

<별첨 10>

4차산업혁명을 선도하는 CODE형 SW 인재 양성

Capstone Design(종합설계) 결과보고서

교과목명		빅데이터캡스톤디자인		학부(과)명	빅데이터 전공		
작품명(한글)		그게 무슨 약인지 알약?					
작품명(영문)		What kind of drug is that?					
팀 명		강약중강약					
지도교수		김태운					
참여 학생	대표	소속학과	학번	성명	연락처		
		빅데이터	20175321	윤소영	010-4926-0468		
	팀원	소속학과	성명	소속학과	성명		
		빅데이터	김한나				
		빅데이터	송기룡				
		빅데이터	윤수정				
		빅데이터	함서은				
참여 분야		<input type="checkbox"/> SW융합대학 <input type="checkbox"/> SW융합/연계 전공					
참여기업		(주)HD테크놀로지					
지식재산권		유형	<input type="checkbox"/> 특허 <input type="checkbox"/> 실용신안 <input type="checkbox"/> 상표 <input type="checkbox"/> 디자인 <input type="checkbox"/> 기타()				
		명칭(출원명)				번호	
깃허브주소		https://github.com/songkiryong/Capstone_design					

과제 요약 (한글)
<p>사용자가 알약을 찍어 업로드하면 해당 알약에 대한 상세 정보를 나타내주는 알약 분석 앱을 개발함.</p> <p>yolov5를 기반으로 알약의 모양을 detect하였고 (원형, 장방형, 타원형), OpenCV로 알약의 색깔을 검출함.</p> <p>Tesseract OCR 로 알약에 있는 문자를 인식하여 결과값을 네이버 의약품 지식 백과 URL로 넘겨줌. 해당 알약에 대한 세부 사항을 파싱하여 사용자에게 정보를 제공함.</p>
과제 요약(영문)
<p>When the user takes a picture of the pill and uploads it, We developed a pill analysis app that displays detailed information about the pill.</p> <p>The shape of the pill was detected based on yolov5(round,oblong,oval), and the color of the pill was detected with OpenCV.</p> <p>Tesseract OCR recognizes the characters in the pill and sends the result to the Naver Pharmaceutical Knowledge Encyclopedia URL.</p> <p>It parses the details about that pill and provides information to the user.</p>
오픈소스 사용내역
<ul style="list-style-type: none"> - Expo : 안드로이드 어플 제작 - 네이버 의약품 지식백과 : 알약의 세부사항 - YOLOv5 ,OpenCV, Tesseract OCR
과제 목적
<p>1)우리는 질환으로 병원을 내원하거나 갑작스러운 통증으로 약을 처방받거나 따로 제품을 사서 복용한다. 하지만 약의 정보가 적혀있는 처방서나 설명서를 잃어버리는 경우 약의 형태와 종류가 비슷하여 어떤 약인지 주관적으로 판단하기 때문에 잘못 복용했을 경우, 약물의 오남용이 발생할 수 있다. 이러한 경우 약물에 대한 정확한 정보가 필요하다.</p> <p>2)약학정보원이나 네이버, 구글, 알약 검색 앱 등 쉽게 알약을 검색해주는 다양한 플랫폼들이 있다. 하지만 사용자가 직접 알약의 생김새와 색깔, 식별문자가 무엇인지 제대로 인식해야 하며 인식한 토대로 항목을 체크하는 방식으로 결과를 도출해내는 사이트나 앱들이 대다수이다. 그래서 사용자들이 쉽고, 정확한 정보를 얻을 수 있도록 하는 프로그램이 필요하다고 생각했다.</p> <p>1),2) 의 필요성에 의해서 우리는 알약 분석 프로그램을 개발하고자 한다.</p> <p>일상 생활에서 자주 접할 수 있는 알약의 종류(감기약, 생리통 약, 소화제 등)의 알약 이미지 사진을 찍어 데이터셋을 확보한다.</p> <p>그리고 확보한 데이터셋으로 데이터의 특징을 추출하여 특징들의 패턴을 파악하고 object detection 모델의 대표적인 기술인 YOLO를 사용하여 해당 모델을 서버에 저장, 배포, 처리하여 알약 분석 웹페이지와 어플을 구성한다.</p>

