



OpenCV를 활용한 고연령 맞춤형 키오스크 시스템



원동진

#AA #PM #기획



최희철

#DA #기술설계 #개발



손미연

#DA #기술설계



윤대호

#TA #개발



채규비

#AA #기획 #PPT

CONTENTS

I 프로젝트 개요

1. 제안 배경 및 필요성
2. 목적 및 방향성
3. 서비스 구상도

II 개발

1. 기술 개요
2. 데이터
3. 데모 시연

III 활용

1. 활용 방안

I. 프로젝트 개요

01. 제안 배경 및 필요성

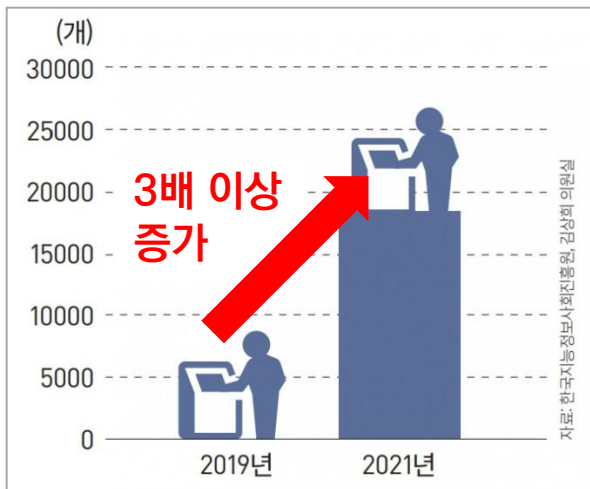
인건비 상승 및 COVID-19 등으로 오프라인 매장의 **키오스크 설치율 증가 추세**

65세 이상 고연령의 키오스크 이용 경험은 절반 이하

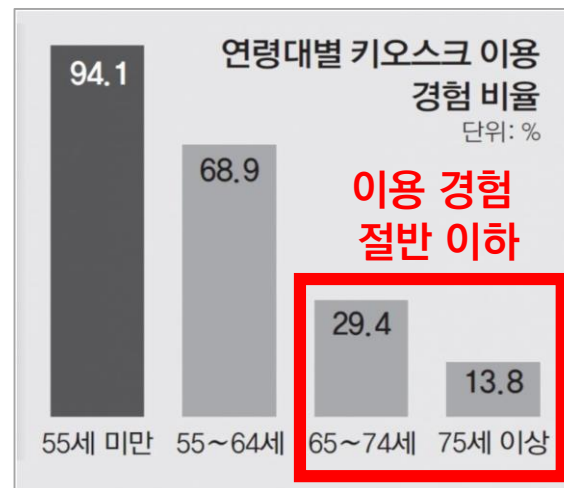
키오스크 시스템의 복잡성으로 **키오스크 사용에 고연령 어려움 경험**

고연령을 위한
키오스크 시스템
간소화 필요

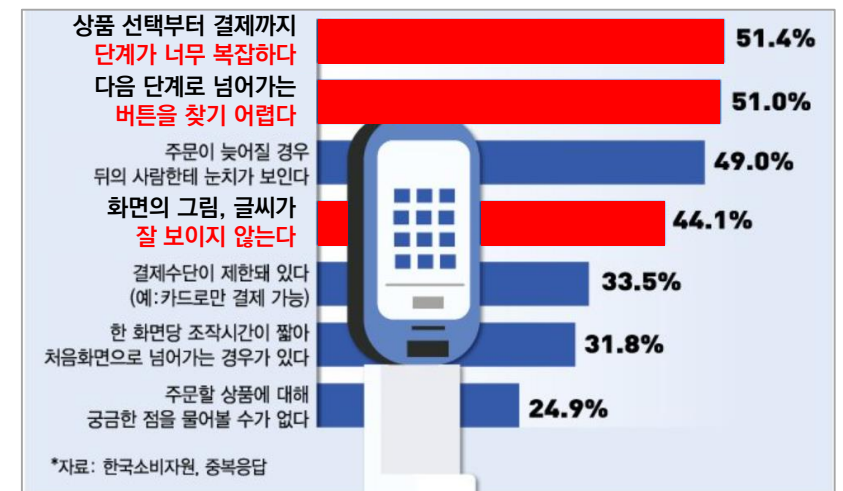
[최근 2년 키오스크 설치 현황]



