

Mysk-마이스크⁺

오픈소스 기초설계 나반 7팀

권지성 20211723

박시은 20211732

이지연 20211741

장민준 20213091



AIM & BACKGROUND

1

배경 설명

위드코로나, 마스크 착용 강화

2

목표

마스크 착용 판별

3

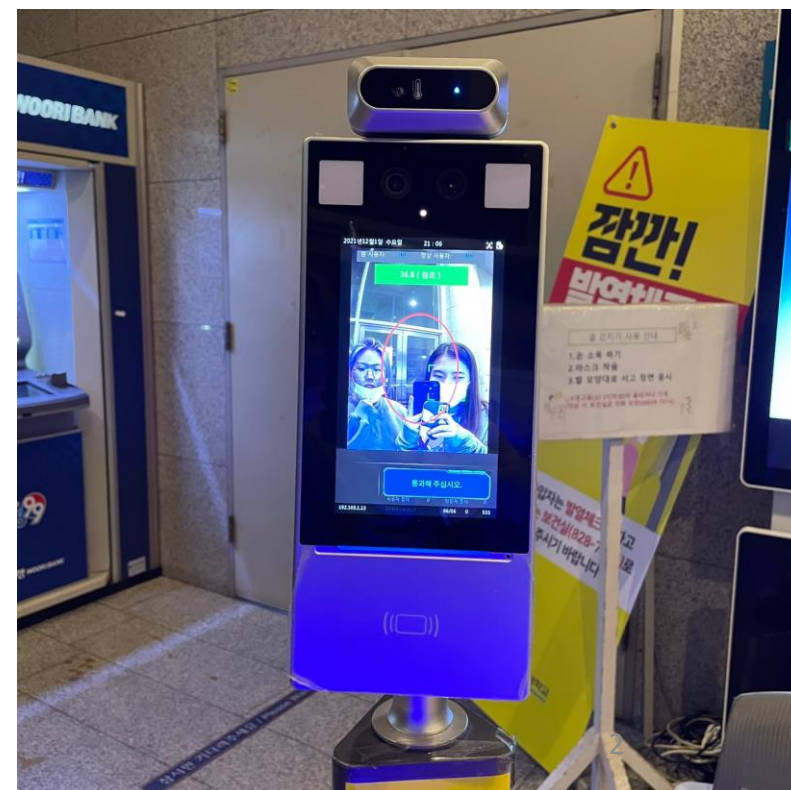
사례 분석

건물 출입구의 기계