과제 내용

1) 일정수립

세부내용	수행기간(월)																비고
	3				4				5				6				
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1. 계획수립 및 자료조사																	
2. 딥러닝 모델 학습 및 서버 구현																	
3. html, 자바스크립트 등을 이용한 프론트엔드 구현																	
4. 프로젝트 마무리 및 발표 준비																	

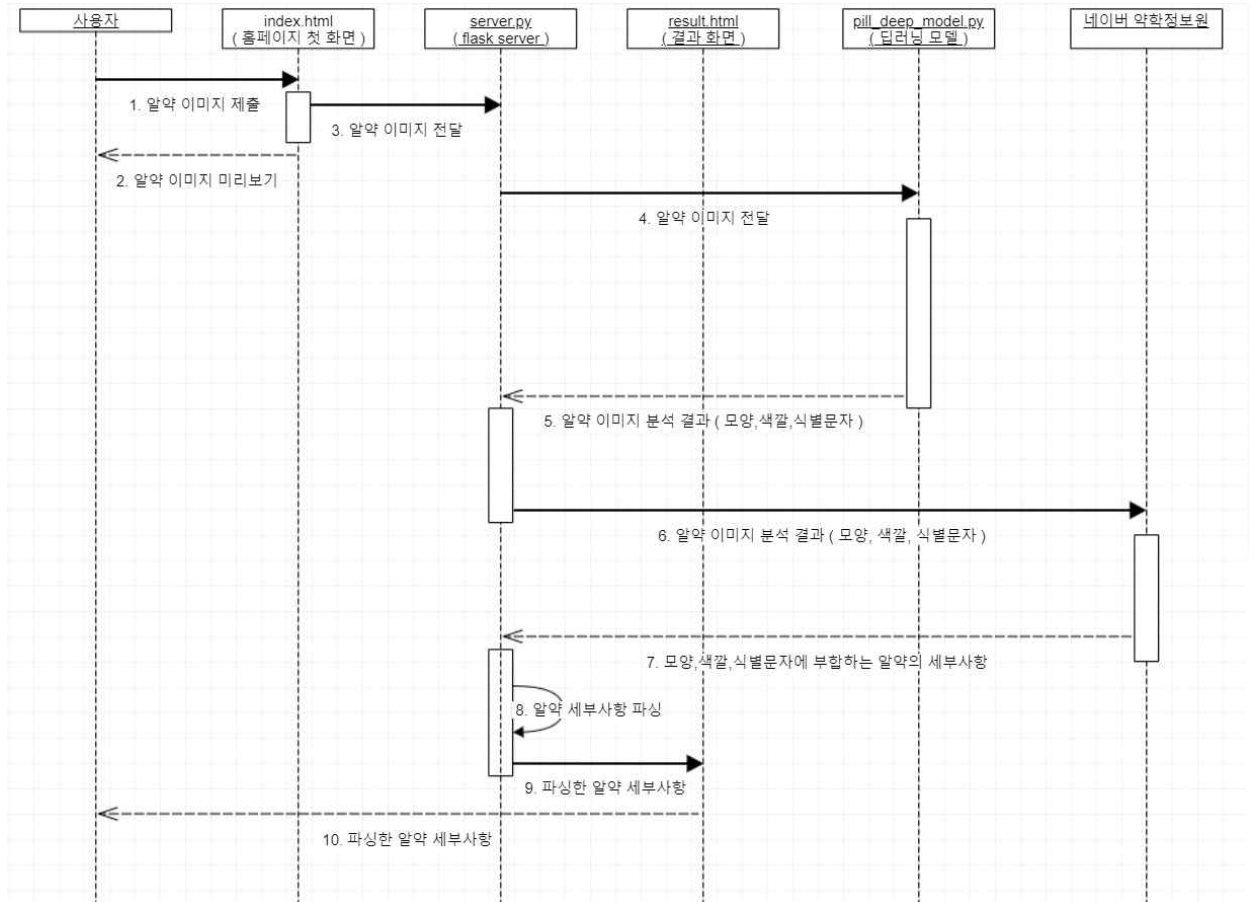
2) 요구사항 분석

요 구 사 항 명 세 서

항목	내용	
반응형 웹	pc에서 볼 때 모든 내용이 정상적으로 출력	
	스마트폰에서 볼 때 모든 내용이 정상적으로 출력	
	태블릿에서 볼 때 모든 내용이 정상적으로 출력	
알약정보 입력	사진 파일을 첨부할 수 있음	
	첨부한 파일을 제출할 수 있음	
	첨부한 파일의 경로가 텍스트폼에 나타남	
	첨부한 파일의 미리보기가 하단에 나타남	
	파일의 확장자가 jpg, png 가 아니면 오류 메시지 출력	
	10mb이상의 파일이면 오류 메시지 출력	
결과화면	결과 값이 하나일 때	사용자 사진 출력
		알약 사진 출력
		알약 이름 출력
		'상세내용보기'를 누르면 숨겨져있던 내용이 펼쳐짐
		'상세내용'은 외형정보, 성분정보, 저장방법, 효능효과를 포함
	결과 값이 여러개일 때	'네이버 약 정보'로 링크 연결
		모든 결과값 을 출력, 각 결과 값은 위와 같은 정보를 포함
		사용자가 알약 식별문자를 직접 입력할 수 있는 텍스트폼을 생성
