

시각장애인을 위한 딥러닝과 이미지인식을 이용한 스마트옷장

최소희⁰ 김주하 박준성 오재동 공기석 전광일

한국산업기술대학교 컴퓨터공학과

{cde9030⁰, suic97, pjsung0426, 2015156021, kskong, gijeon}@kpu.ac.kr

A Smart Closet Using Deep Learning and Image Recognition for the Blind

So-Hee Choi⁰ Ju-Ha Kim Jun-Seong Park Jae-Dong Oh Ki-Sok Kong Gwang-Il Jeon
Dept. of Computer Engineering, Korea Polytechnic University

요 약

시각장애인의 대다수는 독립적인 의생활을 하는데 어려움을 겪는다. 본 논문에서는 시각장애인의 독립적인 의생활을 지원하기 위해 옷장 내부를 관리하는 기능, 음성 대화를 통해 정보를 요청하는 음성인식 기능 그리고 CNN 알고리즘을 이용한 옷 정보에 대한 인식 기능을 가진 스마트 옷장을 제안한다. 본 논문에서는 옷을 인식하는 과정에서 정확도를 높이기 위해 모델 층의 개수를 변경하고 Maxpooling을 조정하여 모델을 생성하였다. 모델 생성 시 Early Stopping Callbacks 옵션을 적용하여 학습 정확도를 보장해주었다. 과적합을 방지해주기 위하여 Dropout을 추가했다. 이러한 과정으로 만들어진 최종 모델은 옷 인식 정확도가 80%가 되는 것을 확인할 수 있다.

1. 서론

시각 장애인들의 90.3%는 후천적인 원인^[1]으로 인해 장애가 발생한다. 후천적으로 장애가 발생한 경우 여러 가지 생활을 하는데 더 큰 어려움을 겪는다. 특히 시각장애인들의 대부분은 의생활에 불편함을 겪는다.

시각장애인은 대부분은 촉감에 주로 의지하여 옷을 고른다. 그 외의 도움을 받기 위해서는 점자가 달린 옷 혹은 코디네이터 봉사자^[2]를 통해 옷을 고른다. 하지만 시각장애인을 위한 점자 달린 옷과 코디네이터 봉사자는 현재까지 적기 때문에 그들은 독립적인 의생활을 하는데 어려움을 겪는다. 그래서 그들은 같은 종류의 옷을 입거나, 양말을 다르게 신는 등^[3]의 문제를 겪는다.

본 논문에서는 앞서 언급한 스마트 가전 기술을 활용해 시각장애인의 독립적인 의생활을 지원해주는 제품을 개발하였다. 시각장애인의 원활한 의생활을 돕는 스마트 옷장은 다음과 같은 기능을 한다. 사용자가 고른 의상을 걸어 놓고, 대화를 통해 원하는 기능을 요청한다. 옷장은 음성 판독과 의상 촬영 등을 수행하여 요청에 따른 결과를 음성으로 출력해준다. 이러한 기능들을 수행하는 스마트 옷장의 이름은 '뷰클로젯 (ViewCloset)'이라고 명명한다.

2. 관련 연구

2.1 기존 유사 제품과의 비교

스마트 가전 시장이 커지면서 개발된 본 논문과 유사한 스마트 옷장에 대해서 찾아보았다. 아래 표 1은 본 논문과 유사한 제품을 정리한 표이다.

표 1. 유사 제품과의 비교

제품명	강점	약점
세븐드림머즈 런드로이드 ^[4]	- 자동으로 옷을 개어 주고 수납하는 기능 - 자동 옷장 내부제어	- 비장애인에게 초점이 맞춰짐 - 옷에 대한 정보를 인식하지 않음
한이음 시각장애인을 위한 스마트옷장 ^[5]	- 손가락으로 옷을 가리켜 옷에 대한 인식 - 옷에 대한 시각적인 정보를 청각으로 출력	- 옷의 종류 및 속성에 대한 인식은 하지 못함 - 손가락을 통해 기능을 제어함

위의 두 제품은 각각의 장점과 부족한 점이 모두 존재하였다. 뷰클로젯은 이러한 제품들의 장점은 살리고 단점은 보완하여 개발하였다.

