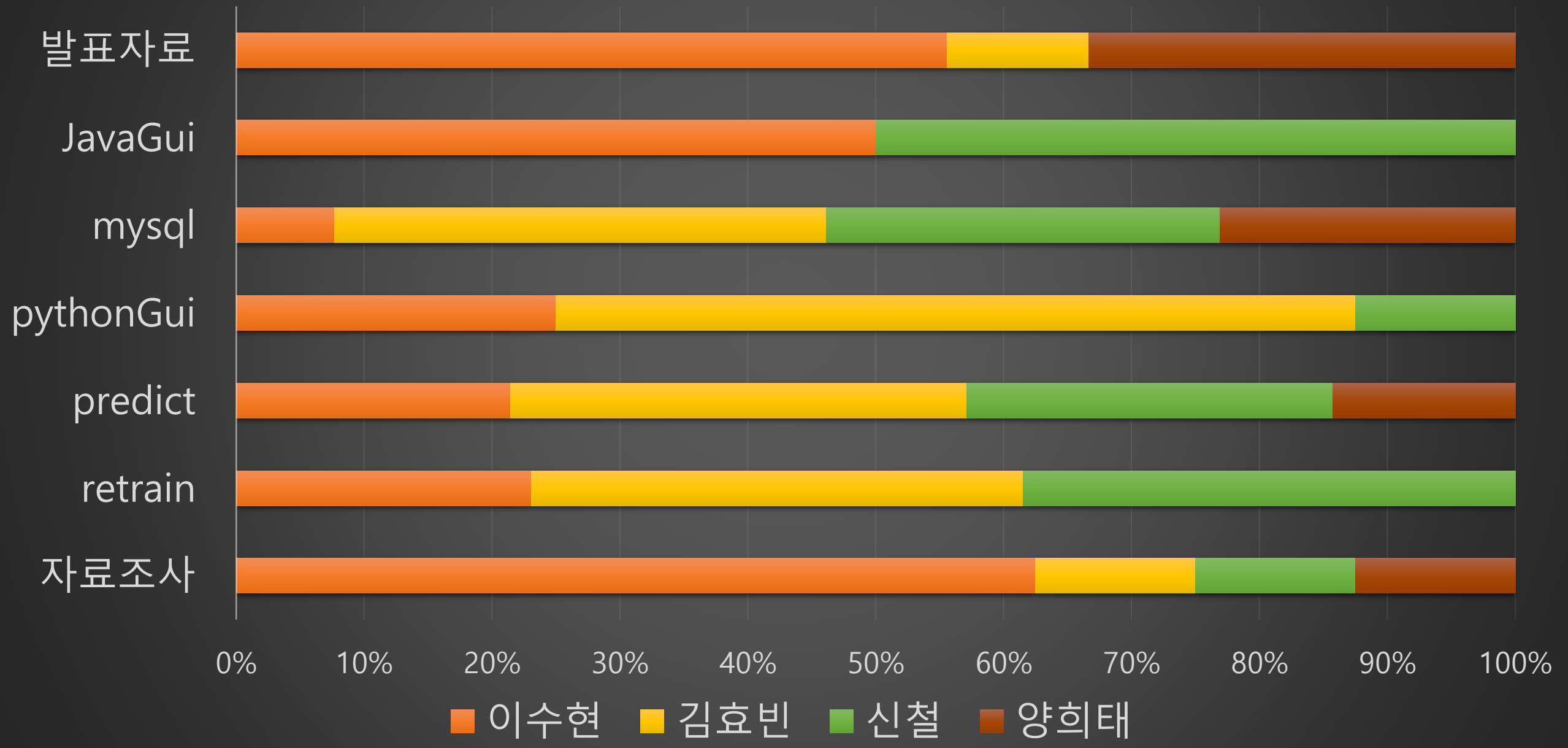
Inception 기반 Image 예측과 분석 AI 시스템 구현

(부제: 과일 자동 분류 시스템)

