

# 종합설계 프로젝트 수행 보고서

프로젝트명	딥 러닝 기반 알약 이미지 식별 서비스
팀번호	S3-11
문서제목	수행계획서( ) 2차발표 중간보고서( ) 3차발표 중간보고서( ) 4차발표 중간보고서( ) 최종결과보고서( O )

2020.12.04

팀원 : 안현진 (팀장)  
배선영  
양소영  
이상현

지도교수 : 전광일 교수 (인)  
지도교수 : 박정민 교수 (인)

## 문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2020.01.22	안현진(팀장)	1.0	수행계획서	최초작성
2020.02.26.	양소영(팀원)	2.0	2차발표 중간보고서	설계서추가
2020.04.27.	배선영(팀원)	3.0	3차발표 중간보고서	Prototype 추가
2020.06.27	이상현(팀원)	4.0	4차발표 중간보고서	시험결과추가
2020.12.04	안현진(팀장)	5.0	최종결과보고서	결론추가

## 문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (4월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I II III
	II. 본론 (1~3)	II. 본론 (1~4)	II. 본론 (1~5)	II. 본론 (1~7)	
	참고자료	참고자료	참고자료	참고자료	

이 문서는 한국산업기술대학교 컴퓨터공학부의 “종합설계” 교과목에서  
프로젝트 “딥 러닝 기반 알약 이미지 식별 서비스”를 수행하는  
(S3-11, 안현진, 배선영, 양소영, 이상현)들이 작성한 것으로 사용하기  
위해서는 팀원들의 허락이 필요합니다.

# 목 차

## I. 서론

1. 작품선정 배경 및 필요성 .....
2. 기존 연구/기술동향 분석 .....
3. 개발 목표 .....
4. 팀 역할 분담 .....
5. 개발 일정 .....
6. 개발 환경 .....

## II. 본론

1. 개발 내용 .....
2. 문제 및 해결방안 .....
3. 시험시나리오 .....
4. 상세 설계 .....
5. Prototype 구현 .....
6. 시험/ 테스트 결과 .....
7. Coding & DEMO .....

## III. 결론

1. 연구 결과 .....
2. 작품제작 소요재료 목록 .....

참고자료 .....

# I . 서론

## 1. 작품선정 배경 및 필요성

필요성	내 용
수요배경	<ul style="list-style-type: none"> <li>• 의학, 화학 기술의 발전으로 여러 종류의 알약이 만들어지고 시중에 유통되고 있음.</li> <li>• 저시력자나 노인층이 알약 섭취 시 작은 글씨로 인해 약 정보를 얻기 어려움.</li> <li>• 색맹인 사람의 경우 특정 약의 색깔을 인지하기 어려워 약을 구분하기 어려움.</li> <li>• 많아지는 알약의 개수에 비해 알약의 효능과 부작용에 대한 정보의 접근성이 낮아 약을 오용하는 상황이 발생함.</li> </ul>
필요성 1	<ul style="list-style-type: none"> <li>• 약은 중요하고 부작용의 위험이 크므로 분류하고 관리하는 것이 매우 중요함.</li> <li>• 많은 사람들이 건강제, 보조제 등의 여러 약품을 취급하지만 보관상의 실수로 인해 약을 재분류해야 하는 경우가 있음.</li> </ul>
필요성 2	<ul style="list-style-type: none"> <li>• 약은 그 효과와 부작용에 대한 정보를 아는 것이 매우 중요함.</li> <li>• 약에 대한 정보 제공 서비스가 부족하고 정보 접근 능력에 따른 정보 획득의 격차가 커 쉽게 약의 정보에 접근할 수단이 필요함.</li> </ul>
필요성 3	<ul style="list-style-type: none"> <li>• 약에 대한 접근성은 높아지고 있지만 이를 분류하고 이에 대한 정보를 취급하는 것이 어렵기 때문에 약에 대한 분류 및 정보 제공 시스템이 필요함.</li> </ul>

## 2. 기존 연구 / 기술동향 분석

개발내용	내 용
국내 관련연구 소개, 장점, 단점	<ul style="list-style-type: none"> <li>■ 휴대형 근적외선/가시광선 분광기를 이용한 의약품 분류기법               <ul style="list-style-type: none"> <li>• (소개) 휴대형 분광기를 사용한 의약품 분류기법에 대해 소개함.</li> <li>• (장점) 저가의 소형 센서인 LinkSquare 분광기를 사용하므로 제품의 휴대성과 가격경쟁력을 높임.</li> <li>• (단점) 전처리된 데이터를 학습하는데 SVM 기법만을 사용했기에 정확도와 처리속도 개선이 요구됨.</li> </ul> </li> <li>■ 알약 인식을 위해 색 특징정보를 이용한 CBIRS/TB               <ul style="list-style-type: none"> <li>• (소개) CBIRS/TB 모델을 이용하여 다양한 색상 및 형태를 갖는 알약의 분류법을 소개함.</li> <li>• (장점) 영상 특징값을 추출하는 과정에서 전체 영상의 특징정보와 부분영역에 대한 특징정보를 동시에 이용하여 검색 정확도가 높음.</li> <li>• (단점) 영상 전체 픽셀에 대한 연산으로 인한 처리시간의 증가로 시스템 서비스로의 접근성이 낮음.</li> </ul> </li> <li>■ 알약 자동 인식을 위한 딥 러닝 모델 간 비교 및 검증               <ul style="list-style-type: none"> <li>• (소개) YOLO, Faster R-CNN, RetianNet과 같은 다양한 딥 러닝 모델들을 사용하여 알약 분류에 적합한 딥 러닝 모델을 소개함.</li> <li>• (장점) 알약 분류를 위한 최적의 딥 러닝 모델은 Faster R-CNN임을 보임.</li> <li>• (단점) 세 가지 모델들이 공통적으로 흰색의 원형 알약의 분류에 낮은 정확도를 보임.</li> </ul> </li> </ul>
국외 관련연구 특징, 장점, 단점	<ul style="list-style-type: none"> <li>■ Automatic text imprint analysis from pill images               <ul style="list-style-type: none"> <li>• (소개) 서로 다른 약 모양으로 인해 약사가 겪는 어려움을 해소하고자 알약에 인쇄된 글자를 학습 후 자동으로 인식 및 분석해주는 시스템을 제안함.</li> <li>• (장점) 알약에 인쇄된 글자를 5단계(정규화-임프린트 식별-Oust's-노이즈제거 및 클러스터링-Tesseract에 입력) 구성을 통해 시스템의 인식도를 향상시킴.</li> <li>• (단점) 데이터 셋의 부족으로 시스템의 알약 인식도가 0.773 정도로 상용화되기에 시스템의 정확도가 낮음.</li> </ul> </li> <li>■ Accurate system for automatic pill recognition using imprint information               <ul style="list-style-type: none"> <li>• (소개) 기존에 사용하던 인쇄된 글자 추출 알고리즘에서 보이는 노이즈와 프래그먼트의 휘도 변화를 예방하기 위한 새로운 알고리즘이 제안된 알약에 인쇄된 글자를 분석해주는 시스템을 제안함.</li> <li>• (장점) 2단계 샘플링 거리 세트(TSDS)라는 디스크립터를 거리 세트에 기초하여 시스템에 적용해 알약에 인쇄된 글자의 파티션에 따른 특징점을 추가하여 노이즈 및 불필요한 프래그먼트의 교란을 배제하는 방식으로 정확도를 향상시킴.</li> <li>• (단점) 이미지의 상태가 좋지 않을 경우 인식률이 급격하게 낮아지는 현상을 보임.</li> </ul> </li> <li>■ Automatic Drug Pills Detection based on Convolution Neural Network               <ul style="list-style-type: none"> <li>• (소개) 딥 러닝 및 CNN(Convolution Neural Network)을 기반으로 알약의 특징을 추출하여 특징 파라미터를 구성해 계층화된 구조를 사용하는 시스템을 제안함</li> <li>• (장점) 계층화된 새로운 구조를 사용함으로써 약물 폐기에 소요되는 많은 인적 자원을 줄임.</li> <li>• (단점) 특정한 틀에서 찍힌 정형화된 사진만을 인식하여 실생활에 사용할 수 있는 편리성이 부족함.</li> </ul> </li> </ul>