딥러닝	텍스트 제출 시 정확한 결과 값 하나만을 출력	
	딥러닝 모델이 5000장 이상의 이미지를 학습	
	YOLO 모델을 사용하여 알약의 모양을 판별	
	OpenCV 모델을 사용하여 알약의 색깔을 판별	
서버	Google Tesseract를 사용하여 알약의 텍스트를 인식	
	html에서 받은 알약 사진을 딥러닝 모델에 전송	
	추출한 알약의 모양, 색, 텍스트를 조합해 사용자가 원하는 알약 링크 크롤링	
	알약 개수가 하나일 때, 알약의 세부내용 크롤링	
	크롤링해 얻은 알약 정보를 HTML에 전송	

3) 설계

시퀀스 다이어그램



4) 구현

딥러닝 팀)

1. YOLOv5를 이용하여 알약의 위치와 모양 디텍트

YOLO란, 딥러닝을 기반으로한 실시간 이미지 검출 시스템으로 영상과 이미지에서 물체를 감지하고 분류하는데에 특화된 기술.

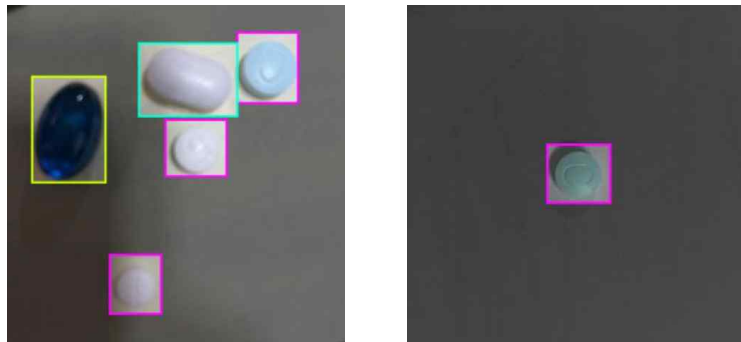
기존의 YOLOv3가 FPS(Frame per seconds)는 높은 반면에 mAP(mean Average Precision)은 비교적 낮은 모델이었던 것에 비해 YOLOv5는 FPS와 mAP 측면에서 모두 뛰어난 성능을 발휘함. 또한 다른 YOLO 모델들과 달리 크기별로 Pre-trained 모델이 나뉘져있는 형태.

먼저, 딥러닝에 필요한 데이터셋을 만듦. YOLO는 직접 데이터 이미지에 bounding box를 그려 물체가 위치해 있는 데이터 라벨을 txt 형식으로 저장하여 데이터셋을 구성해야함.

이를 수행하기 위해 머신러닝에 쓰이는 데이터셋을 가공할 수 있는 부분 유료 플랫폼인 roboflow.com을 사용함.

