

# 딥러닝을 기반 마스크 착용 감지 시스템

## 최종 보고서

2021 2학기 캡스톤디자인2

2015104186 안병훈

### 요약

최근 계속해서 코로나 바이러스의 확산이 지속되고 있다. 특히 계속되는 변이 바이러스의 등장으로 인해 백신을 접종 받더라도 감염되는 돌파 감염의 확률이 점점 늘어나고 있다. 따라서 딥러닝을 활용하여 마스크의 착용 여부와 착용 상태에 대해서 확인하고 알려줌으로써 코로나 바이러스의 감염을 예방할 수 있는 시스템을 구축하는 것에 대해 설명하고자 한다.

### 1. 서론

#### 1.1. 연구배경

코로나 바이러스 19로 인해 우리의 일상생활에 제약이 걸린지 어언 2년이 되어간다. 백신 접종접종시작부터 많은 시간이 흐른 만큼 국민 대다수가 백신접종을 완료했다. 이렇게 백신접종률이 높음에도 불구하고 바이러스의 확산속도를 줄어줄 기미를 보이지 않는다. 오히려 계속해서 변이 바이러스가 출현하고있고, 백신을 접종했음에도 바이러스에 감염되는 돌파 감염이 증가하고 있다. 이에 따라 일일 확진자가 최고치를 기록하고 있다. 따라서 부스트 샷을 접종하고는 있지만 근본적인 해결책은 아니다.

이처럼 변이 바이러스의 출현과 줄어들지 않는 확진자 수로 마스크의 중요성은 계속해서 부각되고 있다. 따라서 정부에서는 실내에서 마스크 착용을 의무화 하고 미착용시 과태료를 부과한다. 이에 대해서 대부분의 시민들은 마스크를 잘 착용하고 있지만 코를 내놓고 마스크를 착용하거나 마스크를 턱에 걸치고, 착용하지 않는 경우도 있다. 따라서 출입자에 대해서 마스크 착용여부를 확인하고 알려줄 수 있는 시스템을 구성하고자 한다.

## 1.2. 연구목표

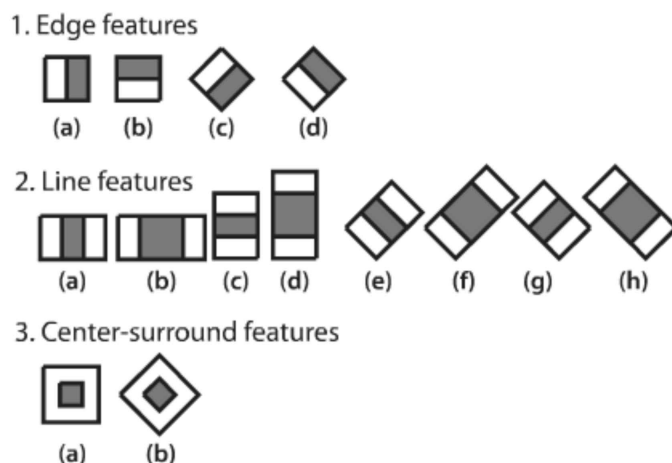
이번 프로젝트의 목표는 마스크의 착용여부를 확인하고 알려줄 수 있는 시스템을 만들고자 한다. 카메라로부터 사람의 얼굴을 실시간으로 검출하여 마스크의 착용여부를 확인할 수 있도록 구현한다. 또한 마스크를 착용했더라도 코를 내놓거나 턱에 걸쳐 쓰는 등 정확하게 착용하지 않은 경우에 대해서도 알려주도록 하여 제대로 착용할 수 있도록 유도하는 기능을 추가한다.