[고연령의 키오스크 이용 경험 현황]

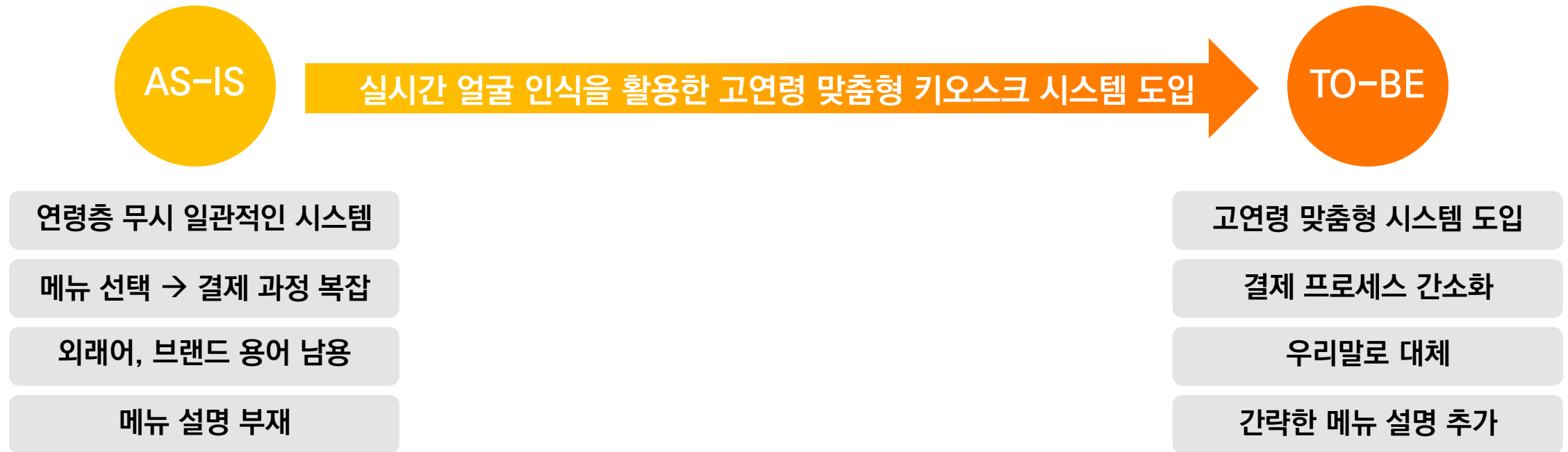


[고연령(65세 이상) 키오스크 이용 불편사항]