세븐드림머즈의 '런드로이드'는 시각장애인에게 적합하지 않으며 옷을 수납하고 관리하는 곳의 초점을 두었다. 뷰클로젯은 옷을 수납 및 관리하는 것뿐만 아니라 시각장애인에게 적합하도록 음성인식 및 옷 인식 등을 하여 이를 보완하였다. 한이음의 '시각장애인을 위한 스마트 옷장'은 시각장애인에게 적합하지만 옷장을 손가락으로 제어하고 자세한 옷의 구체적인 정보를 알지 못한다는 약점이 있다. 뷰클로젯은 이를 보완하여 모든 기능을 음성으로 수행하고, 옷에 대해 색상, 종류 등의 구체적인 정보들을 안내하도록 구현하였다.

3. 세부 설계 및 구현

3.1 개발환경

본 논문에서는 아두이노와 라즈베리파이를 기반으로 '뷰클로젯'의 하

드웨어를 구현한다. 또한, 마그네틱 도어 센서와 온습도 센서를 이용하여 옷장의 문 열림 상태에 따른 LED 점등 상태를 관리한다. 온.습도 센서를 통해 옷장의 습도를 측정해 유지 및 제어하도록 개발한다.

아래의 표 2는 하드웨어 개발환경에 대한 표이다.

표 2. 하드웨어 개발 환경

하드웨어 부품		기능
	Arduino UNO R3 ^[6] (아두이노 보드)	아두이노 보드
	Raspberry Pi 4B ^[7] (라즈베리파이 보드)	라즈베리파이 보드
	MC-38 ^[8] (마그네틱 도어센서)	문 열림 여부를 측정하는 센서
	DHT11 ^[9] 온습도센서	옷장 내부의 온도와 습도를 측정하는 센서
	LED655 ^[10] (LED)	옷장 내부를 밝혀주는 LED 램프
	CT-12025L12R ^[11] (팬모터)	옷장 내부의 습도를 낮추는 팬모터
	OV5640 ^[12] (USB 카메라)	옷을 촬영하는 카메라
	BT042 ^[13] (마이크)	음성인식을 위한 마이크
	스피커 ^[14]	결과 출력을 위한 스피커

본 논문에서는 Linux 운영체제를 기반으로 소프트웨어를 개발한다. 또한, 서버는 아마존 웹 서비스 AWS에서 제공하는 인스턴스 EC2^[15]를 서버로 이용한다. 기본적으로 Python언어로 Anaconda의 PyCham pro IDE^[16]를 이용해 개발하였다.

아래의 표 3는 소프트웨어 개발환경에 대한 표이다.

표 3. 소프트웨어 개발 환경

종류	소프트웨어	버전
OS	Linux	ver 16.04 LTS
Server	AWS EC2	ver p2.xlarge
IDE	PyCharm(Pro)	ver Anaconda 5.01
Programming Language	Python	ver 3.6.9
Library	Keras	ver 2.3.1
Library	Tensorflow	ver 2.2.0

3.2 시스템 구성

시스템은 여러 센서가 연결된 Arduino UNO R3, Raspberry Pi 4B와 딥러닝 모델이 저장된 서버로 구성되어 있다. Arduino UNO R3는 Raspberry Pi 4B와 Serial 통신^[17]과 I2C 통신^[18]을 통해, Raspberry Pi와 서버는 Socket 통신^[19]을 이용해 데이터를 송수신 한다.

아래의 그림 1은 하드웨어와 소프트웨어의 시스템 설계도이다.

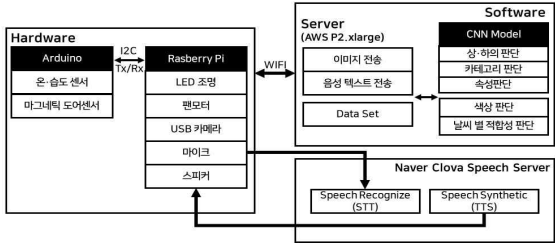


그림 1. 하드웨어와 소프트웨어 시스템 설계도

하드웨어는 아두이노를 통해 옷장 내부 관리를 하며, 라즈베리 파이를 통해 외부와의 통신을 하여 카메라, 마이크, 스피커 등을 관리한다. 소프트웨어는 CNN 알고리즘을 이용한 옷 정보들에 대한 인식 모델과 색상 이미지 인식, 날씨별 적합성 판단 기능들이 서버에서 기능을 수행한다. 또한, Speech-To-Text와 Text-To-Speech 기능은 네이버의 Clova^[20]를 이용해 수행한다.