## 2. 관련연구

### 2.1. 얼굴 검출

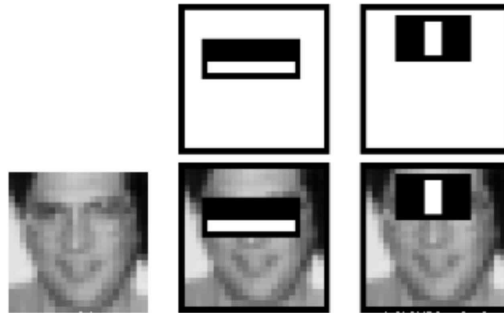
#### 2.1.1. Haar Cascade

Haar Cascade는 머신러닝 기반의 오브젝트 검출 알고리즘이다. 직사각형 영역으로 구성되는 특징을 사용하기 때문에 픽셀을 직접 사용할때 보다 동작속도가 빠르다. 찾으려는 얼굴이 포함된 이미지와 오브젝트가 없는 이미지를 사용하여 Haar Cascade Classifier를 학습시키고 분류기를 사용하여 오브젝트를 검출한다. 이 알고리즘은 분류자를 훈련시키기 위해 매우 많은 훈련용 이미지가 필요하다. 다음 단계는 특징들을 추출해야 한다. 이를 위해, 아래 그림에서의 Haar 특징이 사용된다. 이는 컨볼루션 커널과 같다. 각 특징은 하얀색 사각형에서의 픽셀값의 합을 검정색 사각형 영역의 픽셀값의 합에서 뺀 값이다.



[그림 1]

많은 특징을 계산하기 위해 이미지에 적용할 각 커널에 대한 모든 가능한 크기와 위치를 고려해야 한다. 각 특징을 계산하기 위해서, 하얀색 영역과 검정색 영역의 픽셀의 합을 얻어야 한다 이를 위해 인테그랄 이미지를 사용하고 이는 알고리즘 속도를 매우 빨라진다.

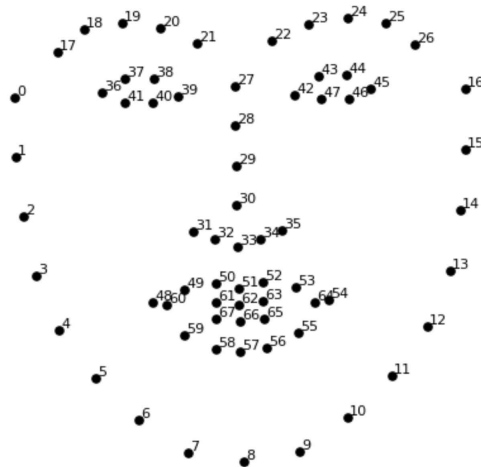


[그림 2]

이를 위해서, 모든 훈련 이미지들에 대해 각각의 모든 특징을 적용한다. 각 특징에 대해, 이미지 상에 얼굴이 있는지 없는지를 분류될 최고의 임계값을 찾는다. 최소한의 에러율로 특징을 선택하는데, 이는 얼굴 이미지인지, 얼굴이 아닌 이미지인지를 아주 잘 분류한다는 것을 의미한다. 최종 분류자는 이러한 약한 분류자들의 가중치 합이다. 약하다고 하는 이유는 분류자 하나만으로 이미지를 분류하지 못하고 분류자들이 모여야만 제대로 이미지를 분류할 수 있기 때문이다. 특징을 이미지에 적용하고 얼굴이 있는지 없는지 검사한다. 그런데, 이 방식은 시간 소모도 많고 비효율적이다. 이를 해결하기 위해 다단계 분류자를 사용한다. 윈도우에 대한 모든 특징을 적용하는 대신, 분류자의 다른 단계로 특징을 묶고 하나씩 하나씩 적용하는 것이다. 만약 윈도우가 첫번째 단계에서 실패하면 버리고 나머지 특징들은 더 이상 고려하지 않는다. 만약 통과되면 특징의 두번째 단계로 넘어가고 이를 계속 반복합니다. 모든 단계를 통과한 윈도우를 얼굴로 인식한다.

## 2.1.2. Face Landmark Detection

사전 훈련된 모델을 사용하여 얼굴 랜드마크 감지기를 초기화한다. 모델이 연속 숫자를 예측하기 때문에 모델은 앙상블 회귀 트리를 기반으로 합니다. 여기에서 모델에 대한 세부 정보를 읽을 수 있다. 이 모델은 iBUG-300 W 데이터 세트에서 훈련되었으며 여기에는 이미지와 해당 68개의 얼굴 랜드마크 포인트가 포함되어 있다. 일반적으로 이러한 랜드마크 포인트는 코, 눈, 입 및 얼굴 가장자리에 속한다.