4

해결 방안

다수의 인물 동시 인식



| CONTENTS



Flask

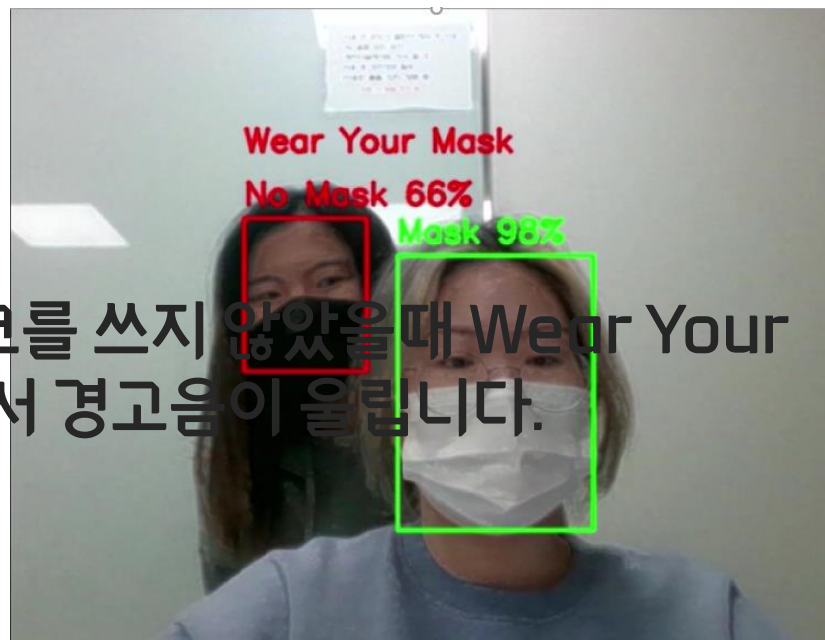
web development,
one drop at a time

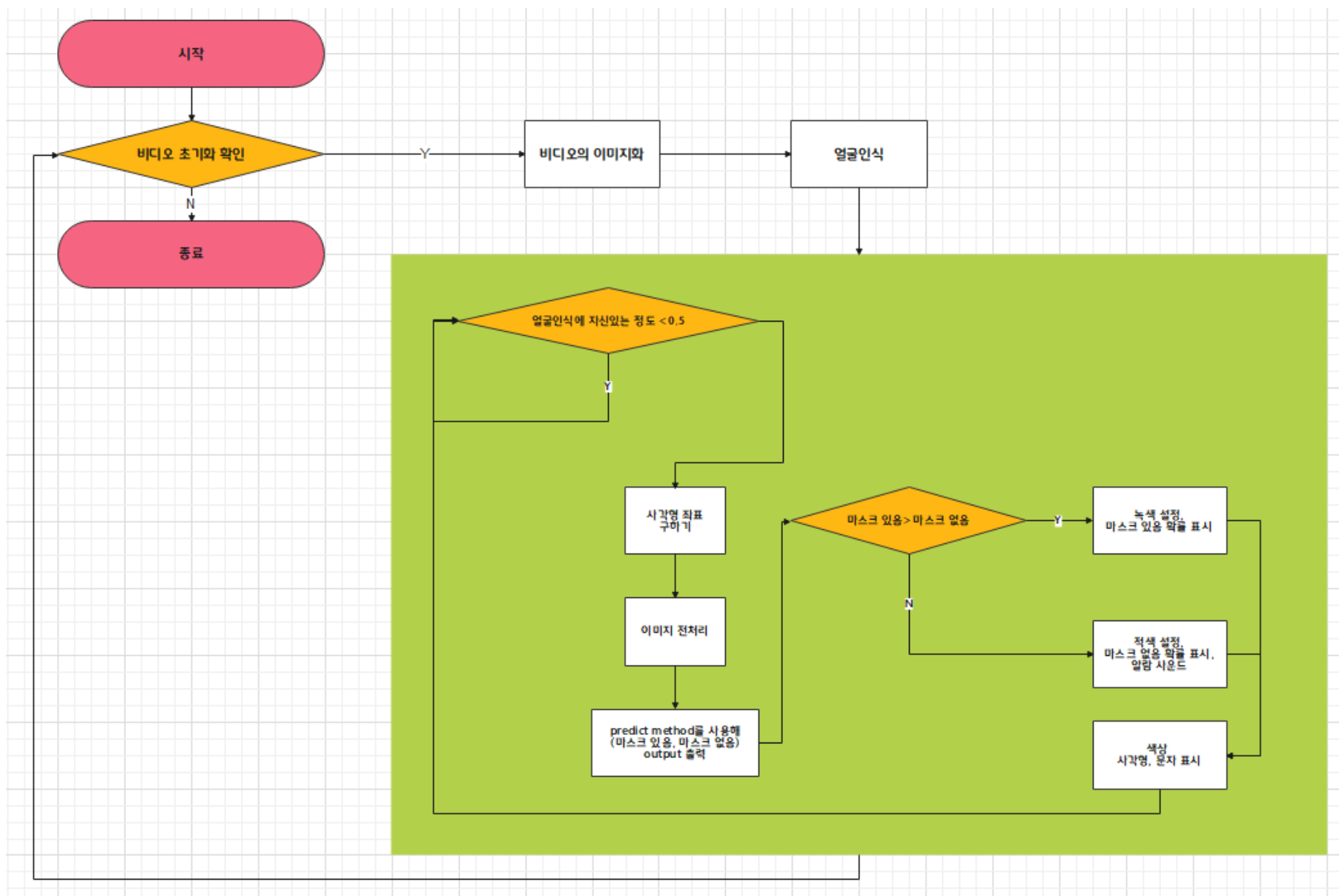
1 사용언어 : 파이썬 HTML CSS FLASK

2 SW 구현

3 웹 구현 :