Super Fruit

조원 역활


발표자료
JavaGui
mysql
pythonGui
predict
retrain
자료조사
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
이수현
김효빈
신철
양희태

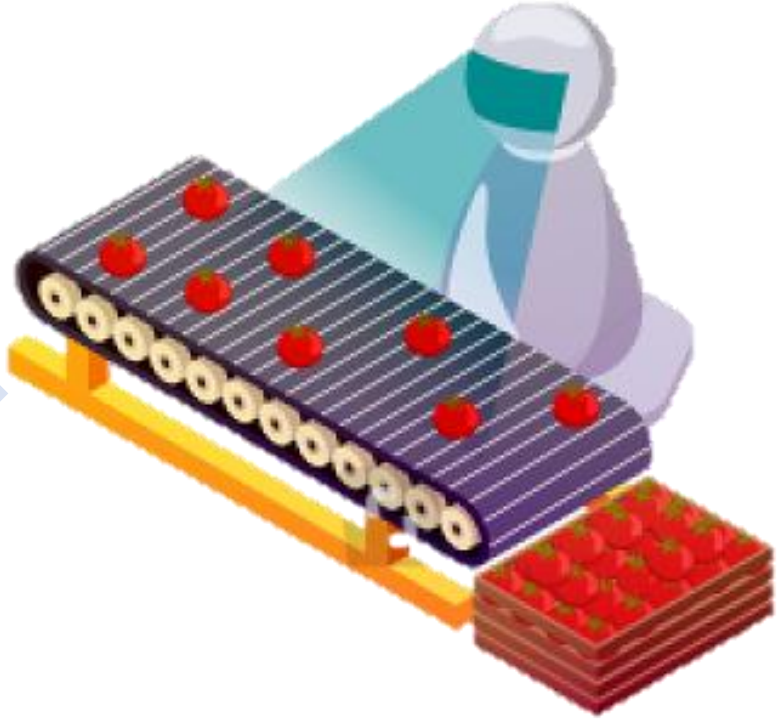
목차


LONGINES
AUTOMATIC

Are You Ready for

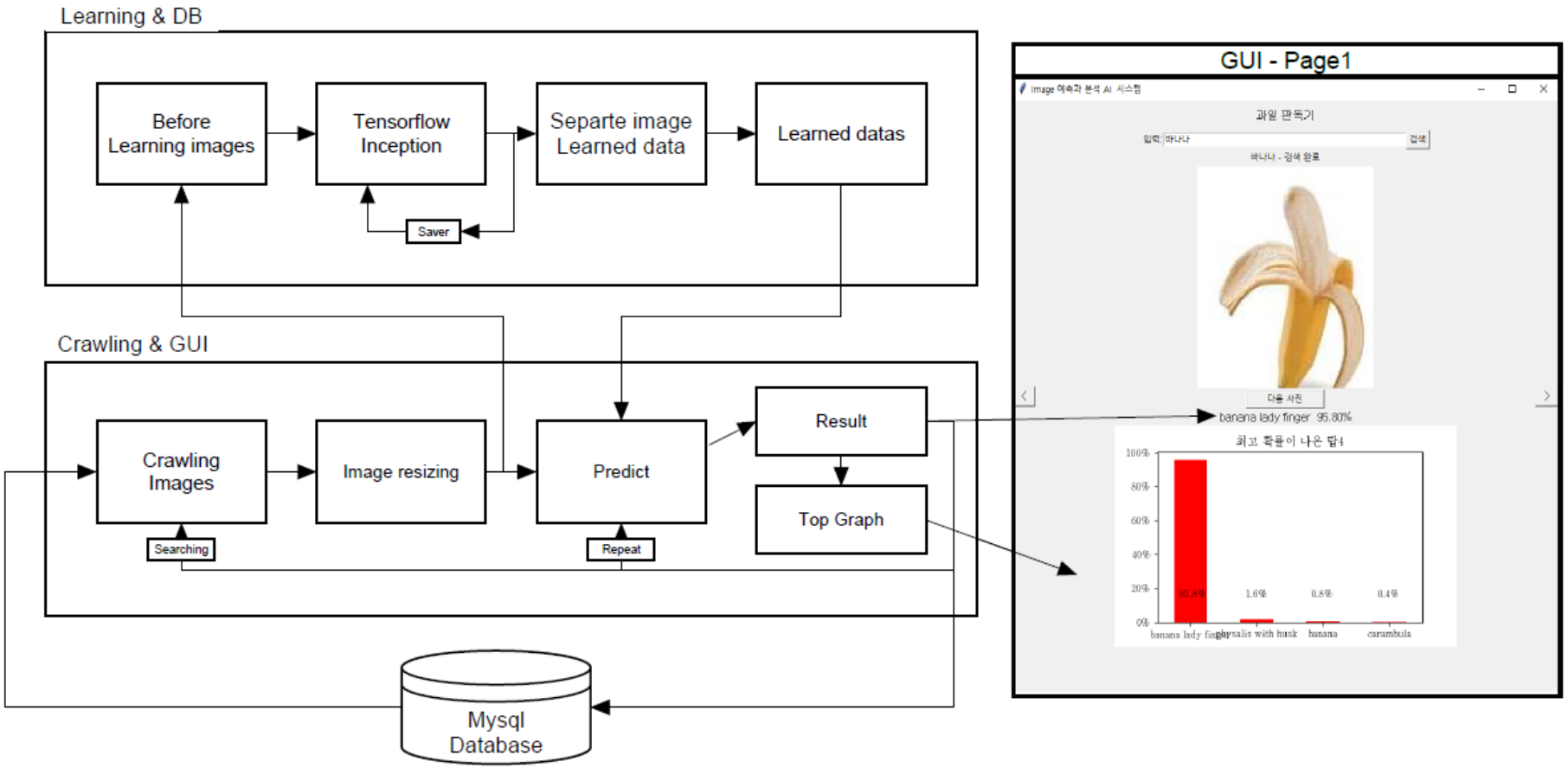
The Fourth Industrial Revolution?

