

# 안드로이드 애플리케이션 프로그래밍

## - 악성코드 탐지 모델 및 안드로이드 앱 개발

텐서플로 라이트 모델을 활용한 악성코드 탐지 모델 개발



부산대학교 정보·의생명공학대학  
정보컴퓨터공학부



# 이론

# 강의 목표와 구성

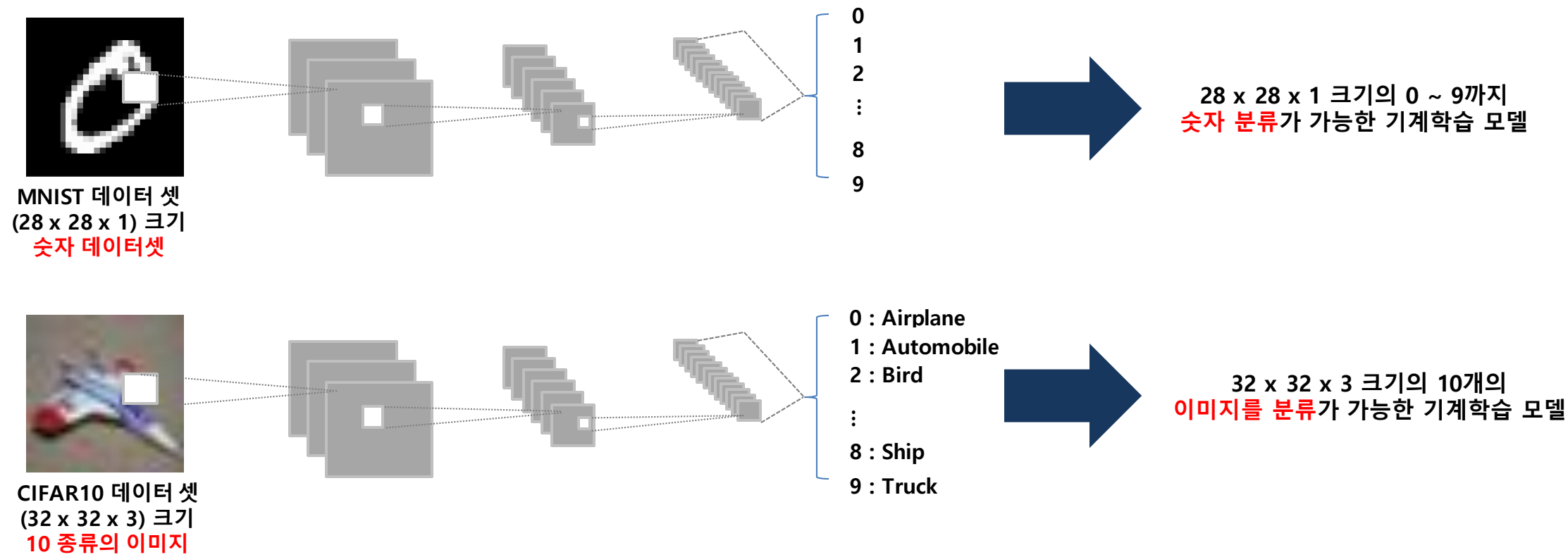
## ❖ 악성코드 탐지 모델 및 안드로이드 앱 개발

- 기계학습 모델 활용
- 기계학습 기반의 악성코드 탐지 모델 개발
- 기계학습 모델 기반의 악성코드 분류 앱 개발

# 기계학습 모델 활용

## ❖ 기계학습 모델 응용

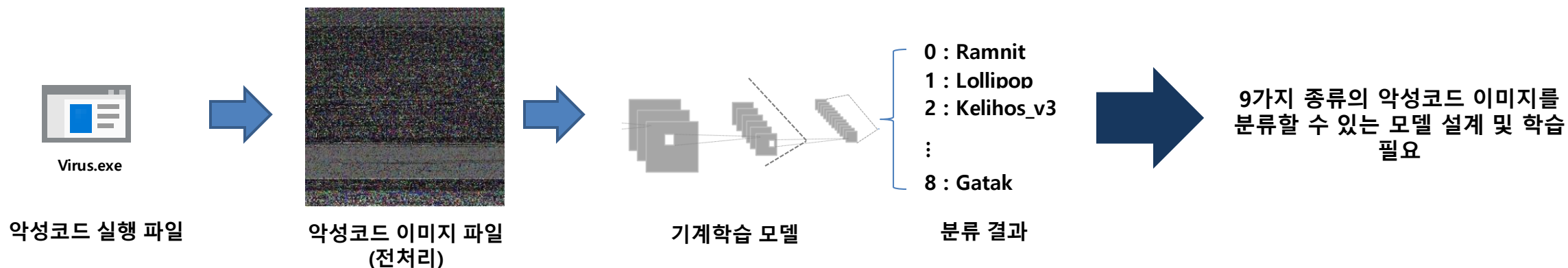
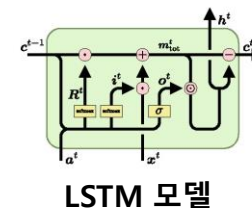
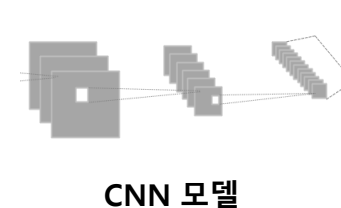
- 기계학습 모델은 데이터로부터 주어진 레이블을 추론하도록 학습 가능
- 학습 데이터와 레이블 변경을 통해 모델의 응용 가능