아래 그림 2는 옷 정보 인식에 대한 기능 구성도이다.

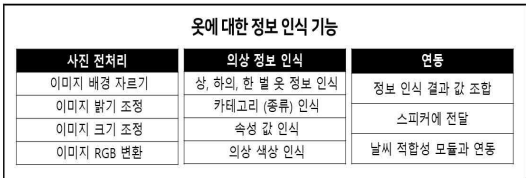


그림 2. 옷 정보 인식 기능 구성도

뷰클로젯은 사용자가 촬영한 이미지를 모델에 적합한 형태로 변환하기 위한 사진 전처리 기능, 의상 정보 인식 기능, 외부 모듈과 상호작용하기 위한 연동 기능으로 구현하였다.

3.3 기능 수행 흐름도

본 논문에서는 라즈베리와 AWS 서버가 통신을 하며 하드웨어와 소프트웨어 기능을 수행한다. 시각장애인 사용자가 불편하지 않도록 행동을 통한 기능은 최대한 줄이고, 음성을 통해 기능이 이루어진다.

옷 인식 기능을 수행하기 위해서 Hotword Detection으로 로제를 불러 뷰클로젯을 활성화 시킨다. 활성화 이후 음성녹음과 Speech -To- Text 과정을 통해 사용자가 원하는 기능을 판단한다. 옷에 대한 정보를 요청하면 촬영을 하고 이미지 전처리를 수행한다. 사용자가 요청한 정보를 CNN 모델을 통해 판독한다. 판독한 정보를 라즈베리파이로 보내 음성으로 사용자에게 알려준다.

아래의 그림 3는 소프트웨어 옷 인식 기능을 수행하는 시나리오 흐름도이다.

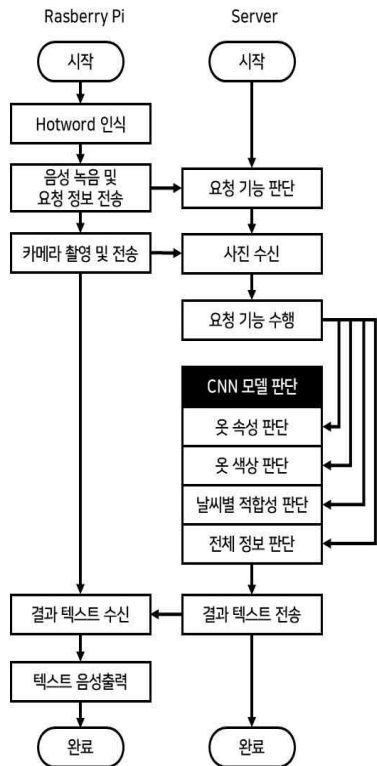


그림 3. 옷 인식 기능 수행 시나리오 흐름도

3.4 옷 정보 인식 모델 구현

본 논문에서는 옷 인식 모델은 크게 두 가지로 나뉜다. 촬영한 이미지를 기반을 통한 이미지 인식 모델과 CNN 알고리즘을 통해 옷의 정보를 인식하는 딥러닝 모델이다.

뷰클로젯의 옷 이미지 인식 모델은 옷 색상 인식 모델에서 사용된다. 이미지 색상 인식 모델은 딥러닝을 사용하지 않고 픽셀 값을 얻어 이미지의 색상을 추출해 낸다.

아래의 그림 4는 옷 색상 인식에 대한 수행 시나리오 이다.



그림 4. 옷 정보 인식 기능 구성도

촬영한 사진을 받아 Grap-cut^[21] 알고리즘을 통해 배경을 제거한 후 가장 옷이 잘 나타나는 이미지의 중간 부분 잘라낸다. 그 후 각 픽셀 단위로 RGB 값으로 저장하고 평균 RGB 값을 판단한다. 이 후 webcolors^[22] 라이브러리를 통해 평균 RGB와 이름이 저장된 색상 RGB 사이의 거리를 유클리디안 거리측정^[23]하여 가장 유사한 색상을 판단한다.

뷰클로젯의 딥러닝 옷 인식 모델은 CNN 알고리즘을 이용하여 옷을 인식하는데 사용된다. 딥러닝 옷 인식 모델은 각 layer에 노드의 개수를

배수 단위로 늘려갔다. 또한 각 층마다 Dropout층과 Maxpooling2D층을 추가하여 학습 시 발생할 수 있는 Overfitting 문제를 방지하였다. 아래의 표 4는 옷에 대한 정보 인식 모델 구조이다.