I. 프로젝트 개요

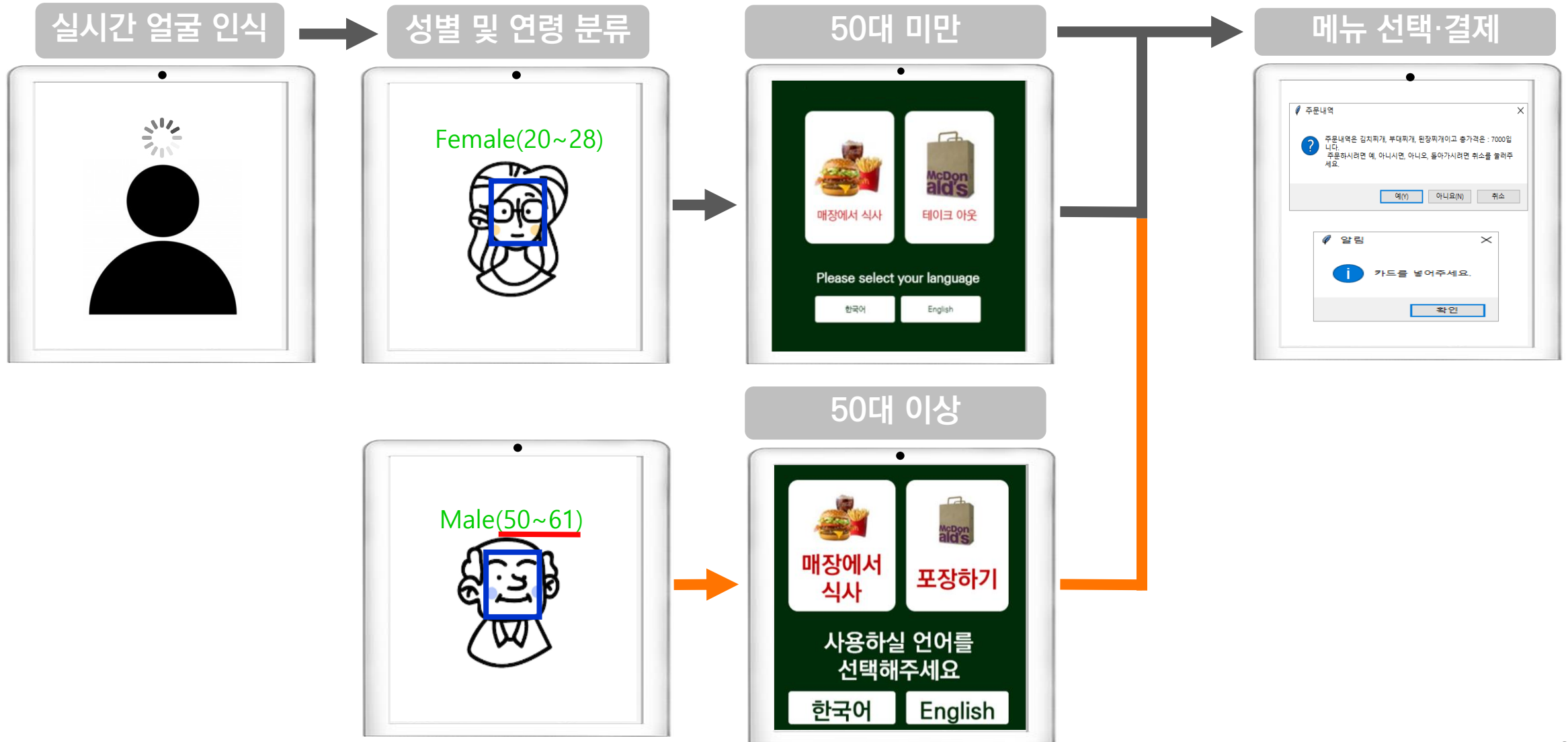
02. 목적 및 방향성



“ 고연령도 쉽게 사용 가능한 **위드 실버 키오스크 시스템** ”

I. 프로젝트 개요

03. 서비스 구상도



II. 개발

01. 기술 개요

실시간 얼굴 인식

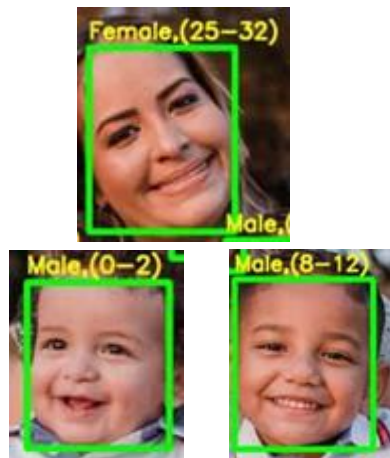
사전 훈련된
하르 캐스케이드 분류기



openCV
CascadeClassifier(xml)

성별 및 연령 분류

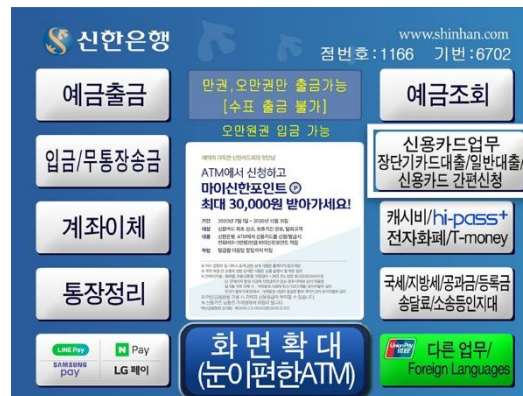
사전 훈련된
성별 분류기와 연령 분류기



```
readNet(  
    'age_net.caffemodel',  
    'deploy_age.prototxt')  
readNet(  
    'gender_net.caffemodel',  
    'deploy_gender.prototxt')
```