[그림 3]

## 2.2. 모델 학습

### 2.2.1. Selenium

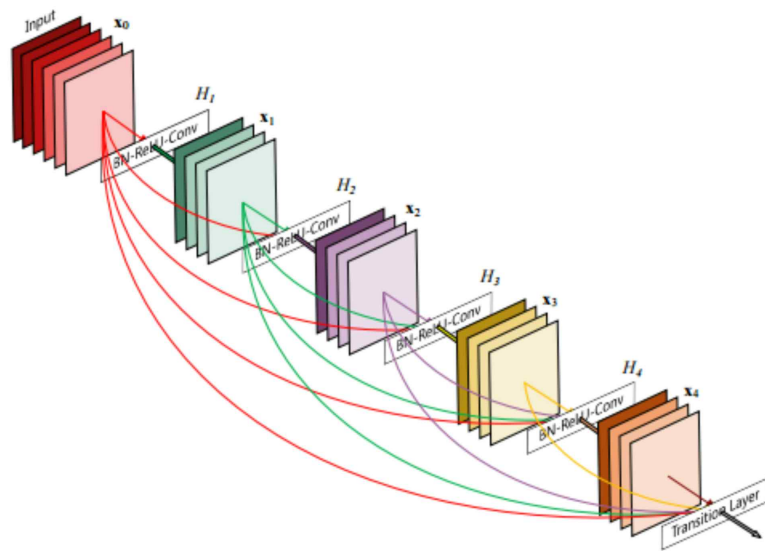
Selenium은 웹 애플리케이션 테스트를 위한 포터블 프레임워크이다. Selenium은 테스트 스크립트 언어를 학습할 필요 없이 기능 테스트를 만들기 위한 플레이백 도구를 제공한다. (Selenium IDE) C 샤프, 그루비, 자바, 펄, PHP, 파이썬, 루비, 스칼라 등 수많은 유명 프로그래밍 언어들에서 테스트를 작성하기 위한 테스트 도메인 특화 언어(Selenese)를 제공한다. 이 테스트들은 현대의 대부분의 웹 브라우저에서 수행이 가능하다. 셀레늄은 윈도우, 리눅스, macOS 플랫폼에서 배포된다. 아파치 2.0 라이선스로 배포되는 오픈 소스 소프트웨어이다.



[그림 4]

## 2.2.2. DenseNet

DenseNet은 ResNet과 Pre-Activation ResNet보다 적은 파라미터 수로 더 높은 성능을 가진 모델이다. DenseNet은 모든 레이어의 피쳐맵을 연결한다. 이전 레이어의 피쳐맵을 그 이후의 모든 레이어의 피쳐맵에 연결한다. 연결할 때는 ResNet과 다르게 덧셈이 아니라 concatenate을 수행한다. 따라서 연결할 때는 피쳐맵 크기가 동일해야 한다. 피쳐맵을 계속해서 연결하면 채널 수가 많아질 수 있기 때문에 각 레이어의 피쳐맵 채널 수는 굉장히 작은 값을 사용한다.



**Figure 1:** A 5-layer dense block with a growth rate of  $k = 4$ . Each layer takes all preceding feature-maps as input.

[그림 5]

## 2.3. 기존방법

2021년 4월 12일부터 실내 마스크 착용 의무화 조치가 시행되고 있다. 또한 사회적 거리두기 단계와 상관없이 모든 실내 마스크 의무화가 실시되었다. 이에 따라 마스크 착용을 하지 않은 사람들을 확인할 필요가 생기게 되었다. 따라서 현재 사람이 직접 마스크 착용 유무를 확인하고 있는 상황이다.

### 2.3.1. 기존방법의 문제점

기존 방법에서는 사람이 마스크의 착용 여부를 직접 확인하게 된다. 이런 경우 인력이 추가적으로 필요할 수 있다. 또한 마스크를 착용하지 않은 사람이 코로나 확진자였을 경우 감염 확률이 올라갈 수 있게 된다.