표 4. 옷에 대한 정보 인식 모델 층 구조

level	Layer	Activation
1층	Convolution2D(16, 3, 3)	relu
	MaxPooling2D(2, 2)	
	Dropout(0.2)	
2층	Convolution2D(32, 3, 3)	relu
	MaxPooling2D(2, 2)	
	Dropout(0.2)	
3층	Convolution2D(64, 3, 3)	relu
	MaxPooling2D(2, 2)	
	Dropout(0.2)	
4층	Convolution2D(128, 3, 3)	relu
	MaxPooling2D(2, 2)	
	Dropout(0.2)	
	GlobalMaxPooling2D	
	Dropout(0.3)	
출력층	Dense(output_node_num)	softmax

상하의 예측 모델 구축에 있어서는 4층 모델을 사용하였다. 하지만 속성 인식 모델과 카테고리 인식 모델 구축에 있어서는 6층 모델이 인식 정확도가 더 높아 6층 모델을 사용하였다.

3.5 구현 결과

본 논문에서는 사용자가 촬영한 사진에 대한 인식을 구축한 학습 모델이 정확하게 수행 하는지 확인하기 위해 모델 층 개수 변경, Dropout 조정, Maxpooling 조정을 통한 실험을 진행하였다. 모델을 구축할 때 모델의 층수를 1층부터 시작하여 한 층씩 늘려가는 방식으로 구축하였다. 층마다 Maxpooling을 적용하였다. 이러한 구축, 비교하는 과정을 거쳐 정확도가 높게 나온 모델에 대해 Dropout을 층마다 추가해줌으로써 모델 구축을 완료하였다. 아래의 그림 5은 모델의 학습 진행에 따른 정확도와 손실도를 시각화하여 비교한 그래프이다.

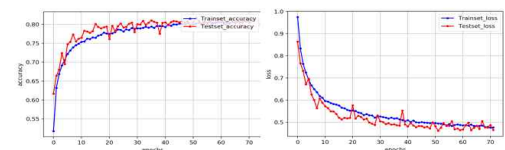


그림 5. 옷 정보 인식 모델 정확도(좌)와 손실도(우) 그래프

Early Stopping Callbacks^[24] 옵션을 적용하여 학습을 15회 진행하는 동안 테스트 세트의 정확도가 높아지지 않으면 학습을 중단 하도록 하여 정확도를 보장해 주었다. 위의 과정을 통해 학습을 진행하였고, 70회를 진행 했을 때 가장 적합한 정확도와 손실도 모습을 보였다.

아래의 표 5는 옷 정보 인식 모델의 층수와 Dropout 정도에 따른 모델의 정확도를 나타낸 표이다.

표 5. 모델 layer 수와 Dropout에 따른 테스트 셋 정확도

	Layer	Dropout	테스트셋 정확도
No.1	1층	x	0.63
No.2	2층	x	0.73
No.3	3층	x	0.74
No.4	4층	x	0.81
No.5	4층	0.125	0.80
No.6	4층	0.2	0.81
No.7	4층	0.3	0.81

Test set과 Training set의 정확도가 0.8까지 올라간 것을 확인할 수 있다. 표5에서 확인된 모델 중 테스트 셋 정확도가 0.8 이상으로 비교적 높게 나온 모델을 가지고 실제로 예측을 수행해보았다. 예측 수행은 구글 사이트에 옷을 검색해 웹 크롤링^[25]을 하여 얻어진 이미지를 통해 진행한다. 아래의 표 6은 웹 크롤링을 통해 얻은 테스트 셋에 대한 정확도가 0.8 이상인 모델별 예측 정확도 표이다.

본 논문에서는 가장 정확도가 높은 모델인 5번 모델을 이용하여 뷰클로젯의 옷 인식 모델을 구현하였다.

