

팀 프로젝트 지도 활동 보고서

교과목명	AI및데이터분석의 기초		
강사	한상호 교수		
활동 일자	2020년 10월 29일 (목요일)		
학과	성명	연락처	서명
스마트시스템SW학과	정학제	010-3895-1306	
스마트시스템SW학과	정재윤	010-9008-4847	
스마트시스템SW학과	정해문	010-7176-4289	
스마트시스템SW학과	조재홍	010-4062-9523	
활동 내용	- 미팅 결과물에 대한 URL: https://github.com/Hott-J/AI-BigData-Basic 팀장 github 계정 아래 project repo를 하나 만들고 (repo 이름은 project 취지에 맞게 작명), 코드 파일들, PPT 파일 등을 commit 하고 해당 파일 접근을 위한 URL을 여기에 작성하기		
	2페이지에 적었습니다.		

< 정학제 팀장 >

OpenCV를 활용한 얼굴탐지 관련 자료를 팀원들에게 주고, 직접 시행 및 코드 리뷰를 해보도록 하였습니다. 자료를 찾고, 팀원이 오류가 나면 오류에 대해 해결책을 제시해주었고, 간단한 코드 설명을 해주었습니다.

<정재윤 팀원 >

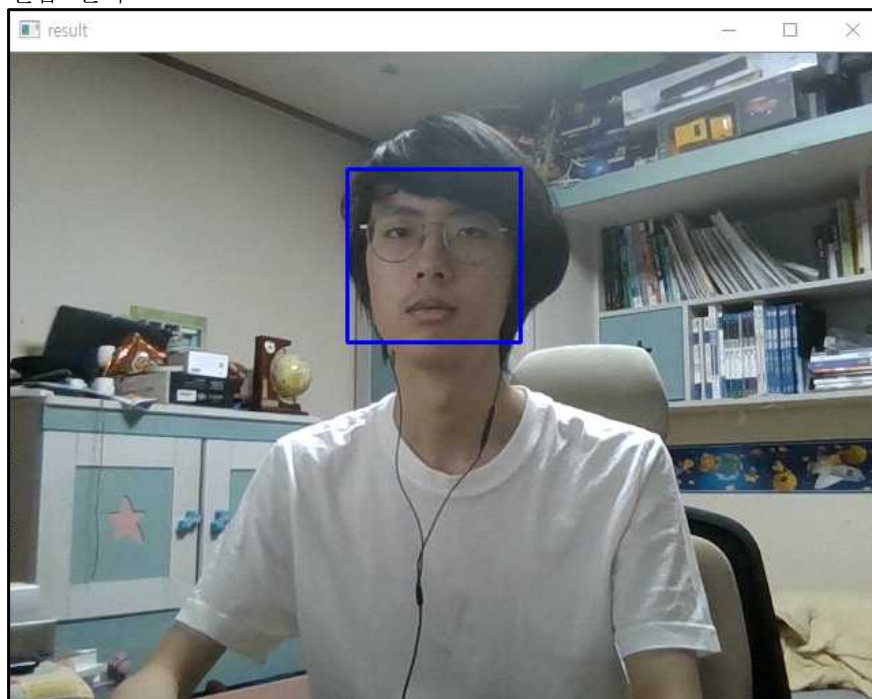
1.오픈CV 실습 및 코드 분석

```
Anaconda Prompt (anaconda3)

(base) C:\Users\LastDefiance>pip install opencv-python
Collecting opencv-python
  Downloading opencv-python-4.4.0.44-cp37m-win_amd64.whl (39.5 MB)
    |#####| 39.5 MB 140 kB/s
Requirement already satisfied: numpy>=1.14.5 in c:\users\lastdefiance\anaconda3\lib\site-packages (from opencv-python) (1.18.1)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.4.0.44

(base) C:\Users\LastDefiance>
```