# 기계학습 기반의 악성코드 탐지 모델 개발

## ❖ 악성코드 분류를 위한 기계학습 모델

- 악성코드 분류 모델 학습을 위한 **데이터 셋** 필요
  - 실제 악성코드 데이터셋은 악용 및 보안사고의 문제로 신뢰성 있는 데이터셋을 확보하기가 매우 어려움
  - 실습을 위해 Kaggle에서 제공하는 Microsoft Malware Classification Challenge (BIG 2015)을 활용
- 악성코드 분류 모델을 선정하기 위한 **모델 선정** 필요
  - 목적 및 상황에 따라 다양한 기계학습 알고리즘을 적용 및 설계 가능
  - 실습에서는 편의를 위해 이전 실습에서 사용하였던 CNN을 사용
- 악성코드 분류 모델 입력 및 성능 개선을 위한 **전처리** 필요
  - CNN 모델의 경우 입력 값으로 이미지 형태의 벡터를 요구함
  - 실습에서 사용하는 BIG2015 데이터셋에 포함된 데이터는 이진수로 동작하는 프로그램 코드이기 때문에 전처리 (변환)이 요구됨



# 기계학습 기반의 악성코드 탐지 모델 개발

## ❖ 악성코드 분류를 위한 실습데이터 생성

### ■ 데이터셋 : Microsoft Malware Classification Challenge (BIG 2015)

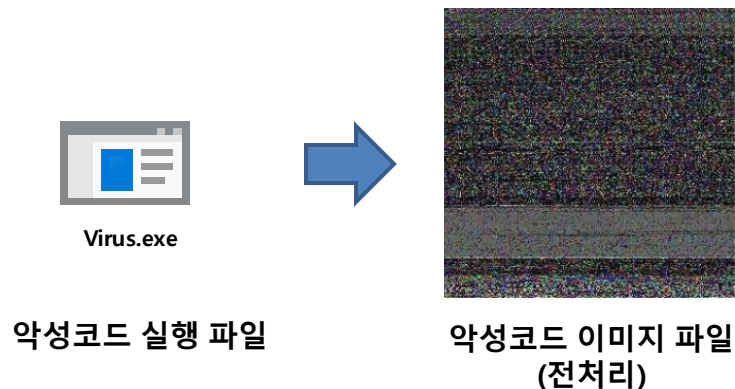
- 9 종류의 악성코드로 구성
- 실제 실행가능한 바이트 파일 제공
- 약 10000여개의 훈련 데이터셋



Microsoft Malware Classification Challenge (BIG 2015) 데이터셋

### ■ 전처리 과정

- 기계학습 모델 설계(CNN, LSTM, GNN 등)에 적합한 데이터 형태의 입력 데이터를 생성함
- 실습에서 사용하는 BIG 2015 데이터는 실제 실행파일을 포함하고 있으므로 CNN 모델에 적합한 이미지로 변환

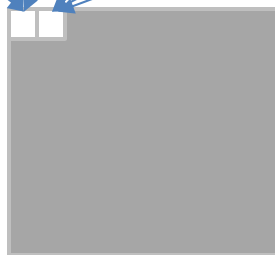


# 기계학습 기반의 악성코드 탐지 모델 개발

## ❖ 입력 데이터 전처리 (1/2)

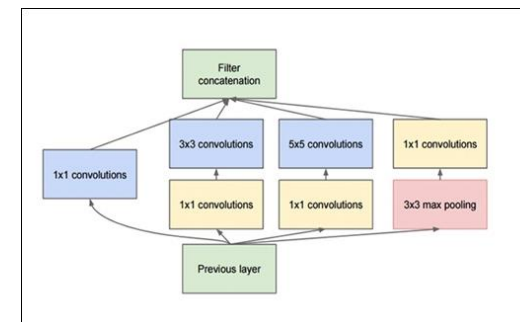
- 악성코드는 바이너리 코드(00~FF)들의 집합으로 구성되어 있음
- 악성코드를 CNN의 입력으로 사용하기 위해, 악성코드 이미지화가 요구됨
- 바이너리 코드(00~FF)는 하나의 픽셀과 RGB 값(00~FF)에 간단히 매칭 가능

55 8B EC 51 51 56



## ❖ 기계학습 모델 설계

- 기계학습의 모델 구조를 문제에 적합한 형태로 변경함으로써 다양한 성능 변화를 얻을 수 있음
- 실습의 경우, 편의를 위해 기존의 모델 구조를 활용



CNN 구조의 예) Inception V3 모델

```
55 8B EC 51 51 56 33 F6 FF 15
90 D0 6E 6D 6A 07 8D 4D F8 51
68 04 10 00 00 50 FF 15 8C D0
6E 6D 85 C0 74 1F 8A 4D F8
```

.....

실제 프로그램의 바이너리 코드 예시

# 기계학습 기반의 악성코드 탐지 모델 개발

## ❖ 입력데이터 전처리 (2/2)

- CNN 모델의 입력 데이터 크기는 같음 Ex.) 28x28x1
- 악성코드의 경우 다양한 크기를 가질 수 있는 프로그램이기 때문에 입력 데이터 크기가 모두 다름 => **변환 필요**
- 일반적으로 악성코드의 크기가 입력으로 **정한 사이즈보다 작을 경우 제로 패딩 (Zero Padding)**을 붙여주고, 클 경우 인코딩을 통해 사이즈를 줄이는 방식을 사용

바이너리 이미지 전처리 코드 (제로패딩 설정)

```
image_size = width * height * 3
if code_len < image_size :
    pad_len = image_size - code_len
    zeroPad = bytes(padLen)
    image = bytearray(BinarySrcCode) + bytes(zeroPad)
```

