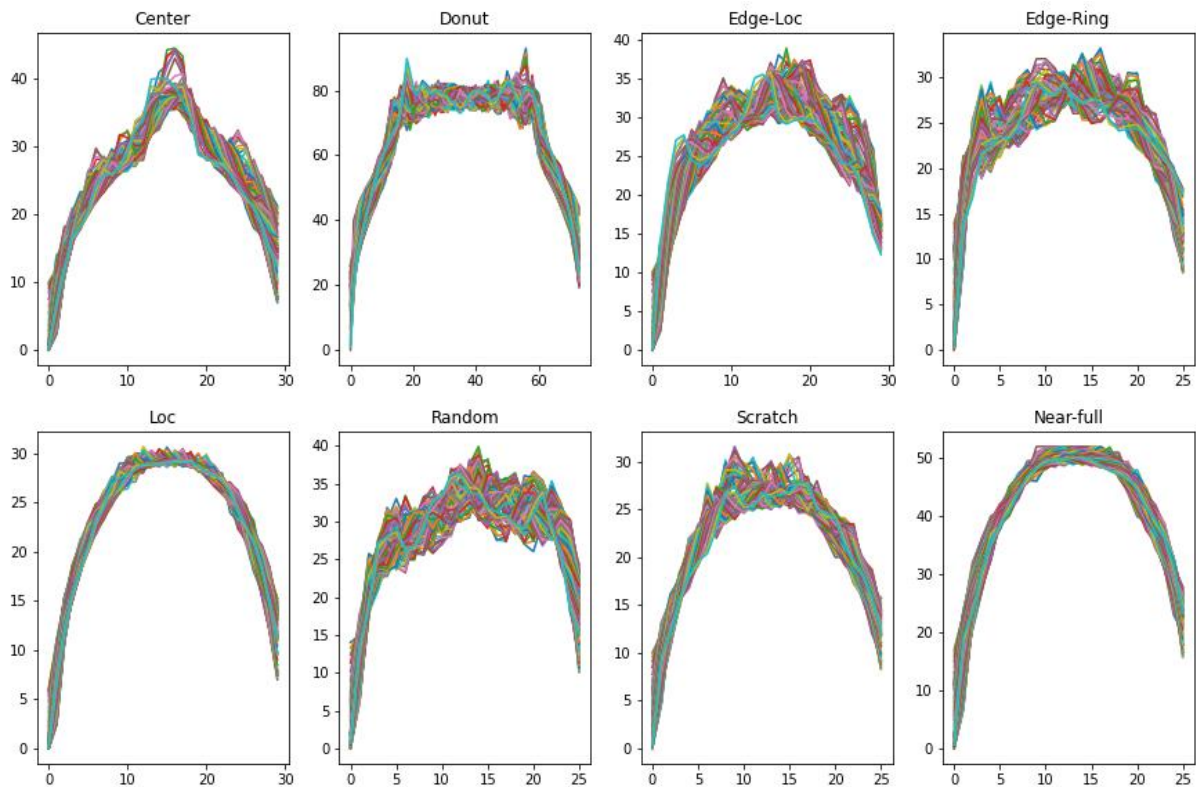
    plt.title(labels2[i])

    for k in range(radon_projection.shape[1]):

        radon_transform = [radon_projection[j][k] for j in range(radon_projection.shape[0])]

        plt.plot(radon_transform)
```

- python에서 제공하는 `skimage.transform`을 통해서 이미지에 대한 Radon Transform을 수행할 수 있다. Radon Transform을 위해서는 Projection하기 위한 각도( $\theta$ )를 정해야 하는데, 본 논문에서는 0도부터 178도까지 2도씩 증가시킨 각도를 사용하였다.



**Figure 5. Results of Radon Transform**

- Figure 5는 각도 별로 Radon Transform한 결과를 도식화한 것이다. Failure Pattern마다 총 90개의 결과가 겹쳐져서 나오게 된다. 또한 Pattern에 따라서 Radon Transform의 결과의 형태가 다른 것을 알 수 있다.

## 2.4 1D interpolation

- Radon Transform의 결과들의 크기는 Wafer Map의 크기에 따라서 달라지기 때문에 Figure 5의 x축을 보았을 때 길이가 모두 다른 것을 볼 수 있다. 따라서 본 논문에서는 1D interpolation을 통해서 -15 부터 15까지 30의 길이를 갖도록 변환해 주었

다.

```
img = data_pattern_copy.waferMap[x[1]]
blur = cv2.medianBlur(img , 3)

theta = np.arange(0., 180., 2)

radon_projection = radon(blur , theta = theta)

for k in range(radon_projection.shape[1]):

    radon_transform = [radon_projection[j][k] for j in range(radon_projection.shape[0])]

    x_ = np.arange(-int(len(radon_transform)/2),int(len(radon_transform)/2) , 1)
    y = radon_transform
    f = interpolate.interp1d(x_ , y , kind = "linear")
    x__ = np.arange(-15,14,1)
    y__ = f(x__)

