- 최종 보고서 -

스마트폰 실측자 개발

지도 교수 : 이도훈 교수님 (B분반)
소속: 정보컴퓨터공학부
팀명 : 3G세대
팀원 : 201424436 김지홍
201424466 백지호
201424471 서지원

목차

I 개요

1.1 개발 배경

물체를 인식하고 측정하는 기술은 다양하게 응용된다. 기초적인 길이 측정부터 Machine learning과 통합되거나 AR 기술과 합쳐지기도 한다. User가 필요로 하는 Object 정보를 제공하기 위해선 물체를 정확히 인식하고 정밀하게 수치 값을 측정할 필요가 있다.

하지만 다양한 분야에 응용되는 측정 기술에는 한계가 있다. 일직선이 아닌 물체는 인식률이 떨어져 실측에 실패하기도 한다. 또 측정 위치에 따라 측정 각도, 측정 거리 등의 제약으로 측정값이 달라져 물체를 측정할 수 있는 환경에 제한이 생긴다. 기초적인 측정 기술의 한계는 측정을 바탕으로 하는 응용 기술에도 악영향을 미친다.

따라서 기초적인 실측 과정에서 물체 인식률을 높이고 길이를 측정할 수 있는 환경의 조건 범위를 확대할 필요가 있다. Real-time Estimate Application의 개발 과정에서 object measurement 기술이 갖고 있는 근본적인 한계를 개선한다면 이를 활용한 응요 기술의 활용도를 향상시키는데 도움이 될 것이다.

1.2 과제 목표

스마트폰으로 찍은 이미지에서 실제 크기를 알려주는 시스템.

1.3 유사 시스템 현황

ⓐ 작물 생육 측정 기반 스마트팜 2.0

(개발: KIST 강릉분원 SFS 융합연구단 김형석 선임연구원 연구팀, 한국전자통신연구원 등)

작물의 생육 정보를 수집하고 분석, 관리하는 Application이다. 스마트팜 2.0은 사진을 활용해 과실의 가로, 세로 직경을 분석하고 크기를 측정한다. 3D 카메라를 통해 정확한 부피를 측정할 수 있다. 이로 인해 줄기 직경, 생장 길이, 꽃의 수, 과실 수 등 총 10개 지표의 측정값을 자동으로 인식할 수 있다.

ⓑ 신발 사이즈 측정, Perfit

(개발: 기업 펄핏 Perfit)

Deep learning과 Image processing 기술을 이용해 발에 맞는 신발을 찾아주는 서비스다. User의 발을 스캐닝해서 영상분석 알고리즘을 활영해 발가락 모양, 아치, 꺾이는 각도 등을 측정할 수 있다. 신발 치수 데이터를 조합해 최적의 사이즈를 추천해준다.

ⓒ 에코룩

(개발: 아마존 Amazon)

아마존 에코룩(Echo Look)은 360도 모든 방향을 3D로 스캔한 뒤 Machine learning을 통해 사용자에게 어떤 옷이 잘 어울리는지 조언해주는 Application이다. 물체의 깊이(depth) 정보를 읽을 수 있는 카메라를 탑재해 사물을 입체적으로 인식할 수 있다.

Ⅱ 설계 과정

2.1 요구 조건

ⓐ 물체의 길이를 측정한다.

-스마트폰의 동영상을 통해 Real-time 측정이 가능하도록 한다.

ⓑ 물체 측정 과정에서 길이 측정값에 발생하는 오차의 범위를 줄인다.

-물체 인식의 정밀도 향상, 물체 깊이(depth) 측정을 통해 인식 개선

ⓒ Application을 이용한 측정 환경에 제약 조건을 개선한다.

-측정 각도, 측정 거리 등의 제한 범위를 넓힌다.