기존의 바이너리 코드

제로패딩 추가

55 8B EC 51 51 56 33 F6 FF 15

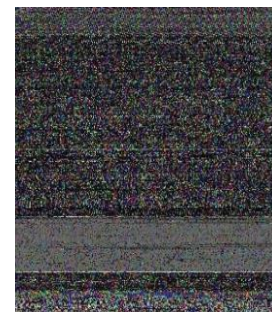
⋮

1F 8A 4D F8 F6 FF 15

+

00 00 00

모델의 입력 크기에 맞추기 위해  
제로 패딩 추가



원본 코드 이미지



인코딩 이미지



# 기계학습 모델 기반의 악성코드 분류 앱 개발

## ❖ 모델 파라미터 변경 및 학습

- 기계학습의 모델 구조를 문제에 적합한 형태로 변경함으로써 다양한 성능 변화를 얻을 수 있음
- (28 x 28 x 1) 크기의 이미지는 28 x 28 x 1 = 784 바이트의 코드 밖에 담지 못함
- 실제 실행 프로그램의 단위인 KB, MB를 고려하였을 때 모델 사이즈 확장이 필요
- 실습에서는 학습 시간을 고려하여, 100 x 100 x 3 = 30000 바이트의 코드를 담을 수 있는 모델 파라미터를 설정

... 8A 4D F8 F6 FF 15 + 00 00 00  
제로패딩 추가

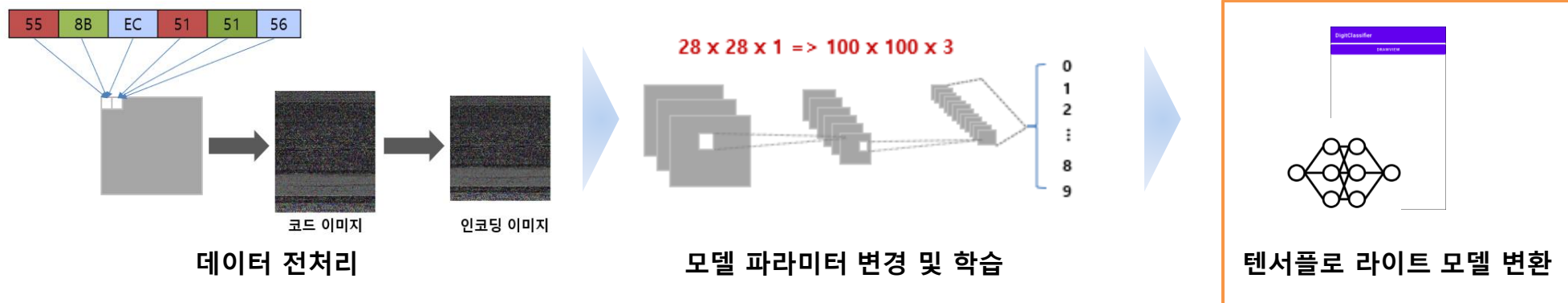
Conv2D(32, (3, 3), input\_shape=(28, 28, 1))



Conv2D(32, (3, 3), input\_shape=(100, 100, 3))

모델의 입력 파라미터 변경

## ❖ 텐서플로 라이트 모델 변환



# 실습

# 실습 목표와 구성

## 1. 기초(따라하기) – 예제 1

- 악성코드 학습 및 모델 변환

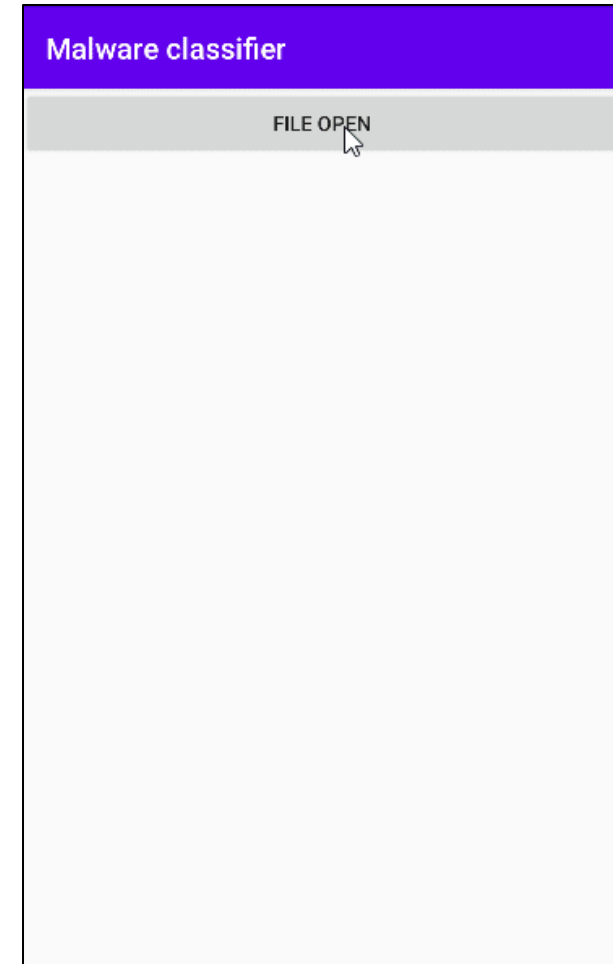
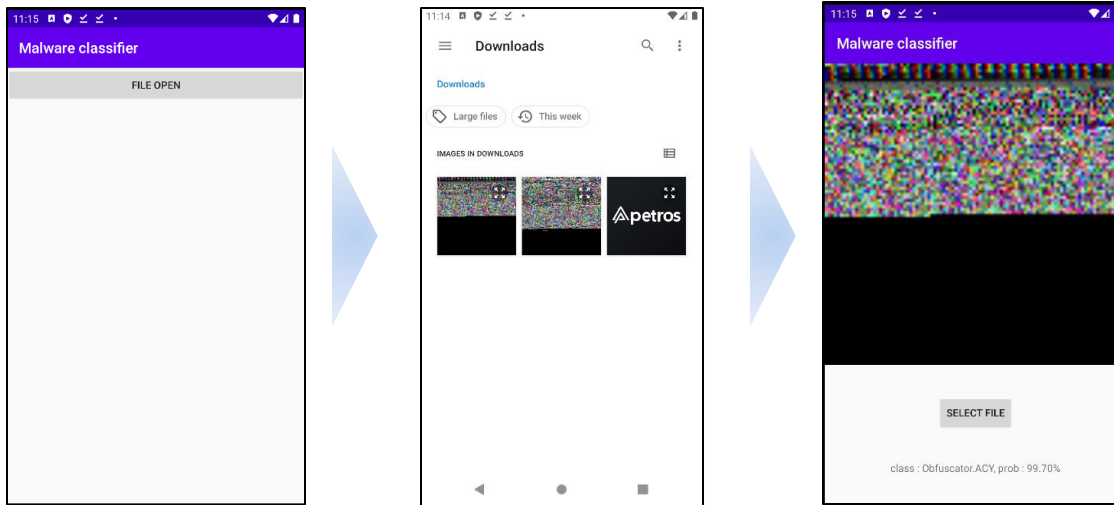
## 2. 응용(로직구현) – 예제 2

