

Seloco, Inc.

얼굴인식 모듈 자료조사

작성일자 : 2021-02-10

인턴 송진호

인턴 이우림

Index

01 Face Recognition

02 OpenFace

Face Recognition 소개

- ✓ https://github.com/ageitgey/face_recognition
- ✓ 세계에서 가장 간단한 얼굴 인식 라이브러리로 알려져 있으며, Python 또는 CLI를 통해서 얼굴을 인식하고 조작할 수 있습니다.
- ✓ 딥러닝 기반으로 구축된 dlib의 최첨단 얼굴인식 기능을 사용하여 빌드됩니다. 이 모델은 'Labeled Faces in the Wild' 벤치마크 기준으로 99.38%의 정확도를 가집니다.

01

Face Recognition 기능

- 이미지로부터 얼굴 찾기



Input



Output

01

Face Recognition 기능

- 이미지로부터 얼굴의 특징을 찾고 조작하기



01

Face Recognition 기능

- 이미지로부터 얼굴의 윤곽을 찾고 디지털 메이크업



Face Recognition 기능

- KNN(K-Nearest Neighbors) 알고리즘을 이용한 이미지 속 인물 신원확인

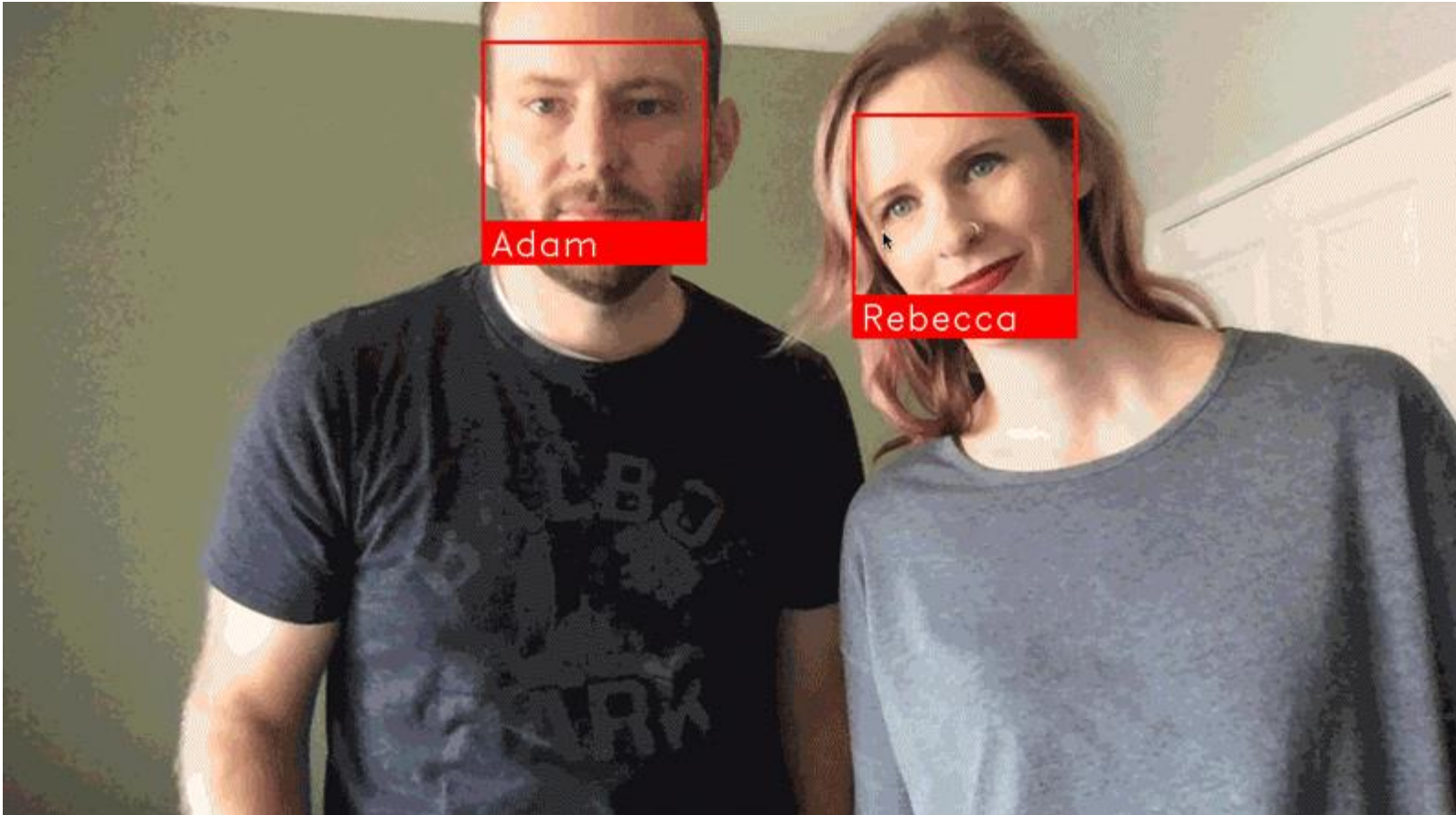
- ✓ KNN 알고리즘은 머신러닝에서 사용되는 분류(Classification) 알고리즘이다.