Anaconda Prompt에서 pip를 이용해 openCV 다운로드
openCV haarcascades xmi 파일 다운로드
실습 결과



코드리뷰(주석으로 되어있다)

```
In [52]: import numpy as np
import cv2

#얼굴 인식 캐스케이드 파일을 읽는다, 다운받은 xml 위치를 잘 파악할것
#xml은 얼굴과 눈을 검출하기 위해 미리 학습해놓은 포맷이다.
xml = 'haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(xml)

cap = cv2.VideoCapture(0) # 노트북 웹캠을 카메라로 사용, 0은 추가적인 카메라가 없을때 default 값이다.
cap.set(3,640) # 너비
cap.set(4,480) # 높이

while(True):
    #이미지 읽어 들이기
    ret, frame = cap.read()
    # 좌우 대칭
    frame = cv2.flip(frame, 1)

    #그레이스케일로 변환시키기
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #얼굴 인식 실행하기(이미지에서 얼굴을 검출한다)
    faces = face_cascade.detectMultiScale(gray,1.05, 5)
    print("Number of faces detected: " + str(len(faces)))

    #얼굴이 검출되면
    if len(faces):
        #원본 이미지에 얼굴의 위치를 표시한다
        for (x,y,w,h) in faces:
            cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)

    #검출 결과를 화면에 보여준다
    cv2.imshow('result', frame)

    # Esc 키를 누르면 종료
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break #break문으로 while문을 빠져나간다

cap.release()
cv2.destroyAllWindows()
```

```
import numpy as np
import cv2
```

```
#얼굴 인식 캐스케이드 파일을 읽는다, 다운받은 xml 위치를 잘 파악할것
#xml은 얼굴과 눈을 검출하기 위해 미리 학습해놓은 포맷이다.
xml = 'haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(xml)
```

```
cap = cv2.VideoCapture(0) # 노트북 웹캠을 카메라로 사용, 0은 추가적인 카메라가 없을때
default 값이다.
cap.set(3,640) # 너비
cap.set(4,480) # 높이
```

```
while(True):
    #이미지 읽어 들이기
    ret, frame = cap.read()
    # 좌우 대칭
    frame = cv2.flip(frame, 1)

    #그레이스케일로 변환시키기
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #얼굴 인식 실행하기(이미지에서 얼굴을 검출한다)
    faces = face_cascade.detectMultiScale(gray,1.05, 5)
    print("Number of faces detected: " + str(len(faces)))

    #얼굴이 검출되면
    if len(faces):
        #원본 이미지에 얼굴의 위치를 표시한다
        for (x,y,w,h) in faces:
```

```

cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)

#검출 결과를 화면에 보여준다
cv2.imshow('result', frame)

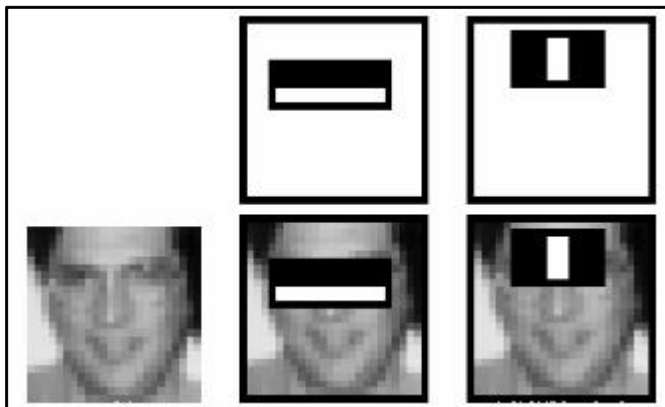
# Esc 키를 누르면 종료
k = cv2.waitKey(30) & 0xff
if k == 27:
    break #break문으로 while문을 빠져나간다

cap.release()
cv2.destroyAllWindows()

```

여기서 사용된 Haar Cascade 알고리즘(간단하게)

특징(feature)을 기반으로 비디오 또는 이미지에서 오브젝트를 검출하기 위해 사용됩니다. 직사각형 영역으로 구성되는 특징을 사용하기 때문에 픽셀을 직접 사용할 때 보다 동작 속도가 빠르다는 장점이 있다.

