**OpenCV를 활용하여, 얼굴인식을 통한 유사도 측정**

|  |  |
| --- | --- |
| 과목 : | 오픈소스SW프로젝트 |
| 분반 : | 01 |
| 교수 : | 나인섭 교수님 |
| 학과 : | 컴퓨터공학과 |
| 학번 : | 20165062 |
| 이름 : | 이윤혁 |

**개발 환경 : Windows 10 Home**

**개발 언어 : Jupyter Notebook(5.7.4)-Python(3.6.6)**

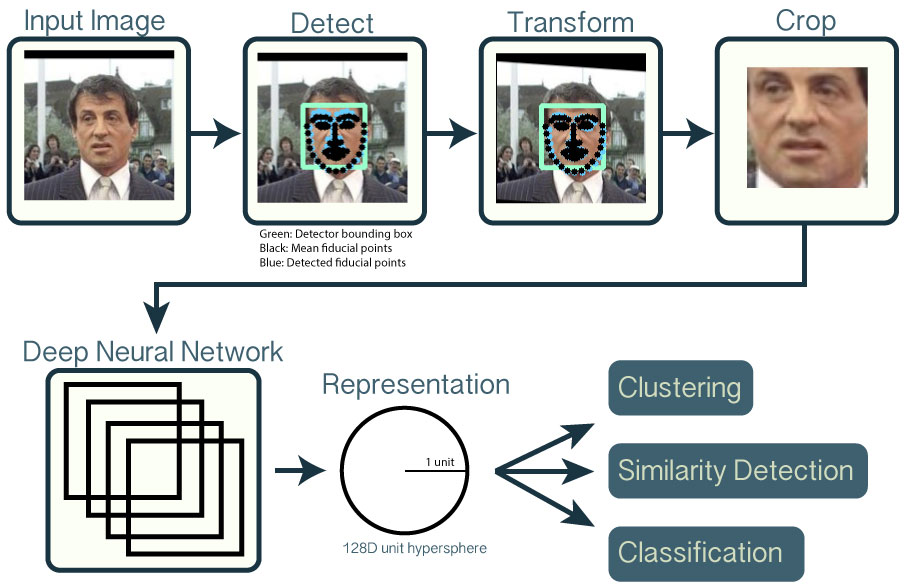
**오픈 소스 : OpenCV(3.4.5-BSD), TensorFlow(1.10.0-Apache2.0)**

***아파치 2.0 오픈소스 라이센스****는 누구나 해당 소프트웨어에서 파생된 프로그램을 제작할 수 있으며 저작권을 양도, 전송할 수 있는 라이센스 규정을 의미한다. 아파치 라이센스에 따르면 누구든 자유롭게 아파치 소프트웨어를 다운 받아 부분 혹은 전체를 개인적 혹은 상업적 목적으로 이용할 수 있으며 재 배포시에는 원본 소스 코드 또는 수정한 소스 코드를 반드시 포함시켜야 하는 것은 아니고 아파치 라이센스 버전 2.0을 포함시켜야 하며 아파치 소프트웨어 재단에 개발된 소프트웨어라는 것을 명확하게 밝혀야 한다.*

***BSD(Berkeley Software Distribution) 라이센스****는 소프트웨어 라이선스 라고도 할 수 없을 만큼 미약하여, 해당 소프트웨어는 아무나 개작할 수 있고, 수정한 것을 제한 없이 배포할 수 있습니다. 수정본의 재배포는 의무적인 사항이 아니므로 BSD 라이센스를 갖는 프로그램은 공개하지 않아도 되는 상용 소프트웨어에서도 사용할 수 있습니다.*

**OpenCV(Open Source Computer Vision Library - 3.4.5)**

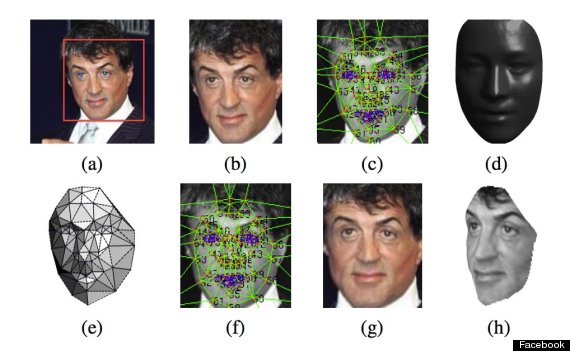
* 수백 개의 컴퓨터 비전 알고리즘(이미지 프로세싱, 비디오 분석, 카메라 캘리브레이션 및 3D 재구성, 2D 기능 프레임 워크, 객체 감지, 고수준 GUI, 비디오 I/O 등…)을 포함하는 오픈 소스 BSD 라이센스 라이브러리 이다.
* **OpenFace(Apache 2.0)**



**Detect (인식단계) :** 입력 이미지에서 얼굴을 찾는 단계이다. OpenCV를 통해 학습된 모델을 이용하여 얼굴을 인식한다. 이 때, 사용하는 알고리즘은 HOG(Histogram of Oriented) 알고리즘을 사용한다**.**

