

모두의 손에 딥러닝 툴을...

전미정, 김태영

2018.02.24.

목차

- iOS와 딥러닝의 만남
- iOS에 딥러닝 심기

- iOS와 딥러닝의 만남
- iOS에 딥러닝 심기

모바일에 왜 ML을 담나요?



User Privacy



Data Cost



Server Cost



Always Available

모바일에서 뭘 할 수 있나요?

Sentiment Analysis

Scene Classification

Predicting Text

Translation

Music Tagging

Handwriting Recognition

DEEP
NEURAL NETWORKS

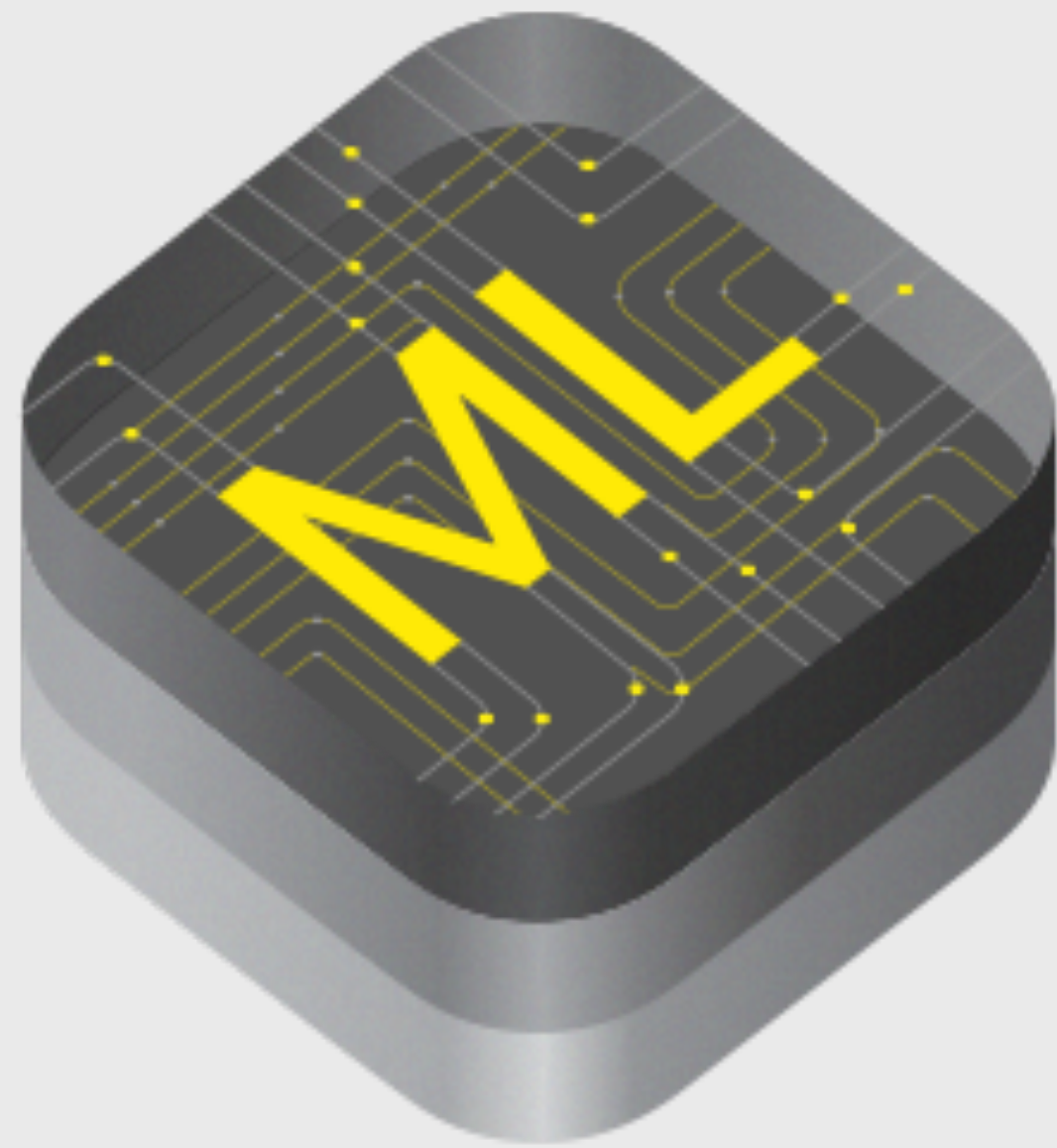
CONVOLUTION
NEURAL NETWORKS

RECURRENT
NEURAL NETWORKS

모바일에서 뭘 할 수 있나요?

