딥러닝을 이용한 손검출에 관한 연구

박명숙, 김상훈*
한경대학교 전기전자제어공학과
e-mail:kimsh@hknu.ac.kr

A Study on Hand Detection using Deep Learning

Myeong-Suk Pak, Sang-Hoon Kim*
Dept of Electrical, Electronic and Control Engineering, Hankyong National University

요 약

딥러닝은 이미지 분류 및 객체 검출과 같은 여러 컴퓨터 비전 관련 작업에 성공적으로 사용되었다. 손 검출은 인간 컴퓨터 상호작용 분야에서 손 분류 및 손 동작 인식을 위한 매우 중요한 부분이며 딥러닝을 사용하여 시도되었다. 본 연구에서는 손 데이터 셋을 이용하여 컨볼루션 신경망을 훈련시킨 다음 학습된 특징을 시각화하고, CNN 아키텍처와 손 데이터 셋의 결과를 각각 살펴보며 손 검출에 대한 이해를 제공한다.

1. 서론

손 동작 의사소통을 사용하는 인터페이스는 여러 응용 분야에서 지난 30년 동안 많은 연구자들에 의해 연구되고 개발되었다[1]. 손을 장치로 사용하면 사람들이 보다 직관적인 방식으로 컴퓨터와 통신할 수 있다[2]. 손 검출은 손 분류와 손 동작 인식을 위한 이전 단계에서 중요한 부분이며 인간 컴퓨터 상호작용, 운전자 행동 모니터링 및 가상현실 분야에서 관련 연구가 수행되고 있다.

여러 종류의 시각적 특징과 색, 모양, 움직임 등의 조합을 활용하는 많은 방법이 문헌에서 제안되어왔다[2]. 그러나 손 모양과 다양한 조명 환경의 변화로 인해 손을 검출하기가 어려운데, 딥러닝을 이용한 손 검출이 시도되어 좋은 성능을 보였다[3,4].

최근에는 딥러닝 아키텍처를 이해하고 개선하기 위해 시각화에 대한 연구가 수행되었다. 본 연구에서는 두개의 데이터 셋으로 합성곱 신경망(Convolutional Neural Network, CNN)을 훈련시킨다. 그리고 시각화 기술을 적용하여 네트워크가 어떻게 손을 학습하는지를 보여주고 두개의 CNN 아키텍처와 두개의 데이터 셋의 결과를 살펴보고 손 검출에 대한 이해를 제공한다.

2. CNN을 이용한 손 검출

손 검출을 구현하기 위해서는 객체 검출 방법 및 데이터가 필요하다. 이 장에서는 최근 CNN 아키텍처를 요약하고 손 데이터 셋을 소개한다.

2.1 CNN 아키텍처

최근의 객체 검출을 위한 CNN 방법은 2단계 검출과 단일 단계 검출로 나눌 수 있다. 2단계 검출에는 R-CNN[5], Fast R-CNN[6], Faster R-CNN[7] 및 그 변형들이 있고, 단일 단계 검출에는 SSD[8]와 YOLO[9]가 있다. Faster R-CNN은 region proposal이 Fast R-CNN에서 계산 시간을 소비하는 문제를 해결하기 위해 Region Proposal Network(RPN)를 도입하고, 컨볼루션 특징(convolutional features)를 공유하여 RPN과 Fast R-CNN을 단일 네트워크로 병합한다. YOLO는 실시간 객체 검출 시스템으로 단일 컨볼루션 네트워크를 통해 여러 경계상자(bounding box)와 클래스 확률을 동시에 예측한다. YOLOv2[10]는 정확도와 속도를 향상시키고, 19개의 컨볼루션 레이어와 5개의 맥스 풀링 레이어가 있는 darknet-19라는 새로운 네트워크를 사용한다.

2.2 데이터 셋

CNN 아키텍처를 훈련하기 위해 두개의 인기 있는 손 데이터 셋을 사용한다.

VIVA hand database[11]는 조명 변화, 큰 손 움직임 및 일반적인 폐색에 대한 자연스러운 운전 설정에서 수집된 54개의 비디오에서 운전자 및 승객 주위의 2D 경계상자로 구성된다. 1인칭 관점을 포함하여 7가지 가능한 관점이 있다. 일부 데이터는 테스트 베드에서 캡처된 것이고 일부는 YouTube에서 제공한 것이다. 챌린지 평가 프로토콜에서 표준 평가 셋은 5,500개의 훈련 이미지와 5,500개의 테스트 이미지로 구성된다.

