**중간 발표 대본**

**슬라이드1**

**안녕하세요. 저는 5조 발표를 맡은 정호현이라고 합니다.**

**저희는 리테일앤인사이트의 체적측량을 주제로 한 기업 연계 프로젝트를 진행하게 되었습니다. 발표 시작에 앞서 전체적인 순서에 대해 말씀드리겠습니다.**

**슬라이드2**

**저희의 발표 순서는 구성원 역할 소개와 프로젝트 목표 및 아이디어, 알고리즘과 결과물, flask로 구현한 데모 앱과 제한사항, 향후 보완 아이디어에 대해 소개할 예정입니다.**

**슬라이드3**

**저희는 총 3명이었지만 전 팀장이었던 윤석님의 인턴십 합격으로 남은 기간 저와 재호님 두 명이서 프로젝트를 마무리하게 되었습니다.. 재호님은 github 관리, notion 작성, PPT 작성, 중간 발표, 알고리즘과 프론트 개발 등을 담당하셨습니다.**

**저는 notion 정리와 알고리즘 개발 및 분석, 아이디어 및 솔루션 제공을 담당하였습니다.**

**슬라이드4**

**저희의 주제는 상품 이미지를 통해 가로, 세로, 높이를 구하는 체적 측량 프로그램 개발입니다.**

**사람들이 흔히 쓰는 스마트 폰과 같이 간단한 방법으로 이미지를 체적 측량하는 프로그램을 개발하는 것을 목표로 하였습니다..**

**슬라이드5**

**다음은 저희가 프로젝트에서 수행중인 아이디어입니다.**

**길이가 4cm인 2D Aruco 마커를 사용하여 마커와 상품의 Pixel수 비교를 통해서 상품의 가로, 세로 길이를 cm로 출력해 줍니다.**

**또한 상품의 형태에 따라 원기둥은 1장의 이미지로 직육면체는 2장의 이미지로 측량을 하게 됩니다. 직육면체의 경우 정면 이미지와 상면 이미지의 가로 길이를 비교하여 오차가 적다면 가로 세로 깊이를 출력해줍니다.**