- 악성코드 탐지 앱 구현

# 기계학습 모델 기반의 악성코드 탐지 어플리케이션

## ❖ 기계학습 기반의 악성코드 분류 어플리케이션 개발

- File Provider를 통한 악성코드 업로드 페이지 개발
- 텐서플로 라이트 모델을 통한 악성코드 분류 페이지 개발
- 분류 결과 출력 화면 개발



# 기초(따라하기) – 예제 1

## ❖ 악성코드 학습 및 모델 변환

1. 악성코드 전처리 코드
2. 학습 및 텐서플로 라이트 변환 코드

# 기초(따라하기) – 예제 1

## 1. 악성코드 전처리 코드

- (1-9 line) : 이미지 변환을 위한 관련 라이브러리 import
- (11-15 line) : 이미지를 저장하기 위한 변수 선언

malware\_to\_image.py

```
1  from PIL import Image
2  from tqdm import tqdm
3  import binascii
4  import glob
5  import os
6
7  import numpy as np
8
9  from keras.preprocessing.image import img_to_array
10
11  dataCnt = 0
12  dataNum = 9000
13  width = 100
14  height = 100
15  channels = 3
16  dataset = np.zeros(shape=(dataNum , height, width, channels), dtype=np.float32)
```

# 기초(따라하기) – 예제 1

## 1. 악성코드 전처리 코드

- (20-21 line): 악성코드 파일이 저장된 폴더 리스트 로드
- (26 line): 악성코드 폴더 리스트 수 만큼(악성코드 종류 만큼 반복)
- (27 line): 악성코드 폴더 별 파일 리스트 로드
- (28 line): 악성코드 파일 수(리스트) 만큼 반복
- (29-30 line): 파일이 비었을 경우 제외
- (32-33 line): 악성코드 파일 read
- (35-41 line): 여러 라인으로 구성된 바이너리 코드를 한 줄로 병합
- (43 line): string 형식의 바이너리를 hex 형식으로 변환
- (44 line): 소스코드의 길이 계산
- (45 line): 제로패딩을 위한 패딩 길이 계산
- (47 line): 변환 과정 중 오류로 인한 예외처리
- (49 line): 만약 100x100x3을 넘어갈 경우 학습에서 제외

malware\_to\_image.py

```
18 def main():
19     global dataCnt
20     malware_path = './bytes'
21     malware_categories = os.listdir(malware_path+'/')# bin file load
22
23     dataCnt = 0
24     cnt = 0
25
26     for malware_family in tqdm(malware_categories):
27         malware_files = glob.glob(malware_path+'/'+malware_family+'/*.txt')
28         for malware_file in tqdm(malware_files):
29             if not malware_file:
30                 continue
31             try:
32                 f = open(malware_file, 'r')
33                 code = f.readlines()
34                 one_line = ''
35                 for line in code:
36                     line = line.replace("+", "")
37                     line = line.replace("\n", "")
38
39                     if len(line) != 2:
40                         print(str(len(line)) + "### " + line)
41                     one_line = one_line + line
42
43                 BinarySrcCode = binascii.unhexlify(one_line)
44                 code_len = len(BinarySrcCode)
45                 padLen = width * height*3 - code_len
46
47                 if code_len == 0:
48                     continue
49                 if padLen < 0:
50                     print("file is over")
51                     cnt +=1
52                     continue
```

# 기초(따라하기) – 예제 1

## 1. 악성코드 전처리 코드

- (54-55 line) : 소스코드에 제로 패딩 추가
- (56 line) : RGB 이미지 변환
- (57 line) : \*PNG 형식으로 이미지 저장 (반드시 무손실 압축으로 저장)
- (60-63 line) : 예외처리
- 실행 전 해당 경로에 “malware\_images” 폴더와 그 아래 “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9” 폴더들이 존재해야 함

malware\_to\_image.py

```
54     zeroPad = bytes(padLen)
55     image = bytearray(BinarySrcCode) + bytes(zeroPad)
56     img = Image.frombytes("RGB", (height, width), bytes(list(image)))
57     img.save("./malware_images/" + malware_family + "/" + malware_file[17:-4] + ".PNG", 'PNG')
58     dataCnt += 1
59
60     except Exception as ex :
61         print(malware_file)
62         print(ex)
63         return
64
65     print("num of exceeded sol : ", cnt)
66     print("num of total vul : ", dataCnt)
67
68     return
```



