

텐서플로우 기반 이미지 프로세싱에 대한 연구: 알약분류 중심으로

조수형¹, 강진구², 김정훈³, 이승준⁴, {김계영, 김영중}^{*}
^{1,2,3,4,*}승실대학교 소프트웨어학부

e-mail:vinis12@naver.com, xortm854@naver.com,
jhkim3619@naver.com, vinis12@naver.com, {gykim11, youngjong}@ssu.ac.kr

A Study on TensorFlow based Image Processing: Focusing by Pill Classification

Soo-Hyoung Joe¹, Jin-Goo Kang², Jung-Hoon Kim³, Sung-Jun Lee⁴,
{Gyeyoung Kim, Youngjong Kim}^{*}

^{1,2,3,4,*}School of Software, Soongsil University

요 약

이미지 프로세싱이란 기존의 이미지에 대해 컴퓨터를 이용하여 새로운 이미지로 창작하거나 수정하는 일련의 작업 과정이다. 우리는 알약의 이미지를 가져와 machine이 인지 할 수 있도록 수정한 후, 사진에 찍힌 알약을 구별하고 사용자 에게 그 알약의 정보들을 제공 할 수 있는 텐서플로우 기반의 이미지 프로세싱 방법에 대해 연구 하였다.

1. 서론

누구나 예고없이 아픈 상황이 올 때가 있다. 이때 자연스럽게 하는 행동 중 하나가 집에 있는 약들을 찾아보는 것이다. 집에서 찾은 약들의 종류를 판단하는데 많은 방법이 있지만 가장 쉽게 할 수 있는 방법은 웹을 통해 검색하는 것이다. 하지만 이러한 방법은 모양, 색, 글씨 등이 필요하기 때문에 절차가 복잡하고 이용하는데 불편함이 존재한다. 따라서 우리는 이러한 기존의 불편한 방법을 간소화하여 사진 한장으로 약의 정보를 알 수 있는 플랫폼을 기획하게 되었다.

먼저 OpenCV를 이용한 이미지 프로세싱을 통해 사진에서 알약부분을 검출한다. OpenCV의 Saliency Map라이브러리를 이용하여 사진의 특이점을 검출하여 알약의 위치를 판단한다. 이 과정에서 정확도를 높이기 위해 PyrMeanShiftFiltering 필터를 사용하여 노이즈를 제거하여 알약부분을 검출한다. 이 후 알약부분이 검출된 이미지를 LAB 색상 공간으로 변형시킨다. 변형된 이미지를 "Canny Edge" 를 이용하여 윤곽선을 검출한다. 검출된 윤곽선으로 모양을 알 수 있다. 이 윤곽선에 따라 알약부분이 검

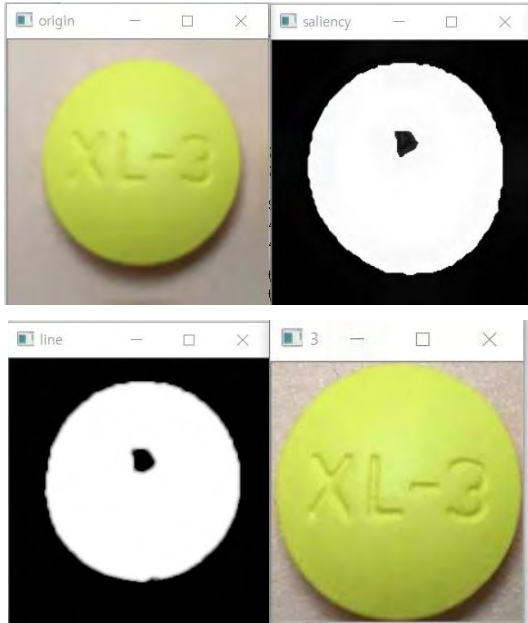
출된 이미지에서 알약만을 추출한다. 추출된 알약 이미지에서 Gradient를 이용해 글자만을 남기고, "Bounding Box"로 글자 해당하는 부분을 이미지로 저장한다. 추출한 모형과 글씨 이미지를 텐서플로우가 학습할 수 있는 TFRecord 파일로 전처리를 거친 후 CNN 딥러닝 이미지 비교 학습을 통해 완성된 모델을 통해 이미지의 모형과 글씨를 판단한다. 이 후 판단된 알약의 모형과 글씨를 데이터베이스에 입력하여 저장된 알약의 정보를 가져오는 것이다.