<https://coreml.store>

모바일에 어떻게 심나요?



Core ML

- Apple Framework(2017년)
- iOS, watchOS, macOS, tvOS
- iOS 11 이상 지원

Core ML, 뭘 할 수 있나요?

Sentiment Analysis

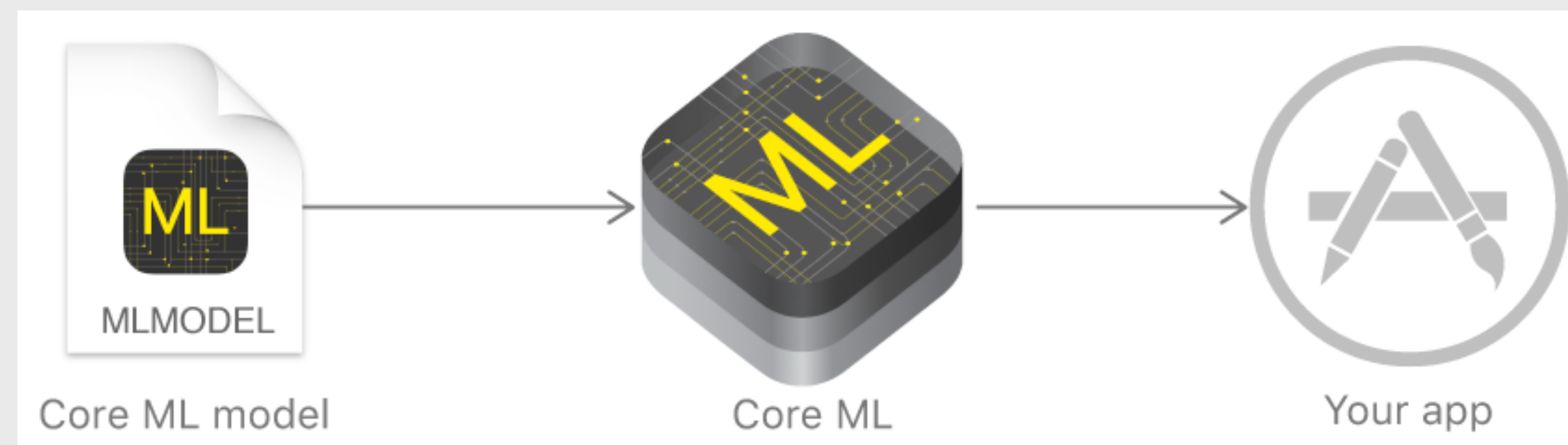
Scene Classification

Predicting Text

Translation

Music Tagging

Handwriting Recognition



Core ML, 장점이 뭐죠?



SIMPLE



PERFORMANT



COMPATIBLE

Core ML, 한계도 있죠?



SIZE



NO TRAINING



CONVERSION

어떤 모델을 사용 할 수있나요?