Oxford hand dataset[12]은 다양한 공개 이미지 데이터 셋 출처에서 수집된다. 총 13,050개의 손 인스턴스에 주석이 추가된다. 경계상자의 고정된 영역(1,500 평방 픽셀)보다 큰 손 인스턴스는 검출을 위해 충분히 큰 것으로

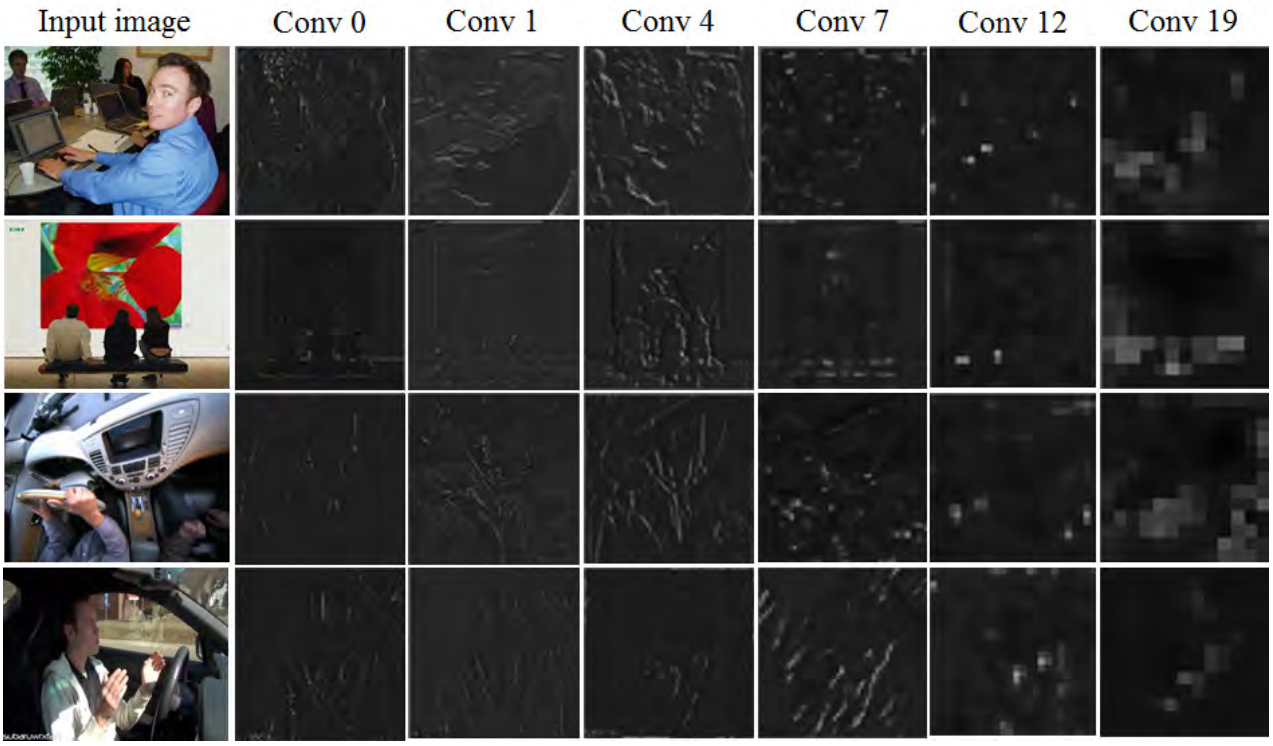
간주되어 평가에 사용된다. 약 4,170개의 고품질 손 인스턴스를 제공한다. 데이터를 수집하는 동안 사람의 포즈 또는 가시성에 대한 제한이 없으며 환경에 부과된 제한도 없었다. 각 이미지에서 사람이 분명히 인식할 수 있는 모든 손은 주석처리 된다. 주석은 손목을 기준으로 축 정렬될 필요가 없는 경계 사각형으로 구성된다.

3. 실험분석

이 장에서는 손 검출을 위한 실험적 분석을 제시한다. 먼저, 특징맵(feature map)의 시각화를 통한 특징 학습 과정을 제시하고, 그런 다음 손 검출 성능에 대해 살펴본다.

3.1 특징 시각화

[13]의 소스코드를 사용하여 컨볼루션 계층의 특징맵을 시각화했다. 그림 1은 YOLO를 사용하여 훈련된 모델을 시각화한 예이다. 하위계층에서 상위계층으로 관련 없는 정보가 제거되고 구별되는 부분이 유지된다. 첫번째 열은 입력 이미지이고 다음 열들은 YOLO의 컨볼루션 레이어의 시각화를 보여준다.



(그림 1) VIVA 및 Oxford 데이터 셋의 시각화

3.2 손 검출 성능

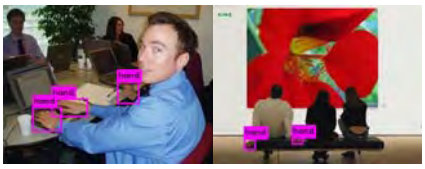
CNN을 이용한 손 감지의 성능을 조사하기 위해 Faster R-CNN과 YOLOv2에 대해 각각 [14]와 [15]를 컴파일하였다. 실험은 Intel i5 6600 3.3GHz CPU, NVIDIA GTX 1060 GPU가 장착된 PC에서 수행되었다. 표 1은 2개의 CNN및 Oxford hand dataset에 대한 손 검출 성능을 보여준다. 821개의 이미지를 가지는 test dataset에 대하여 Faster R-CNN의 평균 정밀도(average precision, AP)는 74.5%로 더 높고, YOLO는 속도가 41 FPS로 더 빠르다.