### 3. 개발 목표

#### 최종목표

- 딥 러닝(Deep Learning)을 이용해 알약을 식별하는 서비스를 구현하여 저시력자들이 약을 복용하는 상황에서 보다 편리하게 약을 구분하고 복용 정보를 획득할 수 있게 함.

단계	내 용
1단계	<ul style="list-style-type: none"> <li>■ 딥 러닝 기반의 이미지 식별 서비스 구현을 위한 조사 <ul style="list-style-type: none"> <li>• 구현 기반이 되는 이미지의 특징을 추출하는 방법 및 딥 러닝 학습에 대해 조사함.</li> <li>• 딥 러닝 학습을 위한 데이터 셋을 수집함.</li> <li>• TTS(Text to Speech) 기술에 대해 조사함.</li> </ul> </li> </ul>
2단계	<ul style="list-style-type: none"> <li>■ 딥 러닝 기반의 이미지 식별 서비스 구현을 위한 설계 <ul style="list-style-type: none"> <li>• 소프트웨어 공학 기반의 애플리케이션 구현 프로세스를 수립함.</li> <li>• 수집한 데이터를 이용하여 딥 러닝 학습을 설계함.</li> <li>• 알약 이미지에서 문자 및 특징을 추출할 수 있는 방법을 설계함.</li> </ul> </li> </ul>
3단계	<ul style="list-style-type: none"> <li>■ 딥 러닝 기반의 이미지 식별 서비스 구현 <ul style="list-style-type: none"> <li>• open API의 데이터 셋을 활용하여 알약 이미지를 학습함.</li> <li>• 딥 러닝을 통한 알약 이미지의 상세 특징 정보 추출을 학습함.</li> <li>• 추출 정보를 바탕으로 해당 알약을 데이터베이스에서 검색함.</li> <li>• 검색된 결과를 애플리케이션 상에서 시각 및 음성으로 제공함.</li> </ul> </li> </ul>
4단계	<ul style="list-style-type: none"> <li>■ 구현한 서비스에 대한 테스트 <ul style="list-style-type: none"> <li>• 저시력자가 알약을 식별하는 여러 상황에 대한 테스트 시나리오를 작성함.</li> <li>• 테스트 시나리오에 대한 단위 테스트를 진행 및 데이터를 재학습함.</li> </ul> </li> </ul>

### 4. 팀 역할 분담

이 름	학 번	사용가능 언어	역 할
안현진 (조장)	2015150020	• C++, Java	• 서버(DB) 구축
배선영 (조원1)	2015150016	• Python, C	• 딥 러닝 시스템 구현
양소영 (조원2)	2017154020	• Java, Python	• 안드로이드 애플리케이션 구현
이상현 (조원3)	2017154028	• Java, C++	• 딥 러닝 시스템 구현

## 5. 개발 일정

항목	추진사항	12월	01월	02월	03월	04월	05월	06월	07월	08월	09월	10월
요구사항 정의 및 분석	<ul style="list-style-type: none"> <li>요구사항 정의 및 분석</li> <li>요구사항 명세</li> </ul>	■										
시스템설계 및 상세설계	<ul style="list-style-type: none"> <li>시스템 설계</li> <li>상세 설계</li> </ul>	■	■									
구현	<ul style="list-style-type: none"> <li>코딩</li> </ul>		■	■	■	■						
시험 및 데모	<ul style="list-style-type: none"> <li>시스템 통합시험</li> <li>졸업작품 완성성 보강</li> </ul>					■	■	■				
문서화 및 발표	<ul style="list-style-type: none"> <li>졸업작품 중간 보고서 작성 (중간보고서, 사용자 매뉴얼 작성)</li> <li>발표(전시회, 산업기술대전)</li> </ul>							■	■		■	
산업기술대전	<ul style="list-style-type: none"> <li>산업 기술대전 참가</li> </ul>										■	
졸업작품 최종 보고서 작성 및 패키징	<ul style="list-style-type: none"> <li>졸업작품 최종보고서 작성</li> <li>CD 패키징(문서, 사용법, 프로그램, 개발환경, 데모 동영상 등)</li> </ul>											■

## 6. 개발 환경

개발 환경	상 세
개발 언어	Java, Python, Javascript
사용 프레임워크	Android Studio, Pycharm, NodeJS
주요 라이브러리	Numpy, TensorFlow, OpenCV, YOLO
서버 프로세서	Linux Ubuntu 18.04.02
DB	MySQL