집에 있는 여분의 알약들과 그것들 외에 필요한 알약들을 직접 구매하여 카메라로 촬영 후 사진들을 확보함.

그리고 roboflow.com에 이미지들을 올려 크기는 416X416 사이즈로 설정하고 하나하나 bounding box를 쳐서 데이터셋을 구성하였고, bounding box의 클래스들은 알약의 모양으로 설정하여 round(둥근형), oval(타원형), oblong(장방형)의 세 개의 클래스를 만들어 bounding box를 그려냄.



이 때, 여러개의 다양한 알약들을 함께 찍어 우리가 만드는 딥러닝 모델의 정확도를 확보하고자 함. 확실히 알약 하나만 찍어서 데이터셋을 구성했을 때보다 여러 모양의 알약들을 찍어서 구성했을 때 모델의 정확도가 더 높았기 때문.

이렇게 총 4685장의 이미지들을 가공하여 데이터라벨들을 얻었고 train / test split은 train 이미지 70% (3.3k), validation 데이터 21% (971), test 데이터 9% (434)의 비율로 설정하여 완성된 데이터셋을 YOLOv5 pytorch 형식으로 export 받아 Colab에서 curl 명령어로 데이터셋을 원격으로 전송받아 사용함.

데이터셋을 구성하여 업로드한 후 Colab에 YOLOv5 깃헙 폴더를 클론하여 모델을 받아오고 필요한 라이브러리들을 코드로 설치.

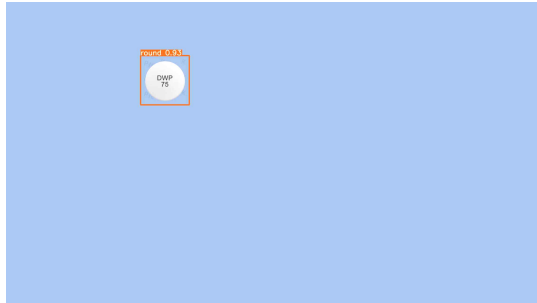
그리고 Pre-trained 모델들을 받아와 직접 만들어놓은 알약 데이터셋을 넣어 딥러닝을 진행함.

당연히 x가 성능이 제일 낮지만 FPS가 가장 높고, x가 성능이 제일 높지만 FPS는 가장 낮음. 모든 모델로 딥러닝을 해보았을 때 정확도의 차이를 느끼지 못하여 가장 FPS가 빠른 YOLOv5s로 선택하여 딥러닝을 진행하였음.

이미지 크기는 416X416, 학습이 완료된 weight 파일(pre-trained 파일)은 YOLOv5s, batch 사이즈는 16으로 설정하였고 총 epoch수는 50으로 설정하여 테스트를 진행하였음.

이렇게 직접 만든 데이터셋을 넣어 다시 만들어진 weight파일(custom weight 파일)을 다운받아 그것을 바탕으로 알약의 상세정보를 검출하는 프로세스에 사용하여 보다 빠르고 직관적인 단계를 거치게 하였음. custom weight 파일은 가장 성능이 좋은 best.pt와

최신 파일인 last.pt가 나오게 되는데 정확도를 우선시하여 이 중 best.pt 파일을 사용하였음.



그리고 얻은 best.pt(custom weight 파일) 와 정보를 검출할 알약 이미지를 넣어 object detection을 최종 수행함.

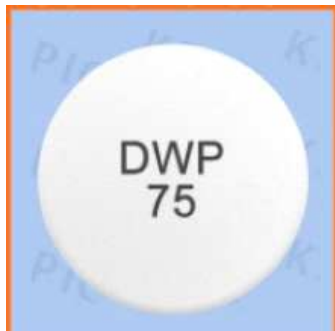
여기서 나오는 딥러닝 결과를 텍스트 파일로 파싱받아 이 파일에 포함되어 있는 모양정보를 다시 파싱받기 위한 fileread 함수를 구성하여 알약의 모양을 str 변수로 저장함.