Caffe

K Keras

dmlc
XGBoost

scikit
learn

turi 

LIBSVM

Core ML Tools

 python™

ML

MLMODEL

- iOS와 딥러닝의 만남
- iOS에 딥러닝 심기

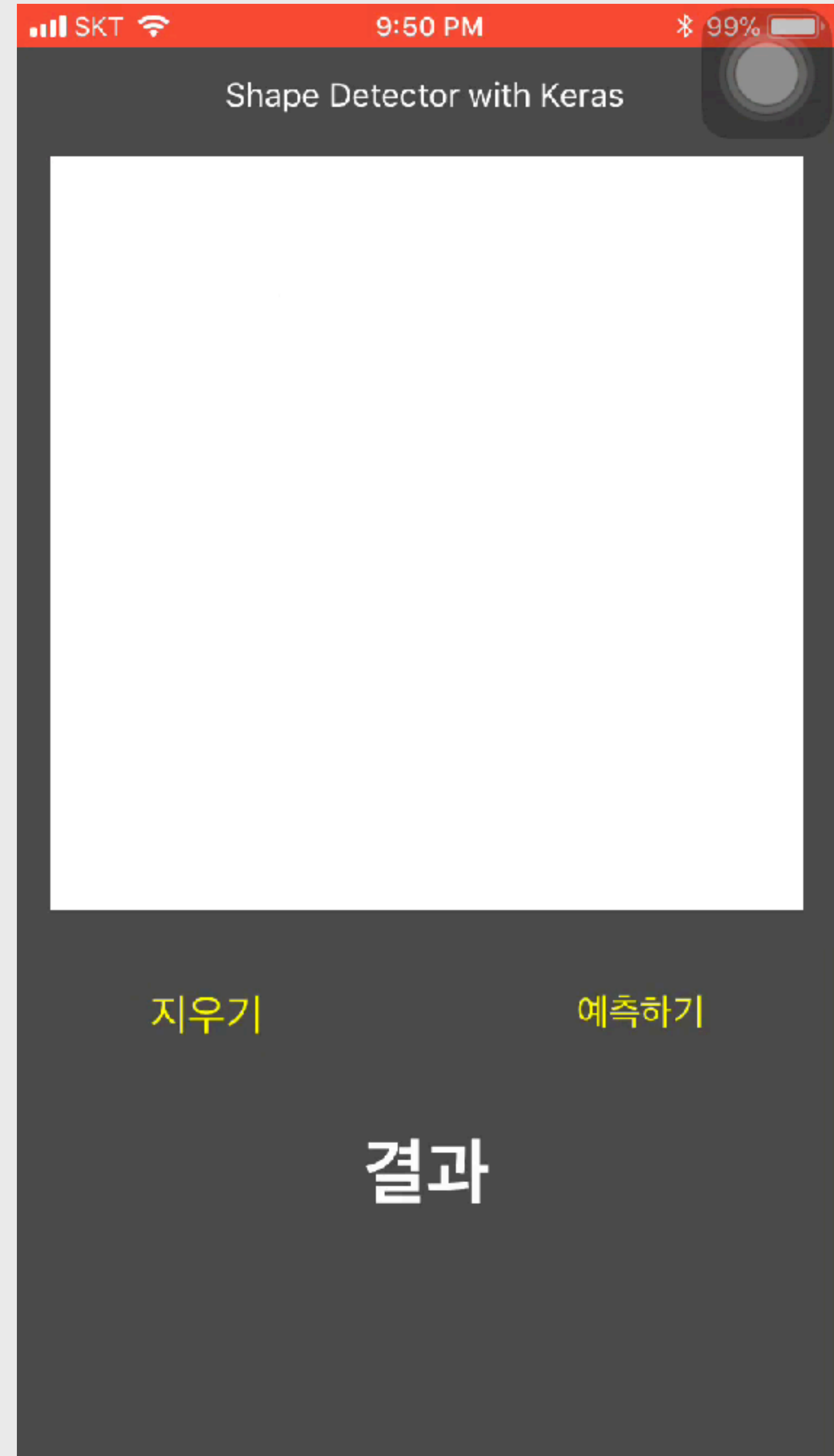
문제 정의

데이터셋 준비

Keras 모델 학습

CoreML Tool

CoreML 적용



문제 정의

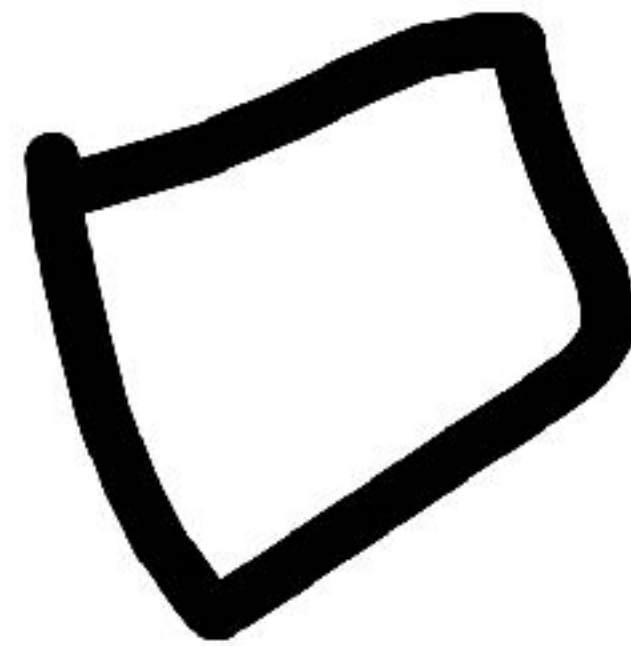
데이터셋 준비

Keras 모델 학습

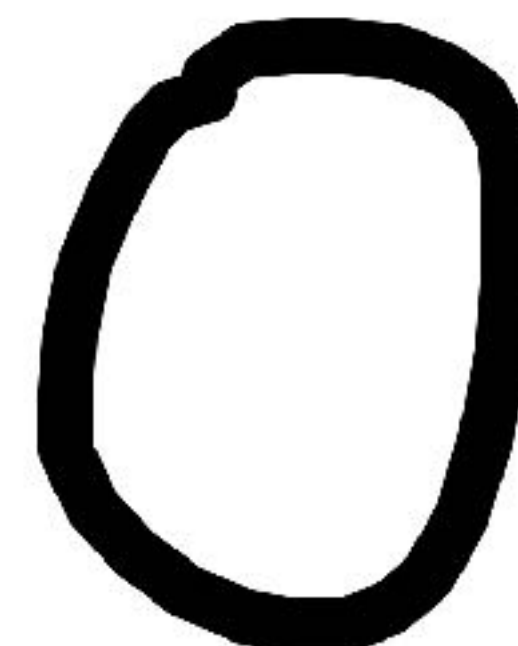
CoreML Tool

CoreML 적용