## II. 본론

### 1. 개발 내용

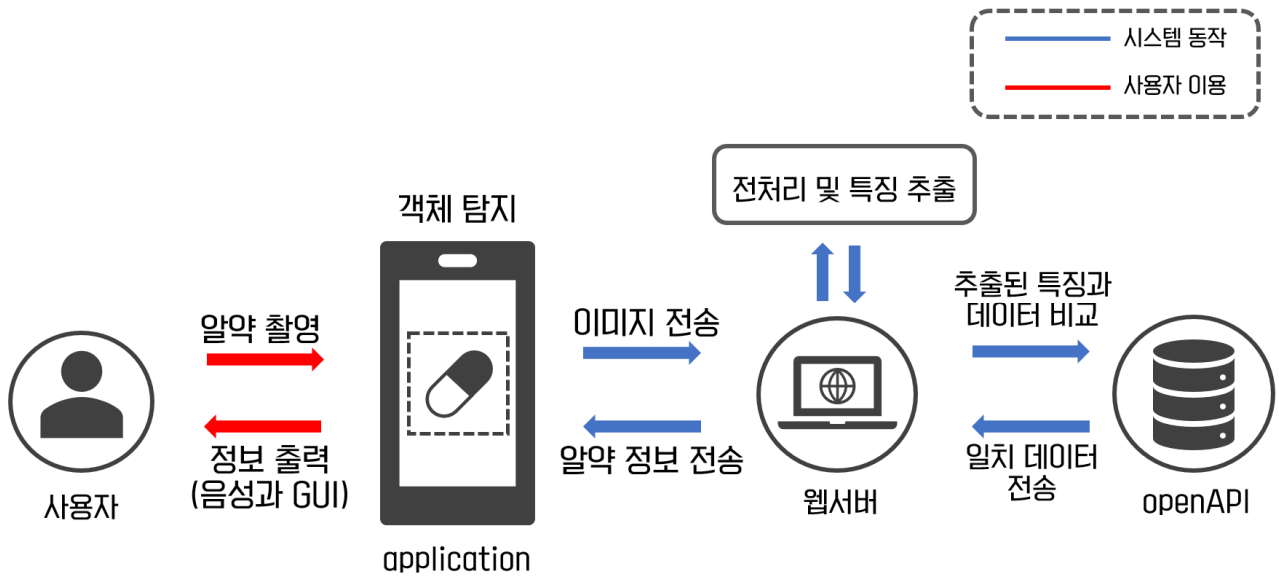
순번	개발 내용
1	<ul style="list-style-type: none"> <li>■ 입력된 사진 이미지의 특징을 추출하는 딥 러닝 시스템 개발 <ul style="list-style-type: none"> <li>• OpenCV의 Bilateral Filter를 기반으로 이미지 화질을 개선함.</li> <li>• 알약의 색상만 추출하기 위해 배경색을 제거해주는 Grap-Cut 알고리즘과 오브젝트가 위치하는 개별 영역을 파악해주는 Grad-CAM 알고리즘 이용해 알약 색상을 추출함.</li> <li>• 알약 이미지에 대해 템플릿 매칭(Template Matching) 알고리즘을 이용하여 알약의 모양을 추출함.</li> <li>• 알약 이미지에 대해 OpenCV를 활용해 윤곽(Contour)을 추출하고 Inception-v3 모델로 학습하여 알약의 글자를 추출함</li> <li>• 각 특징 추출 알고리즘을 종합한 하나의 딥 러닝 시스템을 개발함.</li> <li>• 가공되지 않은 데이터 셋에 대한 딥 러닝 결과를 웹 서버에 저장함.</li> </ul> </li> </ul>
2	<ul style="list-style-type: none"> <li>■ 사용자 인터페이스를 제공하는 애플리케이션 개발 <ul style="list-style-type: none"> <li>• YOLO를 이용하여 객체를 인식하는 카메라 기능을 구현함.</li> <li>• 웹 서버에서 전송받은 결과 값을 GUI로 출력할 수 있도록 구현함.</li> <li>• 사용자가 요청한 추가정보에 대해 음성으로 출력함.</li> </ul> </li> </ul>
3	<ul style="list-style-type: none"> <li>■ 사용자와의 통신 및 데이터 처리를 위한 웹 서버 구축 <ul style="list-style-type: none"> <li>• 추출된 데이터와 데이터베이스에 저장된 데이터를 비교하여 결과 값을 애플리케이션에 전송함.</li> </ul> </li> </ul>
4	<ul style="list-style-type: none"> <li>■ 딥 러닝 학습 결과 및 공공 알약 API에 대한 데이터베이스 구축 <ul style="list-style-type: none"> <li>• 딥 러닝 학습 결과를 저장함.</li> <li>• 공공데이터 API를 통해 얻어온 알약에 대한 데이터를 저장함.</li> </ul> </li> </ul>

### 2. 문제 및 해결방안

	상세
문제점	<ul style="list-style-type: none"> <li>■ 알약 검색 시스템의 낮은 서비스 접근성 <ul style="list-style-type: none"> <li>• 기존의 시스템은 컴퓨터에서만 실행되는 서비스로 다양한 상황에서 시스템 사용이 불가능함.</li> </ul> </li> <li>■ 데이터 셋의 부족으로 인한 시스템의 낮은 알약 인식률 <ul style="list-style-type: none"> <li>• 머신 러닝 학습은 많은 데이터 셋을 필요로 하는데 시스템의 높은 인식률을 위한 알약에 대한 데이터가 부족함.</li> </ul> </li> </ul>
해결방안 1	<ul style="list-style-type: none"> <li>■ 서비스 접근성이 높고 사용이 편리한 알약 식별 서비스를 제안 <ul style="list-style-type: none"> <li>• 알약 식별 애플리케이션을 구현하여 사용자에게 편리함을 제공하는 시스템을 구축함.</li> <li>• 사용자가 요청한 알약의 사진을 분석하는 서버를 구축하고 분석 결과를 애플리케이션에서 가시화하여 사용자에게 제공함.</li> </ul> </li> </ul>
해결방안 2	<ul style="list-style-type: none"> <li>■ 높은 인식률을 위한 딥 러닝 기반의 다각도 학습을 제안 <ul style="list-style-type: none"> <li>• open API의 데이터 셋 뿐만이 아닌 여러 각도에서 촬영한 사진을 추가로 학습시켜 인식률을 높임.</li> </ul> </li> </ul>