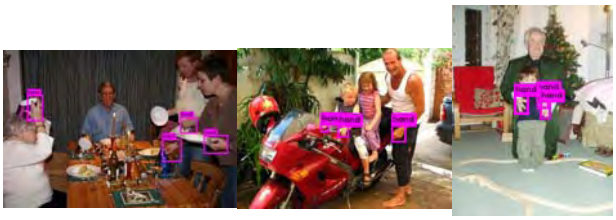
<표 1> 손 검출 성능

Methods	AP	FPS
Faster R-CNN[7]	74.5	6
YOLOv2[10]	73.2	41

깊은 신경망은 많은 양의 학습데이터가 필요하며 성능은 데이터 획득 조건에 따라 달라질 수 있다. 우리는 실시간 검출 성능을 보여준 YOLO에 대해 두 가지 데이터 셋을 훈련시켰고, 각 테스트 데이터에 대한 결과가 그림 2에 나와 있다. 다중 손을 잘 감지할 수 있지만 다른 객체와 겹치거나(그림 2-a의 두 번째 행) 손 모양의 특징이 부족하면(그림 2-b의 두 번째 행) 완전히 검출할 수 없다.

(a) VIVA database





(b) Oxford dataset

(그림 2) YOLO를 이용한 손 검출 결과

4. 결론

최근 몇 년 동안 객체 검출에 딥러닝 접근법이 적용되었다. 손 검출을 이해하기 위해 인기 있는 손 데이터 셋으로 컨볼루션 신경망을 훈련했다. 훈련된 모델을 이용하여 콘볼루션 레이어의 특징맵을 시각화하고 손 분류 과정을 살펴보았다. 실험에서 2단계 객체 검출기인 Faster R-CNN은 손 검출 정확도가 더 높았고, 단일 단계 객체 검출기인 YOLO는 실시간 손 검출 성능을 보였다. 본 연구의 결과를 통하여 모델을 개선하고 신속하고 정확한 손 검출 방법을 연구할 수 있을 것으로 기대한다.

감사의 글

이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임(No. 2015R1D1A1A01057518).

참고문헌

[1] Wachs J.P., Kolsch M., Stern H., Edan Y., "Vision-based hand-gesture applications", Communications of the ACM, vol.54, 2011, pp.60-71

[2] Rautary S.S., Agrawal A., "Vision based hand gesture recognition for human computer interaction:a survey", Springer Transaction on Artificial Intelligence Review, 2012, pp.1-54

[3] Chen T.Y., Wu M.Y., Hsieh Y.H., Fu L.C., "Deep learning for integrated hand detection and pose estimation", International Conference on Pattern Recognition, 2016, pp.4-8

[4] Le T.H.N., Quach G., Zhu C., Duong C.N., Luu K., Savvides M., "Robust Hand Detection and Classification in Vehicles and in the Wild", IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp.1203-1210

[5] Girshick R., Donahue J., Darrell T., Malik J., "Rich feature hierarchies for accurate object detection and semantic segmentation", IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp.580-587

[6] Girshick R., "Fast r-cnn", IEEE International Conference on Computer Vision, 2015

[7] Ren S., He K., Girshick R., Sun J., "Faster r-cnn: Towards real-time object detection with region proposal networks", Proceedings of the 28th International Conference on Neural Information Processing Systems, 2015, pp.91-99

[8] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.Y., Berg A.C., "SSD: Single Shot MultiBox Detector", European Conference on Computer Vision, 2016, pp.21-37

[9] Redmon J., Divvala S., Girshick R., Farhadi A., "You only look once: Unified, real-time object detection", IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp.779-788

[10] Redmon J., Farhadi A., "Yolo9000: Better,faster,stronger", arXiv:1612.08242, 2016

[11] The VIVA Hand Detection Challenge, http://cvrr.ucsd.edu/vivachallenge/index.php/hands/hand-detection/

[12] Mittal A., Zisserman A., Torr P.H.S., "Hand detection using multiple proposals", British Machine Vision Conference, 2011

[13] darknet neural network addon for openFrameworks, https://github.com/mrzl/ofxDarknet/tree/master/example-features

[14] py-faster-rcnn that can compile on windows directly,https://github.com/MrGF/py-faster-rcnn-windows

[15] Windows version of Yolo v2 for object detection, https://github.com/unsky/yolo-for-windows-v2