졸업자격실험보고서

머신러닝을 활용한 안면인식

Facial Recognition

using Machine Learning

지도교수 강 동 현

동국대학교 과학기술대학 컴퓨터공학과

노 형 직

2 0 2 0

본 보고서를 졸업자격 실험보고서로 제출함.

2020 년 월 일

노 형 직의 졸업자격 실험보고 통과를 인준함.

2020 년 월 일

주 심

|  |
| --- |
|  |

(인)

부 심

|  |
| --- |
|  |

(인)

부 심

|  |
| --- |
|  |

(인)

동국대학교 과학기술대학 컴퓨터공학과

**목 차**

**1.서론**

**2.OpenCV**

2.1 OpenCV의 정의

**3. 안드로이드 어플리케이션**

3.1 구현 범위

3.2 설계

3.2.1 User InterFace

3.2.2 AsyncTask

3.2.3 얼굴 촬영

**4. 라즈베리파이**

4.1 구현 범위

4.2 결과 수신 및 제어

**5. 서버와 클라이언트**

5.1 서버 구축

5.2 서버 구성파일

**6. 결론**

**참고문헌**

**표 목 차**

[표 5-1] AWS EC2 t2.micro 사양17

**그 림 목 차**

[그림 2-1] OpenCV2

[그림 3-1] 어플리케이션 UI

[그림 3-2] AsyncTask의 문법

[그림 3-3] 카메라 설정과 인터넷 전송 허용

[그림 4-1] 라즈베리파이와 구성품의 연결

[그림 4-2] 실제 구성 모습

[그림 4-3] 결과를 수신받는 receive.py파일

[그림 5-1] 서버와 클라이언트의 구성 요소

[그림 5-2] 서버에 업로드된 폴더와 파일

**1. 서론**

인공지능 분야에 관심이 있어 공부를 하던 도중, OpenCV를 통해이미지를 학습시키는 방법에 대해 흥미를 느꼇다. 이제는 흔하게 찾아보지만, 직접 머신러닝에 대해 이해하고 그것을 구현하고 싶었다.

최근 '알파고'가 화재가 되어서 머신러닝이 대중들에게 알려지게 된것만은 아니다. 과거에는 하드웨어의 낮은 사양으로 인해서 복잡한 연산에 오랜 시간이 걸렸지만, 원활한 GPU를 바탕으로 전체적인 하드웨어 사양이 높아지며 이에 따른 연산 시간이 줄어들게 되었다. 또한, 인터넷의 발달로 인해 과거와는 다르게 학습에 필요한 데이터를 구하는 것이 수월해졌다. 위와 같은 과정들로 인해서 머신러닝은 대중들에게 알려지게 됐다.

본 연구에서 다루는 것은 머신러닝을 활용한 안면인식 기능이다. 머신러닝을 통해서 사용자의 얼굴을 학습하고 비교하여 안면인식기능을 구현한다. 구현된 안면인식기능을 라즈베리파이와 제공되는 안드로이드 어플리케이션을 통해서 얼굴을 판별 할 수 있다. 이는 다양한 환경에서 보안이나 가정 시스템에서도 훌륭한 방법으로 쓰임새가 많을 것이라 단언한다.

본 논문의 구성은 2장에서 OpenCV에 대한 설명과 구현에 대해서 보여주며, 3장에서는 어플리케이션의 설계에 대해서 설명한다. 4장에서는 라즈베리파이의 설계에 대해서, 5장에서는 서버와 클라이언트사이의 통신을 위한 서버설계에 대해서 설명하며, 마지막 장에서는 달성한 연구 목적과 문제점, 개선점을 설명하며 마친다.

**2. OpenCV**

2.1 OpenCV의 정의

Open Source Computer Vision의 약자로, 다양한 영상 및 동영상 처리에 사용 할 수 있는 오픈소스 라이브러리를 말한다.