2. 알약 구역 크롭

알약을 디텍트한 결과 이미지에서 bounding box의 각 모서리 좌표를 얻는 과정을 거쳐 얻은 좌표값으로 이미지를 알약 부분만 크롭함.

좌표를 얻는 과정에서 torch.hub.load() 함수를 통해 미리 학습된 모델을 불러와서 사용하는 model() 함수를 구성하였고 디텍트한 결과 이미지를 넣어 bounding box의 네 개의 좌표를 얻어내었음.

이 좌표들은 tensor 형식으로 이루어져있어 이미지를 크롭하는 함수에 값을 넣을 수 없었기 때문에 값을 float 형식으로 바꿔주는 item() 함수로 한번 더 처리해주었음.



얻은 네 개의 좌표값과 crop() 함수를 이용하여 알약이 있는 구역만 크롭한 cropped_img를 얻어내는 과정을 거침.

3. GrabCut 알고리즘

GrabCut 알고리즘은 이미지에서 배경이 아닌 전경에 해당하는 이미지를 추출하는 방법. 이미지에서 한 번에 전경을 추출해 내는 것이 아닌 사용자와의 상호 작용을 통해 단계

적으로 전경을 추출함. 이미지에서 전경이 포함되는 영역을 사각형으로 대략적으로 지정함.

이러한 grabcut 알고리즘에 대한 OpenCV 의 구현은 `cv2.grabCut()`.

모든 요소가 0인 마스크 이미지를 생성하고 이 알고리즘의 수행 과정 중 활용할 메모리 두 개를 만들었으며 이미지에서 전경이 포함된 사각영역은 `rect` 변수에 지정하였고 이 `rect` 변수를 이용해 전경을 추출하는 `cv2.grabCut()` 함수를 실행함.

```
cv2.grabCut(grabcut_img,mask,rect,bgdModel,fgdModel,5,cv2.GC_INIT_WITH_RECT)
```

함수의 인자를 살펴보면 첫 번째는 입력할 이미지, 두 번째는 먼저 생성해 놓았던 모든 요소가 0인 마스크 이미지, 세 번째는 `rect`-사각영역, 네 번째와 다섯 번째는 활용할 메모리들로 구성되어 있음. 여섯 번째는 사각형 영역의 지정을 통한 GrabCut 일 경우 `cv2.GC_INIT_WITH_RECT`를 지정함.

이를 이용해 원본 이미지에 마스크 처리를 해서 전경만을 표시.



GrabCut 알고리즘을 거쳐 물체만을 추출하여 `grabcut_img`를 얻어냄.

4. 알약 텍스트 검출

다음으로 처음에 얻어내었던 `cropped_img`에서 알약에 써져있는 텍스트 정보를 얻어내는 과정을 수행함.

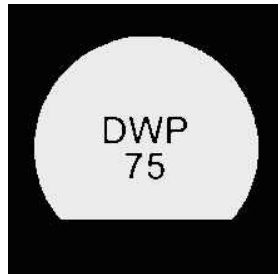
대표적인 OCR(Optical Character Recognition) 기술인 `pytesseract`를 설치하여 이미지 상의 글자를 인식시킴. 이 때, 음각되어있는 글자는 거의 인식을 하지 못하여 글자가 프린트 되어 있는 알약들을 뽑아 진행하였음. 음각되어있는 글자를 인식시키기 위해 OpenCV 함수 중 경계선을 검출하는 `Canny()` 함수와 이미지의 임계값을 처리하여 이진화 시켜주는 `threshold()` 함수도 써보았지만 기술의 한계로 이러한 선택을 하게 되었음.

5. K-Means clustering 이용한 image segmentation(이미지 분할)

이미지 분할(segmentation)은 컴퓨터 비전에서 물체를 인식하고 분리하는데 기초가 되는 중요한 이미지 처리 방법. `cv2`에 `K_means` 함수가 포함되어 있기 때문에 비교적 간편하게 작업을 수행할 수 있었음.