4 주요 결과 : 마스크를 쓰지 않았을때 Wear Your Mask 라고 뜨면서 경고음이 울립니다.

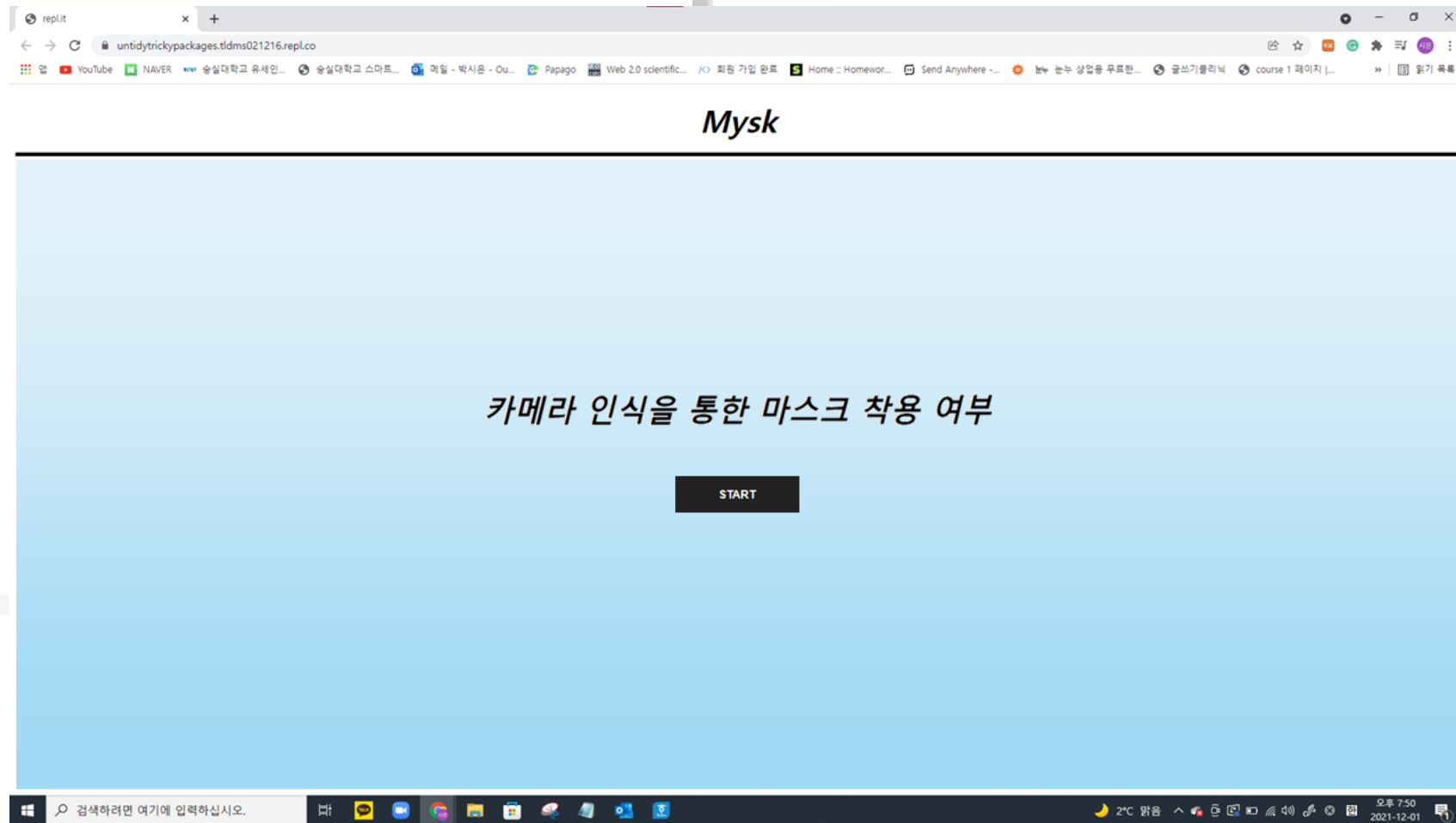




```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width">
6 <title>repl.it</title>
7 <link href="style.css" rel="stylesheet" type="text/css" />
8 <style>
9
10 .body {
11   margin:0;
12   height:100%;
13   <background-image:linear-gradient(to top, #d9afd9 0%, #97d9e1 100%);
14   background-repeat: no-repeat;
15   background-size: cover;
16   background-attachment: fixed;
17 }
18 .linear_gradient {
19   width: 1888px;
20   height: 1000px;
21   background: linear-gradient(to top, aqua, pink);
22 }
23
24 .menu {
25   height: 70px;
26   border-bottom: 5px solid black;
27   margin-top: 20px;
28   text-align: center;
29   font-weight: bold;
30   font-style: italic;
31   font-size: 40px;
32 }
33 .button_container {
34   position: absolute;
35   text-align: center;
36   left: 0;
37   right: 0;
38   top: 30px;
39   margin-top: 300px
40 }