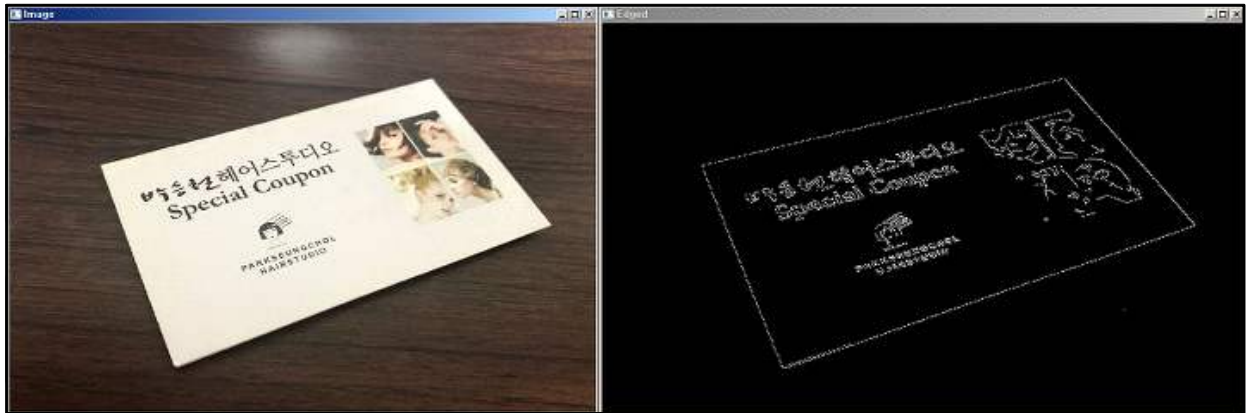


예시로 여기서 보이는 두가지 특징은 눈 주위에서만 의미있는 특징이다. 왼쪽부터 2번째 사진 두장은 코와 뺨보다 눈 부분이 더욱 어둡다는 특성을 잡아내 학습된 특징 블록이고, 3번째는 중앙에 있는 코보다 양쪽에 있는 눈 부분이 더욱 어둡다는 특성을 학습한 블록이다.

이러한 큰 특징 블록들을 많이 학습해서 실제 데이터에서 빠르게 검출해내는 방식의 알고리즘입니다.

적용사례 및 활용방안 조사

명함 인식 및 검출



카메라앱의 기능 (얼굴 인식 후 토끼 귀 같은거 달아주기, 사진 자동 보정 등)



투영변환(문서 스캔)

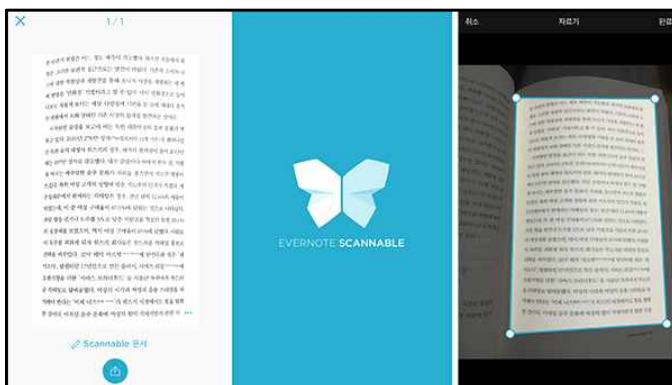


사진 속 글씨 인식하기 (예로 빅스비 비전에 외국에서 외국어로 되어 있는 표지판 같은 것을 찍으면 글자를 검출해 번역해주는 기능이 있다)



차량검출 및 차선검출



등등의 적용사례 및 활용방안이 있을 것 같다.

< 조재홍 팀원 >

OpenCv : 실시간 컴퓨터 비전을 목적으로 한 영상처리 프로그래밍을 위한 라이브러리이다. 영상처리를 위한 여러 함수들이 포함되어 있다.

OpenCv 적용 / 활용 사례

1. 얼굴인식 - 성별 및 연령 판별
2. 명함 인식 / 글자 인식 - 차량 번호판 추출, 사진 속 글자 검출
3. 모션 분석 - 눈동자(동공) 모션 인식을 통한 의사소통 구현
4. 유전자 분석 - 바이러스의 활동 / 움직임을 확인하여 바이러스의 종류를 구분할 수 있다.