이 한 pixel 당 red, green, blue의 세가지 색상이 0~255의 값을 가져 color를 완성하게 됨. 이 색깔들을 K개의 cluster로 뽑아낸다면, 비슷한 색깔을 지닌 pixel들을 묶어

낼 수 있기 때문에 k-means clustering 으로 grabcut_img의 색깔을 구분하였음. clustering은 비슷한 픽셀 값을 가지는 것들을 분류하는 것.



따라서 cluster의 개수인 K가 증가할수록 이미지 컬러의 세분화가 증가하는데 알약 하나의 색깔은 다양하지 않고 단색이거나 최대 두 개의 색깔을 지닌 알약들이 많아 cluster의 개수를 2로 설정하여 작업하였고 충분한 결과를 얻어낼 수 있었음.

6. 색 (rgb) 추출

이미지 컬러를 세분화 시킨 k_means_img에서 이미지 가운데의 10x를 크롭하는 과정을 거쳤음. 먼저 처리하기 용이하도록 256X256의 사이즈로 resize한 후 크롭할 영역을 (127, 127, 128, 128)로 설정하였고, crop() 함수에 적용하여 가운데 1px만 점과 같은 모양인 이미지로 추출하였음.

(237, 237, 237)

그리고 크롭한 1px의 이미지에서 해당 좌표의 색상 값을 가져오는 getpixel()함수를 적용하여 rgb 값을 추출하였음.

```
if (rgb[0] >= 215 and rgb[1] >= 205 and rgb[2] >= 185):  
    color = "white"  
elif (rgb[0] >= 180 and rgb[1] >= 180 and rgb[2] >= 160):  
    color = "white"  
elif (rgb[0] >= 140 and rgb[1] <= 30 and rgb[2] <= 30):  
    color = "red" #red  
elif (rgb[0] >= 220 and rgb[1] >= 120 and rgb[2] >= 0):  
    color = "orange"  
elif (rgb[0] >= 180 and rgb[1] >= 200 and rgb[2] >= 200):  
    color = "blue"  
elif (rgb[0] >= 110 and rgb[1] >= 130 and rgb[2] >= 150):  
    color = "blue"  
elif (rgb[0] >= 150 and rgb[1] >= 150 and rgb[2] >= 0):  
    color = "green" #green  
elif (rgb[0] >= 0 and rgb[1] >= 0 and rgb[2] >= 50):  
    color = "blue"
```

이렇게 추출한 rgb 값에서 색 범위를 직접 설정하여 색깔을 분류하고 나온 색상정보를 str로 저장하였음.

7. 모양, 색깔, 글자 정보를 딕셔너리 값으로 저장

지금까지 처리한 모양, 색깔, 글자 정보들을 서버에서 사용하기 용이하도록 함수를 설정하고 그 안에 모든 정보를 딕셔너리 값으로 저장하여 이 값을 return.

서버팀)

0. flask로 구동한 홈페이지의 url을 react native의 webview 기능을 이용하여 어플로 제작
1. React Native(Expo)를 이용해 만든 apk를 다운받음.
2. 앱을 실행하면 서버에서 루트에 존재하는 페이지인 'mainsite.html'을 render_template 함수를 통해 렌더링을 해준다. 메인 페이지에서 사용자는 알약 사진을 넣고 제출 버튼을 클릭시, HTML의 form태그 중 action='/post'를 통해 현재 주소에서 '/post'가 추가된 주소로 이동이 되며 그 하단에 존재하는 post함수를 실행한다.
3. post함수에서는 request.files['pill_file']를 통해 HTML에서 준 pill_file 파일을 받는다. 파일을 save 명령어를 통해 저장 후, 딥러닝 학습을 위해 yolo5의 detect.py로 가져온 파일을 텍스트 형식(msg.txt)으로 변환시켜준다. flask에서 딥러닝모델(yolov5,opencv,pytesseract)을 import as pmodel로 하여 받은 후, 모델 파일에 존재하는 detectPill함수를 통해 해당 알약의 모양,색깔,글자를 딕셔너리 형태로 저장한다.
- 4.BeautifulSoup 라이브러리를 이용하여 해당 딕셔너리를 parser 함수를 통해 해당 분석값들을 네이버 의약품 사전에 쿼리를 넣어 나온 결과들을 가져오게 한다. 네이버 의약품 사전 링크에 조건들을 추가한 링크는 BeautifulSoup을 통해 크롤링을 한다. 크롤링을 통해 가져올 값을 알약의 이름과 링크 그리고 약 제품의 이미지를 받아온다.
5. 만약 파싱한 알약의 개수가 2개 이상일 경우엔 HTML로부터 식별문자를 입력받아 재 크롤링을 하고, 값이 없는 경우에는 다시 메인페이지로 띄우도록 HTML을 렌더링한다. 알약 결과가 한 개일 경우에만 해당 알약의 세부사항을 파싱하여 원하는 정보를 추출 후, HTML에 값을 전달한다.