```



Reference

<4p>

html, css 아이콘, 수학과와 좌총우돌 프로그래밍 (tistory), 2019.1.22, <https://ssungkang.tistory.com/entry/html-%EC%9D%B4%EB%AF%B8%EC%A7%80%EB%A5%BC-%EB%A7%8C%EB%93%9C%EB%8A%94-%ED%83%9C%EA%B7%B8>

flask 아이콘, Wikimedia commons, 2021.12.01(접속일자),
https://commons.wikimedia.org/wiki/File:Flask_logo.svg

python 아이콘, 개발자 김철수, 2020.06.30, <https://blog.fakecoding.com/archives/python-cycle/>

<코드 참조>

빵형의 개발도상국, “마스크 썼는지 안썼는지 알아내는 인공지능”, 2020.06.12, 2021.12.02 ,
<https://youtu.be/nclyy1doSJ8>

Appendix

코드설명

```
1  from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
2  from tensorflow.keras.models import load_model
3  import numpy as np
4  import cv2
5  import matplotlib.pyplot as plt
6  import os
7  from pygame import mixer
8  import time
9
10 facenet = cv2.dnn.readNet('models/deploy.prototxt', 'models/res10_300x300_ssd_iter_140000.caffemodel') #얼굴 인지모델(face detection) 미리 학습해둔 파일을 받아옴
11 model = load_model('models/mask_detector.model') #마스크 모델은 keras모델 -> 설명 사이트 https://hazel-developer.tistory.com/97
12
13 cap = cv2.VideoCapture(0) # videocapture 객체 생성 (0 = 노트북 내장 웹캠)
14 ret, img = cap.read() # 프레임별로 이미지 캡처 ret = 프레임을 정상적으로 읽었는가 확인
15
16 fourcc = cv2.VideoWriter_fourcc('m', 'p', '4', 'v') # 비디오 캡처 저장 객체
17 out = cv2.VideoWriter('output.mp4', fourcc, cap.get(cv2.CAP_PROP_FPS), (img.shape[1], img.shape[0]))
18 #cv2.VideoWriter(파일이름, fourcc, fps초당프레임수, frameSize(width, height))
19 mixer.init()
20 sound = mixer.Sound('alarm.wav')
```

