

프로젝트 #1 중간 발표

2022. 5. 25

충북대학교 산업인공지능학과

[20-1조] 최원희, 손의걸

수행방법 및 기여도

수행방법

- 수시로 자료를 공유하며, 코딩을 수행하고, 결과를 취합
- *Anaconda3 Jupyter Notebook* 으로 프로그램 코딩
- 결과 취합 및 자료 정리

업무분장 및 기여도

이름	비중	수행내용	비고
최원희	60%	<ul style="list-style-type: none">• 자료 검색 및 취합• 데이터 증량• 주제발표, 결과발표	
손의걸	40%	<ul style="list-style-type: none">• 데이터 증량• 코딩/학습• 중간 발표	

- 비중은 총합이 100%일 것

환경 설정

Anaconda3 설치

- <https://www.anaconda.com>
- Jupyter Notebook 사용

Tenforflow 설치

- *pip install tensorflow*

AVX 미지원 CPU 사용으로 파이썬 3.6버전 Tenforflow 1.5버전으로 설정

- *conda install python=3.6*
- *pip install tensorflow==1.5.0*

사용 컴퓨터

- *Windows 10 Pro*
- *11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz*
- *8GB ram*
- *Intel Iris(R) Xe Graphics*

CNN 모델 구조

```
def create_model():
    input_shape = (26, 26, 3)
    input_tensor = Input(input_shape) # (80000, 26, 26, 3)

    Conv_1 = layers.Conv2D(16, (3,3), activation='relu',padding='same')(input_tensor)
    Pool_2 = layers.MaxPooling2D(pool_size=(2,2))(Conv_1)
    Conv_3 = layers.Conv2D(16, (3,3), activation='relu',padding='same')(Pool_2)

    Conv_4 = layers.Conv2D(32, (3,3), activation='relu',padding='same')(Conv_3)
    Pool_5 = layers.MaxPooling2D(pool_size=(2,2))(Conv_4)
    Conv_6 = layers.Conv2D(32, (3,3), activation='relu',padding='same')(Pool_5)

    Conv_7 = layers.Conv2D(64, (3,3), activation='relu',padding='same')(Conv_6)
    Pool_8 = layers.MaxPooling2D(pool_size=(2,2))(Conv_7)
    Conv_9 = layers.Conv2D(64, (3,3), activation='relu',padding='same')(Pool_8)

    Conv_10 = layers.Conv2D(128, (3,3), activation='relu',padding='same')(Conv_9)
    Pool_11 = layers.MaxPooling2D(pool_size=(2,2))(Conv_10)
    Conv_12 = layers.Conv2D(128, (3,3), activation='relu',padding='same')(Pool_11)

    drop2D_13 = layers.SpatialDropout2D(0.2)(Conv_12)
    Pool_14 = layers.MaxPooling2D(pool_size=(2,2))(drop2D_13)

    flat = layers.Flatten()(Pool_14)

    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'])

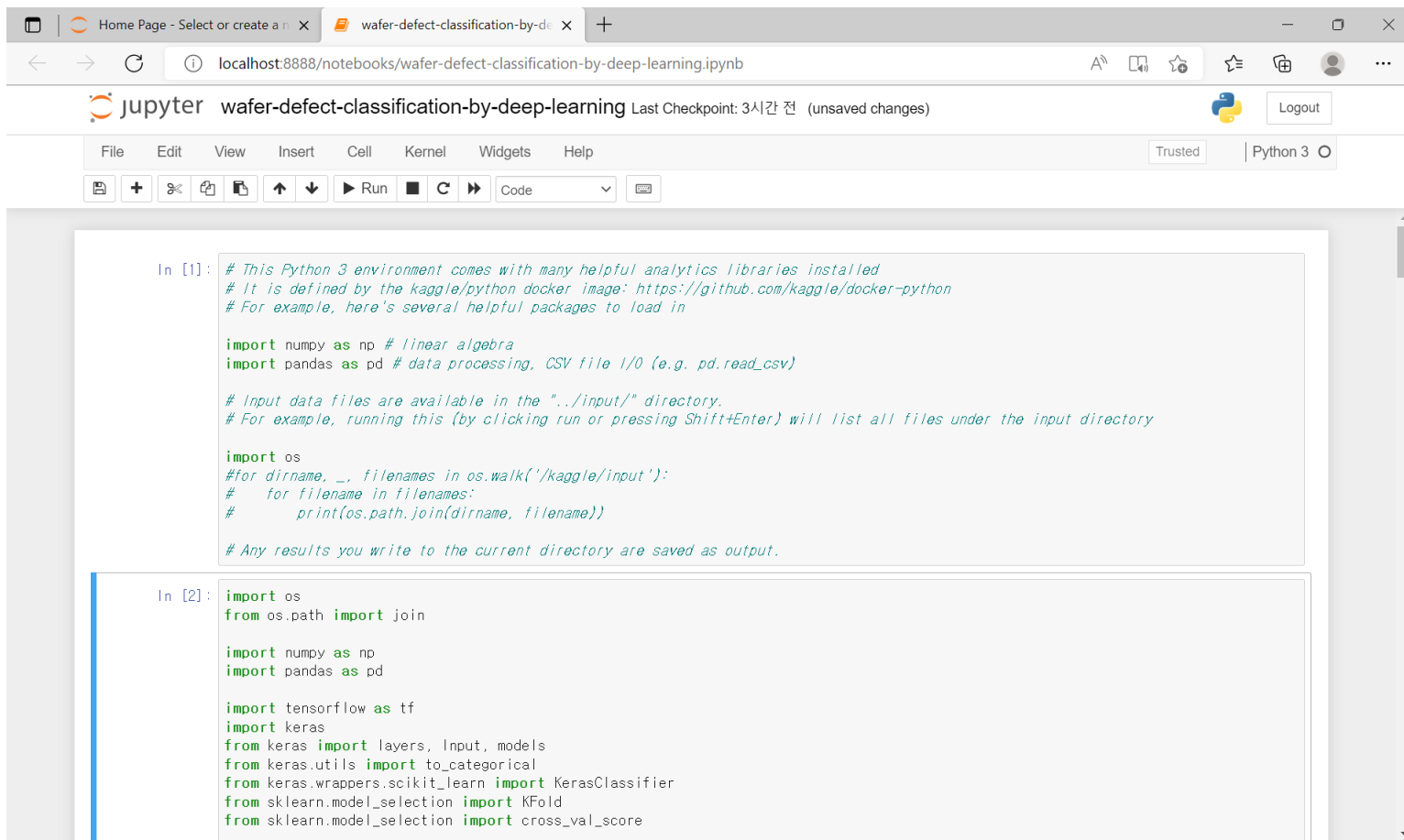
    return model
```