# 기초(따라하기) – 예제 1

## 2. 학습 및 텐서플로 라이트 변환 코드

- (1-2 line) : 학습을 위한 관련 라이브러리 import
- (5 line) : 학습할 파일의 경로 설정
- (6 line) : 학습할 파일을 normalization
- (8-18 line) : 학습 및 검증 파일들을 텐서 형태로 불러옴

keras\_model.py

```
1 from keras.preprocessing.image import ImageDataGenerator
2 import tensorflow as tf
3
4 # create generator
5 data_dir = './malware_images/'
6 datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
7 # prepare an iterators for each dataset
8 train_it = datagen.flow_from_directory(
9     data_dir,
10    class_mode='categorical',
11    target_size=(100, 100),
12    batch_size=32)
13
14 val_it = validation_generator = datagen.flow_from_directory(
15    data_dir, # same directory as training data
16    target_size=(100, 100),
17    batch_size=32,
18    subset='validation')
```

# 기초(따라하기) – 예제 1

## 2. 학습 및 텐서플로 라이트 변환 코드

- (20-28 line) : 학습 모델의 레이어 및 파라미터 설정
- (30-32 line) : 모델의 optimizer, loss 함수, metric 설정 후, 모델 컴파일

keras\_model.py

```
20 cnn_model = tf.keras.models.Sequential([
21     tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(100, 100, 3)),
22     tf.keras.layers.MaxPooling2D((2, 2)),
23     tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
24     tf.keras.layers.MaxPooling2D((2, 2)),
25     tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
26     tf.keras.layers.Flatten(),
27     tf.keras.layers.Dense(64, activation='relu'),
28     tf.keras.layers.Dense(9, activation='softmax')])
29
30 cnn_model.compile(optimizer='adam',
31                  loss='categorical_crossentropy',
32                  metrics=['accuracy'])
```

# 기초(따라하기) – 예제 1

## 2. 학습 및 텐서플로 라이트 변환 코드

- (34 line) : 모델 학습을 위한 인자로 학습 데이터 및 epoch 설정
- (35 line) : 검증 데이터 셋을 통한 학습 완료된 모델의 성능 평가
- (38-42 line) : 학습 완료된 모델을 텐서플로 라이트 모델로 변환

keras\_model.py

```
34 cnn_model.fit(train_it, epochs=10)
35 test_eval_result = cnn_model.evaluate(val_it)
36 print(test_eval_result)
37
38 converter = tf.lite.TFLiteConverter.from_keras_model(cnn_model)
39 tflite_model = converter.convert()
40
41 with open('./keras_model_cnn_malware.tflite', 'wb') as f:
42     f.write(tflite_model)
```

# 응용(로직구현) – 예제 2

## ❖ 악성코드 탐지 앱 구현

1. 프로젝트 설정
2. Assets 파일 관리

# 응용(로직구현) – 예제 2

## 1. 프로젝트 설정

- AndroidDraw 라이브러리 사용을 위해, Jitpack 레포지터리 추가

build.gradle (Project: DigitClassifier)

```
15 allprojects {
16     repositories {
17         google()
18         mavenCentral()
19         maven { url 'https://jitpack.io' }
20     }
21 }
22
23 task clean(type: Delete) {
24     delete rootProject.buildDir
25 }
```

- AndroidView 의존성 추가
- 텐서플로 라이트 라이브러리 의존성 추가

build.gradle (Module: DigitClassifier.app)

```
35 dependencies {
36     implementation 'com.github.divyanshub024:AndroidDraw:v0.1'
37     implementation 'org.tensorflow:tensorflow-lite:2.7.0'
38     implementation 'org.tensorflow:tensorflow-lite-support:0.3.0'
39     implementation 'androidx.appcompat:appcompat:1.2.0'
```

- 액티비티 추가

AndroidManifest.xml

```
11 android:theme="@style/Theme.Week14">
12     <activity android:name=".GalleryActivity"/>
13     <activity
14         android:name=".MainActivity"
```

