

효율적인 CNN을 이용한 눈 개폐 검출에 관한 연구

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

A Study on Open/Closed Eye Detection using Efficient CNN

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

요 약

눈의 개폐 검출은 졸음 운전 감지, 온라인 강의에서 수강자 모니터링, 인간 컴퓨터 상호작용(HCI) 등에 적용될 수 있다. 최근에는 모바일 장치에 적용 가능한 효율적인 기법들이 연구되고 있으며 객체 검출 기법과 결합하여 좋은 결과를 보여주고 있다. 본 논문에서는 임베디드 환경에서 적용할 수 있는 가볍고 빠른 딥러닝 방법을 살펴보고, 눈 개폐 검출에 적용하는 방법에 대해 검토한다.

1. 서론

눈의 개폐 검출은 졸음 운전 감지, 온라인 강의에서 수강자 모니터링, 인간 컴퓨터 상호작용(HCI) 등에 적용될 수 있는 분야이다. 눈의 깜빡임 속도에 따라 심리 상태 분석이나 졸음을 감지할 수 있고, 의도적인 깜빡임 패턴을 이용하여 로봇이나 장치를 제어할 수 있다. 영상기반 눈 개폐 검출은 물리적 장치를 사용하지 않고 비교적 먼 거리에서 가능하다는 장점이 있다.

객체 검출은 영상 내에 객체가 있는지 결정하고 그 위치를 찾는 것으로 어려운 작업이다. 지난 몇 년 동안 객체 검출에 뛰어난 성능을 보여주는 딥러닝 기법들이 발표되었다. 그런데 기존의 딥러닝 객체 검출 기법은 임베디드 환경에 적합한 성능을 보여주지 못한다. 임베디드 환경에서는 검출 정확도를 유지하면서도 속도가 빠르고 모델 크기가 작아야 한다. 최근에는 모바일 장치에 적용 가능한 효율적인 기법들이 연구되고 있으며 객체 검출 기법과 결합하여 좋은 결과를 보여주고 있다. 본 연구에서는 로봇과 같은 임베디드 환경에서 눈의 개폐를 검출할 수 있는 딥러닝 기법을 살펴보고 적합한 방법에 검토한다.

2. 효율적인 딥러닝 기법을 이용한 눈 개폐 검출

2.1 효율적인 CNN 아키텍처

모바일 환경에 맞는 아키텍처를 위해 MobilNetV1[1]이 발표되었고, 이 방법은 depthwise 컨볼루션과 pointwise 컨볼루션을 이용하여 연산량을 줄인 네트워크이다. 이를 개선한 MobilNetV2[2]는 기존의 bottleneck 구조와 다르게 선 확장(Expansion)하는 방향으로 변경하여 연산량을 개

선하였다. 이 방법은 정확도가 높으면서 메모리 및 속도에 강점을 가지고 있다.

ShuffleNetV2[3]는 기존의 모바일 환경을 위한 많은 아키텍처들이 FLOPs를 고려하여 설계되었지만 속도와 지연 시간과 같은 direct metric을 고려해야 한다고 강조하면서 효율적인 네트워크 설계를 위한 가이드라인을 제시한다. 채널의 수를 동일하게 유지하고, 그룹 컨볼루션으로 인한 비용을 인지하고, fragmentation을 줄이고, elementwise 연산을 줄이는 것이다. 실험에서 대부분의 지표에서 가장 좋은 성능을 보여주었으며, 객체 검출에서도 좋은 성능을 보여주었다.

2.2 객체검출을 위한 CNN 기법

객체 검출을 위한 CNN 기법에서 1단계 검출기가 성능과 속도를 함께 만족시키는 결과를 보여주었다.

SSD[4]는 객체의 후보영역을 검출하고 픽셀이나 특징을 리샘플(resampling)하던 2단계 검출방법과는 다르게 이 과정들을 제거하여 계산 시간을 줄여 실시간 객체 검출 성능을 보여준다. 높은 검출 정확도를 얻기 위해 여러 크기의 특징 맵과 다른 중형비의 디폴트 상자(default box)를 사용한다. 이러한 설계로 속도와 정확도의 균형을 향상시킨다. MobilNetV2[2]와 결합하여 객체 검출에서 좋은 성능을 나타내었다.