2.2 개발 환경

Android studio 3.4.2

삼성 갤럭시 S9, Android version 9.0

삼성 갤럭시 노트5, Android version 7.0

Knox API level 27

Knox API level 26

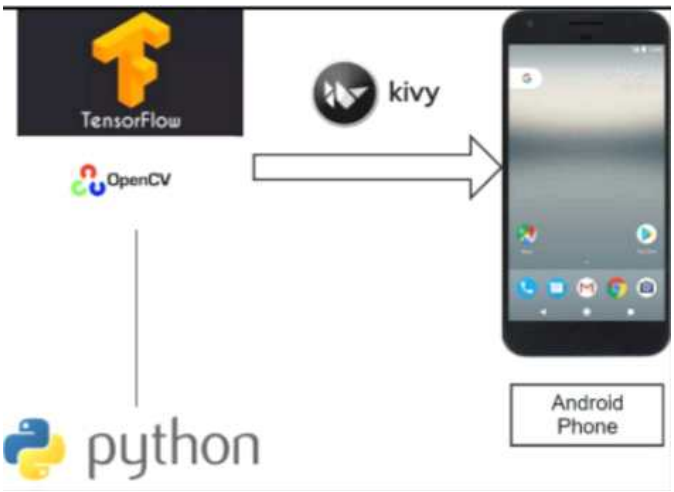
OpenCV Library 3.4

JNI Library

Ⅲ 제약 및 수정 사항

3.1 제약 사항

ⓐ Python을 개발언어로 kivy(*)를 Application 개발 소프트웨어로 사용하려고 했으나 Application layout 구현에 한계가 있고, Application 동작을 구현하는 reference가 부족해 사용에 제약이 있었다.



(*) kivy는 User Interface 를 이용한 모바일 앱과 다중터치 Application 소프트웨어를 개발하는데 사용되는 Open source Python 라이브러리다.

Android studio에서 Python 파일을 연동할 경우 제약 사항이 많기 때문에 Kivy를 통해 필요에 따라 Android studio를 거치지 않고 apk 파일을 생성할 수 있다.

ⓑ Tensorflow를 활용한 reference 인식으로 Object detection을 개선하려 했다. 하지만 모바일에서 동작하기에 Tensorflow는 무거운 프로그램으로서 메모리를 많이 차지해 Application 성능이 저하됐다.

ⓒ Template matching을 이용하려 했지만 각도에 따른 물체 인식에 어려움이 있었다. Template matching은 기기에 사진을 넣어 물체를 학습시키는 것이다. 하지만 스마트폰과 물체 사이의 각도가 변하면서 입력시킨 물체를 인지하지 못하는 문제가 발생했다. 기능 개선을 시도했으나 입력한 물체만 인식하는 Template matching 기술의 한계가 있었다.

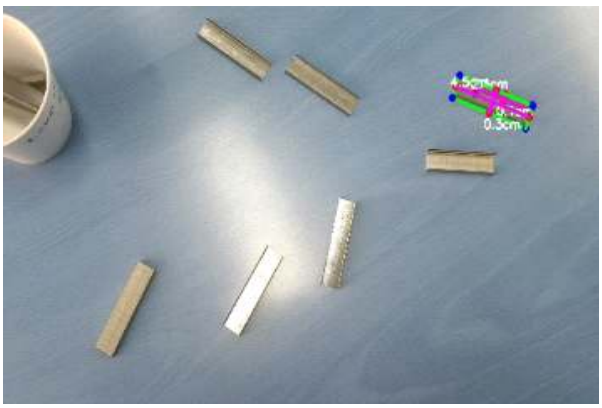
ⓓ 하얀색 바탕이어야 물체를 인식할 수 있는 제약 사항이 있다. 물체의 배경이 하얀 배경이 아니면 물체의 경계를 인식하는 과정에서 배경과 물체를 혼동할 수 있다. 또한 edge detection 과정에서 물체의 테두리가 둥글면 인식이 안될 수 있다.

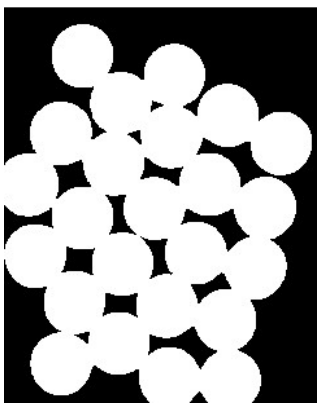
Ⅳ 수행 내용 및 실행 결과

4.1 물체 인식

ⓐ Binarization & Edge deteection

물체와 배경을 이진화를 통해 구분한다. 그리고 openCV의 threshold, findContours 라이브러리 등을 이용해 Binary 이미지에서 경계선을 검출한다. 경계선 검출을 끝내면 approxPolyDP 함수를 통해 다각형(polygon)을 검출할 수 있다. 해당 도형의 경계에서 꼭지점을 기준으로 물체를 인식한다.



