열 및 통계학 기말고사 2021-06-17

https://github.com/ksimel/digit_classifier

글자인식 어플 만들기

허성우

요 약

본 보고서는 안드로이드 어플 콘텐츠 개발 시 여러 학문의 융합과 게임개발에 관해 개발자의 입장에서 고찰하고자 한다. 스마트폰의 대중화가 가속화됨에 따라 모바일 어플은 다양한 형태의 어플들이 모든 연령의 유저들에게 어필되는 시기가 되고 있다. 본 연구에서는 모바일 어플을 제작하는 기초설립을 설명하고자 한다.

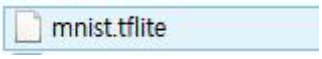
1. 서론

스마트폰이 대중화되기 이전에는 모바일 어플은 매우 단순한 어플에 불과했다. 그러나 스마트폰의 급속한 발전으로 대중화되면서 메신저, 게임, 계산기, 시계 등 편리한 어플이 뛰어나게 발전 되었고 곧, 모바일 어플의 전성시대가 되었다.

시간과 장소에 구애 받지 않고 어디서든 어플을 즐길 수 있게 되면서 어플의 필수적인 요소로 자리 잡았다고 할 수 있다. 본 보고서는 남녀노소 구분 없이 사용할 수 있는 글자인식 어플이다. 손으로 쓴 숫자를 판별해주는 지도학습 유형인 mnist라는 숫자 판별기(digit classifier)딥러닝 모델을 이용해서 이것을 모바일 어플로 제작되는 과정을 보여준다. 우리는 이 모델을 모바일 어플에서 사용할 수 있도록 변환하여 최종적으로 앱에서 숫자를 작성한 후에. 손으로 숫자를 제대로 판별할 수 있는 지를 보여주는 간단한 앱을 만들었다.

2. 본론

본 보고서는 colab을 기반으로 코드를 입력하여 mnist의 tflite파일을 만들고



Java기반으로 Android Studio를 활용하여 어플의 기본 구조를 설계하고 다양한 기능을 접목시켰다. Android Studio 내의 어플로 디버그 시켜 숫자를 판별할 수 있는 어플을 구축하였다.

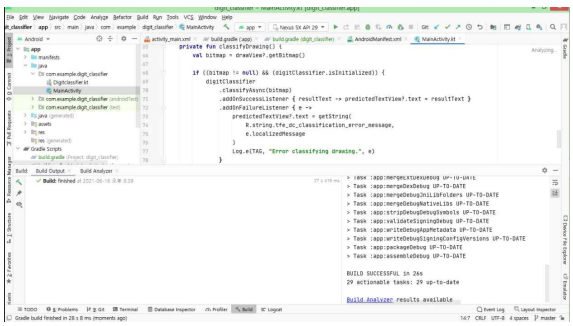
2.1 Mnist.tflite (colab)

Mnist 파일을 만들기 위해 우선적으로 colab에 접속을 한다. import 코드를 입력 후 이미지 화면을 보여주는 함수코드 -show sample을 추가한다. 총 25개의 학습용 샘플을 받게 되는데 딥러닝 모델을 정의하고 컴파일한 후에 학습용 데이터로 실행한다. 모델을 구축하여 테스트 데이터와 정확도를 확인 할 수 있다. 이후 tensiortflow lite 모델로 변환하는 코드를 추가하여 mnist.tflite 파일을 저장한다.

2.2 Android Studio

Android Studio는 Java언어를 이용해 코딩을 하고 어플을 통해서 실행이 가능하다.

Digit Classifier라는 이름으로 kotin프로젝트를 생성한다. app/res/layout/activity_main.xml을 열어 화면에 필요한 코드 https://github.com/tensorflow/examples/tree/master/lite/examples/digit_classifier 에 참조해서 가져온다. 여기서 한가지 안드로이드 스튜디오의 빨간색 코드로 칠해져 있는 것은 고쳐줘야 할 부분이다. 아래의 value/string들도 똑같이 코드를 가져온다. 이후 똑같이 bulid.gradle의 코드를 추가해준다. 추가를 해준 후 코드를 적용하기 위해선 sync now 버튼을 누른다. 위의 2-1에서 만든 mnist.tflite파일을 app/res/raw/main/assts폴더에 파일을 넣어준다. DigitClassfier.kt를 생성해주어 화면이 얼어붙지 않기 위해 코드를 https://developers.google.com/android/guides/setup 에서 가져온다. 이후 main activity.kt 의 코드를 불러와 작성하고 bulid을 해준다. 에러가 뜰 시 해결 방법은 minSOK 16>19로 변경을 해주고, Duplicate class에러가 뜰 시 gradle.propertise에서 android.useAndroidX=true, android.enableJetifier=true를 추가를 해준다. 이러면 성공적으로 Andriod studio의 bulid를 완성 할 수 있다.