- 데이터셋의 이미지들을 26x26으로 resizing하여 사용

데이터셋

Data augmentation/전처리

- Kaggle에서 제공하는 wafer-defect-classification-by-deep-learning 사용



```
In [1]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
#for dirname, _, filenames in os.walk('/kaggle/input'):
#    for filename in filenames:
#        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.

In [2]: import os
from os.path import join

import numpy as np
import pandas as pd

import tensorflow as tf
import keras
from keras import layers, Input, models
from keras.utils import to_categorical
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
```

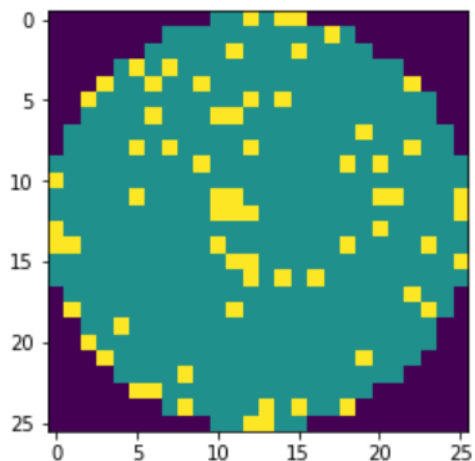
데이터셋

수행 결과

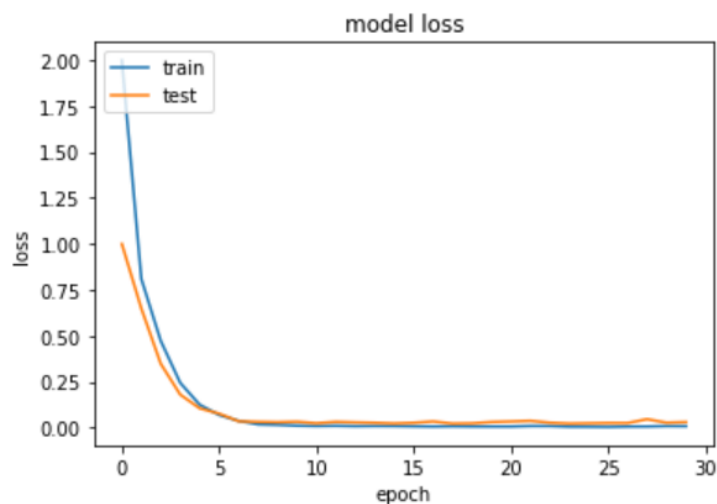
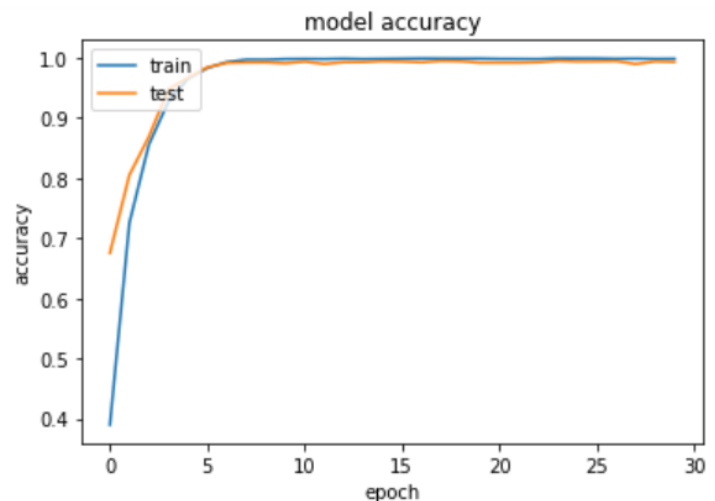
- Kaggle에서 제공하는 wafer-defect-classification-by-deep-learning 의 수행 결과임

```
In [14]: # plot 1st data
plt.imshow(x[2040])
plt.show()

# check faulty case
print('Faulty case : {}'.format(y[2040]))
```



Faulty case : ['none']



향후 계획

CNN 구조의 대한 스터디

- CNN의 구조에 대한 스터디와 정리된 자료를 가지고 진행 방향 토의

학습

- 데이터 증량 코딩 및 수행

결과 및 토의

- 결과 및 토의 자료 작성
- 발표 자료 작성

개선 방향

- 추후 개선 방향 도출

감사합니다