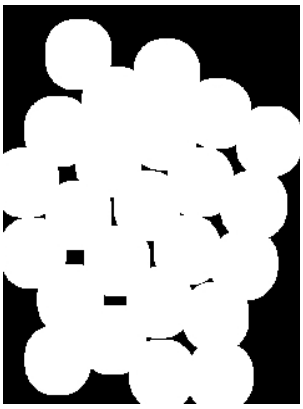
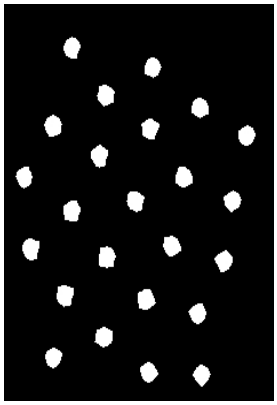
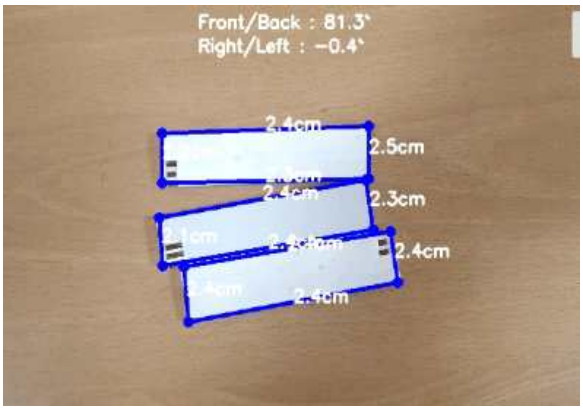


4.2 인식 성능 향상

ⓐ Image segmentation

Image segmentation을 이용해 윤곽선 검출을 더 정밀하게 할 수 있도록 했다. 이를 통해 영상에서 물체와 배경의 경계를 선뿐만 아니라 곡선까지 구분할 수 있도록 확대했다. 기존의 edge 이미지를 morphology(*) 함수로 형태를 뭉개어 정밀한 측정이 가능하다.

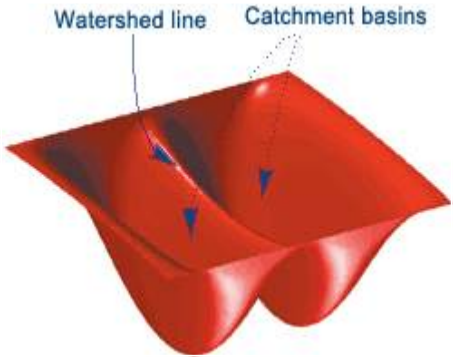
(*)morphology 함수는 침식 연산과 팽창 연산을 통해 영상에서 잡음을 제거한다.
침식 연산은 이미지의 크기를 작게 하고 팽창 연산은 객체의 크기를 키우며 작은 틈을 메운다.

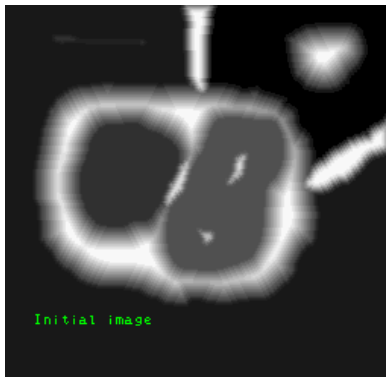




(morphology 함수로 edge를 뭉갠 뒤 이진화를 거리에 맞춰 이진화를 적용한 그림)

ⓑ Watershed algorithm

Watershed algorithm은 영상이 위상에 따른 입체감을 보여준다는 아이디어에 따라 물체의 높이(depth)를 측정하는 알고리즘이다. 물체의 depth에 따라 gray 이미지에는 픽셀 값의 차이가 나는데 해당 값의 차이를 0을 기준으로 계산해 높이를 알 수 있다.





4.3 물체 인식 환경 개선

ⓐ 자이로 센서를 이용한 object warping

모바일 기기에 탑재되어 있는 자이로 센서를 이용해 물체와 스마트폰 사이에 각도를 측정했다. 측정값을 물체 와핑(warping)에 접목해 스마트폰으로 물체를 비스듬히 측정할 수 있도록 환경을 개선했다. 이를 통해 물체와 스마트폰이 수직이 아닌 상태에서도 물체의 길이를 정확히 인식할 수 있도록 했다.

ⓑ Image rotation

Image rotation을 활용해 warping으로 인한 화면 깨짐을 보완했다. Object warping을 시도하면 물체뿐만 아니라 스마트폰 화면도 왜곡되는 현상이 발생했다. 이

를 해결하고자 Image rotation을 이용해 User가 화면을 볼 때 느낄 수 있는 불편함을 해소했다.

(비스듬히 측정하면 인식되지 않았던 물체가 와핑 이후에 명확하게 인식되었다.)

4.4 실험

ⓐ 실험 목적

물체 인식과 환경 개선을 하기전과 현재 성능이 어떤 차이가 있는지 확인하고 평가한다.

ⓑ 실험 방법

1) 하얀 백그라운드를 준비한다.

2) 레퍼런스 물체를 준비한다.

3) 레퍼런스 물체를 하얀 곳에 놔두고 카드, 지갑, 지우개의 길이를 90° 에서 측정한다.

4) 3)번의 내용을 각도(80° , 70° , 60°)만 다르게 하여 측정한다.