PROGRAMS



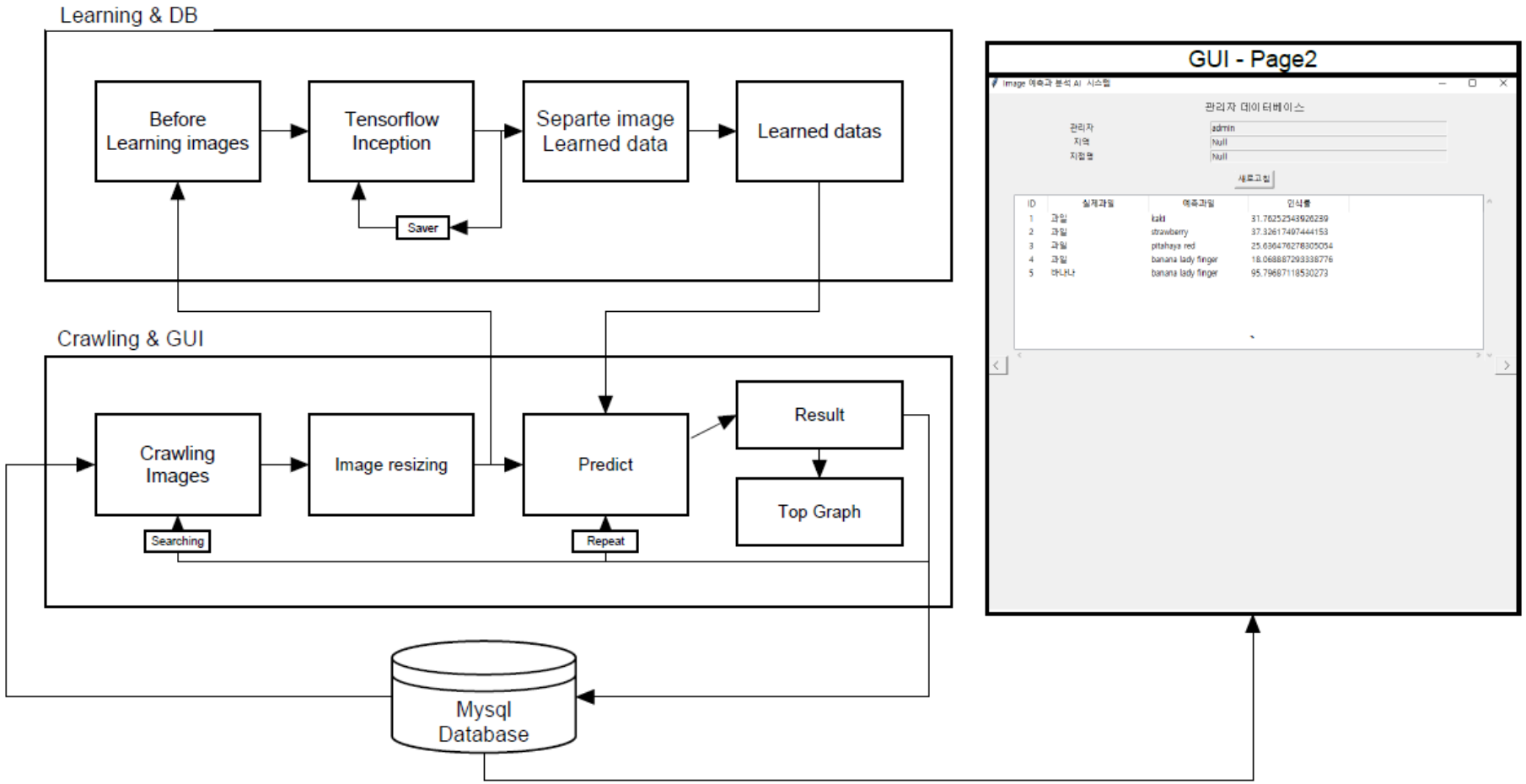
Flow Chart

Super Fruit Diagram

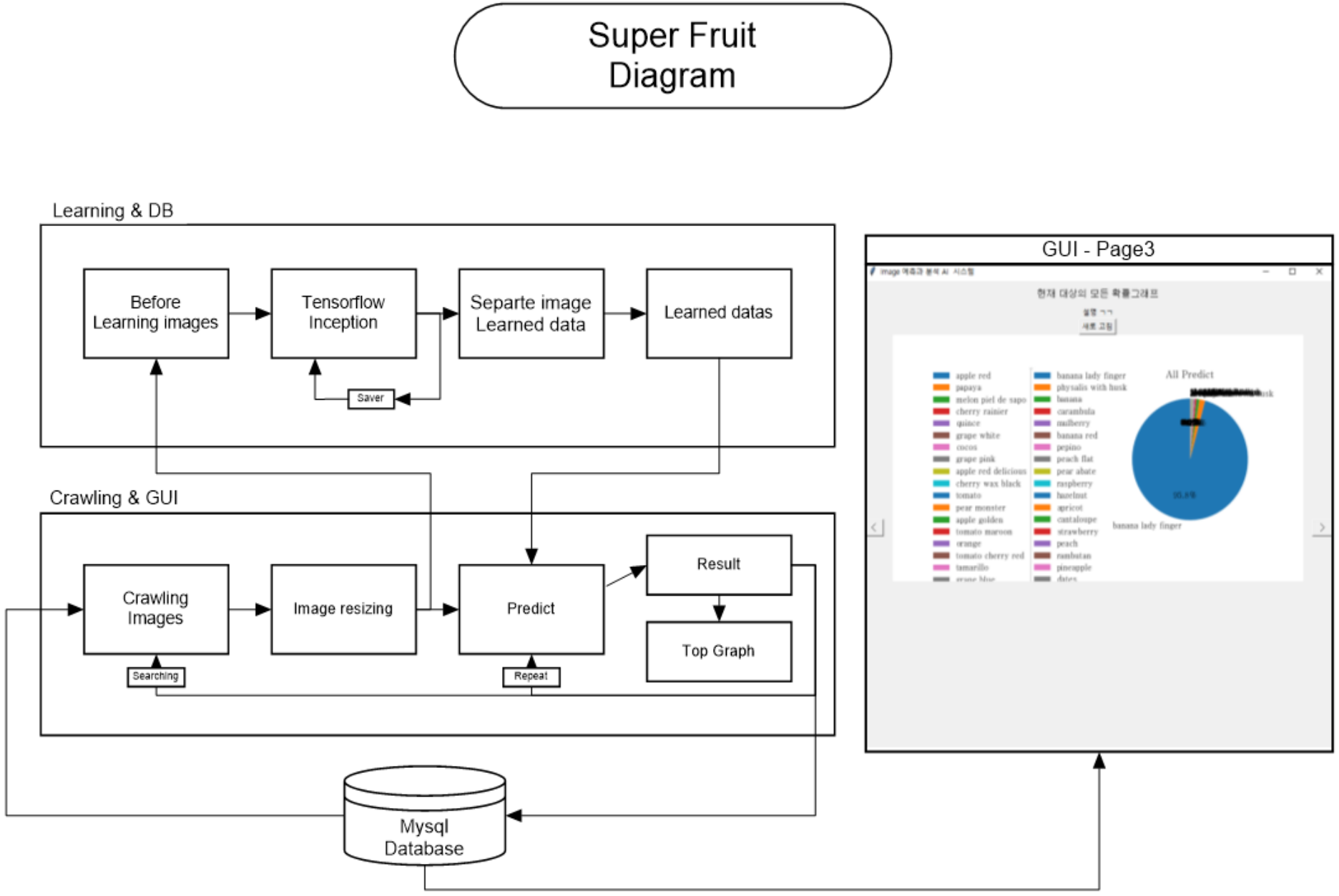


Flow Chart

Super Fruit Diagram



Flow Chart


Super Fruit Diagram
Learning & DB
Before Learning images
Tensorflow Inception
Separte image Learned data
Learned datas
Saver
Crawling & GUI
Crawling Images
Image resizing
Predict
Result
Top Graph
Searching
Repeat
Mysql Database
GUI - Page3
All Predict

Dataset

432

Fruits 360 dataset

A dataset with 65429 images of 95 fruits

Mihai Oltean • updated 2 months ago (Version 44)

image data, food and drink, multiclass classification

Description

Fruits 360 dataset: A dataset of images containing fruits

Version: 2019.01.16.0

Content

The following fruits are included: Apples (different varieties: Golden, Red Yellow, Granny Smith, Red, Red Delicious), Apricot, Avocado, Avocado ripe, Banana (Yellow, Red, Lady Finger), Cactus fruit, Cantaloupe (2 varieties), Carambula, Cherry (different varieties, Rainier), Cherry Wax (Yellow, Red, Black), Chestnut, Clementine, Cocos, Dates, Granadilla, Grape (Blue, Pink, White (different varieties)),

Data (658 MB)

Data Sources

fruits-360_dataset.zip

About this file

A high-quality, dataset of images containing fruits. The following fruits are included:
Apples (different varieties: Golden, Golden-Red, Granny Smith, Red, Red Delicious)

이름	수정한 날짜	유형	크기
bottlenecks	2019-02-15 오후...	파일 폴더	
fruits_photos	2019-02-15 오후...	파일 폴더	
inception	2019-02-15 오후...	파일 폴더	
test-multiple_fruits	2019-02-12 오후...	파일 폴더	
Crawling.py	2019-02-19 오후...	PY 파일	2KB
fruits_graph.pb	2019-02-15 오후...	PB 파일	86,002KB
fruits_labels.txt	2019-02-15 오후...	텍스트 문서	1KB
fruits_photos.zip	2019-02-18 오후...	ALZip ZIP File	219,676KB
test.py	2019-02-21 오전...	PY 파일	0KB