    #len(x_)
    plt.plot(x__ , y__)
```

- scipy에서 제공하는 interpolate 함수를 통해서 기존의 x, y 값을 interpolation하고 변환할 x값에 대하여 새로운 y를 생성하여 최종적으로 x\_\_, y\_\_ 를 feature로서 생성한다.

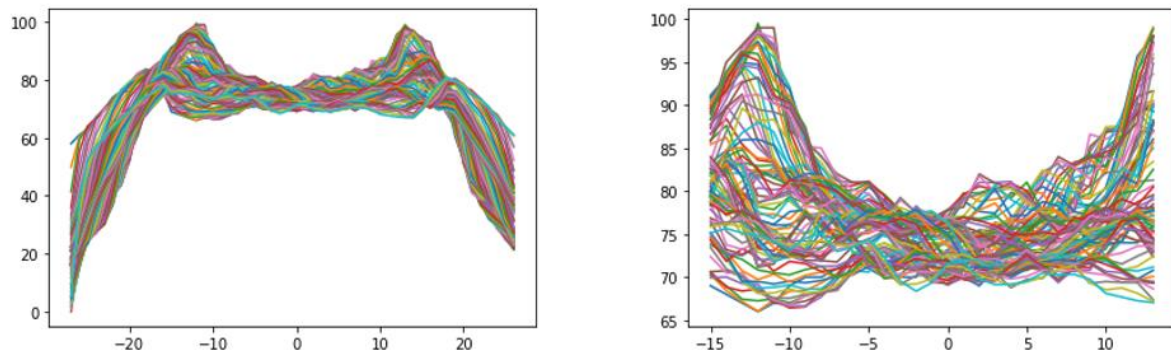
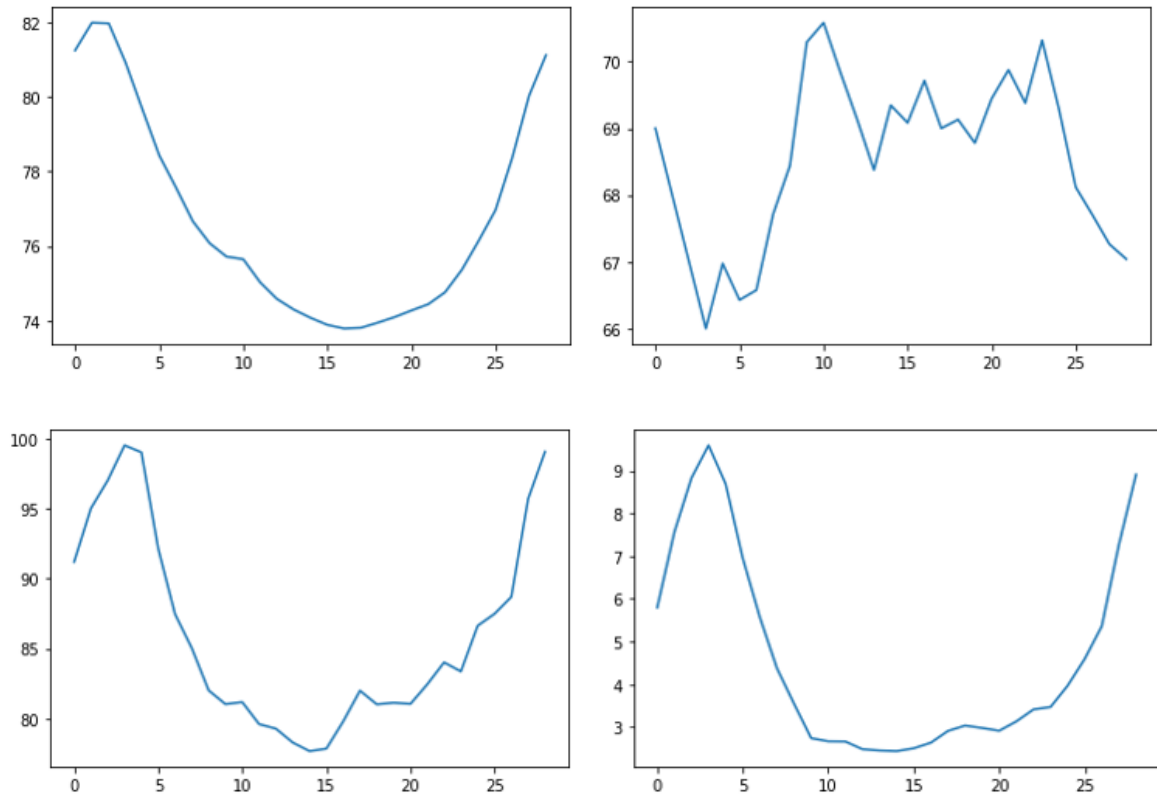


Figure 6 Results of interpolation

- 위 그림은 Donut pattern의 Radon Transform 결과를 1D interpolation을 적용한 결과이다. 다른 Pattern들에 대해서도 마찬가지로 똑같이 적용하였다.

## 2.4 Training data generation



**Figure 7 Examples of Training data generation**

- Figure 7은 interpolation까지 끝난 결과를 이용하여 각도마다 **Average, Min, Max, Standard deviation**을 통해 4가지의 데이터를 생성한다. 생성한 데이터를 Decision Tree의 변수로 활용하여 Failure Pattern recognition을 수행한다.

### 3 Conclusions

- 본 논문은 Raw wafer map의 Feature extraction 및 Dimensionality reduction을 위해서 Radon Transform 방식을 이용하였다. Radon Transform은 의료기기에 사용되는 수학적 원리이지만 Defect Patterns을 효율적으로 Recognition하기 위한 방법으로 사용되었습니다. Median Filter를 이용하여 Denoising하여 Defect pattern을 더 부각시키고 Radon Transform을 적용하였는데, Defect Patterns을 Maximal Area로 판단하여 더 정확하게 처리할 수 있는 Region labeling algorithm을 사용한다면 더 좋은 결과를 기대해볼 수 있다. 그리고 Wafer Map 뿐만 아니라 다양한 이미지를 Recognition 하기



위한 Feature extraction & Dimensionality reduction 방법론으로 사용해볼 수 있을 것이다.