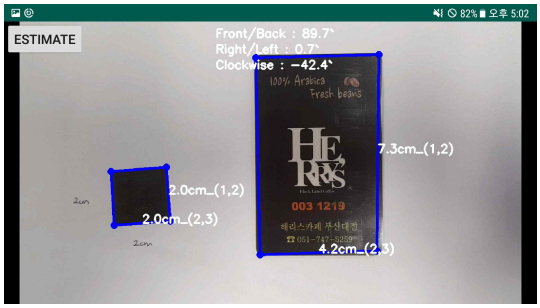
5) 3),4)번을 와핑을 사용하여 측정한다.

6) 각도를 크게 기울여 몇° 까지 물체 인식이 가능한지 확인한다.

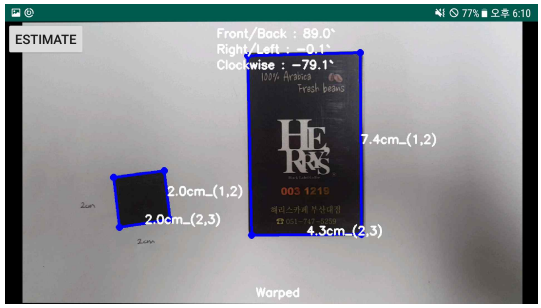
ⓒ 실험 데이터

물체 길이 측정 실험

	와핑 전/후	카드(실제크기- 4.5 x 7.3)	지갑 (9.7 x 7.6)	지우개 (5.5 x 13.7)
90°	before	4.2 x 7.3	9.6 x 8.6	5.2 x 14.3
	after	4.3 x 7.4	9.7 x 8.4	5.3 x 14.1
80°	before	4.4 x 7.1	10 x 8.3	5.4 x 13.8
	after	4.5 x 7.2	9.7 x 8.5	5.3 x 13.9
70°	before	4.4 x 6.9	10.4 x 8	6 x 13.6
	after	4.4 x 7.2	9.8 x 8.4	5.6 x 14.1
60°	before	4.5 x 7	9.9 x 8.8	5.9 x 13.7
	after	4.3 x 7.1	9.7 x 8.4	5.7 x 13.8