OpenCv 활용 방안

- i. 카메라(CCTV)의 화질이 더욱 선명해지고 명확해지면 불분명한 사람의 얼굴이나, 문신 등의 특징을 추출하여 그 사람의 얼굴을 예상하고 구현할 수 있다. - 도둑 / 범인 의 불분명한 사진으로도 범인의 얼굴을 예상 가능
- ii. 정자세의 모습을 촬영함으로써, 골격의 형태를 건강한 사람의 골격과 비교함으로써, 그 사람의 뼈가 틀어진 정도와 휘어짐의 정도를 어느 부위의 뼈가 이상이 있는지 예상할 수 있다.

얼굴 인식의 과정 :

```
import numpy as np
import cv2

xml = 'haarcascades/haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(xml) #얼굴인식 프로그램 실행

cap = cv2.VideoCapture(0) # 카메라 사용
cap.set(3,640) # 너비
cap.set(4,480) # 높이

while(True): #while문이 본격적인 얼굴 인식 과정
    ret, frame = cap.read()
    frame = cv2.flip(frame, 1) # 좌우 대칭
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #색상을 회색으로 설정

    faces = face_cascade.detectMultiScale(gray,1.05, 5)
    print("Number of faces detected: " + str(len(faces)))

    if len(faces):
        for (x,y,w,h) in faces:
            cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)

    cv2.imshow('result', frame)

    k = cv2.waitKey(30) & 0xff #16진수(아스키 코드값)로 프로그램을 탈출하기 위한 조건 설정
    if k == 27: # Esc 키를 누르면 종료
        break

cap.release() #카메라를 끄기
cv2.destroyAllWindows() #프로그램 창 닫기
```

< 정해문 팀원 >

얼굴인식 opencv (주석 참고)


```

import numpy as np
import cv2

xml = 'haarcascades/haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(xml) #얼굴 인식용 xml 파일

cap = cv2.VideoCapture(0) # 노트북 웹캠을 카메라로 사용
cap.set(3,640) # 너비
cap.set(4,480) # 높이

while(True):
    ret, frame = cap.read() #카메라로부터 사진 1장 읽기
    frame = cv2.flip(frame, 1) # 좌우 대칭
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = face_cascade.detectMultiScale(gray,1.05, 5) #이미지내에서 얼굴 검색
    print("Number of faces detected: " + str(len(faces))) #이미지에서 얼굴의 개수 출력

    # cv2.rectangle 와 faces에 들어있는 x,y,w,h 값을 통해 이미지에 박스를 치는 방식으로 확인함.
    if len(faces):
        for (x,y,w,h) in faces:
            cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2) #원본 영상에 위치 표시

    cv2.imshow('result', frame)

    k = cv2.waitKey(30) & 0xff
    if k == 27: # Esc 키를 누르면 종료
        break

cap.release()
cv2.destroyAllWindows() #윈도우를 닫음

```

적용/활용 사례

- 1)cctv를 이용해서 실시간으로 특정인물의 얼굴을 인식함
(비슷하게 주행하는 자동차들의 번호판을 인식하는 카메라가 있을 수 있다)



- 2) 얼굴인식 기능을 이용한 도어락



3) image inpainting
(아래와 같이 손상된 사진의 손상된 부분을 제거할 수 있다)



이미지에서 지우고 싶은 선 또는 잡음을 포함한 객체를 지우고,
주변 픽셀이 가지는 색상으로 지워진 부분을 채운다.

4)QR코드 인식



5)불량품 검출



소수의 정상품 이미지와 불량품 이미지를 모아 딥러닝 알고리즘을 학습시키면
인공지능 신경망이 결함의 특징성을 스스로 찾아냄

팀프로젝트 지도 수행과 관련하여 상기와 같이 활동하였음을 확인합니다.

2020 년 10월 29 일

담당 교수 : 한 상 호 (인)