표 6. 웹 크롤링 테스트 셋에 대한 모델별 예측 표

	상의	하의	한 벌 옷
No.4	85%	70%	70%
No.5	95%	70%	60%
No.6	85%	60%	55%
No.7	85%	70%	70%

4. 결론 및 향후 연구과제

기존에 개발되어 있는 스마트 옷장은 비장애인에게 초점을 맞추어 옷 자체를 관리하는 기능이 중점인 스마트 옷장이 대다수이다. 본 논문의 뷰클로젯은 음성 인식, 옷 정보 인식 기능 등의 다양한 기능을 통해 시각장애인의 독립적인 의생활에 도움을 줄 수 있을 것이다. 특히 뷰클로젯의 옷에 대한 종류, 패턴, 재질, 상.하의, 색상 등에 대한 인식 모델은 실험 결과 정확도가 80%로 높다. 더불어 옷을 인식하는 과정에서 정확도를 높이기 위해 모델 층의 개수와 Maxpooling을 조정하고, 정확도를 보장하기위해 Early Stopping Callback, Dropout을 적용해 과적합을 방지해주어 모델을 만드는 방법은 옷 인식뿐 아니라 다양한 분야의 이미지 인식 딥러닝 모델 생성 연구에 도움을 줄 수 있을 것으로 보인다.

또한, 딥러닝 모델에 대한 인식 정확도에 대한 추가적인 개선은 학습 과정에서 사용한 'Deep Fashion Dataset'의 라벨링의 정확도를 높이거나, 새로운 데이터 셋을 학습을 통해 개선 될 것으로 판단된다. 더불어 CNN 알고리즘의 발전된 알고리즘인 Mask R-CNN을 이용한다면 이동 중에 발견하는 옷에 대한 인식도 가능할 것이다.

참고문헌

[1] Woo-ri Choi, "You can be blind, too.", The Hankyoreh, Nov. 02, 2012.
<https://url.kr/agDCM2>
 [2] Ji-yeon Lee, "Blind people choose clothes like this.", The Ohmynews, Feb. 25, 2017.
<https://url.kr/zDvIf9>

[3] Seul-gi Lee, "the inconvenience of the deaf and the blind who feel sadder.", The Ablenews, Sept. 27, 2018.
<https://url.kr/7yWzu8>
 [4] You-min Jeun, "Laundroid, smart wardrobe for people who are too lazy to fold their laundry", The JoinsNews, Oct. 28, 2018.
<https://news.joins.com/article/23071312>
 [5] Sae-Young Heo, Sang-Wook Park, Soon-Ho Jung, Jong-Ho Woo, "Implementation of Smart Wardrobe based on IoT", Dept. of Computer Engineering, Pukyong National University, Proceedings Vol. 25, No. 2, pp.949-952, 2018.
<https://url.kr/kAvHCn>
 [6] Arduino UNO R3, <https://url.kr/sacBzd>
 [7] RaspberryPi 4B, <https://url.kr/Deqj7f>
 [8] Magnetic door sensor MC-38,
<https://url.kr/bmBGQu>
 [9] DHT11 sensor, <https://url.kr/gEfZrb>
 [10] LED BAR LED655, <https://url.kr/xvBoPa>
 [11] FAN moter CT-12025L12R-2P,
<https://url.kr/5jgv8G>
 [12] USB Camera OV5640, <https://url.kr/9HO78f>
 [13] USB MIC BT042, <https://url.kr/ervDnH>
 [14] USB Speaker, <https://url.kr/UjV8pa>
 [15] AWS EC2 P2 instance,
<https://aws.amazon.com/ko/ec2/instance-types/p2>
 [16] Anaconda PyCham pro IDE,
<https://www.jetbrains.com/pycharm/>
 [17] Serial UART , <https://url.kr/WNZQT5>
 [18] linux I2C Tools documentation,
<https://i2c.wiki.kernel.org/index.php/I2C>
 [19] sys/socket.h, <https://url.kr/TBWKX2>
 [20] Naver Clova API, <https://clova.ai/ko>
 [21] Grap Cut, <http://www.gisdeveloper.co.kr/?p=6747>
 [22] Webcolors Library, <https://pypi.org/project/webcolors/>
 [23] Jong-chan Yoon, Hak-chul Kim, Jon-jin Kim, Sung-dae Yoon. "A Study on the CBR Pattern using Similarity and the Euclidean Calculation Pattern", KCI, 14(4), pp.875-885, 2010.
<https://url.kr/vMqDrA>
 [24] Jason Brownlee , "Use Early Stopping to Halt the Training of Neural Networks At the Right Time.", Machine Learning Mastery, Dec. 10, 2018.
<https://url.kr/yaoCvz>
 [25] Dong-min Seo, Han-min Jeong, "Intelligent Web Crawler for Supporting Big Data Analysis Services". The Korea Contents Association, 13(12), 575-584, 2013.
<https://url.kr/cTUiVW>