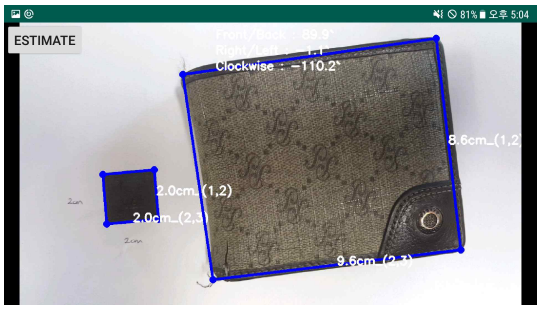


default

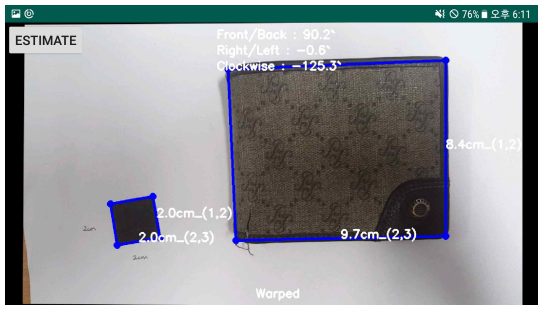


warped

90° 로 카드 측정

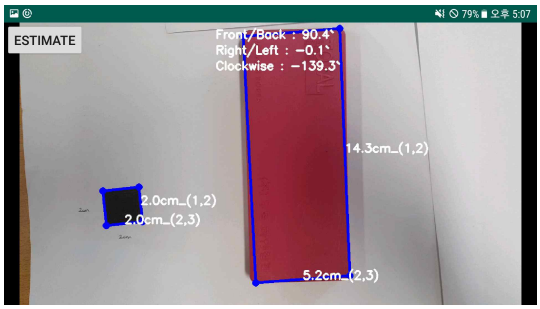


default

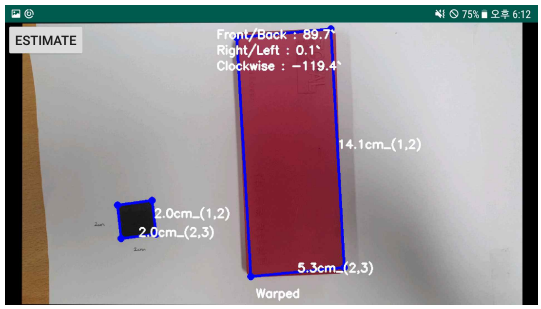


warped

90° 로 지갑 측정

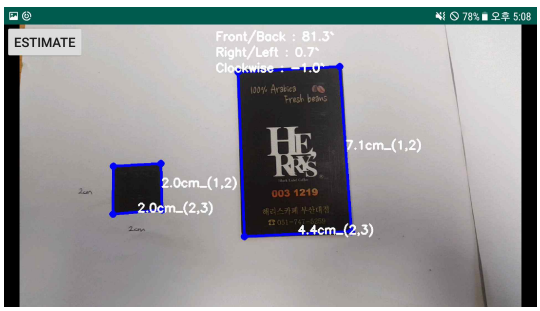


default

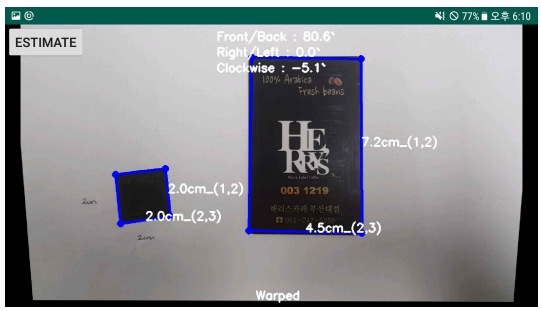


warped

90° 로 지우개 측정

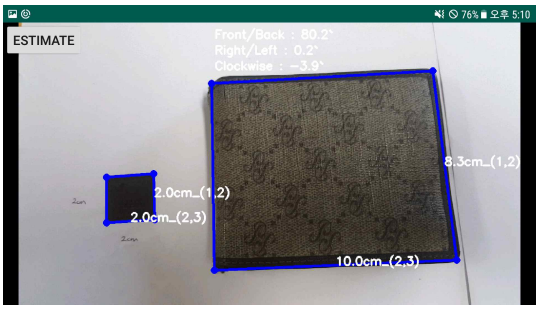


default

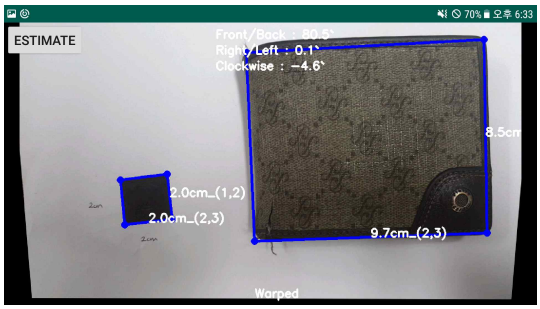


warped

80° 로 카드 측정

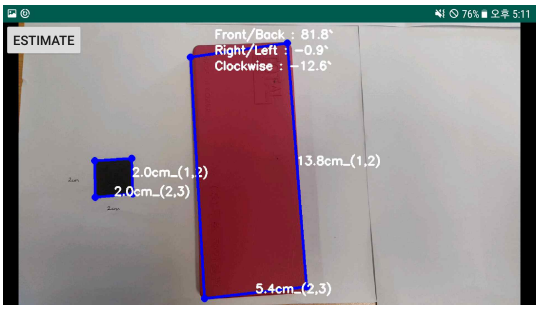


default

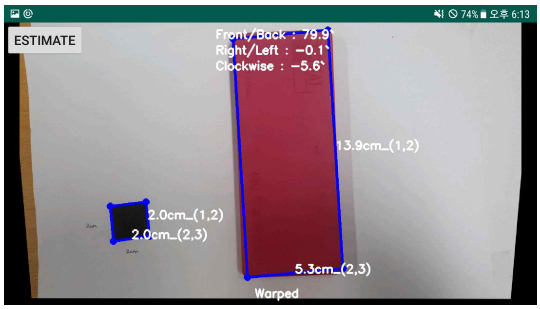


warped

80° 로 지갑 측정

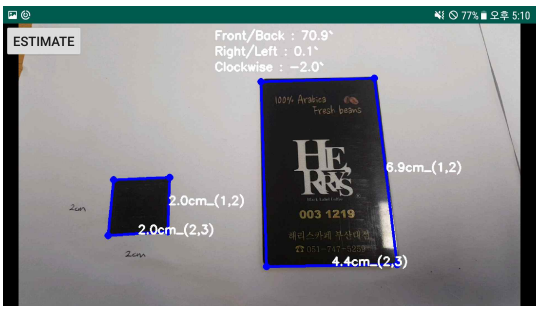


default

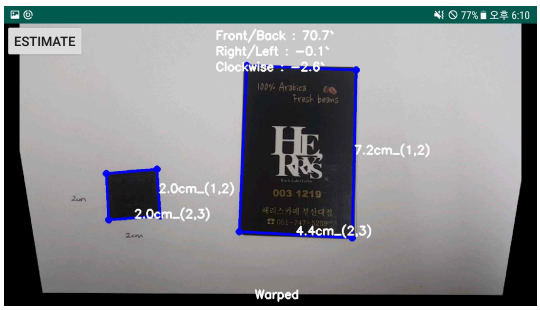


warped

80° 로 지우개 측정