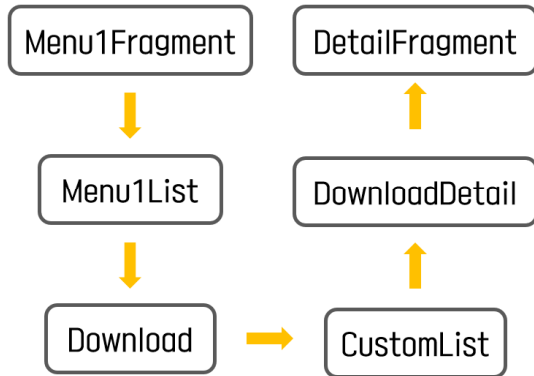


### 3. 시험시나리오



## 4. 상세 설계

### 1) 검색 모듈 및 통신 모듈



#### Menu1Fragment

검색화면을 보여주고 사용자의 검색어 옵션을 menu1\_list에 전달함

#### CustomList

웹서버에서 가져온 데이터를 보여줌

#### Download

웹서버 통신을 통해 검색 결과를 받아옴

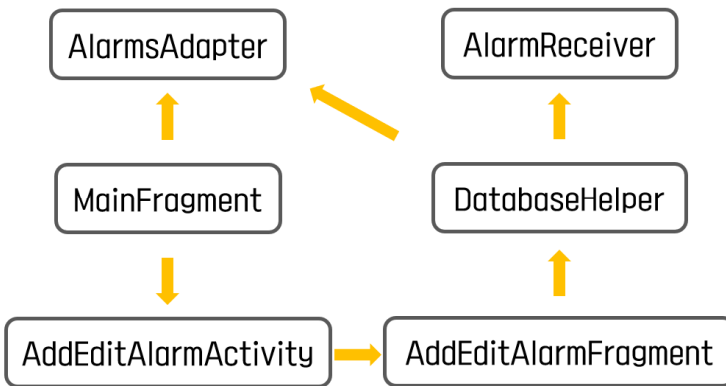
#### Menu1List

menu1Fragment에서 받아온 파라미터 값을 통해 download를 실행함

#### DownloadDetail

특정 알약의 상세정보를 웹서버에서 받아옴

### 2) 알람 모듈



#### MainFragment

데이터베이스에 저장된 내용을 adapter를 통해 보여줌

#### AddEditAlarmFragment

알람 추가, 편집 화면을 보여줌

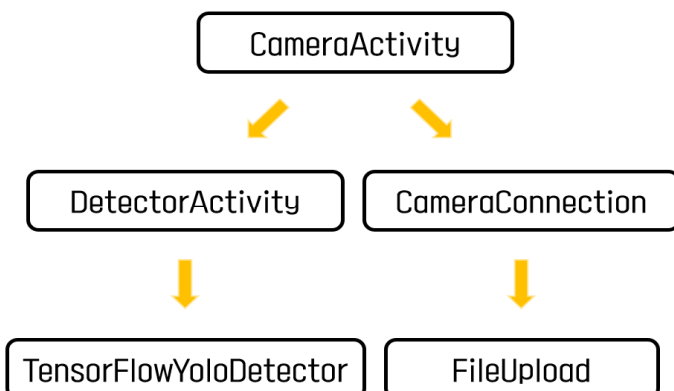
#### AlarmReceiver

데이터베이스에 저장된 알람을 등록함

#### DatabaseHelper

Alarm객체를 데이터베이스에 저장함

### 3) 객체 탐지 모듈



#### CameraActivity

사용자에 따른 카메라 및 프리뷰 크기 설정, CameraConnection 객체 생성

#### CameraConnection

카메라 작동 시작 및 프리뷰 생성, 이미지 캡처

#### DetectorActivity

TensorflowYoloDetector 객체 생성, 인식률이 80%이상인 객체에 대한 좌표 측정

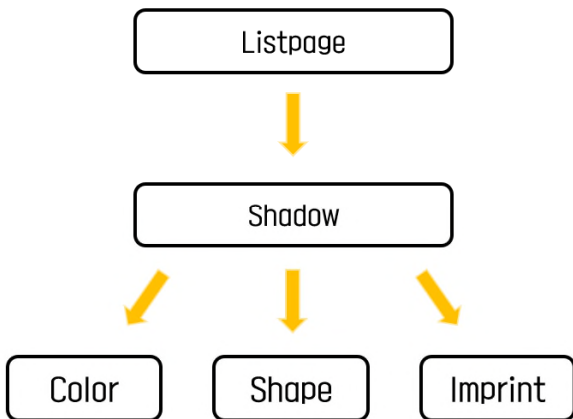
#### TensorFlowYoloDetector

객체 클래스 개수와 라벨 설정, 학습된 데이터를 가지고 객체 Detection

#### FileUpload

imageReader로 부터 받아온 이미지와 알약의 좌표를 웹 서버에 전송

#### 4) 처리 모듈



#### Listpage

앱으로부터 이미지 수신 및 모듈 호출 후 결과 전송

#### Shadow

Remove.Background API를 사용하여 이미지 전처리 과정 수행

#### Color

전처리된 이미지에 대해서 색상 평균 계산 후 색상 값 반환

#### Shape

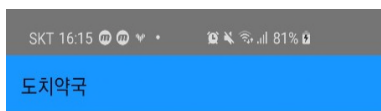
전처리된 이미지에 대해서 Contour 탐지를 통해 모양 값 반환

#### Imprint

전처리된 이미지에 대해서 히스토그램 분석을 통한 글자 영역 추출 후 딥러닝 학습 결과값 반환