- ✓ 유사한 특성을 가진 데이터는 유사한 범주에 속하는 경향이 있다는 가정하에 사용한다.



Face Recognition 기능

- 웹캠을 이용한 실시간 얼굴 인식 및 신원확인

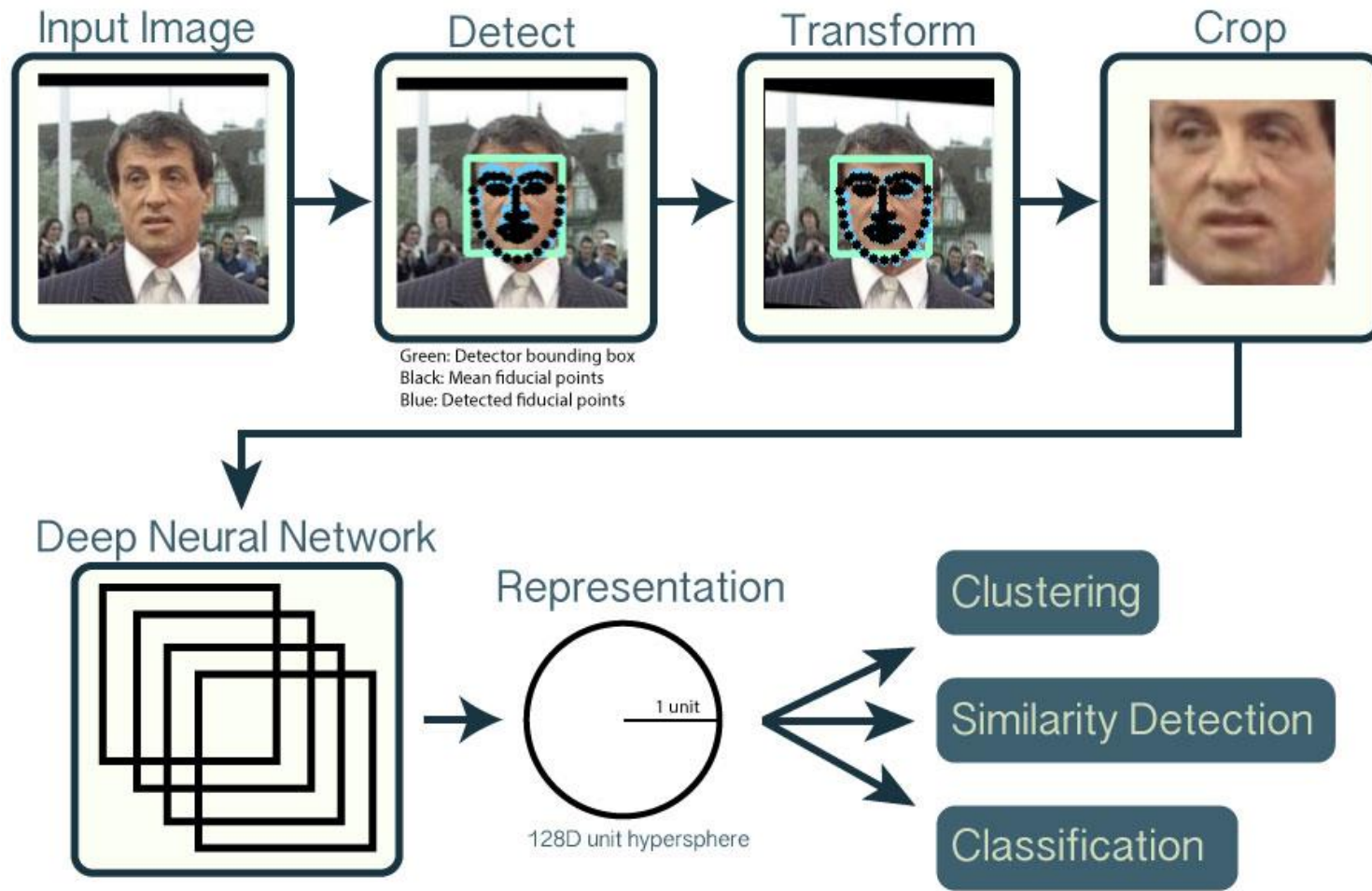


OpenFace

- ✓ <https://github.com/cmusatyalab/openface>
- ✓ 심층 신경망을 기반으로 얼굴 인식을 구현한 무료 오픈소스 라이브러리
- ✓ Google의 Florian Schroff, Dmitry Kalenichenko, James Philbin이 작성한 CVPR 2015 논문 ‘FaceNet : 얼굴 인식 및 클러스터링 을 위한 통합 임베딩’을 기반으로 합니다.