CNN 러닝 실행과정

fruits [C:₩Users₩user₩PycharmProjects₩fruits] - ...₩CNN.py [fruits] - PyCharm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

fruits 〉 CNN.py 〉 predict_v5

predict.py | predict1.py | predict_v4.py | predict_v5.py | CNN.py | retrain.py | MY.py | 1.py

mnist — Match Case — Words — Regex — 6 matches

```
L3 = tf.nn.dropout(L3, keep_drop)
'''

L3 = tf.contrib.layers.flatten(L2)
L3 = tf.layers.dense(L3, 256, activation=tf.nn.relu)
L3 = tf.layers.dropout(L3, keep_drop, is_correct)
# 출력값 10를 만듬
'''
W4 = tf.Variable(tf.random_normal([256, 10], stddev=0.01))

model = tf.matmul(L3, W4)
'''
model = tf.layers.dense(L3, 10, activation=None)
# https://tykimos.github.io/2017/01/27/MLP_Layer_Talk/   (dense에 관한 정리)

# 손실값
cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(logits=model, labels=y))
optimizer = tf.train.AdamOptimizer(0.001).minimize(cost)

# 학습시키고 결과 확인
init = tf.global_variables_initializer()

sess = tf.Session()
```

Run: CNN | predict_v5

```
Epoch: 0013 Avg. cost= 0.004237
Epoch: 0014 Avg. cost= 0.001582
Epoch: 0015 Avg. cost= 0.004417
최적화 완료
2019-02-27 16:52:54.586433: W tensorflow/core/framework/allocator.cc:122] Allocation of 1730560000 exceeds 10% of system memory.
정확도: 0.982
```

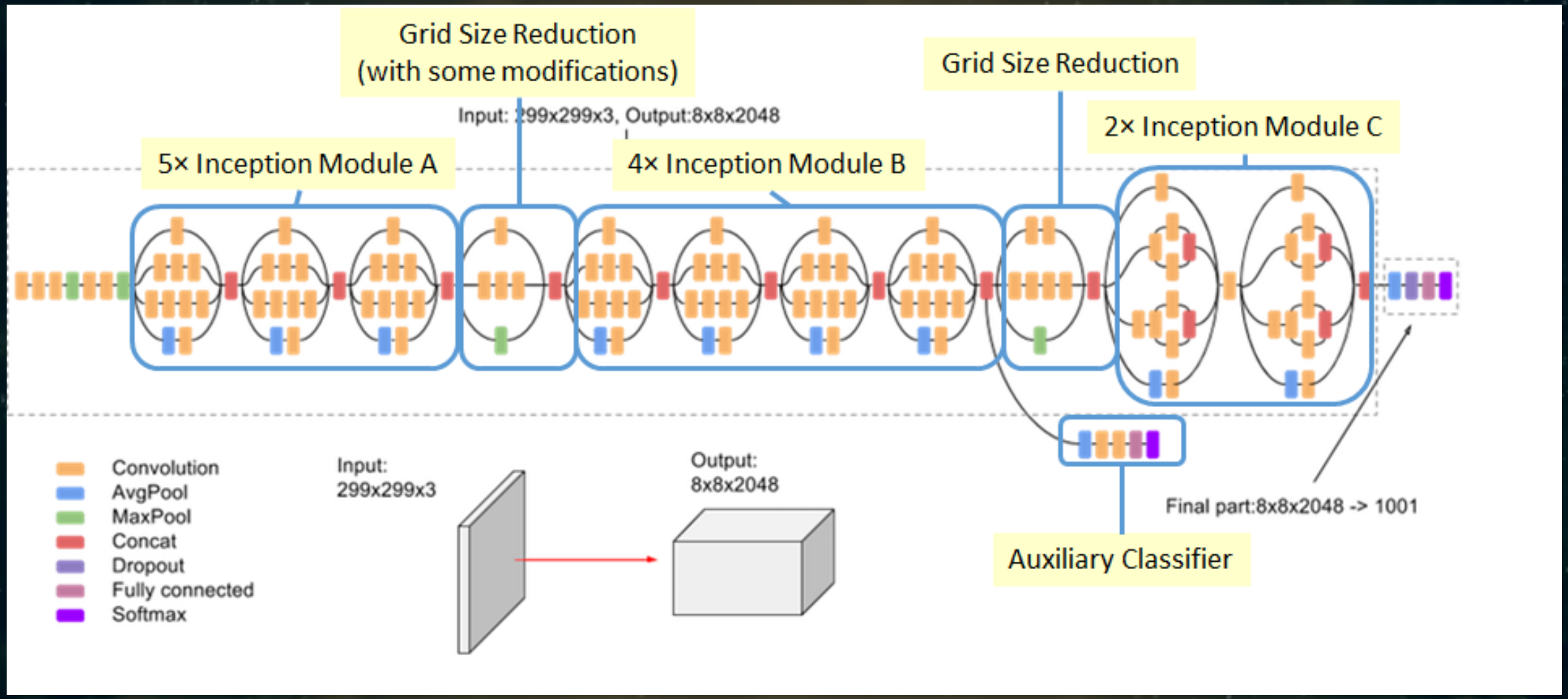
Python Versions Compatibility
Your source code contains __future__ imports.
Would you like to enable Code compatibility...