### 2.3.2. 기존방법과의 차이점 및 해결방안

QR체크인과 같은 방식으로 사람이 직접 확인하지 않고 카메라를 통해 마스크 착용여부를 체크할 수 있도록 한다. 따라서 사람과의 밀접 접촉확률이 떨어지게 된다. 또한 사람이 확인할 경우 마스크를 제대로 착용하지 않은 사람들에 대해서 무심코 넘어갈 수도 있지만 이번 프로젝트에서는 이에 대해서도 확인하여 알려줄 수 있도록 한다.

## 3. 프로젝트 내용

### 3.1. 프로젝트 소개

이번 프로젝트는 마스크 착용 감지를 위해 DenseNet으로 착용, 미착용, 오착용의 결과가 나올 수 있도록 이미지 분류모델을 학습하고, 학습된 모델을 사용하여 사용자의 얼굴에서 마스크 착용 여부를 확인할 수 있도록 한다.

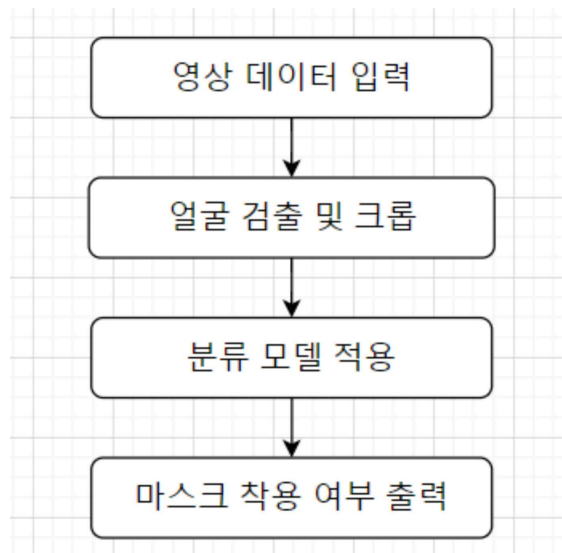
### 3.2. 요구사항

#### 3.2.1 분류모델에 대한 요구사항

- 시스템이 카메라로부터 실시간으로 영상을 받아들이어서 마스크 착용여부를 확인할 수 있도록 한다.
- 카메라로부터 들어오는 영상에는 다양한 객체들이 존재하기 때문에 먼저 객체들 중에서 사람의 얼굴을 검출하고 크롭할 수 있도록 한다.
- 검출된 사람의 얼굴에서 마스크 착용, 마스크 미착용, 부적절한 마스크 착용으로 나누어서 보여줄 수 있도록 한다.

### 3.3. 시스템 설계

#### 3.3.1. Activity Diagram



[그림 6]

### 3.4. 구현

#### 3.4.1. 데이터셋 생성

```
VAL_BATCH_SIZE = 128
BATCH_SIZE = VAL_BATCH_SIZE * 8

train_generator = train_datagen.flow_from_dataframe(
    dataframe=train_df,
    target_size=(128,128), # 최대 입력 이미지 크기에 맞춤
    class_mode='categorical',
    batch_size=BATCH_SIZE) # 출력 값 2가지: 마스크 착용, 마스크 미착용
val_generator = val_test_datagen.flow_from_dataframe(
    dataframe=val_df,
    target_size=(128,128),
    class_mode='categorical',
    batch_size=VAL_BATCH_SIZE)
test_generator = val_test_datagen.flow_from_dataframe(
    dataframe=test_df,
    target_size=(128,128),
    class_mode='categorical',
    batch_size=VAL_BATCH_SIZE)
```

### 3.4.2. 학습 모델 구성

```
pretrained_base = applications.DenseNet121(  
    include_top=False,  
    input_shape=(128,128,3),  
    pooling='avg')  
  
pretrained_base.trainable = False  
  
model = Sequential([  
    pretrained_base,  
    Flatten(),  
    Dense(3, activation='softmax')])  
  
model.compile(  
    optimizer='adam',  
    loss='categorical_crossentropy',  
    metrics=['accuracy'])
```