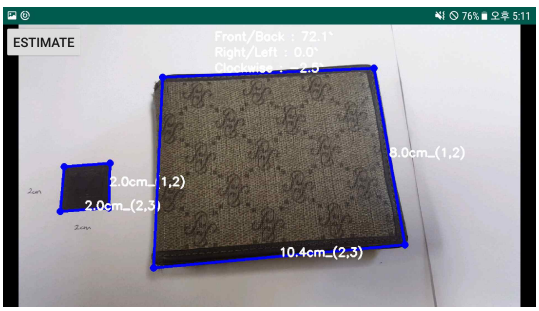


default

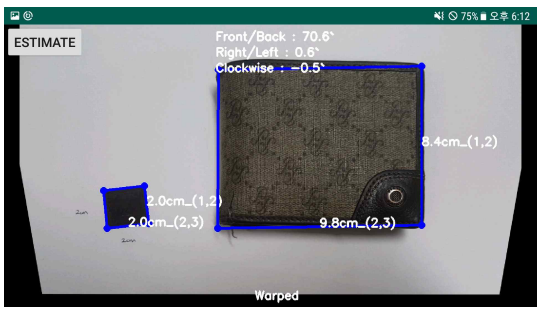


warped

70° 로 카드 측정

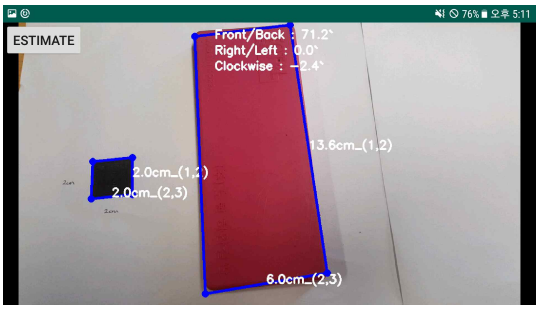


default

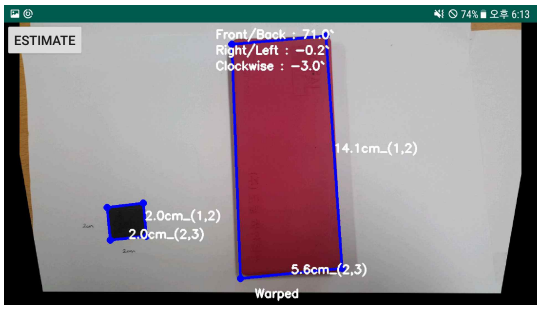


warped

70° 로 지갑 측정

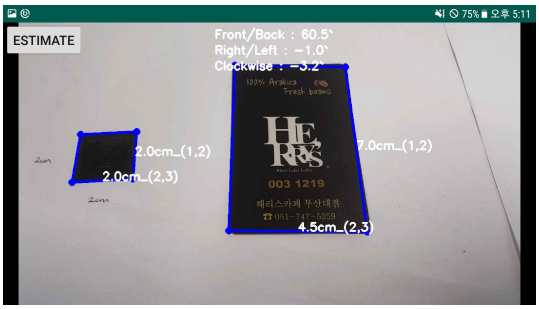


default

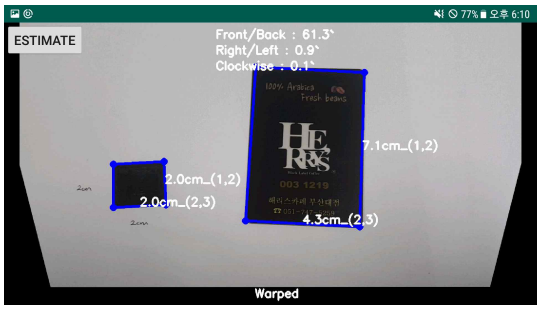


warped

70° 로 지우개 측정

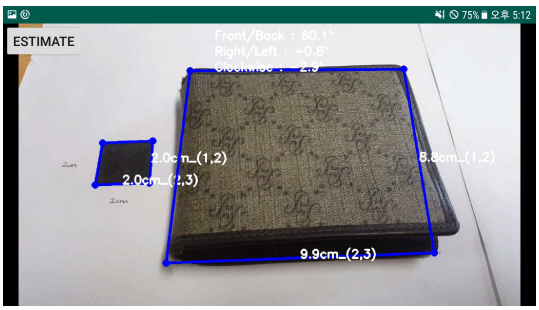


default

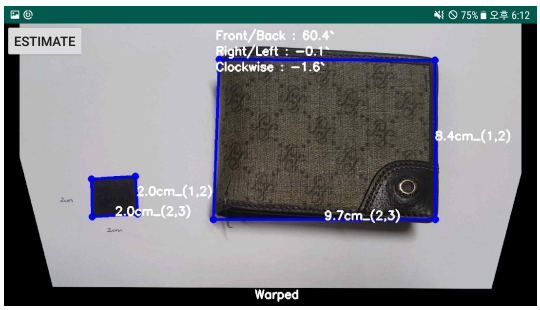


warped

60° 로 카드 측정

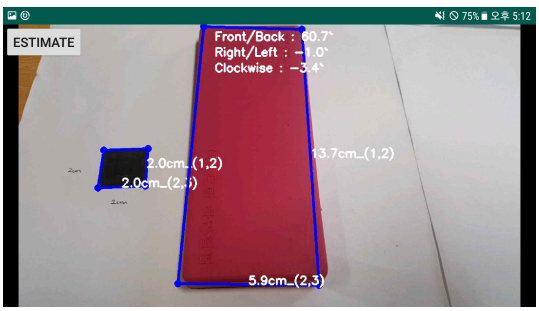


default

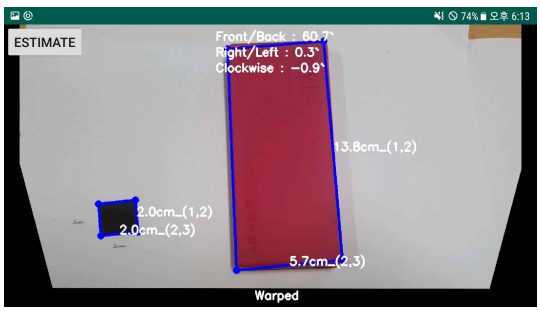


warped

60° 로 지갑 측정



default

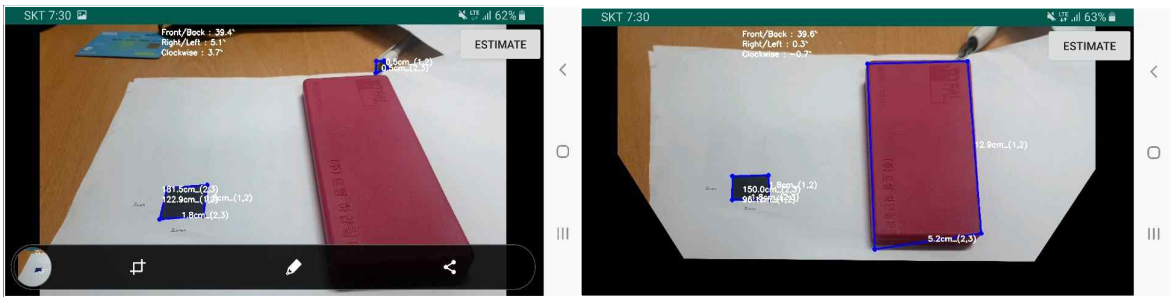


warped

60° 로 지우개 측정

각도 측정 범위 개선 결과

	와핑 전/후	인식할 물체