4: Run | 6: TODO | Terminal | Python Console | Event Log

Install packages failed: Installing packages: error occurred. Details... (today 오후 4:49)

25:37 CRLF UTF-8

Inception


Grid Size Reduction
(with some modifications)
Input: 299x299x3, Output:8x8x2048
Grid Size Reduction
2× Inception Module C
5× Inception Module A
4× Inception Module B
Convolution
AvgPool
MaxPool
Concat
Dropout
Fully connected
Softmax
Input:
299x299x3
Output:
8x8x2048
Final part:8x8x2048 -> 1001
Auxiliary Classifier

러닝 실행과정

retrain1

```
eating bottleneck at ./workspace/bottleneck₩Tomato₩1_100- (265).jpg.txt
90500 bottleneck files created.
Creating bottleneck at ./workspace/bottleneck₩Tomato₩1_100- (266).jpg.txt
Creating bottleneck at ./workspace/bottleneck₩Tomato₩1_100- (268).jpg.txt
Creating bottleneck at ./workspace/bottleneck₩Tomato₩1_100- (27).jpg.txt
Creating bottleneck at ./workspace/bottleneck₩Tomato₩1_100- (270).jpg.txt
Creating bottleneck at ./workspace/bottleneck₩Tomato₩1_100- (271).jpg.txt
Creating bottleneck at ./workspace/bottleneck₩Tomato₩1_100- (272).jpg.txt
Creating bottleneck at ./workspace/bottleneck₩Tomato₩1_100- (274).jpg.txt
```

accuracy = 99.8% (N=10032)

retrain1.py

```python
import ...

FLAGS = None

# These are all parameters that are tied to the particular model architecture
# we're using for Inception v3. These include things like tensor names and their
# sizes. If you want to adapt this script to work with another model, you will
# need to update these to reflect the values in the network you're using.
# pylint: disable=line-too-long
DATA_URL = 'http://download.tensorflow.org/models/image/imagenet/inception-2015-12-05.tgz'
# pylint: enable=line-too-long
BOTTLENECK_TENSOR_NAME = 'pool_3/_reshape:0'
BOTTLENECK_TENSOR_SIZE = 2048
MODEL_INPUT_WIDTH = 299
MODEL_INPUT_HEIGHT = 299
MODEL_INPUT_DEPTH = 3
JPEG_DATA_TENSOR_NAME = 'DecodeJpeg/contents:0'
RESIZED_INPUT_TENSOR_NAME = 'ResizeBilinear:0'
MAX_NUM_IMAGES_PER_CLASS = 2 ** 27 - 1  # ~134M

def create_image_lists(image_dir, testing_percentage, validation_percentage):
    """Builds a list of training images from the file system.
    Analyzes the sub folders in the image directory, splits them into stable
    training, testing, and validation sets, and returns a data structure
    describing the lists of images for each label and their paths.
    Args:
      image_dir: String path to a folder containing subfolders of images.
      testing_percentage: Integer percentage of the images to reserve for tests.
      validation_percentage: Integer percentage of images reserved for validation.
    Returns:
      A dictionary containing an entry for each label subfolder, with images split
      into training, testing, and validation sets within each label.
    """
    if not gfile.Exists(image_dir):
        print("Image directory '" + image_dir + "' not found.")
```