Appendix

코드설명

```
22 while cap.isOpened(): # cap이 초기화 되었나 확인
23     ret, img = cap.read()
24     if not ret:
25         break
26
27     h, w = img.shape[:2] #이미지의 높이와 너비 저장
28
29     blob = cv2.dnn.blobFromImage(img, scalefactor=1., size=(300, 300), mean=(104., 177., 123.)) #네트워크 입력 볼륨(blob)으로 이미지 만들기
30     #(입력영상, 입력영상-픽셀에 곱할 값으로 기본값은 1, 출력영상크기, 입력 영상 각 채널에서 뺀 평균 값)
31     facenet.setInput(blob) #모델에 들어가는 input
32     dets = facenet.forward() #결과 추론
33
34     result_img = img.copy() #결과값 저장
```


Appendix

코드설명

```

36 for i in range(dets.shape[2]): # 사진속 얼굴 개수가 여러 개 있을 수 있으니 반복문 사용
37     confidence = dets[0, 0, i, 2] #결과값이 얼마나 자신있느냐
38     if confidence < 0.5: #값을 0.5로 정함
39         continue # 0.5 미만인 값들은 모두 넘김
40
41     x1 = int(dets[0, 0, i, 3] * w)
42     y1 = int(dets[0, 0, i, 4] * h)
43     x2 = int(dets[0, 0, i, 5] * w)
44     y2 = int(dets[0, 0, i, 6] * h) # x와 y의 바운딩 박스 값 구해줌
45
46     face = img[y1:y2, x1:x2] # 얼굴 사진만 자르기
47     # 전처리 구간
48     face_input = cv2.resize(face, dsize=(224, 224)) #이미지 크기 변경
49     face_input = cv2.cvtColor(face_input, cv2.COLOR_BGR2RGB) #컬러 시스템을 BGR에서 RGB로 변환
50     face_input = preprocess_input(face_input) #전처리 저장값 shape(224,224,3)
51     face_input = np.expand_dims(face_input, axis=0) # 모델에 넣을때 (1,224,224,3)이므로 차원 추가
52
53     mask, nomask = model.predict(face_input).squeeze() #predicr method를 사용해 아웃풋 출력 (마스크 있음, 마스크 없음)
54     # 마스크 확률 출력
55     if mask > nomask:
56         color = (0, 255, 0)
57         label = 'Mask %d%%' % (mask * 100)
58     else:
59         color = (0, 0, 255)
60         label = 'No Mask %d%%' % (nomask * 100)
61         sound.play()
62         time.sleep(1)
63     cv2.putText(result_img, text='Wear Your Mask', org=(x1, y1 - 50), fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.8, color=color, thickness=2, lineType=cv2.LINE_AA)

```

Appendix

코드설명

```
65     cv2.rectangle(result_img, pt1=(x1, y1), pt2=(x2, y2), thickness=2, color=color, lineType=cv2.LINE_AA)
66     #사각형 표시 (그림 그릴 이미지, 시작좌표, 종료좌표, 선두께)
67     cv2.putText(result_img, text=label, org=(x1, y1 - 10), fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.8, color=color, thickness=2, lineType=cv2.LINE_AA)
68     #문자 표시 (그릴 이미지, 출력문자, 출력문자 시작위치 좌표, 출력 문자 시작위치 좌표, 폰트, 폰트 크기, 색상, 두께, 선종류)
69     out.write(result_img)
70     cv2.imshow('result', result_img) #이미지를 모니터에 보여줌
71     if cv2.waitKey(1) == ord('q'): #아스키 값으로 어떤 키를 눌렀는지 보여주는 값, q값을 누르면 영상 종료
72         break
73
74 out.release() # out 객체 해제
75 cap.release() # cap 객체 해제
```

Appendix

프로그램 구현한거는 옵션으로 링크 하고싶으면 해라.(깃허브로)