## 5. Prototype 구현

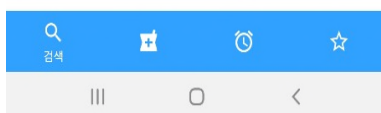
### 1. 검색



텍스트로 검색



사진으로 검색



알약이름

식별표시

☐ 포함 ☐ 일치

색상

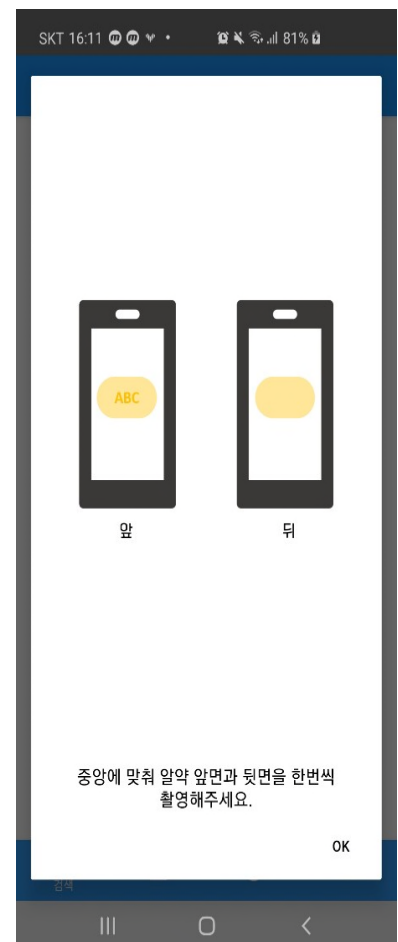


모양

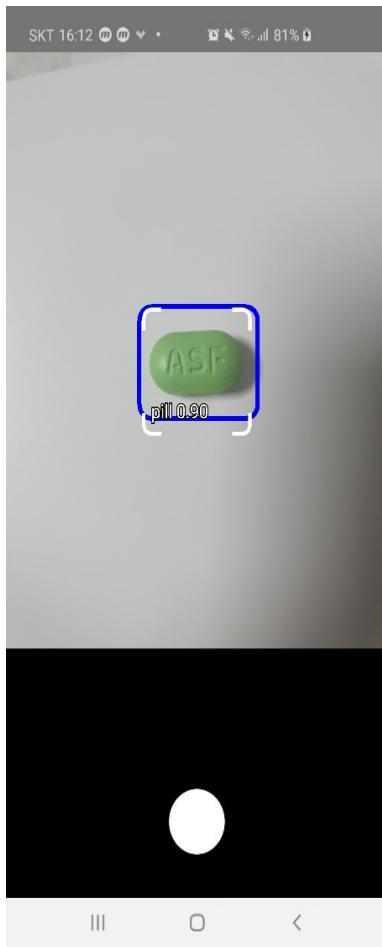


초기화

검색



## 2. 객체 탐지



```
{ connection: 'keep-alive',
  enctype: 'multipart/form-data',
  'content-type': 'multipart/form-data;boundary=*****',
  'user-agent':
    'Dalvik/2.1.0 (Linux; U; Android 10; SM-G973N Build/QP1A.190711.020)',
  host: '59.14.252.86:3000',
  'accept-encoding': 'gzip',
  'content-length': '57740' }
[ { fieldname: 'files',
  originalname: 'camera.jpg',
  encoding: '7bit',
  mimetype: 'text/plain',
  destination: 'images',
  filename: 'camera.jpg',
  path: 'images/camera.jpg',
  size: 57547 } ]
```

Green

POST /listpage 200 213.306 ms - 5

```
{ connection: 'Keep-Alive',
  enctype: 'multipart/form-data',
  'content-type': 'multipart/form-data;boundary=*****',
  'user-agent':
    'Dalvik/2.1.0 (Linux; U; Android 10; SM-G973N Build/QP1A.190711.020)',
  host: '59.14.252.86:3000',
  'accept-encoding': 'gzip',
  'content-length': '57884' }
[ { fieldname: 'files',
  originalname: 'camera2.jpg',
  encoding: '7bit',
  mimetype: 'text/plain',
  destination: 'images',
  filename: 'camera2.jpg',
  path: 'images/camera2.jpg',
  size: 57690 } ]
```