Java

Eclipse IDE

Run Window Help

MySQLtoGraph.java

```java
package com.oracle;

import java.sql.Connection;

public class MySQLtoGraph extends ApplicationFrame {                    //MySQLtoGraph 클래스는 ApplicationFrame을 상속 받음
    static ArrayList <String> name = new ArrayList<String> ();
    static ArrayList <String> score = new ArrayList<String> ();
    public static void main(String[] args) throws SQLException{
        // db connection
        String jdbc_driver     = "com.mysql.jdbc.Driver";
        String jdbc_url     = "jdbc:mysql://localhost:3306/kobis";
        String user         = "root";
        String pwd            = "1234";

        Connection con = null;
        Statement stmt = null;
        ResultSet rs = null;
        ResultSetMetaData metaData = null;

        try{
            // mysql 커넥션 설정
            Class.forName(jdbc_driver);                                 //Class.forName() 을 이용해서 드라이버 로드
            con = DriverManager.getConnection(jdbc_url, user, pwd);     //DriverManager.getConnection() 으로 연결 얻기
            stmt = con.createStatement();                               //Connection 인스턴스를 이용해서 Statement 객체 생성
            String sql = "select * from fruits3";                       //MySQL에서 사용할 쿼리문 작성
            rs = stmt.executeQuery(sql);                                //쿼리문 실행 결과를 ResultSet에 받기
            metaData = rs.getMetaData();                                //ResultSet과 관련된 메타 데이터를 얻어 ResultSetMetaDa

            // 각 행을 읽어 리스트에 저장한다.
            int sizeOfcolumn = metaData.getColumnCount();               //getColumnCount메서드로 ResultSet의 총 필드수를 반환히
            String column;
            List<Map<String,Object>> list = new ArrayList<Map<String,Object>>();
            Map<String, Object> map;

            while(rs.next()){                                           //rs.next()는 ResultSet에 다음 열이 존재하는지 검사한 후
                map = new HashMap<String,Object>();                     //HashMap의 key값은 String타입으로 받고, value값은 Obj

                for(int indexOfcolumn=0; indexOfcolumn<sizeOfcolumn; indexOfcolumn++){
                    column = metaData.getColumnName(indexOfcolumn + 1);     //String 타입의 column변수에 첫번째 열이름, 즉 DB의 필드명
                    map.put(column, rs.getString(column));                  //map객체에 필드명과 주어진 행과 주어진 열에 해당하는 필드의 값
                }
                list.add(map);                                          //리스트에 map객체를 받음
            }

            // 테스트 출력
            for( Map<String, Object> map1 : list ){                     //리스트에 저장된 map객체를 map1으로 명명하여 가져옴
                Iterator<String> it = map1.keySet().iterator();         //HashMap타입인 map1의 key값만 모아서 it객체에 저장
                while(it.hasNext()){                                    //it객체의 커서 다음 값이 존재하면
                    String key = it.next();                             //그 다음 값을 key값에 저장
                    if(key.equals("name")) {                            //필드명이 name이면
                        String value = (String)map1.get(key);           //필드명이 name인 데이터 값을
```

Database Systems

PROGRAMMING

Python Java

eclipse

PC

쉽다 안정적 경제적


Q&A