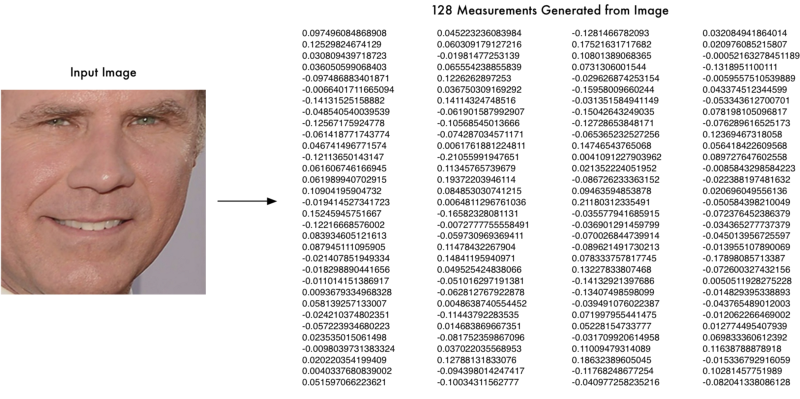
**Transform(변환) & Crop(자르기) :** 입력 이미지에서 인식된 얼굴 이미지를 이미지 분류기 학습에 사용할 수 있도록 표준화 하는 단계다.

Ex) 대표적인 표준화 예시. 페이스북의 DeepFace



동일한 사람이더라도, 정면과 측면에서 보는 모습은 다르다. 컴퓨터가 동일한 사람으로 분류하도록 학습을 하기 위해서는 눈, 코, 입이 사진의 동일한 위치로 오도록 변환해야 한다. 얼굴 특징 점(눈, 코, 입)을 찾는 알고리즘을 얼굴 특징 점 추정(Face Landmark Estimation) 이다. 올바른 배치로 변환하는 알고리즘은 OpenCV의 아핀 변환 알고리즘(2D 변환)을 이용한다.

**Representation(수치화) :** 변환된 얼굴 사진을 학습된 DNN(Deep Neural Network) 모델을 기반으로 수치화하는 단계다.



* DNN 모델 : 입력 이미지를 128차원 숫자값으로 변환할 수 있는 모델이다.
* 분류 모델 : 128차원 숫자값을 입력으로 사용하여, 얼굴을 분류할 수 있는 모델이다. 일반적인 분류 학습 알고리즘을 사용할 수 있다.

**TensorFlow(1.10.0)**

* 데이터 흐름 그래프를 사용하는 수치 연산용 오픈소스 소프트웨어 라이브러리입니다.
* 수학 계산과 데이터의 흐름을 노드(Node)와 엣지(Edge)를 사용한 방향 그래프(Directed Graph)로 표현합니다.
* 노드는 수학적 계산, 데이터 입/출력, 그리고 데이터의 읽기/저장 등의 작업을 수행한다.
* 엣지는 노드들 간 데이터의 입출력 관계를 나타낸다.
* **특징**
* 데이터 플로우 그래프를 통한 풍부한 표현력
* 코드 수정 없이 CPU/GPU 모드로 동작
* 아이디어 테스트에서 서비스 단계까지 이용 가능
* 계산 구조와 목표 함수만 정의하면 자동으로 미분 계산을 처리

**사용할 DB 설명 :**

**상위 방법 :**

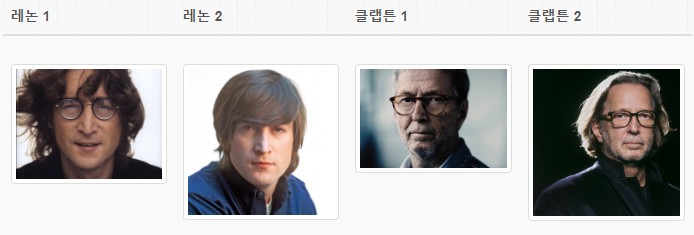
**전체 흐름도(구성도) :**

1. **필요성 및 기대효과 :**

**우리가 영화를 볼 때, 흔히 CCTV를 통해 특정한 인물을 찾거나, 마스크나 모자를 쓴 범인들을 찾는 장면을 볼 수 있다. 실제 범죄자들은 마스크와 모자를 착용 함으로서, 얼굴을 보이지 않으려 하기 때문에 얼굴인식을 하기 어렵다. 이를 통해 얼굴의 윤곽이나, 다양한 얼굴 정보들을 가지고 비교를 함으로서 유사한 인물을 찾는다면, 범죄예방이 가능 할 것입니다.**

1. **기존 연구 :**
2. 두 이미지 비교

* 비교 데모를 통해 그들의 표현 간의 제곱거리를 산출하여 양면의 예측 유사성 스코어를 출력한다. 이 때, 낮은 점수는 두 사람의 얼굴이 같은 사람의 가능성이 높다는 것을 나타낸다. 점수는 0~4로 부여.





거리는 두 사람을 구별 할 수 있는 거리의 임계 값을 보여준다.

1. **제안 방법 :**
2. **실험 및 결과(비교실험) :**
3. **고찰(토론) : 실패 케이스와 그 이유….**
4. **참고 문헌 :**

[1] 저자명, “제목”, 출처(출판사 등), 판차, 년도.

1. **느낀점 :**

* 기존연구에서 소개된 내용을 실험에서 비교하기
* 참고한 자료가 있는 곳 문장 끝에 참고 문헌에 기록된 번호를 기입하기