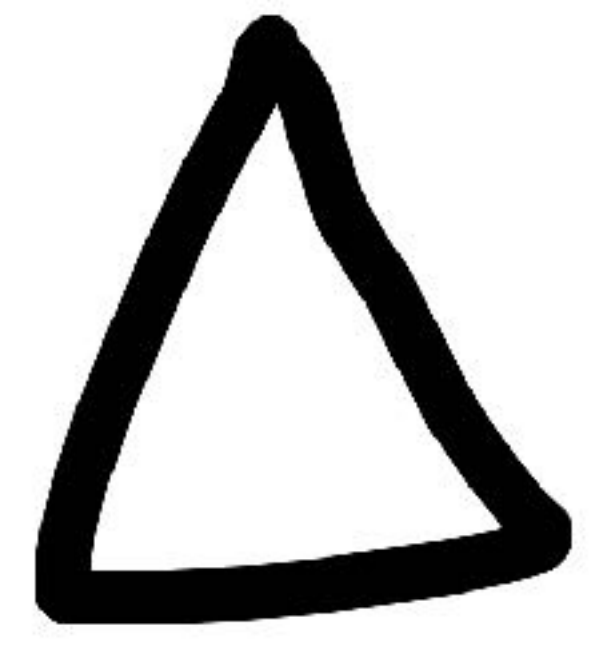
모바일 디바이스에서 손으로 그린 모형을
분류해 String으로 반환하는 어플



Rectangle



Circle



Triangle

만약 알고리즘으로 푼다면...!?

훈련셋, 평가셋 데이터 준비

문제 정의

데이터셋 준비

Keras 모델 학습

CoreML Tool

CoreML 적용



- 28 x 28 pixel
- 훈련셋 각 200장, 평가셋 각 30장

문제 정의

데이터셋 준비

Keras 모델 학습

CoreML Tool

CoreML 적용

웹기반 파이썬 개발환경

jupyter notebook

데이터 포맷 변환

* 여기서부터 python 코드입니다.