HTML & CSS)

1. 접속 시 팝업 창

먼저 접속하게 되면 팝업창이 뜰 수 있는 모션을 주었다. 팝업창 이미지와 로고 이미지는 adobe photoshop으로 디자인 하였으며 function closePopUp() 함수를 사용하여 사용자가 팝업창을 닫고 싶을 때 닫을 수 있게 구현하였다.

2. 레이아웃

가장 위에는 약간의 여백이 있고 그 아래에 로고를 출력하였다. 로고는 adobe photoshop을 사용하여 직접 디자인하고 제작했다. 로고 아래에는 사진파일을 첨부할 수 있는 text폼과 제출버튼을 구현하였다. 제출하면 결과화면이 그 아래로 출력되게 된다. 가장 아래에는

footer를 삽입하여 프로젝트 팀명과 팀원 이름들을 볼 수 있게 하였다. 또한 디자인의 디테일을 위해 하이퍼링크의 밑줄을 없애고 클릭시 색이 변하는 것을 방지하기 위하여 a:hover, a:visited { color: #32373a; text-decoration: none } 를 사용했다.

3. 사진파일 첨부

자바스크립트를 주로 이용하였으며, 우선 파일을 불러오면 확장자를 검사하기 위해 substring 함수를 사용하여 "."을 기준으로 파일명과 확장자를 구분한다. 이후 확장자가 jpg나 png인지 검사하여 이미지 파일이 아니면 alert를 띄우도록 설정하였다. 또한 10mb를 초과하지않는지 검사하여, 10mg 이상의 파일은 alert를 띄우고 사용자가 업로드 한 파일을 초기화 시키도록 하였다. 모든 조건을 만족 한다면 frm.preview.src = e.target.result 문을 통하여 업로드 한 이미지의 미리보기를 보여준다. 이후 제출 버튼을 누르면 첨부파일을 딥러닝 모델에게 넘겨준다.

4. 결과 화면

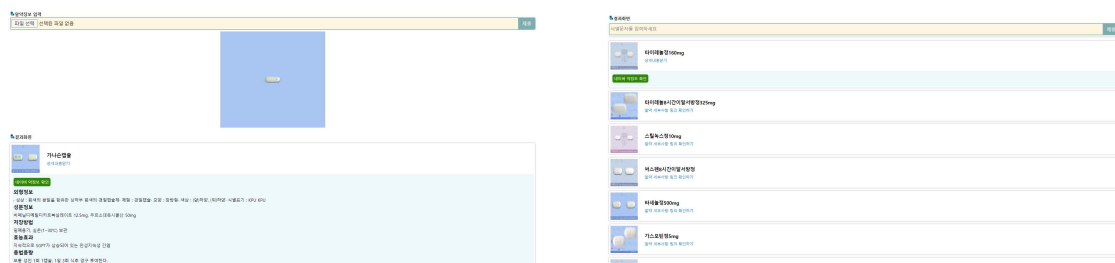
4-1. 결과 값이 하나일 때

사용자가 올린 이미지를 딥러닝 모델이 분석한 결과 값이 하나일 때는 단순히 결과 화면을 출력한다. 결과 화면은 알약 사진, 알약 명칭, 상세내용 보기로 구성되어있다. 상세보기 버튼은 onclick이벤트를 통하여 자세한 내용이 출력되도록 하였으며, 더 자세한 정보를 위해 href를 통해 네이버 약정보 확인으로 넘어갈 수 있는 링크도 구현하였다.