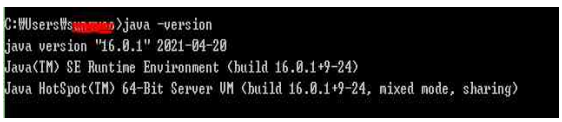


2.3 app

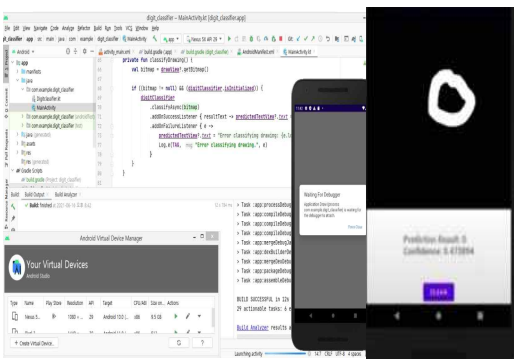
Android Studio 내의 자신의 어플을 test겸 debug시킬 수 있는 툴이 있다. 그 툴을 실행시켜 먼저 최신 버전인 android 10.0.0 target을 만들어줘야 한다. 만들 때 충분한 용량을 확보해준다. 기본적인 툴은 11.1.0이지만 실행할 시 API30이 아직 구현이 되지 않아 에러가 뜨게 된다.

두 번째로 최신버전의 자바를 설치해줘야 한다. http://www.oracle.com/technetwork/java/javase/downloads/index.html?ssSourceSiteId=ocomen 의 링크로 들어가 다운 받은 뒤 제어판으로 들어가 환경변수 설정을 해준다.

Variable name: JAVA_HOME 라고 적고 Variable value: (자신의 JDK 설치경로) 를 적어준다. 여기서 또한 [Path]를 찾아서 [편집]을 클릭하여 ;%JAVA_HOME%\bin; 를 추가적으로 입력해준다. 마지막으로 [System Variables]에서 [New]를 클릭한다. Variable name: CLASSPATH 라고 적고 Variable value: %JAVA_HOME%\lib 를 적어준다.



이제 어플을 실행시킨 뒤 debug를 실행하여 원하던 어플을 만들 수 있게 된다.



3. 결론

본 보고서에서는 안드로이드스튜디오를 기반으로 글자인식 어플을 제작하면서 다양한 기능을 구하고, 각 제작단계별 설명을 했다. 어플에 관심 있는 사람이 코랩의 약간의 기능을 학습한다면 안드로이드기반의 어플 응용콘텐츠를 제작할 수 있음을 보여주고 있다. 본 연구에서는 누구나 부담 없이 즐길 수 있고, 간단한 디자인 및 글자 등을 바꾸어 가면서 즐길 수 있는 글자인식 어플이다. 본 보고서에서는 실제로 숫자판별을 하면서 실제 기능 구사를 할 수 있고, 어떠한 기술이 쓰이는 지를 상세히 기술함으로써 글자 판별을 연구하는 개발자들이 쉽게 참고할 수 있도록 했다. 본 보고서 이후에 확장연구로는 숫자인식이 아닌 이미지 묘사가 가능하고, 기능상으로 이미지 안의 주제를 분류해서 즐길 수 있는 어플 응용콘텐츠를 개발하고자 한다.

참고자료

(1) 동영상 강의
https://www.youtube.com/watch?v=vL0J0Fyfa4k

(2) 안드로이드 소스
https://github.com/tensorflow/examples/tree/master/lite/examples/digit_classifier

(3) drowview (bulid.gradle)
https://github.com/divyanshub024/AndroidDraw
https://tensortflow.org/lite/guide/android
https://developers.google.com/android/guides/setup

(4) JAVA 버전 업데이트
https://t2t2tt.tistory.com/9

(5) github 연동
https://blog.naver.com/beaqon/221278572578