문제 정의

데이터셋 준비

Keras 모델 학습

CoreML Tool

CoreML 적용

```
from keras.preprocessing.image import ImageDataGenerator
```

```
# 1. 데이터셋 준비하기
```

```
train_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_directory(
```

```
    'data_set/train', // 이미지 경로 지정
```

```
    target_size=(28, 28), // 이미지 사이즈
```

```
    batch_size=10, // 배치 크기
```

```
    class_mode='categorical') // 분류 방식
```

데이터 부풀리기

문제 정의

데이터셋 준비

Keras 모델 학습

CoreML Tool

CoreML 적용

```
from keras.preprocessing.image import ImageDataGenerator
```

```
# 1. 데이터셋 준비하기
```

```
train_datagen = train_datagen =
```

```
    ImageDataGenerator(rescale=1./255,  
                        rotation_range=10,  
                        width_shift_range=0.2,  
                        height_shift_range=0.2,  
                        shear_range=0.7,  
                        zoom_range=[0.9, 2.2],  
                        horizontal_flip=True,  
                        vertical_flip=True,  
                        fill_mode='nearest')
```

문제 정의

데이터셋 준비

Keras 모델 학습

CoreML Tool

CoreML 적용

2. 영상 모델 구성하기

```
from keras.models import Sequential
```

```
from keras.layers import Dense, Flatten
```

```
from keras.layers.convolutional import Conv2D, MaxPooling2D
```

```
model = Sequential() // 순차 모델 생성
```

```
model.add(Conv2D(32, // 필터 수
```

```
    kernel_size=(3, 3), // 필터 사이즈
```

```
    activation='relu', // 활성화 함수
```

```
    input_shape=(28, 28, 3)) // (입력 이미지 사이즈, 채널 수)
```

```
model.add(Conv2D(64, (3, 3), activation='relu')) // 은닉층
```

```
model.add(MaxPooling2D(pool_size=(2, 2))) // 사소한 변화 무시
```

```
model.add(Flatten()) // 영상 -> 일차원 변환
```

```
model.add(Dense(128, activation='relu')) // 은닉층
```

```
model.add(Dense(3, activation='softmax')) // 출력층
```

문제 정의

데이터셋 준비

Keras 모델 학습

CoreML Tool

CoreML 적용

3. 모델 학습과정 설정하기

```
model.compile(loss='categorical_crossentropy',  
optimizer='adam', metrics=['accuracy']) // 최적화 알고리즘 설정
```

4. 모델 학습시키기

```
model.fit_generator(  
    train_generator, // 훈련셋 지정  
    steps_per_epoch=60, // 총 훈련셋 수 / 배치 사이즈 (=600/10)  
    epochs=150) // 전체 훈련셋 학습 반복 횟수 지정
```

5. 모델 평가하기

```
score=model.evaluate_generator(test_generator, steps=3)
```

문제 정의

데이터셋 준비

Keras 모델 학습

CoreML Tool

CoreML 적용

6. coreml 모델로 변환하기

```
import coremltools
```

```
coreml_model = coremltools.converters.keras.convert(model,  
input_names='image', // 입력 = 이미지  
output_names='class',  
image_input_names = 'image',  
class_labels = ['circle', 'rectangle', 'triangle'], // 출력 라벨  
is_bgr=True)
```

```
coreml_model.save('shape_detect_with_keras.mlmodel') // 저장
```

문제 정의

데이터셋 준비

Keras 모델 학습

CoreML Tool

CoreML 적용

* 여기서부터 Swift 코드입니다.

```
# 7. coreml 적용
```

```
import CoreML
```

```
let model = shape_detect_with_keras()
```

```
guard let output: shape_detect_with_kerasOutput = try?
```

```
model.prediction(image: image) else { return }
```

You have just found Keras

감사합니다.

https://github.com/MijeongJeon/ShapeDetector_Keras_CoreML/
ninevincentg@gmail.com