YOLO[5]는 단일 컨볼루션 신경망을 통해 이미지 하나에서 경계상자와 클래스 확률을 동시에 예측한다. 영상을 그리드 셀(grid cell)로 나누고, 해당 객체의 중심점이 위치한 그리드에서 경계상자를 예측하고 신뢰도 점수(confidence score)를 계산한다. YOLOv2[6]에서는

Darknet-19, YOLOv3[7]에서는 Darknet-53이라는 새로운 네트워크를 사용하여 이전 버전보다 강력해졌다. 320 x 320 YOLOv3는 22.2 ms에서 28.2 mAP로 실행되고 SSD[4]만큼 정확하면서 3배 빠르다. 가벼운 버전인 Tiny YOLO는 각 버전마다 포함되어 있으며 속도가 빠르고 모델 크기가 작다.

2.3 눈의 개폐 판단

사람의 눈은 크기와 모양이 다르고 깜빡임이나 하품 등의 행동으로 눈의 형태가 달라진다. CNN 기법을 이용하여 개폐를 검출하기 위해서 뜬 눈과 감은 눈의 기준을 두고 데이터 셋을 준비하여 학습시켜야 한다. 이 과정에서 뜬 눈과 감은 눈을 나누고 경계에 있는 경우에 대한 결정이 필요하다. 졸음 상태, 눈의 깜빡임 등을 알고자 할 때 뜬 눈 상태에서 감은 눈 상태로의 변화나 감은 눈에서 뜬 눈 상태로의 변화 과정이 진행되기 때문에 경계에 있는 상태는 배제하거나 흐름에 따라 판단할 수 있다. 운전, 온라인 강의 등의 일상생활에서 눈의 동작에서 변화는 감은 눈을 기준으로 판단할 수 있다.

3. 실험

실험에서 첫 번째 작업으로 감은 눈의 데이터를 훈련하여 검출을 시도하였다. 감은 눈 검출을 위하여 eyeblink8[8]을 이용하여 훈련하였다. eyeblink8는 네 사람에게 대한 8개의 비디오를 제공하며, 640x480 크기의 70,992개의 프레임이 있다. 1,799개의 감은 눈에 대해 주석을 추출 및 편집하여 사용하였다.

그림 1은 Tiny YOLOv3로 훈련한 모델을 이용하여 Talking face[9]의 데이터에서 감은 눈 검출 예를 보여준다. 감은 눈은 어두운 부분이 가로로 길게 분포되어 있고 특징이 뚜렷하지 않아 주석을 만들 때 다른 특징(눈썹 등의 얼굴 영역)을 고려할 필요가 있다.



(그림 1) 감은 눈 검출 예

4. 결론

최근 임베디드 환경에 적합한 가볍고 빠른 딥러닝 기법들이 연구되고 있고 이를 객체 검출에 적용하여 좋은 성능을 보여주었다. 본 연구에서는 로봇과 같은 모바일 장치에서 눈의 개폐를 검출할 수 있는 효율적인 CNN 기법

을 살펴보고, 눈 개폐 검출의 첫 번째 작업으로 이들 기법에서 감은 눈 데이터를 훈련시켜 검출 결과를 살펴보았다. 다음 작업으로 더 다양한 사람과 환경에서 뜬 눈과 감은 눈 데이터를 수집하고 검출에 맞는 주석처리가 필요하며, 성능비교와 적합한 딥러닝 기법에 대한 연구가 이어진다.

참고문헌

[1] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications" arXiv preprint arXiv:1704.04861, 2017

[2] Sandler M., Howard A., Zhu M. Zhmoginov A. and Chen L.C. "MobileNetV2: Inverted Residuals and Linear Bottlenecks" arXiv preprint arXiv:1801.04381, 2018

[3] Ma N., Zhang X., Zheng H.T. and Sun J. "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design" arXiv preprint arXiv:1807.11164, 2018

[4] 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

[5] 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

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

[7] Redmon J. and Farhadi A. "YOLOv3: An incremental improvement" arXiv preprint arXiv:1804.02767, 2018

[8] Dataset Eyeblink8, <https://www.blinkingmatters.com/research>

[9] Talking face video, http://www-prim.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html