90°	before after	측정 가능 가능
70°	before after	가능 가능
40°	before after	불가능 가능 (측정 각도 범위 증가)
30°	before after	불가능 가능 (측정 각도 범위 증가)



ⓓ 실험 결과

실제 크기와 스마트폰으로 측정한 물체의 크기는 거의 동일했다. 그런데 와핑을 적용하면 이전과 큰 차이를 보일 것으로 예상했다. 하지만 실제 실험을 해본 결과 와핑 하기 전과 큰 차이는 없었다. 물체의 크기가 작기 때문에 이런 결과가 나타났을 것으로 예상된다. 반면에 스마트폰을 30°까지 기울이고, 와핑을 적용 하였을 때 물체를 정확하게 인식하는 것을 확인했다.

V 결론 및 개선 방향

5.1 결론

스마트폰을 사용하였고, image가 아닌 영상으로 측정하는 것을 목표로 하였기 때문에 최대한 opencv만 사용하여 개발하였다. 다른 라이브러리(tensorflow 등)를 사용할 경우 스마트폰에 과부화가 생길 수 있기 때문이다. opencv의 image segmentation과 wartershed algorithm을 활용하여 물체를 더 잘 인식 할수 있게 하였고, 스마트폰의 자이로 센서를 활용해 각도를 구했다. 스마트폰의 각도를 이용하여 스마트폰을 기울이더라도 우리가 정면에서 볼 수 있도록 개발하였다. 그 결과 길이 측정이 오차 범위 0.5cm 이내로 정확하게 되는 것을 확인 하였다.