4-2. 결과 값이 두 개 이상일 때

사용자가 올린 이미지를 딥러닝 모델이 분석한 결과 값이 여러 개일 때는 여러 개의 결과 값을 출력함과 동시에 새로운 text폼을 구현한다. 이 text폼에 알약에 적혀있는 문자를 사용자가 직접 입력하여 여러 개의 결과들 중 정확하게 하나의 결과로 추릴 수 있다. 이후의 출력결과는 3-1과 같다.

<사용자 웹 화면 페이지>



알약 정보 입력 칸에 알약 사진을 업로드 하면 해당 알약에 맞는 세부 정보 사항들이 결과화면에 나타나게 됨.

만약 검색한 조건의 알약이 여러개라면, 해당 알약에 네이버 의약품 백과사전 URL로 넘어갈 수 있게끔 설정 (해당 알약의 링크를 띄어줌).

<사용자 어플리케이션 화면>



5) 테스트

테스트 케이스

항목	내용		
반응형 웹	pc에서 볼 때 모든 내용이 정상적으로 출력		✓
	스마트폰에서 볼 때 모든 내용이 정상적으로 출력		✓
	태블릿에서 볼 때 모든 내용이 정상적으로 출력		✓
알약정보 입력	사진 파일을 첨부할 수 있음		✓
	첨부한 파일을 제출할 수 있음		✓
	첨부한 파일의 경로가 텍스트폼에 나타남		✓
	첨부한 파일의 미리보기가 하단에 나타남		✓
	사진 파일이 아니면 오류 메시지 출력		✓
	10mb이상의 파일이면 오류 메시지 출력		✓
결과화면	결과 값이 하나일 때	알약 사진 출력	✓
		알약 이름 출력	✓
		‘상세내용보기’를 누르면 숨겨져있던 내용이 펼쳐짐	✓
		‘상세내용’은 외형정보, 성분정보, 저장방법, 효능효과를 포함	✓
		‘네이버 약 정보’로 링크 연결	✓
	결과 값이 여러개 일 때	모든 결과값 을 출력, 각 결과 값은 위와 같은 정보를 포함	✓
		사용자가 직접 입력할 수 있는 텍스트폼을 생성	✓
		텍스트 제출 시 정확한 결과 값 하나만을 출력	△
딥러닝	딥러닝 모델이 5000장 이상의 이미지를 학습		△
	YOLO 모델을 사용하여 알약의 모양을 판별		✓
	OpenCV 모델을 사용하여 알약의 색깔을 판별		✓
	Google Tesseract를 사용하여 알약의 텍스트를 인식		△
서버	html에서 받은 알약 사진을 딥러닝 모델에 전송		✓
	추출한 알약의 모양, 색, 텍스트를 조합해 사용자가 원하는 알약 링크 크롤링		✓
	알약 개수가 하나일 때, 알약의 세부내용 크롤링		✓
	크롤링해 얻은 알약 정보를 HTML에 전송		✓

활용 방안 및 기대효과

알약은 비슷한 디자인이 많아 일반인은 쉽게 구분하기 어렵다. 예를 들어 하얀색 원형 모형의 알약이나, 색이 똑같은 캡슐형이여도 성능이 다른 경우 등이 있다. 이러한 경우에 알약의 오용이 문제가 될 수 있는데 이 때, 해당 프로그램을 사용하여 자신이 현재 복용하고 있는 약에 대한 상세한 정보를 알 수 있으며, 그로 인하여 올바른 복용을 기대할 수 있다.

또한 약사가 약을 처방하기 전에 해당 프로그램을 통해 처방한 약이 맞는지 한번 더 검토할 수 있다. 또한 약을 폐기하거나 약 성분을 체크할 때, 알약을 재분류 할 때 등과 같은 상황에서 수작업으로 약을 분류 했을 때 보다 더 높은 정확성을 보일 수 있다.

이러한 이점으로 시장에 창출했을 때 제약사, 약대, 보건, 의료 등 여러 분야에서 유용하게 쓰일 것으로 예상된다.