POST /imgback 200 18.288 ms - 16

## 3. 이미지 처리



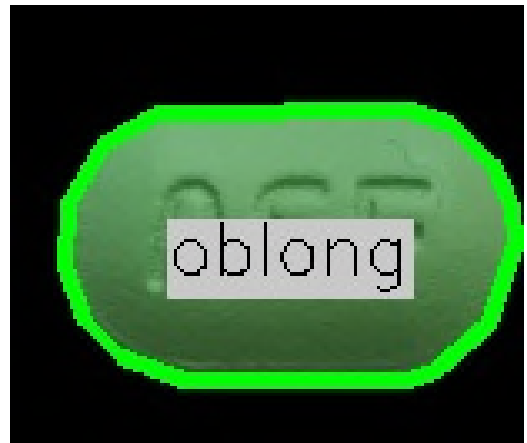
< 이미지 소형화 >



< 배경 제거 >



< 색상 인식 >



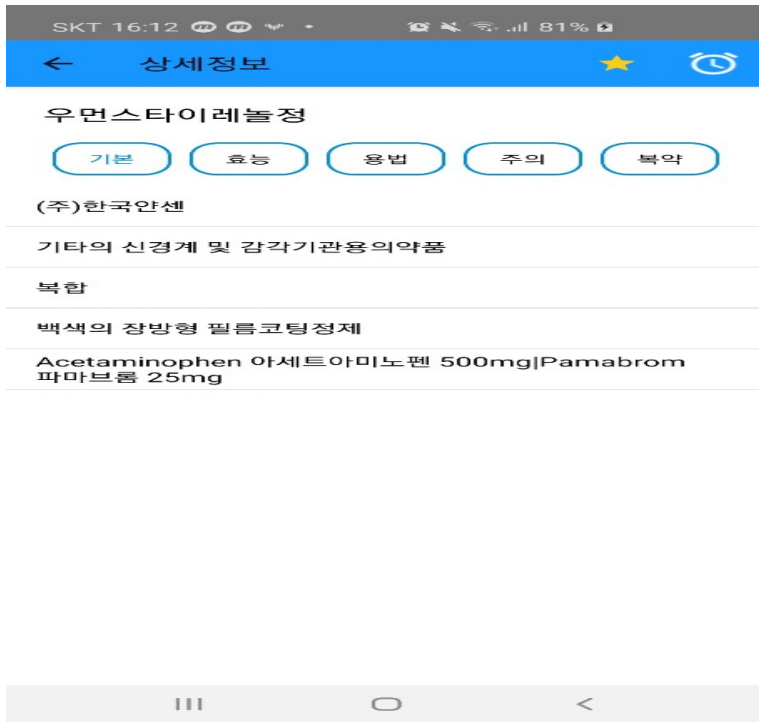
< 모양 인식 >



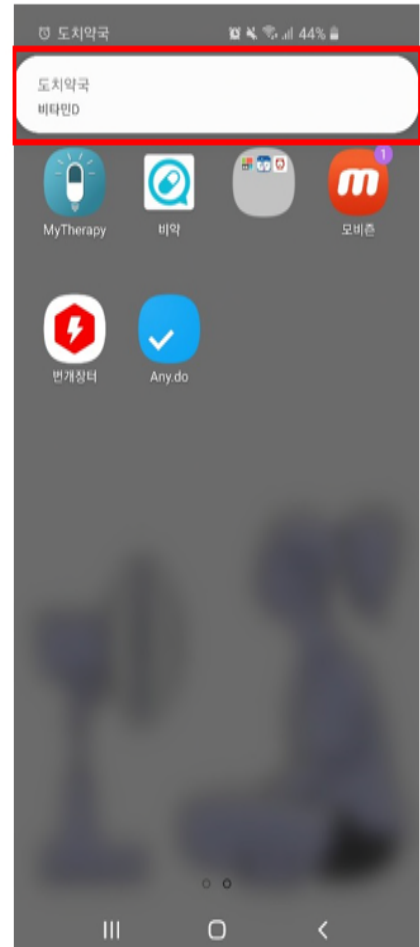
|  
ASF

< 글자 인식 >

#### 4. 결과 확인



#### 5. 알람



## 6. 시험/테스트 결과

PILL DATA	Shape	Color	Imprint	Result	List
image1	oblong	Yellow	NONE	Correct	1
image2	circle	White	NONE	Correct	0
image3	circle	White	NONE	Correct	0
image4	oblong	Pink	CENTRUM	Correct	0
image5	oblong	White	TYLENOL	Correct	3
image6	oblong	Green	ASF	Correct	1
image7	oblong	White	TYLENOL	Correct	3
image8	oblong	Pink	SW	Failure	4

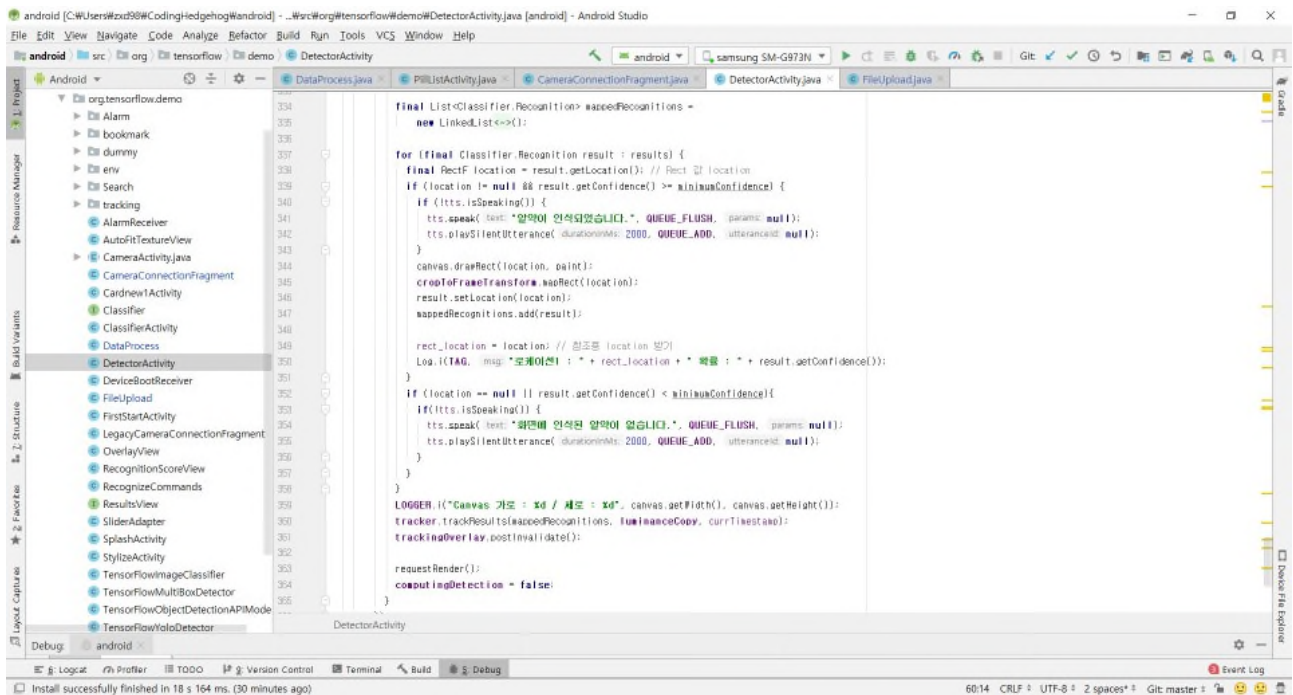
(중략)

image43	oblong	Blue	NONE	Correct	6
image44	oblong	Pink	CENTRUM	Correct	0
image45	oblong	White	TYLENOL	Correct	3
image46	oblong	White	TYLENOL	Correct	3
image47	oblong	Green	ASF	Correct	1
image48	oblong	White	TYLENOL	Correct	3
image49	circle	Green	IDPQ	Correct	0
image50	oblong	Orange	MGB	Correct	0
분석 정확도 : 82.0%					
검색 결과 : 33					