화면 확대 자동 전환

고연령으로 인식할 경우
맞춤형 시스템으로 전환



Tkinter

화면 확대 실행

쉬운 용어 및 큰 글씨 등
보기 쉬운 화면으로 실행



tkFont
cv2.resize

II. 개발

실시간 얼굴 인식을 통한 성별/연령 분류

02. 데이터

실시간 이미지 인식을 위한 웹 카메라 설정

```
# 라이브러리 호출
import numpy as np
import cv2

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

인식된 이미지 값 반환 후 회색으로 스케일링

```
while(True):
    ret, frame = cap.read() # 설정된 캠으로 영상을 송출하고 비디오 frame을 array 형태로 받아옴, ret은 실행도
    frame = cv2.flip(frame, 1) # 좌우대칭
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # 회색으로 스케일링(이미지의 윤곽을 더 명확하게 확인하기

    faces = face_cascade.detectMultiScale(gray, scaleFactor = 1.05, minNeighbors = 3, minSize=(30,30))
```

이미지 값(array)을 4차원 Tensor 형태로 변환

```
for(x,y,w,h) in faces: # 얼굴
    face = frame[int(y):int(y+h),int(x):int(x+h)].copy() # x,y = rectangular가 시작하는 지점 x,y의 좌표
    # w,h = rectangular의 너비와 길이
    # rectangular의 y축 시작지점부터 길이를 더한 값과 x축 시작지점부터 길이를 더한 값을 반환
    # 4차원 텐서 형태로 변환(영상에서 검출한 얼굴을 4차원 텐서 형태로 변환하여 추론하기 위한 함수)
    blob = cv2.dnn.blobFromImage(face, scalefactor = 1.1, size = (227, 227), mean=MODEL_MEAN_VALUES, swapRB=False)
    # 이미지, 이미지를 스케일 사이즈에 맞추기 위한 파라미터, 신경망이 예상하는 이미지의 크기
```

사전 학습된 이미지 데이터, 성별/나이 분류 모델

```
xml = 'haarcascade/haarcascade_frontalface_default.xml' # github 오픈소스 haarcascades 사용
face_cascade = cv2.CascadeClassifier(xml) # 학습된 데이터 셋을 가져옴

# 얼굴을 인식하여 나이와 성별을 학습 시켜놓은 모델을 불러옴
age_net = cv2.dnn.readNet(
    'deploy_age.prototxt',
    'age_net.caffemodel')

gender_net = cv2.dnn.readNet(
    'deploy_gender.prototxt',
    'gender_net.caffemodel')
```

이미지 값에서 성별 및 연령대 구별

```
# gender detection
gender_net.setInput(blob) # net에 blob 데이터를 넣어줌
gender_preds = gender_net.forward() # 순방향으로 학습
gender = gender_preds.argmax() # 학습한 것 중에 제일 높은 값 반환
# Predict age

age_net.setInput(blob) # net에 blob 데이터를 넣어줌
age_preds = age_net.forward() # 순방향으로 학습
age = age_preds.argmax() # 학습한 것 중에 제일 높은 값 반환
info = gender_list[gender] + ' ' + age_list[age]
```

실시간 영상에 인식된 얼굴과 정보 제공

```
cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2) # 이미지
cv2.putText(frame,info,(x,y-15),0, 0.5, (0, 255, 0), 1)

cv2.imshow('result', frame) # 현재 프레임을 윈도우에 출력
```


II. 개발

키오스크 시스템 화면 구현 및 실시간 연령 분류 모델과 연동

02. 데이터

키오스크 화면 구현 라이브러리 설정

```
window = tk.Tk()
window.title("PyPOS Ver 0.1")
window.geometry("600x400+500+300")
window.resizable(False, False)

frame1 = tk.Frame(window, width="600", height="10")
# frame1.pack(fill="both")
```

키오스크 화면 구성 내부 설정