이 라이브러리에는 성능과 속도면에서 뛰어난 정확도를 가진 2500개 이상의 최적화된 알고리즘이 포함되어 있으며, 물체를 식별하고 얼굴을 감지 또는 인식하는등 많은 분야에서 사용되고 있다

|  |
| --- |
|  |

[그림 2-1] OpenCV

2.2 구성

|  |
| --- |
|  |

[그림 2-2] OpenCV의 이미지 서버 전송

**3. 안드로이드 어플리케이션**

3.1 구현 범위

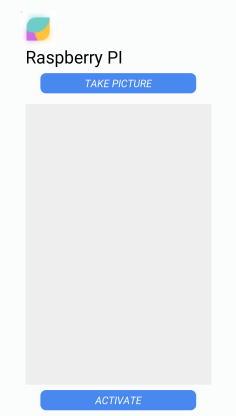
본 연구에서는 사용자의 얼굴 사진을 촬영해서 서버로 전송하는 어플리케이션이 필요하다. 안드로이드에서 인터넷을사용하기 위해서는 AsyncTask를 사용하여 메인스레드가 아닌 백그라운드에서 작업을 하게 설정한다.

(1) 직관적으로 알기 쉬운 UI

(2) 메인 스레드의 부담을 줄이기 위해 AsyncTask 사용

(3) 사용자의 얼굴 촬영을 위해서 카메라 기능 사용

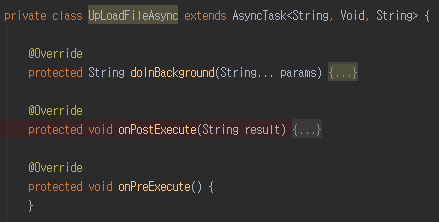
3.2 설계

 3.2.1 User Interface

[그림 3-1] 어플리케이션 UI

3.2.2 AsyncTask

안드로이드에서 인터넷을 사용할 때는 메인 스레드의 부담을 줄이기 위해서 백그라운드에서 작업하는 AsyncTask를 사용해야 한다.



[그림 3-2] AsyncTask의 문법

AsyncTask는 크게 3가지의 필드를 갖고 있다.

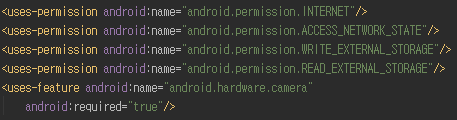
onPreExecute()는 AsyncTask를 사용하기 전 해야 할 작업에 대해서 설정한다. 사용자에게 입력 정보를 받는다거나 작업을 시작하기 전 사용자에게 알림으로 알려주는 행동을 할 수 있다.

doInBackground()는 onPreExecute()의 실행이 끝나면 바로 실행된다. 이는 메인 스레드가 아닌 백그라운드에서 서버와 통신을 하는 등의 작업을 할 수 있다.

onPostExecute()는 서버에서 받은 값 등을 사용자에게 적절하게 변환해서 알려주는 작업을 할 수 있다. 위 세가지 필드를 통해서 메인스레드의 부담을 줄여주는 AsyncTask를 사용할 수 있다.

3.2.3 얼굴 촬영

얼굴을 촬영하기 위해서 카메라 기능을 사용해야 한다. 안드로이드에서는 기본으로 카메라 사용이 꺼져있기 때문에 이를 활성화 해줘야한다. 안드로이드의 기본적인 권한을 설정하는 파일인 AndroidManifest.xml파일을 통해서 카메라사용을 허용해준다. 또한 촬영한 사진을 서버로 전송해야 하기 때문에 데이터를 전송하는 권한도 허용해 준다.



[그림 3-3] 카메라 설정과 인터넷 전송 허용

**4. 라즈베리파이**

|  |
| --- |
|  |

4.1 구현 범위

본 실험에서는 서버에서 진행된 딥러닝의 결과를 받아 출력할 라즈베리파이가 필요하다. 라즈베리파이는 서버에서 머신러닝을 통해서 예측한 결과를 받아서 출력하는 역할을 수행한다.

[그림 4-1] 라즈베리 파이

|  |
| --- |
|  |

4.2 결과 수신 및 제어

라즈베리파이에서 결과를 수신받기 위해서는 목적에 맞는 py파일이 필요하다.

[그림 4-2] reception.py

[그림 4-2]의 코드는 주기적으로 서버에 있는 json파일을 읽는다. 읽은 파일의 timestamp가 변하게 되면 파일을 받아서 확인한다. json파일에는 머신러닝 연산 결과가 담겨 있고 그것을 출력한다.