- Sync Project with Gradle Files 를 통한 Gradle 설정 동기화

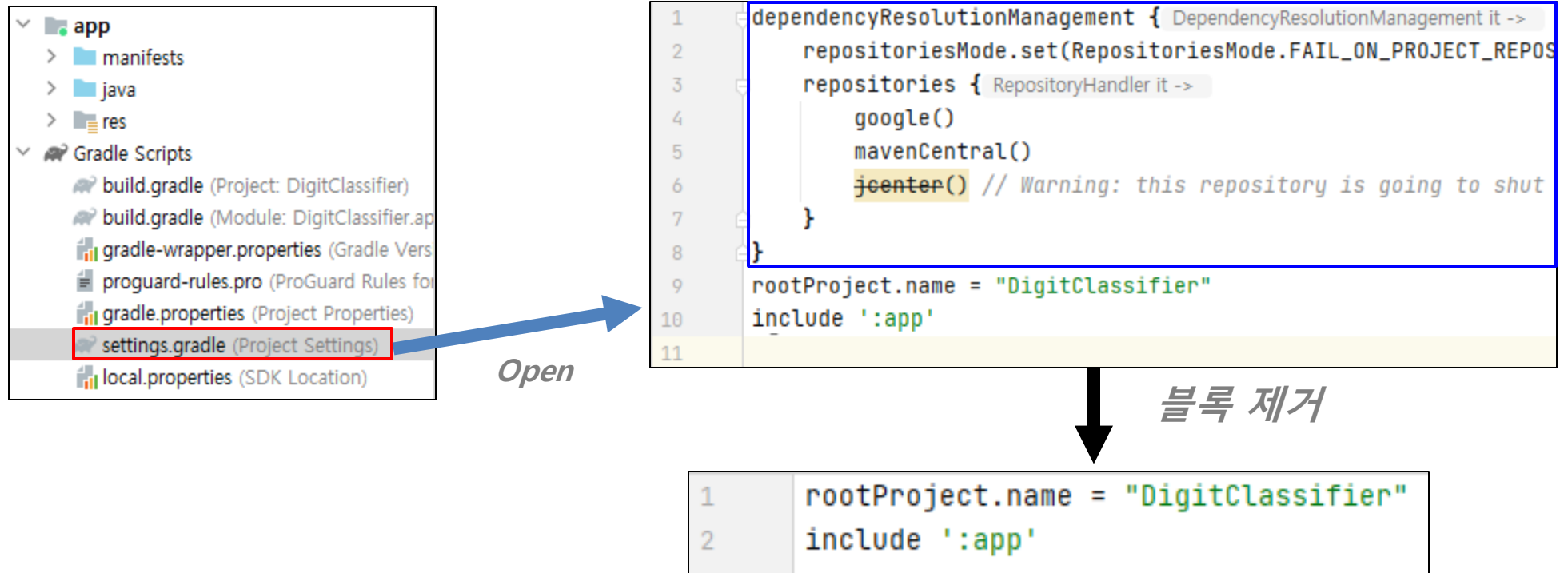


Sync Project with Gradle Files

# 응용(로직구현) – 예제 2

## 1. 프로젝트 설정

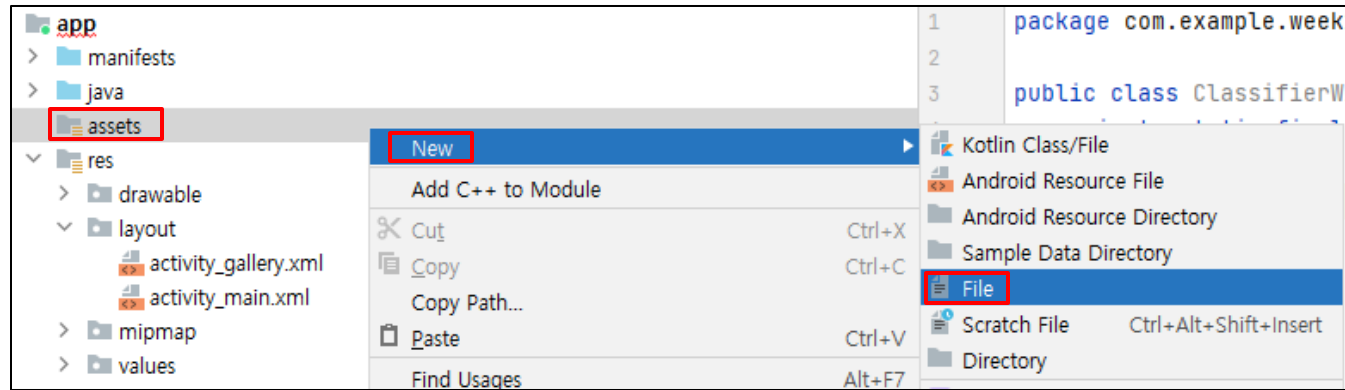
- 에러 발생시 setting.gradle 의 dependencyResolutionManagement 블록 제거
- Sync Project with Gradle Files 재 수행



# 응용(로직구현) – 예제 2

## 2. Assets 파일 관리

- Assets 폴더 생성 후 아래 2개 파일 포함
  - Keras\_model\_cnn\_malware.tflite 복사
  - Labels.txt 생성



The screenshot shows an IDE interface. On the left, a project tree displays the 'app' directory with subfolders 'manifests', 'java', and 'assets'. The 'assets' folder is selected and highlighted with a red box. A context menu is open over the 'assets' folder, with the 'New' option selected and highlighted with a red box. The 'New' submenu is visible, showing options like 'Kotlin Class/File', 'Android Resource File', 'Android Resource Directory', 'Sample Data Directory', 'File' (highlighted with a red box), 'Scratch File', and 'Directory'. To the right of the IDE, a code editor shows a snippet of Java code: 

```
1 package com.example.week
2
3 public class ClassifierW
```

Below the IDE, a 'New File' dialog box is shown with 'labels.txt' entered in the text field and highlighted with a red box. An arrow points from this dialog to the final 'labels.txt' file content.

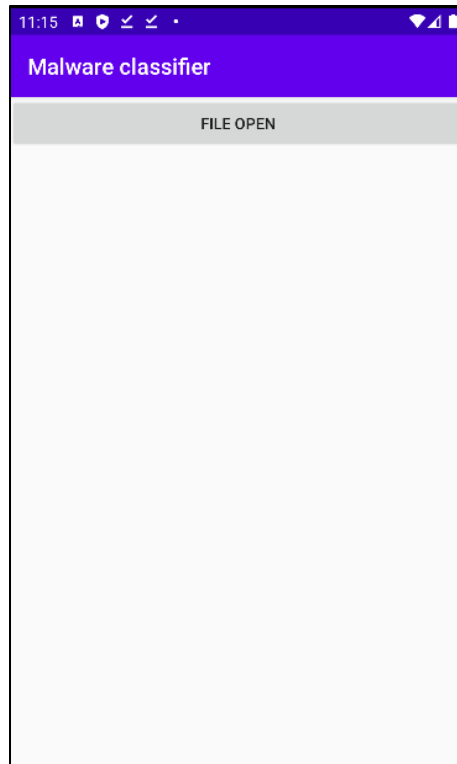
labels.txt

```
1 Ramnit
2 Lollipop
3 Kelihos_ver3
4 Vundo
5 Simda
6 Tracur
7 Kelihos_ver1
8 Obfuscator.ACY
9 Gatak
```