```
label_table = tk.Label(frame1, text="테이블번호 ", padx=10, pady=10, fg="red", font='Arial 15')
label_table.grid(row=0, column=0, padx=10, pady=10)

btn_meal = tk.Button(frame1, text="식사", padx="10", pady="10", bg="yellow", command=show_meal)
btn_meal.grid(row=0, column=1, padx=10, pady=10)

btn_drink = tk.Button(frame1, text="음료", padx="10", pady="10", bg="white", command=show_drink)
btn_drink.grid(row=0, column=2, padx=10, pady=10)

btn_end = tk.Button(frame1, text="주문종료", padx="10", pady="10", command=order_end)
btn_end.grid(row=0, column=3, padx=10, pady=10)
```

설정된 내부 구성요소에 기능 부여

```
# 테이블 선택
def table_select(tno):
    global table_no
    table_no = tno

    print("table number:", tno)
    label_table.configure(text="테이블 " + str(table_no))

    frame5.pack_forget() # 테이블을 선택하면 첫 위젯인 테이블
    frame6.pack_forget() # 결제 프레임 위젯을 감춘다(frame
    print_order() # 테이블 선택 후 나오는 프레임의 구성요소
    print_price() #
    show_meal() #

# 테이블 보여줌
def show_table():
    frame1.pack_forget()
    frame2.pack_forget()
    frame3.pack_forget()
    frame4.pack_forget()
    frame5.pack(fill="both", expand=True)
    frame6.pack_forget()
```

실시간 얼굴 인식 코드를 함수로 제작

```
def facefind(cap, face_cascade, age_net, gender_net, MODEL_MEAN_VALUES, age_list, gender_list):
    while(True):
        ret, frame = cap.read() # 설정된 캠으로 영상을 송출하고 비디오 frame을 array 형태로 받아옴,
        frame = cv2.flip(frame, 1) # 좌우대칭
```

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

return age
```

If문을 이용하여 연령에 맞는 키오스크 파일 실행

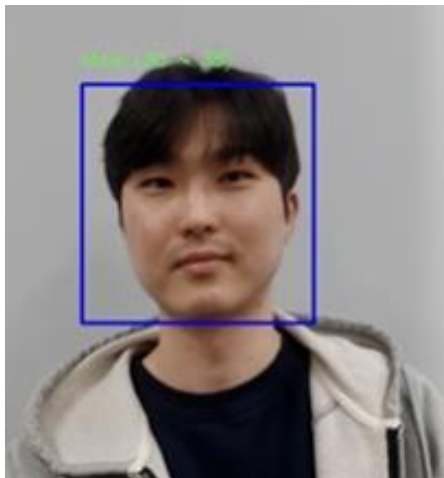
```
age = facefind(cap, face_cascade, age_net, gender_net, MODEL_MEAN_VALUES, age_list, gender_list)

if age < 4: # age_list의 5번째 인덱스보다 작은(50대 미만)
    import os
    os.system("C:/Users/82104/Desktop/카메라로 얼굴인식/order_dist/order.exe") # 기존의 키오스크
else: # 50대 이상
    import os
    os.system("C:/Users/82104/Desktop/카메라로 얼굴인식/older_dist/older.exe") # 기존의 키오스크
```

II. 개발

03. 데모 시연

50세 미만



변경 전

예시 GUI

메뉴를 선택해 주세요

한식	일식	중식
김치찌개 : 7000	돈까스 : 6500	짜장면 : 6000
된장찌개 : 7000	우동 : 5000	짬뽕 : 6500
부대찌개 : 7500		

주문하기



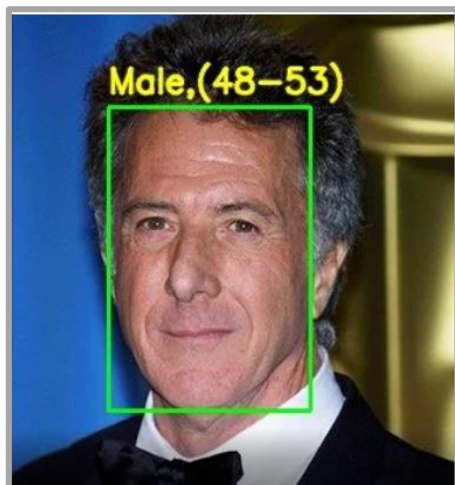
〈결제 화면〉

주문내역

? 주문내역은 김치찌개, 부대찌개, 된장찌개이고 총가격은 : 7000입니다.
주문하시려면 예, 아니시면, 아니오, 돌아가시려면 취소를 눌러주세요.