**5. 서버와 클라이언트**

5.1 서버 구축

서버는 어플리케이션과 라즈베리파이 사이의 통신을 위해서 필요하다. 본 연구에서는 AWS(Amazon Web Service)에서 제공하는 EC2(Elastic Compute Cloud)를 사용하여 서버를 구축하였다. EC2에서 사용할 수 있는 많은 어플리케이션 중 본 연구에서는 LAMP스택을 사용한다. LAMP스택은 Linux운영체제 위에서 Apache서버를 사용하며 MySQL데이터베이스와 PHP를 사용한다. 서버의 인스턴스 유형은 t2.micro를 사용하며 자세한 항목은 [표5-1]로 확인할 수 있다.

[표 5-1] AWS EC2 t2.micro 사양 [5]

|  |  |
| --- | --- |
| CPU | 인텔 AVX 터보 1코어 3.3GHz |
| Memory | 1 Gib |
| Storage | EBS 전용 |
| Network | 낮음에서 중간 |



[그림 5-1] 서버와 클라이언트의 구성 요소

|  |
| --- |
|  |

5.2 서버 구성파일

서버의 역할은 안드로이드 어플리케이션에서 보낸 이미지를 저장하고 이를 바탕으로 학습된 모델을 이용하여 결과를 예측해야 한다. 서버에 업로드된 폴더와 파일은 [그림 5-2]에서 확인할 수 있다.

[그림 5-2] 서버에 업로드된 폴더와 파일

upload폴더에는 안드로이드 어플리케이션에서 보낸 사진이 저장된다. 이 사진은 계속 저장되어 쌓이지 않고, 결과를 예측한 후 삭제되어 사용자의 개인정보가 남아있지 않게 한다.

**6. 결론**

본 연구는 머신러닝을 활용하여 사용자의 얼굴사진을 학습하고 안드로이드 어플리케이션을 이용하여 실시간으로 촬영되어 서버로 전송되는 사진을 비교하여 결과를 예측한다. 예측한 결과값을 서버에 기록하고 라즈베리파이는 기록된 결과값을 받아와서 출력한다. 모든 시스템을 구축한 결과 목적에 맞는 성능을 보여준다.

하지만 학습데이터의 양이 부족한 만큼 취약한 부분 또한 파악할 수 있다. 사용자의 얼굴이 절반이상 가려있거나, 촬영한 장소의 밝기 정도에 따라서 인식률이 떨어지는 결과를 얻었다. 또한 서버의 메모리가 부족하여 결과를 예측하지 못하는 경우가 생기기도 했다.

위 두가지의 문제점은 해결할 수 있는 방법이 있다. 전자의 경우 학습데이터의 양을 늘리면 된다. 하지만 이는 사용자에게 학습을 위해서 촬영을 강요하게 되고 현 수준에서는 40장이 가장 적절한 수준이라고 생각하여 본 연구에서는 40장으로 선택했다. 후자의 경우 서버의 사양을 늘리면 해결할 수 있지만 비용의 문제로 인해서 사양을 늘리지는 못하고 서버를 재부팅시키는 방향으로 설정했다.

본 시스템을 완성시키면서 생각한 앞으로 개선시켜야 할 부분이 몇가지 있다. 서버에서 머신러닝을 이용해서 결과를 예측하는 시간이 현재 평균 5s ~ 7s 정도 기록된다. 이 시간을 줄일 수 있는 CNN 모델의 최적화가 필요하다. 이 부분에 대해서는 추후 연구를 통해서 보완해야 한다.

**참고문헌**

[1] 한국정보통신기술협회, 정보통신용어사전

<http://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=104374-2>

[2] 한국정보통신기술협회, 정보통신용어사전

<http://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=138910-8>

[3] 프랑소와 숄레, 『케라스 창시자에게 배우는 딥러닝』, 박해선(역), 길벗(2018), p75-p77

[4] 김태영, 『블록과 함께하는 파이썬 딮러닝 케라스』, 디지털북스(2017), p42

[5] Amazon Web Serveice, Amazon EC2 인스턴스 유형

<https://aws.amazon.com/ko/ec2/instance-types/>