## 4. 프로젝트 결과

### 4.1. 연구결과

#### 4.1.1. 마스크 착용 데이터의 경우

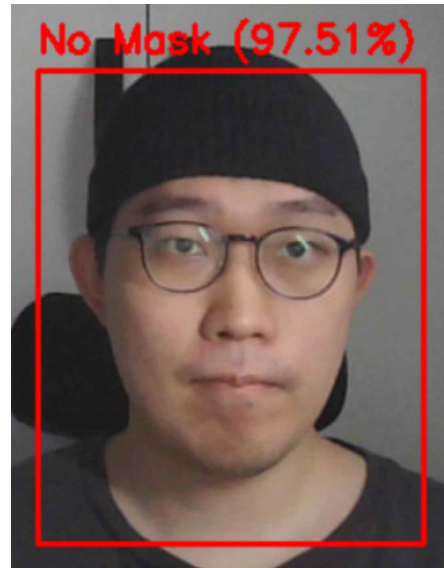
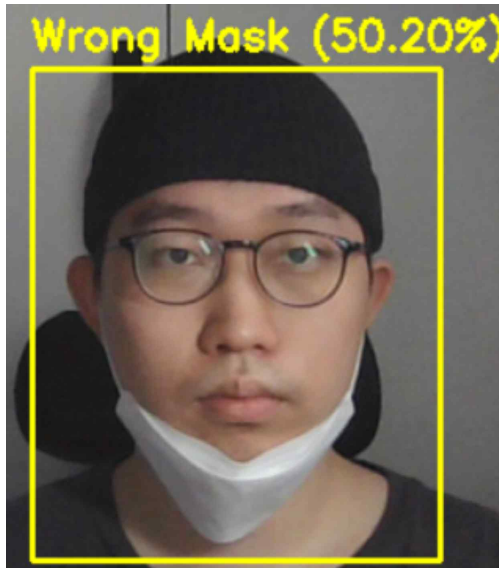
다음은 마스크 착용시 모델에 적용한 결과값이다.





#### 4.1.2. 마스크 미착용, 오착용 데이터의 경우

다음은 마스크 오착용, 미착용시 모델에 적용한 결과값이다.



#### 4.2. 모델 성능 평가

마스크 착용시와 미착용시에서 학습모델이 높은 인식률을 보이지만 마스크 오착용시에는 인식률이 비교적 낮다.

### 5. 결론

#### 5.1. 기대효과

코로나 상황에서 필수적인 마스크의 착용여부를 확인할 수 있는 시스템이다. 이를 통해 추가적인 인력충원 없이도 마스크를 착용하지 않거나 착용방법이 잘못된 사람들에게 마스크를 제대로 착용할 수 있도록 유도할 수 있을 것으로 기대된다. 또한 지금 주로 사용되고 있는 QR체크인과 결합하여, 마스크 착용 여부와 출입여부를 동시에 관리할 수 있도록 하면 코로나 바이러스 추적 관리를 더 효과적으로 할 수 있을 것으로 기대된다.

## 5.2. 추후 연구 방향

지금까지 진행된 프로젝트 연구사항은 입력된 영상을 기반으로 사용자의 마스크 착용 여부를 확인하는 것이다. 마스크 오착용시의 인식률을 높이기 위해 딥러닝 모델을 업데이트 하는 것이 좋아 보인다. 현재 오착용시 인식률이 비교적 낮는데 이는 마스크 오착용시 데이터가 부족하기 때문이다. 따라서 더 다양한 데이터를 수집하는 것이다.

또한 해당 시스템을 기반으로 실제로 적용할 수 있도록 GUI를 구성하는 것이 당면과제이다. 마스크는 코로나가 종식될때까지 필수적으로 착용해야 한다. 따라서 해당 시스템을 이용하는데 있어서 사용자의 편의성을 위해 마스크 미착용한 인원 발생시 직접적으로 알려주는 것 외에도 발생한 시간을 같이 저장하는 것으로 추적관리가 용이할 것이다.

## 6. 참고문헌

- [1] Rapid Object Detection using a Boosted Cascade of Simple Features
- [2] Face Landmark Detection using Python
- [3] <https://ko.wikipedia.org/wiki/selenium>
- [4] Densely Connected Convolutional Networks