OpenFace

- Pipeline : 이미지로부터 얼굴을 인식하기 위한 과정



OpenFace

- 인식(Detect)

- ✓ 입력 이미지로부터 얼굴을 찾는 단계입니다.
- ✓ 기본적으로 HOG(Histogram of Oriented) 알고리즘이 적용된 모델을 사용하였지만 성능이 좋지 않습니다.
- ✓ HOG 알고리즘이 적용된 모델을 사용하는 방법 대신에 OpenCV 또는 dlib를 통해 기학습된 모델을 사용하여 얼굴을 인식하는 것으로 문제점을 해결할 수 있습니다.

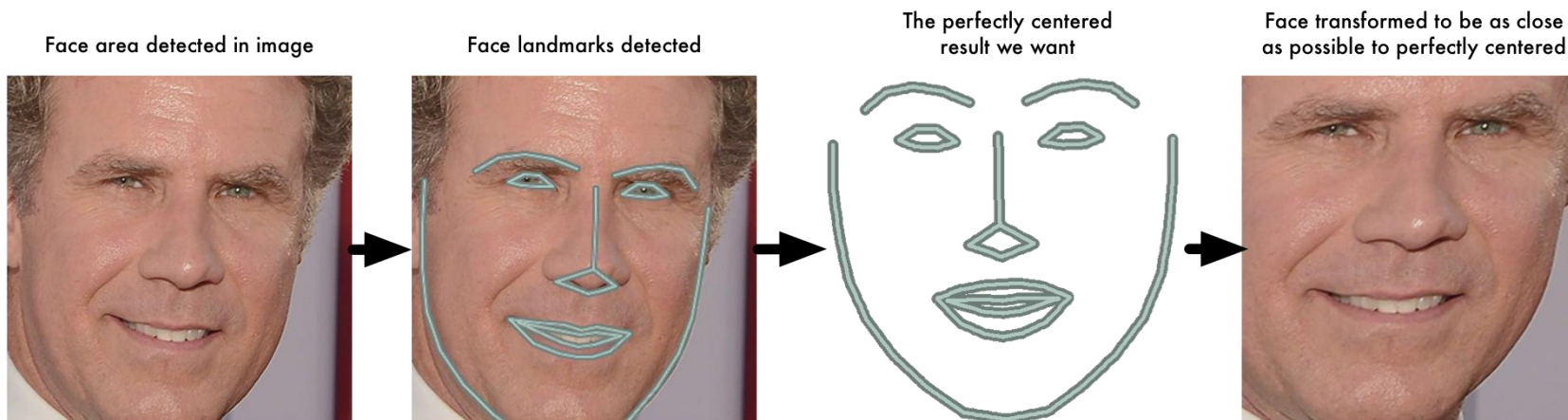
- HOG(Histogram of Oriented Gradients)

- ✓ Object-Tracking에 많이 사용되는 기능 중 하나로, 이미지의 지역적인 Gradient 방향 정보를 히스토그램으로 나타내서 영상의 형태를 표현하는 방법입니다.
- ✓ 전체 영상으로부터 부분적인 영상의 특징을 추출하여 전신을 판단하는 알고리즘으로, 일반적으로 보행자 또는 사람의 형태에 의한 검출에 많이 쓰입니다.

OpenFace

● 변환(Transform) & 자르기(Crop)

- ✓ 입력 이미지로부터 인식된 얼굴 이미지를 분류기 학습에 사용할 수 있도록 표준화하는 단계
- ✓ 동일한 얼굴이라도 정면과 측면에서 보는 모습은 다르기 때문에 컴퓨터가 동일한 사람으로 분류하도록 학습하기 위해서 눈/코/입이 이미지의 동일한 위치로 오도록 변환하는 작업이 필요합니다.
- ✓ 이러한 작업을 **얼굴 특징점 추정(Face Landmark Estimation)**이라고 부릅니다.
- ✓ 얼굴의 특징점이 인식되면 특징점이 사진의 동일한 위치에 오도록 변환하는 과정을 거친다.



OpenFace

- 수치화(Representation)

- ✓ 변환 및 자르기 과정을 거친 얼굴 이미지를 기학습된 DNN(Deep Neutral Network) 모델을 기반으로 수치화해서 분류 모델을 제작하는 단계

- DNN 모델 vs 분류 모델

- ✓ DNN 모델은 입력 이미지를 128차원의 값으로 변환할 수 있는 모델입니다. MNIST 필기체 인식 문제의 경우 CNN(Convolutional Neural Network)을 바탕으로 입력 이미지를 0에서 9까지의 결과로 분류하지만, 이 경우에는 CNN을 바탕으로 입력 이미지를 128차원의 결과값으로 수치화할 수 있습니다.
- ✓ 이처럼 입력 데이터를 고차원의 벡터로 수치화하는 기법을 **임베딩(Embedding)**이라고 하며, 얼굴 이미지를 128차원의 값으로 임베딩되도록 학습시키려면 대량의 데이터와 높은 성능의 컴퓨터 자원을 필요로 합니다.
- ✓ 분류 모델은 128차원의 값을 입력으로 사용하여 각각의 인물을 분류할 수 있는 모델입니다.

OpenFace

- DNN Embedding Example

Input Image



128 Measurements Generated from Image

0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128