**첨부된 이미지는 저희가 예시로 만들어 본 이미지입니다.**

**6번 슬라이드**

**이 아이디어를 구현하기 위한 저희 프로그램의 알고리즘에 대해 설명해드리겠습니다.**

**먼저 알고리즘의 기본 순서에 대해 설명드리겠습니다.**

**첫번째 이미지를 불러옵니다.**

**두번째 불러온 이미지를 영역별로 나눠준 후 배경을 제거 합니다.**

**세번째 배경이 제거된 이미지를 이진화하여 외곽선을 검출합니다.**

**네번째 픽셀수 비교를 통해 cm로 변환시켜 가로, 세로를 출력합니다.**

**다섯번째 직육면체의 경우 정면과 상면의 이미지에서 가로, 세로 높이를 산출해줍니다.  
원기둥의 경우에는 한 장의 이미지로 지름과 높이를 산출해 줍니다.**

**7번 슬라이드**

**이어서 이런 알고리즘을 구현하기위해 사용된 라이브러리들 및 자세한 알고리즘들에 대하여 설명드리겠습니다.**

**먼저, Rembg는 U2net(u square net)이라는 딥러닝 모델을 사용하여 물체를 제외한 배경을 지워줍니다.**

**U2net의 기본 구조는 Unet과 유사하지만, 이름 에서도 어느 정도 유추 가능한 것처럼 Unet을 이중으로 사용한 모델입니다.**

**U2net은 각layer가 Unet의 형태로 되어 있으며 U2net의 전체적인 구조 또한 U자형을 띄고 있습니다.**

**다만 Unet과 다른 점은 upsampling과정에서 각 layer에서 만들어진 이미지들을 최종적으로 합쳐 Unet보다 더욱 정밀한 segmentation map을 제공해 줍니다.**

**8번 슬라이드**

**다음으로 이미지를 영역별로 나눠주는 알고리즘에 대해 말씀드리겠습니다.**

**이미지의 세로길이 가로길이 rgb값을 받아온 후 세로길이와 가로길이를 곱하여 이미지 전체의 크기를 구하고 세로길이, 가로길이, 전체크기를 반환해 줍니다.**

**이미지의 가로길이를 조정하여 사물이 놓여질 영역은 이미지의 가로길이의 2/3지점까지 마커가 놓여질 영역은 2/3지점부터 끝 지점까지로 지정하여 줍니다.**

**9번 슬라이드**

**외곽선을 검출해주는 알고리즘입니다.**

**Rembg를 통해 배경을 제거한 이미지를 이진화를 처리 하였습니다**

**이진화 파라미터의 수치가 높아질수록 노이즈가 많아지고 낮아질수록 이진화된 이미지의 경계가 불분명해져 여러 번의 실험을 통해 31을 최적의 크기로 정했습니다.**

**평균이나 가중평균에서 차감할 값은 많이 쓰이는 5로 지정하였습니다.**

**이진화 처리를 한 이미지를 통해 외곽선을 검출하였고, 리스트에 넣어준 후 외곽선 중 값이 제일 큰 외곽선을 리스트에서 가져왔습니다.**

**10번 슬라이드**

**가로 세로를 구분해주는 알고리즘 입니다.**

**OpenCV의 minAreaRect 함수를 통해 외접하면서 면적이 가장 작은 직사각형을 구하는데 활용하였습니다.**

**minAreaRect의 반환 값으로는 좌 상단 꼭지점 좌표와 가로 세로 폭 그리고 사각형이 기울어진 각도가 반환 됩니다.**

**각도는 (0, 90) 사이에 있으며 0에 가까운 값은 가로와 세로가 그대로 나오고, 90에 가까운 값은 가로와 세로가 뒤바뀌는 문제점이 있었기에 조건문을 사용하여 가로와 세로가 정해지게 하였습니다..**

**11번 슬라이드**

**마지막으로 저희 프로그램의 중심인 aruco marker를 사용한 방법에 대해 설명하겠습니다.**

**aruco 모듈에는 다양한 dictionary sizes와 marker sizes가 사전에 정의되어 있습니다.**

**이처럼 사전에 정의되어 있는 Parameter들을 생성해줍니다.**

**정의되어 있는 Parameter들은 마커를 detect하는데 사용되는 마커의 최소둘레 및 최대둘레 등과 같은 것들이 정의되어 있습니다.**

**마커를 찾을 이미지와 정의된 객체들을 통하여 마커를 detect하고 그 값을 전달해줍니다.**

**실제 마커의 길이는 4x4cm 이므로 전체 외곽선을 16으로 나눠 1cm당 픽셀의 수를 전달합니다.**

**12~13번 슬라이드**

**이런 알고리즘들을 이용해 만들어진 테스트 결과물을 보여드리겠습니다.**

**좌측 하단에 계란과자 이미지를 봐주시겠습니까?**

**첫번째 이미지의 가로는 12.0 세로는 12.3 두번째 이미지의 가로는12.1 세로는 4.9로 측정된 것을 확인해주시면 감사하겠습니다.**

**이어서 표를 봐주시겠습니까?  
표에 계란과자라고 적혀진 것을 보시면 실측 사이즈는 가로 12 세로13 높이 4.9 로 프로그램을 통해 측정 된 결과와 큰 오차가 없음을 알 수 있습니다.**

**계란과자 외에도 표에 보이는 것처럼 가장 작은 물체는 크기가 가로8.1 세로5.1 높이2.9인 치실통부터 가장 큰 물체는 크기가 가로17.7 세로 8.6 높이 25.3인 아이스티박스까지 다양한 크기의 물체들을 실험해 보았고 그 결과 오차가 그리 크지 않은 것을 확인할 수 있었습니다.**

**14~15번 슬라이드**

**이번에는 원기둥 형태의 상품을 측정한 결과 입니다.**

**이번에는 처음 이미지를 보시면 가로 6.9 세로 16.0으로 측정되었습니다.**

**그럼 바로 표에 기네스라고 적힌 부분을 보시면 실측 값은 가로6.5 세로 16.0으로 먼저 보았던 것보다도 오차가 적음을 확인하실 수 있습니다. 직육면체의 경우와 마찬가지로 정말 다양한 크기의 물체들을 실험해보았습니다.**

**다만 저희가 실험한 결과 직육면체의 전체 평균 오차율은 2.5%정도 였고 원기둥의 경우 3.4% 정도의 오차율로 평균적으로는 직육면체 형태의 이미지를 더 잘 측량하는 것으로 판단되었습니다.**

**16~17번 슬라이드**

**플라스크로 개발한 데모앱을 실제 사용한 영상을 보여드리겠습니다.**

**보시는 것처럼 직육면체의 경우 2장의 이미지를 원기둥의 경우 1장의 이미지를 넣어줘야 합니다.**

**이미지가 들어가 결과가 나올 때까지 약2초가량이 요구됩니다.**

**이어서 모바일버전을 보여드리겠습니다.**

**보시는 바와 같이 모바일에서도 잘 동작하는 것을 볼수있습니다.**

**17 ~ 18번 슬라이드**

**실험 기록표와 영상을 통해 보신 것처럼 나쁘지 않은 성능을 보여주지만 이 처럼 잘 측정하기 위해서 제한되는 사항이 4가지가 있습니다.**

1. **상품과 마커가 동일선상에 있어야 합니다.  
   마커가 상품보다 앞이나 뒤에 있다면 그만큼 오차가 생기게 됩니다.**
2. **상품을 찍을 때 찍고자하는 면만을 촬영해야 합니다.  
   가령 상면 또는 측면이 같이 나오게 촬영될 경우 그 만큼 외곽선이 크게 생성되어 오차가 생기게 됩니다.**
3. **상품의 정면 이미지의 가로와 상면이미지의 가로가 일치 해야합니다.  
   이를 지키지 않으면 프로그램 내부에서 가로길이를 비교하기 때문에 가로길이의 오차가 1cm이상으로 클 경우 재촬영을 요구하고  
   설령 1cm미만의 오차로 인해 측정을 해주더라도 잘못된 측정이 이루어지게 될것입니다.**
4. **상품과 마커 영역을 지켜서 촬영해야 합니다.  
   프로그램 알고리즘 상 상품을 detect하는 영역과 마커를 detect하는 영역이 나누어져 있어 이를 지켜 촬영하지 않으면 마커를 인식하지 않거나 상품을 인식하지 않아서 프로그램이 동작하지 않게 됩니다.**

**19번 슬라이드**

**마지막으로 향후 프로그램의 보완 및 업데이트에 관한 아이디어에 대해 말씀드리겠습니다.**

1. **사물의 그림자로 인한 오차를 줄이기 위하여 밝은 조명을 다각도에서 비추거나 촬영시 플래시를 터뜨려 그림자가 지지 않게하는 것이 좋을것입니다.**
2. **스마트폰의 앱으로 개발하게 된다면 사물영역과 마커영역을 구분해주는 선을 보여주가나 사용자가 임의로 선을 조정하여 영역을 나눌 수 있게하여 편의성을 높일 계획입니다.  
   또한 그림자를 제거해주는 알고리즘을 추가하여 그림자로 인한 오차를 제거할 계획입니다.**

**이상으로 발표를 마치겠습니다.**

**지금까지 긴 시간 저희 5조의 발표를 들어주셔서 감사합니다.**