## 7. Coding & Demo

### < 안드로이드 객체탐지 코드>



```

final List<Classifier.Reognition> mappedReconitions =
    new LinkedList<>();

for (final Classifier.Reognition result : results) {
    final RectF location = result.getLocation(); // Rect 값 location
    if (location != null && result.getConfidence() >= minimumConfidence) {
        if (!tts.isSpeaking()) {
            tts.speak( text: "알약이 인식되었습니다.", QUEUE_FLUSH, params: null);
            tts.playSilentUtterance( durationInMs: 2000, QUEUE_ADD, utteranceId: null);
        }
        canvas.drawRect(location, paint);
        cropOfFrameTransform.mapRect(location);
        result.setLocation(location);
        mappedReconitions.add(result);

        rect_location = location; // 참조본 location 받기
        Log.i(TAG, msg: "오제이션 : " + rect_location + " 확률 : " + result.getConfidence());
    }

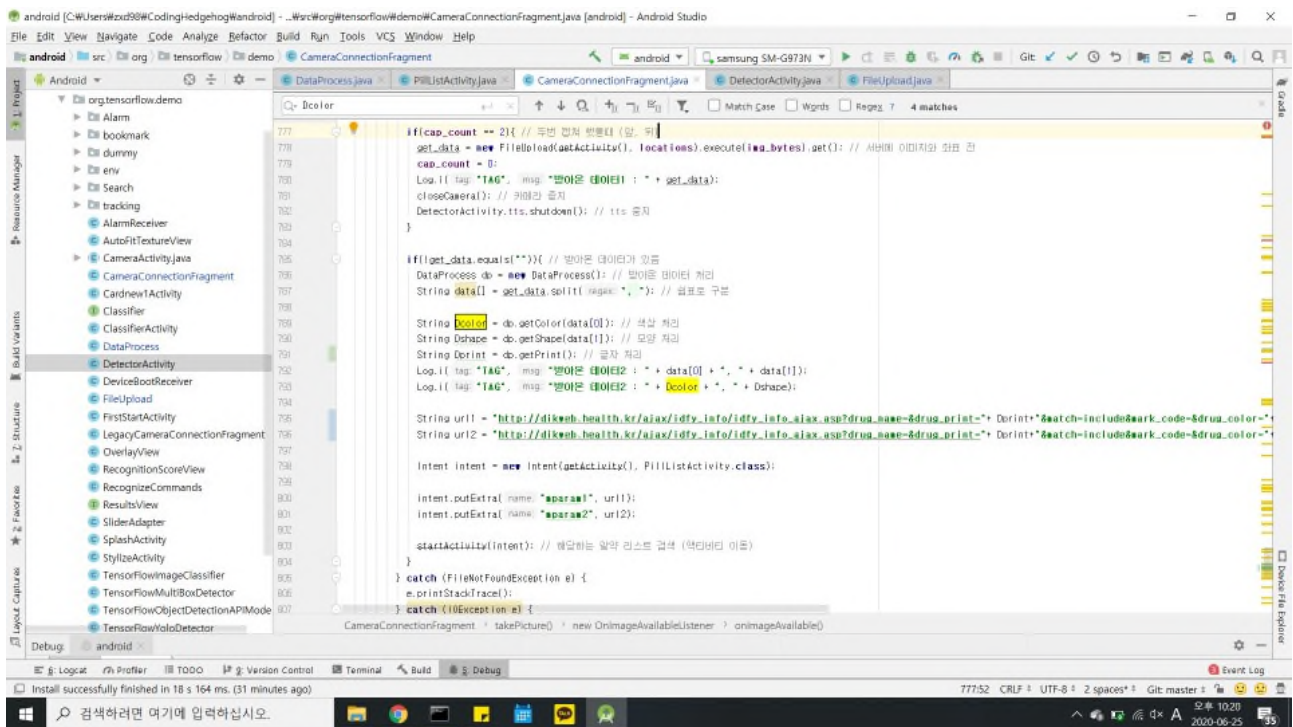
    if (location == null || result.getConfidence() < minimumConfidence) {
        if (tts.isSpeaking()) {
            tts.speak( text: "알약이 인식된 알약이 없습니다.", QUEUE_FLUSH, params: null);
            tts.playSilentUtterance( durationInMs: 2000, QUEUE_ADD, utteranceId: null);
        }
    }

    LOGD.i("Canvas 가로 : $d / 세로 : $d", canvas.getWidth(), canvas.getHeight());
    tracker.trackResults(mappedReconitions, frameInanceCrop, currTimeStep);
    trackingOverlay.postInvalidate();

    requestRender();
    computingDetection = false;
}

```

### < 안드로이드 알약 검색 코드 >



```

if (cap_count == 2) { // 두번 검색 실패한 경우
    get_data = new FileUpload(getActivity(), locations).execute(new byte[] {}); // 서버에 이미지와 좌표 보
    cap_count = 0;
    Log.i("TAG", msg: "방어문 데이터1 : " + get_data);
    closeCamera(); // 카메라 중지
    DetectorActivity.tts.shutdown(); // tts 중지
}

if (get_data.equals("")) { // 방어문 데이터가 없음
    DataProcess dp = new DataProcess(); // 방어문 데이터 처리
    String data[] = get_data.split(" "); // 좌표로 구분

    String color = dp.getColor(data[0]); // 색상 처리
    String Dshape = dp.getShape(data[1]); // 모양 처리
    String Dprint = dp.getDprint(); // 문자 처리
    Log.i("TAG", msg: "방어문 데이터2 : " + data[0] + " , " + data[1]);
    Log.i("TAG", msg: "방어문 데이터2 : " + color + " , " + Dshape);

    String url1 = "http://dtkweb.health.kr/ajax/idfy-info/idfy-info_ajax.asp?drug_name=&drug_print=" + Dprint + "&match-include=&mark_code=&drug_color=" +
    String url2 = "http://dtkweb.health.kr/ajax/idfy-info/idfy-info_ajax.asp?drug_name=&drug_print=" + Dprint + "&match-include=&mark_code=&drug_color=" +

    Intent intent = new Intent(getActivity(), PiListActivity.class);

    intent.putExtra(name: "sparam1", url1);
    intent.putExtra(name: "sparam2", url2);

    startActivity(intent); // 방대한 알약 리스트 검색 (엑티비티 이동)
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
CameraConnectionFragment.takePicture() > new OnImageAvailableListener() > onImageAvailable()

```



## < Node 서버 데이터 처리 코드 >

```
146
147     for i in range(0, 10):
148         start = time.time()
149         COLOR_AVG = colorAverage()
150         end = time.time()
151
152         duration = end - start
153         DURATION_SUM += duration
154
155     DURATION_AVG = DURATION_SUM / 10
156     find_color = rgb2colorname.color_detect(red, green, blue)
157
158     co_width, co_height = color_img.shape[:2]
159     center_x = int(co_height / 2.0)
160     center_y = int(co_width / 2.0)
161     font = cv2.FONT_ITALIC
162     (text_width, text_height), baseline = cv2.getTextSize(find_color, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 1)
163     location = (center_x-30, center_y + int(text_height/2))
164     # cv2.rectangle(color_img, (center_x-50, center_y+baseline), (center_x+text_width-50, center_y-text_height), (200,200,200), cv2.FILLED)
165     cv2.putText(color_img, find_color, location, cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255,0,0), 1, 8)
166     cv2.imwrite('images/color_result2.png', color_img)
167     # print('average color: rgb(%s, %s, %s)' % (COLOR_AVG))
168     # print(round(minWidth/8), round(minHeight/4), minWidth, minHeight)
169
170     img = cv2.imread("images/result.jpg")
171     tmp = img
172     #shadow.shadow_api()
173     rs_width, rs_height = img.shape[:2]
174     shadow.grabcut(5, 10, rs_height-10, rs_width-5)
175     # print("width : ", rs_width-5, "height : ", rs_height-10)
176     find_shape = shape.shape()
177     print(find_color+'', find_shape)
```