# 응용(로직구현) – 예제 2

## 3. 메인 레이아웃 코드

- (11-15 line) : File Open 버튼 설정



activity\_main.xml

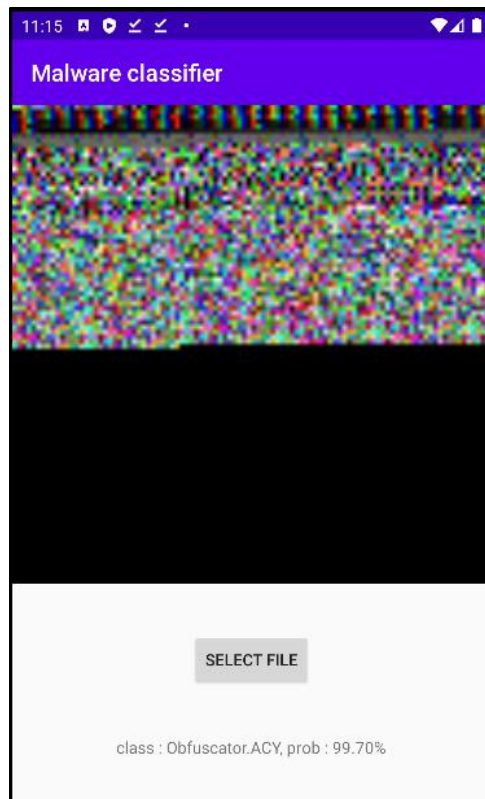
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical"
9     tools:context=".MainActivity">
10
11
12
13
14
15
16
17 </LinearLayout>
```



# 응용(로직구현) – 예제 2

## 4. Gallery 레이아웃 코드

- activity\_gallery.xml 생성
- (10-15 line): 악성코드 이미지를 보여주는 ImageView 배치
- (17-25 line): 실제 파일을 불러오기 위한 버튼 배치
- (27-35 line): 결과를 보여주기 위한 TextView 배치



activity\_gallery.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".GalleryActivity">
9
10     <ImageView...>
11
12
13
14
15
16
17     <Button
18         android:id="@+id/selectBtn"
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="Select File"
22         app:layout_constraintTop_toBottomOf="@id/imageView"
23         app:layout_constraintBottom_toTopOf="@id/textView"
24         app:layout_constraintStart_toStartOf="parent"
25         app:layout_constraintEnd_toEndOf="parent" />
26
27     <TextView...>
28
29
30
31
32
33
34
35
36
37 </androidx.constraintlayout.widget.ConstraintLayout>
```

# 응용(로직구현) – 예제 2

## 5. Classifier 소스 코드

- tflite 패키지 생성 후 ClassifierWithModel 클래스 생성
- (28 line) : 모델이 저장된 tflite 파일
- (29 line) : label이 저장된 txt 파일
- (40 line) : 생성자 정의
- (44 line) : 초기화 함수 정의
- (52 line) : 초기화 상태 플래그 출력 함수

ClassifierWithModel.java

```
27 public class ClassifierWithModel{
28     private static final String MODEL_NAME = "keras_model_cnn_malware.tflite";
29     private static final String LABEL_FILE = "labels.txt";
30
31     Context context;
32     Model model;
33     int modelInputWidth, modelInputHeight, modelInputChannel;
34     TensorImage inputImage;
35     TensorBuffer outputBuffer;
36     private List<String> labels;
37
38     private boolean isInitialized = false;
39
40     public ClassifierWithModel(Context context) { this.context = context; }
41
42
43
44     public void init() throws IOException {
45         model = Model.createModel(context, MODEL_NAME);
46
47         initModelShape();
48         labels = FileUtil.loadLabels(context, LABEL_FILE);
49         isInitialized = true;
50     }
51
52     public boolean isInitialized() { return isInitialized; }
```

# 응용(로직구현) – 예제 2

## 5. Classifier 소스 코드

ClassifierWithModel.java

- (56 line) : 텐서 모델 구조 형성
- (69 line) : 모델 입력 예외처리
- (75 line) : 비트맵 변환 함수

```
56 private void initModelShape() {
57     Tensor inputTensor = model.getInputTensor( inputIndex: 0);
58     int[] shape = inputTensor.shape();
59     modelInputChannel = shape[0];
60     modelInputWidth = shape[1];
61     modelInputHeight = shape[2];
62
63     inputImage = new TensorImage(inputTensor.dataType());
64
65     Tensor outputTensor = model.getOutputTensor( outputIndex: 0);
66     outputBuffer = TensorBuffer.createFixedSize(outputTensor.shape(), outputTensor.dataType());
67 }
68
69 public Size getModelInputSize() {
70     if(!isInitialized)
71         return new Size( width: 0, height: 0);
72     return new Size(modelInputWidth, modelInputHeight);
73 }
74
75 @ private Bitmap convertBitmapToARGB8888(Bitmap bitmap) {
76     return bitmap.copy(Bitmap.Config.ARGB_8888, isMutable: true);
77 }
```