예(Y) 아니요(N) 취소




50세 이상



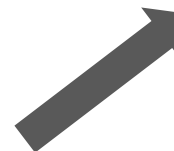
변경 후

예시 GUI

메뉴를 선택해 주세요

김치찌개	된장찌개	부대찌개
		
김치찌개	된장찌개	부대찌개

주문하기



알림

i 카드를 넣어주세요.

확인

III. 활용

III. 활용

연령별 맞춤 키오스크 적용 예시 - 맥도날드

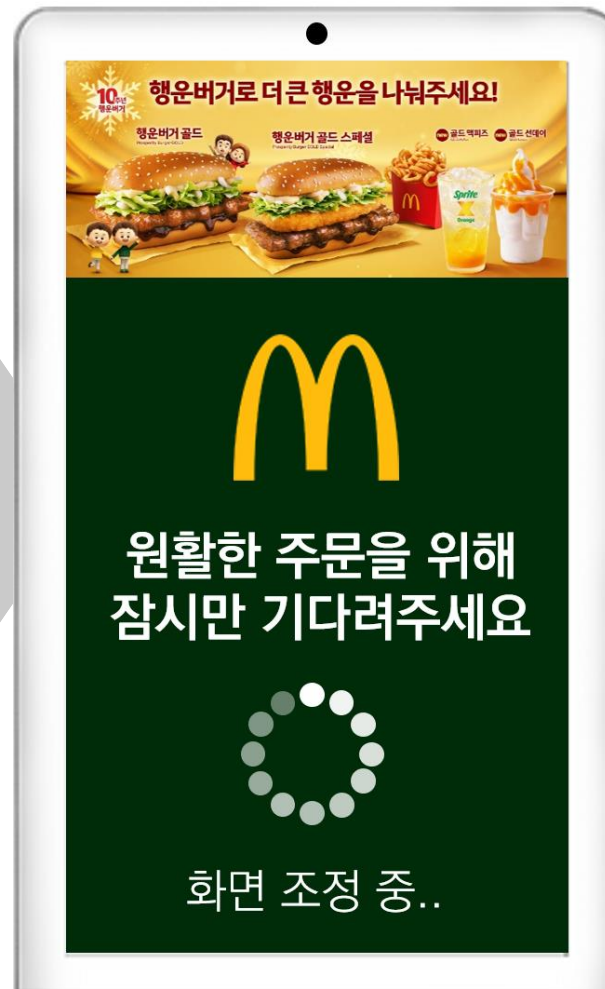
01. 활용 방안

초기 키오스크 화면



키오스크 내
얼굴 인식 카메라
도입 및 활용하여
소비자 연령대 확인

고연령으로 인식할 경우 키오스크 화면



III. 활용

연령별 맞춤 키오스크 적용 예시 - 맥도날드

01. 활용 방안

초기 화면

메뉴 화면

사이즈 선택

장바구니 추가

결제

변경전



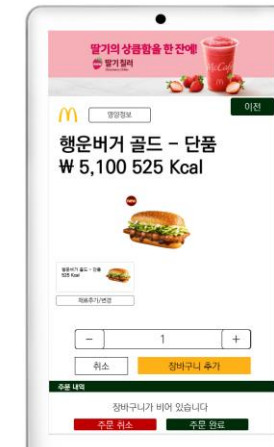
글씨 확대,
불필요한 외래어 삭제



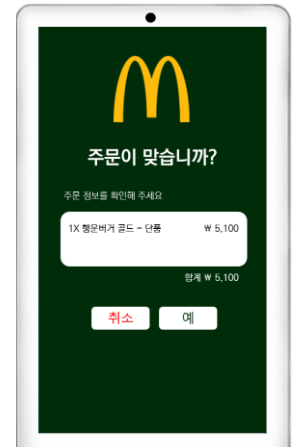
메뉴판 간소화, 간략한 메뉴 설명 추가



주문 방법의 간편화 및 불필요한 장바구니 시스템 삭제



삭제



변경후

Q & A

E.O.D