5.2 개선 방향

현재 물체를 측정 할 때 화면의 백그라운드가 하얀색이어야 한다는 제약 조건이 있는데, yolo 라이브러리를 활용하면 이를 해결 할 수 있다. 스마트폰을 위한 tiny yolo 라이브러리도 출시되어 있다고 하니 이를 사용하여 측정 환경을 개선하여야 한다.

그리고 현재는 레퍼런스와 물체를 한 평면상에 두고 측정 하여야 정확한 값이 측정 되는데, 높이가 다른 경우에도 정확하게 측정 할 수 있는 방법도 찾아보면 좋을 것 같다.

VI 수행 체계

6.1 개발 일정

일자	제목	내용	비고
5/3	사전조사(논문,제품 등)	논문: 센서 활용해 물체 인식하는 방법 제품: 1)작물 생육측정 기반 스마트팜 2.0 2)신발사이즈 측정 3)방어체중 측정 4)google tango 5)Amazon 에코룩	
5/10	openCV환경 설정 논문 추가 조사	openCV 기본적인 동작(카메라, 이미지 출력) 논문: 1)3차원 점군 데이터로 좌표 변환(어파인 변환) 2)스마트 CamRuler(3축 자이좌표, 가속 센서)	
5/17	해외 논문 조사	Harris coner detector	
5/24	착수보고서 초안	착수보고서 초안 교수님 검토	
5/31	착수보고서 완성	-Time table(일정표, 한 일, 해야할 일) 작성 -시중에 있는 제품(Smart Measure, Measure Google 등) 조사하고 Table로 정리 -제품 토대로 착수 목표 설정.	

7/8	물체 인식 기초 작업	그레이스스케일, 이진화를 이용해서 image 탐색, Video 탐색	
7/15	UI 작업 물체 인식 기초	1.물체 길이 측정과 레퍼런스 인식 2. User Interface 설계 및 기초 작업	
7/22	시스템 흐름도 작성	1. 시스템 흐름도 작성 2. UI에 카메라 결합	
7/29	UI에서 측정할 수 있도록 결합	1. UI카메라에서 측정이 가능하도록 결합. 2. UI 디자인 개선	
8/5	python에서 Android studio로 변경		
8/12	실측 환경 완성 실험값 측정	각도 별로 다양한 실험값 측정	
8/21	인식 성능 개선을 위한 목표 변경	tensorflow를 이용해 물체 인식 개선하기로 결정.	tesorflow는 모바일에서 무거울 수 있어서 적합하지 않음(피드백)
8/30	물체 인식 성능 개선	image segmentation과 watershed를 이용해 와핑 시도	
9/6	와핑을 통한 인식 개선	와핑, 화면 비율 개선	
9/13	와핑을 통한 인식 개선	레퍼런스 인식 개선 화면 깨짐 현상 개선	

6.2 역할 분담

김지홍 :

Binarization과 Edge detection 자료 조사, Image detection 구현, 실험 데이터 축적, 중간 및 최종 보고서 검수

백지호:

Binarization과 Edge detection 자료조사, Image detection 자료조사 및 구현, Wateshed 자료조사, 실험 설계 및 데이터 축적, 중간 및 최종 보고서 작성

서지원:

Binarization과 Edge detection 구현, Image detection 구현, Watershed 구현, 실험 설계 및 데이터 축적

Ⅶ 산학협력 활동보고서

팀 명	3G 세대	
팀 원	김지홍, 백지호, 서지원	
과 제 명	스마트폰 실측자 어플리케이션 개발	
산 업 체 멘 토	기 업 명	㈜유캔스타
	성 명	최대길 (대표)

세부 활동 내용
1. 산학협력 멘토링 내용 기본적인 물체의 길이 측정은 완성해냈다. 또한 와핑을 통해 카메라와 수직이 아닌 각도에서도 물체를 잘 인식하도록 했다. 하지만 레퍼런스를 인식하는데는 openCV 라이브러리만으로는 인식이 힘들다. Yolo 등을 통해 reference 인식률을 높일 필요가 있다. 레퍼런스와 다른 높이에 있는 물체도 원근감에 구애받지 않고 길이를 개선하도록 개선할 수도 있다. **2. 산학협력 멘토링 후기** openCV 만으로 레퍼런스를 인식하기 위해 노력했으나, 라이브러리의 한계가 있음을 알게 됐다. 또한 단순히 각도와 측정 거리에 따른 물체 인식률을 개선하는 것 외에 원근감이나 밝기 등 다양한 조건이 물체 인식에 영향을 미침을 깨달았다. 이를 통해 어플리케이션의 발전 방향에 대해 다양하게 생각해볼 수 있는 계기가 됐다.

	
회사 전경	멘토링 회의 모습