2. 텐서플로우 기반의 이미지 프로세싱에 대한 방법

2.1. 이미지 프로세싱

- 이미지 프로세싱 방법 1

- (1) 이미지를 grayImage로 변환.
- (2) Sobel을 통한 image Gradient.
- (3) 임계처리를 통한 이미지 이진화



(좌)이미지 프로세싱 전 알약사진 (우) 이미지 프로세싱 후 알약사진

- 이미지 프로세싱 방법 2

- 1) 원본 사진 노이즈 제거
- 2) saliencyMap + otsu threshold
- 3) 그림자 제거
- 4) 알약 검출

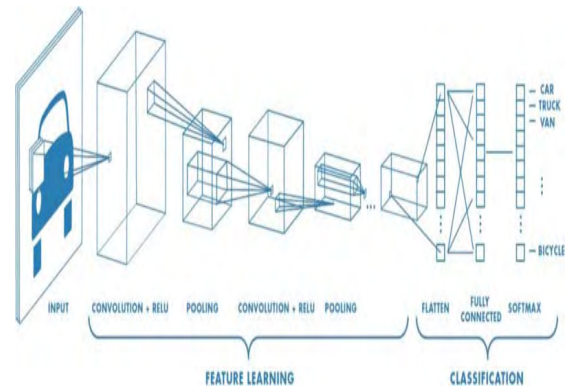


- (1) pyrMeanShiftFiltering을 이용하여 원본 이미지 노이즈 제거
- (2) 알약부분 검출을 위해 saliencyMap을 이용하여 이미지의 특이점을 파악할 수 있는 이진파

일로 변환

- (3) otsu threshold를 이용 노이즈 제거
- (4) hsv로 변환한 후 밝기를 나타내는 v의 값을 0에서 1사이의 값으로 정규화 시킴
- (5) 정규화 시킨 값이 0.5이하이면 그림자로 판단한 후 2.의 이미지에서 해당부분의 값을 0으로 변환
- (6) 4의 이미지에서 findContours를 사용하여 경계선들을 검출
- (7) 6의 경계선들의 통해 알약부분 좌표 검출
- (8) 7의 좌표로 원본 이미지 잘라 hsv영상으로 변환한 후 밝기 조절, 필터 등을 글자추출을 위해 이미지 변환
- (9) 8의 이미지에서 gradient magnitude을 통해 글자를 추출

2.2. CNN딥러닝으로 이미지 비교 학습



CNN딥러닝 기반의 이미지 비교학습과정은 아래와 같다.

1. 이미지 크기 고정 (96,96)으로 (pixel 단위)
2. Feature Learning
 - (1) 이미지에 커널을 합성곱(convolution)
 - 32배, 64배, 128배
 - 각 합성곱마다 Pooling
 - flat하여 다층 신경망으로 만듦
 - (2) Classification
 - 전결합층(ReLU 함수)을 사용하여 학습과정에서의 가중치 조절
 - 출력층에서 활성화함수(SoftMax 함수)를 이용하여 학습모델의 정확도를 확인
- (3) Data

- 이미지 프로세싱 작업 후 텐서플로우가 학습할 수 있도록 TFRecord 파일로 전처리
- Bazel을 통해 이미지 데이터 스크립트 빌드

참고문헌

[1] 알약 이미지 데이터셋:
<https://pillbox.nlm.nih.gov/developer.html#data>

[2] 김대욱. "스마트폰으로 촬영된 알약 영상의 글자 및 형상 인식 방법" 공학석사 학위논문 (2017.8)

[3] 약학정보원: <http://www.health.kr/main.asp>

2.3. 데이터 베이스 구축 및 활용

우선적으로 사용자 친화적인 인터페이스를 구현하기 위한 목적과 시각적으로 편리한 인터페이스를 구현하기 위한 목적이 있기 때문에, JAVA로 패킷 스니핑 프로그램을 구현할 것이다.