## < Node 서버 색상 인식 알고리즘 코드 >

```
1315     # Lookup color name using Hex:ColorName dictionary:
1316     NearestRGB = (RGB[spatial.KDTree(RGB).query(pt)[1]])
1317
1318     # TODO: Calculate Hex from pt. (upper case letters)
1319     # Instead of str(hex(pt[0])[2:]) in Python2, this is Python3 compatible:
1320     s = '#' +
1321         + format(NearestRGB[0], 'x').zfill(2) +
1322         + format(NearestRGB[1], 'x').zfill(2) +
1323         + format(NearestRGB[2], 'x').zfill(2)
1324     ColorHex = s.upper() # "#8B7355" # "#8B7355"
1325     ColorDiff = #
1326         + ('+' + '{0:d}'.format(NearestRGB[0]-pt[0]) +
1327         + ', '+' + '{0:d}'.format(NearestRGB[1]-pt[1]) +
1328         + ', '+' + '{0:d}'.format(NearestRGB[2]-pt[2]) +
1329         + ')')
1330     try: ## TODO: try catch block per https://wiki.python.org/moin/HandlingExceptions
1331         ColorName=HexNameDict[ColorHex]
1332     except:
1333         ColorName="not found"
1334     #print('Nearest color name to input RGB ' +
1335         # + str(pt) +
1336         # + ' is "' + ColorName + '"' +
1337         # + ' ' + ColorHex +
1338         # + ' ' + str(NearestRGB) +
1339         # + ' ' + ColorDiff +
1340         # + ',')
1341     # print(ColorName)
1342     return ColorName
1343     #return ColorName
1344     # print 'Is the above correct?'
```

## < Node 서버 이미지 전처리 코드 >

```
17 def grabcut(ix, iy, x, y):
18     global img, img2, drawing, value, mask, rectangle
19     global rect, rect_or_mask, rect_over
20
21     img = cv2.imread("images/result.jpg")
22     img2 = img.copy()
23
24     width, height = img.shape[:2]
25     mask = np.zeros(img.shape[:2], dtype=np.uint8)
26     output = np.zeros(img.shape, np.uint8)
27
28     count = 0 # 카운트 설정
29     cv2.rectangle(img, (ix, iy), (x, y), RED, 2)
30     rect = (min(ix, x), min(iy, y), abs(ix - x), abs(iy - y))
31     rect_or_mask = 0
32
33     while True:
34
35         k = ord('n')
36         count += 1
37
38         if count > 20: # 20번 수행
39             k = 27
40             mask_inv = cv2.bitwise_not(mask2)
41             empty_img = 255 * np.ones(shape=(width, height, 3), dtype=np.uint8) # 빈 이미지 생성
42             empty_img = cv2.bitwise_and(empty_img, empty_img, mask=mask_inv)
43
44             resize = output[5:int(width)-5, 5:int(height)-5]
45             resize = cv2.resize(resize, dsize=(height, width), interpolation=cv2.INTER_CUBIC)
46
47             bit = cv2.bitwise_and(resize, resize, mask=mask_inv)
48             bit = cv2.bitwise_not(bit)
49             cv2.imwrite("images/edge.jpg", bit) # 엣지 이미지 저장
```

## < 애플리케이션 데모 이미지 >



### III. 결론

#### 1. 연구결과

본 연구는 다양한 약이 개발되고 약에 대한 접근성이 높아지는 현대 속에서 약 오용의 원인이 되는 보관 및 식별 문제를 해결하고자 하였다.

효율적인 결과물을 위해 약을 오용하게 되는 경우를 분석하였고 약을 식별하기에 어려움이 있는 즉, 저시력자나 시각장애인들이 약 오용 문제에 크게 노출 될 것이라 생각하였다. 때문에 저시력자나 시각장애인들에게 더욱 초점을 맞춘 연구의 필요성을 느꼈고 스스로 약을 식별해주고 복용을 도와주는 애플리케이션을 만들기로 하였다. 높은 알약 인식률을 위해 딥러닝(YOLO)과 그림자 제거, 모양 추출, 색상 추출, 글자 인식(OCR)등의 모듈을 사용하였다. 결과적으로 높은 인식률을 보이는 딥 러닝 기반 알약 식별 서비스를 완성하였다.

본 연구는 기존에 존재했던 알약 서비스를 확장한 연구이지만, 기존 서비스가 해결하지 못한 저시력자 및 시각장애인의 알약 식별 서비스의 이용을 해결하려 하였다. 본 연구는 기존의 기술 서비스 이용에 제약을 받는 장애인들을 위한 새로운 기술 서비스를 선보이고 있는 현대 사회의 흐름에 있어 큰 의의를 갖는다.

향후 더 발전된 서비스를 위해서는 연구 중 가장 큰 문제였던 흰색 알약의 글자 인식과 다양성에 대한 문제를 해결하기 위한 연구가 진행되어야 한다.

#### 2. 작품제작 소요재료 목록

이름	내용
Amazon EC2(Ubuntu)	웹 서버, 데이터베이스 등을 올리기 위한 시스템 환경
Removal Background API	이미지 인식률을 높이기 위한 이미지 전처리
YOLO 알고리즘	카메라 프레임 내 알약 존재 여부 파악
색상 추출 알고리즘	이미지 내 알약의 색상을 추출
모양 추출 알고리즘	이미지 내 알약의 모양을 추출
글자 추출 알고리즘	이미지 내 알약의 글자를 추출
TTS 모듈	획득한 텍스트 정보를 음성으로 변환

#### IV. 참고자료

- [1] Hee-won Kim, 'Case Study on Services for the Blind', <http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09294043>
- [2] Object Detection, <https://junyoung-jamong.github.io/>
- [3] Dae-wook Kim, 'Shape and Text Imprint Recognition of Pill Image Taken with a Smartphone',  
<https://s-space.snu.ac.kr/bitstream/10371/137361/1/000000146053.pdf>