# 응용(로직구현) – 예제 2

## 5. Classifier 소스 코드

- (89 line) : 이미지 전처리 파이프라인 구성

ClassifierWithModel.java

```
79 @ private TensorImage loadImage(final Bitmap bitmap, int sensorOrientation) {
80     if(bitmap.getConfig() != Bitmap.Config.ARGB_8888) {
81         inputImage.load(convertBitmapToARGB8888(bitmap));
82     } else {
83         inputImage.load(bitmap);
84     }
85
86     int cropSize = Math.min(bitmap.getWidth(), bitmap.getHeight());
87     int numRotation = sensorOrientation / 90;
88
89     ImageProcessor imageProcessor = new ImageProcessor.Builder()
90         .add(new ResizeWithCropOrPadOp(cropSize, cropSize))
91         .add(new ResizeOp(modelInputWidth, modelInputHeight, NEAREST_NEIGHBOR))
92         .add(new Rot90Op(numRotation))
93         .add(new NormalizeOp(mean: 0.0f, stddev: 255.0f))
94         .build();
95
96     return imageProcessor.process(inputImage);
97 }
```

# 응용(로직구현) – 예제 2

## 5. Classifier 소스 코드

- (108-109 line): 추론 결과 가져오기

ClassifierWithModel.java

```
99 public Pair<String, Float> classify(Bitmap image, int sensorOrientation) {
100     inputImage = loadImage(image, sensorOrientation);
101
102     Object[] inputs = new Object[]{inputImage.getBuffer()};
103     Map<Integer, Object> outputs = new HashMap();
104     outputs.put(0, outputBuffer.getBuffer().rewind());
105
106     model.run(inputs, outputs);
107
108     Map<String, Float> output =
109         new TensorLabel(labels, outputBuffer).getMapWithFloatValue();
110
111     return argmax(output);
112 }
113
114 public Pair<String, Float> classify(Bitmap image) { return classify(image, sensorOrientation: 0); }
```

# 응용(로직구현) – 예제 2

## 5. Classifier 소스 코드

- (118 line) : 추론 결과 자료 구조 변환

ClassifierWithModel.java

```
118 @ private Pair<String, Float> argmax(Map<String, Float> map) {  
119     String maxKey = "";  
120     float maxVal = -1;  
121  
122     for(Map.Entry<String, Float> entry : map.entrySet()) {  
123         float f = entry.getValue();  
124         if(f > maxVal) {  
125             maxKey = entry.getKey();  
126             maxVal = f;  
127         }  
128     }  
129  
130     return new Pair<>(maxKey, maxVal);  
131 }  
132  
133 public void finish() {  
134     if(model != null) {  
135         model.close();  
136     }  
137 }  
138 }
```

# 응용(로직구현) – 예제 2

## 6. GalleryActivity 소스 코드

- GalleryActivity.java 생성
- (28-30 line) : classifier 및 레이아웃 관련 멤버 변수 선언
- (37-41 line) : 레이아웃 관련 요소와 소스코드 연결

GalleryActivity.java

```
24 public class GalleryActivity extends AppCompatActivity {
25     public static final String TAG = "[IC]GalleryActivity";
26     public static final int GALLERY_IMAGE_REQUEST_CODE = 1;
27
28     private ClassifierWithModel cls;
29     private ImageView imageView;
30     private TextView textView;
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.activity_gallery);
36
37         Button selectBtn = findViewById(R.id.selectBtn);
38         selectBtn.setOnClickListener(v -> getImageFromGallery());
39
40         imageView = findViewById(R.id.imageView);
41         textView = findViewById(R.id.textView);
42
43         cls = new ClassifierWithModel(context, this);
44         try {
45             cls.init();
46         } catch (IOException ioe) {
47             ioe.printStackTrace();
48         }
49     }
```

# 응용(로직구현) – 예제 2

## 6. GalleryActivity 소스 코드

- (52-54 line): 앨범 가져오기


GalleryActivity.java

```
51     private void getImageFromGallery(){  
52         Intent intent = new Intent(Intent.ACTION_GET_CONTENT).setType("image/*");  
53  
54         startActivityForResult(intent, GALLERY_IMAGE_REQUEST_CODE);  
55     }
```



## 6. GalleryActivity 소스 코드

- (61-65 line): 전달된 매개변수 값(처리결과) 검증 및 데이터 검증
- (70-80 line): Bitmap 이미지를 가져오기 위해 관련 함수 사용
- (83 line): bitmap 파일 null 여부 체크
- (84 line): String, Float 형태의 반환 값을 받는 classify 함수 실행
- (85-87 line): String, Float 형태의 반환값을 String 형태로 생성
- (89-90 line): textView 및 imageView에 결과값을 출력

```
57  @Override
58  public void onActivityResult(int requestCode, int resultCode, Intent data) {
59
60      super.onActivityResult(requestCode, resultCode, data);
61      if (resultCode == Activity.RESULT_OK &&
62          requestCode == GALLERY_IMAGE_REQUEST_CODE) {
63          if (data == null) {
64              return;
65          }
66
67          Uri selectedImage = data.getData();
68          Bitmap bitmap = null;
69
70          try {
71              if (Build.VERSION.SDK_INT >= 29) {
72                  ImageDecoder.Source src =
73                      ImageDecoder.createSource(getContentResolver(), selectedImage);
74                  bitmap = ImageDecoder.decodeBitmap(src);
75              } else {
76                  bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), selectedImage);
77              }
78          } catch (IOException ioe) {
79              Log.e(TAG, msg: "Failed to read Image", ioe);
80          }
81
82          if (bitmap != null) {
83              
84
85
86
87          textView.setText(resultStr);
88          imageView.setImageBitmap(bitmap);
89      }
90  }
91
92 }
```

# 응용(로직구현) – 예제 2

## 6. GalleryActivity 소스 코드

- (95 line) : Activity 종료 시 수행 내용 구성

GalleryActivity.java

```
94      @Override
95      protected void onDestroy() {
96          cls.finish();
97          super.onDestroy();
98      }
99  }
```