C와 C++을 기반으로 한 pcap을 이용하여 구현했던 기존의 Wire Shark와 달리, JAVA를 기반으로 패킷 스니핑 기술을 구체화해야 한다. 따라서 libpcap과 winpcap의 래퍼 라이브러리인 jnetpcap을 사용하여 프로그램을 구현할 것이다.

```

0000: 7b 22 76 65 72 73 69 6f 6e 22 3a 20 5b 32 2c 20 {"version": [2,
0010: 30 5d 2c 20 22 70 6f 72 74 22 3a 20 31 37 35 30 0], "port": 1750
0020: 30 2c 20 22 68 6f 73 74 5f 69 6e 74 22 3a 20 35 0, "host_int": 5
0030: 39 30 37 36 32 32 36 39 32 39 36 30 34 39 31 39 9876226929684919
0040: 32 34 39 37 30 35 32 37 33 31 36 35 31 31 36 31 2497052731651161
0050: 39 31 38 34 38 2c 20 22 64 69 73 70 6c 61 79 6e 91848, "displayn
0060: 61 6d 65 22 3a 20 22 22 2c 20 22 6e 61 6d 65 73 ame": "", "names
0070: 70 61 63 65 73 22 3a 20 5b 35 30 39 33 36 31 36 paces": [5093616
0080: 39 36 30 2c 20 31 33 33 32 39 38 39 33 34 39 2c 960, 1332089349,
0090: 20 31 30 36 37 30 30 31 39 37 2c 20 31 34 36 37 106700197, 1467
00a0: 38 39 31 30 30 39 2c 20 31 34 32 33 39 32 34 37 891809, 14239247
00b0: 31 33 2c 20 35 34 33 32 31 39 36 35 39 2c 20 31 13, 543219659, 1
00c0: 34 36 37 38 39 31 34 30 34 2c 20 38 38 34 39 38 467891404, 88498
00d0: 30 35 35 39 2c 20 32 38 32 36 37 34 37 33 36 2c 0559, 282674736,
00e0: 20 38 32 31 38 33 36 39 31 34 2c 20 37 37 39 33 821836914, 7793
00f0: 30 36 37 37 32 2c 20 38 35 31 34 30 30 35 30 30 06772, 851400500
0100: 2c 20 31 35 35 32 38 35 32 37 34 33 2c 20 35 30 , 1552852743, 50
0110: 32 31 39 33 32 31 35 5d 7d 2193215]}
    
```

▲캡처한 패킷. - hex사값임을 확인 가능

jnetpcap 라이브러리를 이용하여 캡처한 패킷 정보들은 정제되어있지 않은 hex사값이다. 따라서 필자는 정제되어있지 않은 hex사값에서 IP header 부분을 찾아내어 그 구조에 맞게 분할하여 분석한다. 분석 내용을 바탕으로 TCP프로토콜을 사용하는 패킷만을 남기고 다른 패킷들은 제거한다. 그 후 IP Datagram부분을 TCP header와 data 부분으로 분할하고, TCP header 구조에 맞게 분석한다. 이후 분석한 내용을 사용자가 쉽게 정보들을 파악할 수 있도록 보다 더 사용자 친화적인 인터페이스로서 구현한다.

ACKNOWLEDGMENT

"본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학사업의 연구결과로 수행되었음 (2018-0-00209-001)"