

효율적인 작은 객체 검출을 위한 균형적인 성능의 YOLOv3-tiny

이경민, 송혁, 김제우 *인치호
전자부품연구원, *세명대학교

leegis0904@keti.re.kr, hsong@keti.re.kr, jwkim@keti.re.kr, *ich410@semyung.ac.kr

Balanced performance for Efficient Small Object Detection YOLOv3-tiny

Kyung-Min Lee, Hyok Song, Je Woo Kim, *Chi-Ho Lin
Korea Electronics Technology Institute, *Semyung University

요 약

본 논문에서는 작은 객체를 검출하기 위한 수정 된 YOLOv3-tiny 를 제안한다. 컴퓨터 비전에서 작은 객체 검출은 제한된 해상도와 정보로 검출하기 어렵다. 이 문제를 해결하기 위해 기존 방법의 대부분은 높은 정확도 향상을 위해 속도를 희생한다. 본 논문은 정확도와 속도가 균형적인 성능을 통해 빠른 속도로 작은 객체를 검출하는 것을 목표로 한다. 실험은 WIDER FACE 와 자체 수집한 데이터베이스에서 기존 YOLOv3-tiny 보다 높은 87.48% mAP 를 얻었으며, 속도는 각각 100.5FPS 로 YOLOv3-tiny 보다는 느리지만 높은 정확도와 YOLOv3 보다는 빠르지만 낮은 정확도를 통해 균형적인 성능을 얻을 수 있다.

1. 서론

최근 딥러닝을 활용한 객체 검출 기술이 주목받고 있다. 딥러닝 기술은 다양한 응용 분야에서 활용되고 있으며 특히 영상분야에서 좋은 성능을 보여주고 있다. 이러한 딥러닝 기술은 보다 효율적인 검출을 위한 연구가 활발히 진행되며 그 중에서 Darknet 의 YOLO[1]는 간단한 처리과정으로 속도가 빠르며 기존의 다른 real-time 검출 시스템들과 비교할 때 높은 정확도 성능을 가졌다. 그러나 작은 객체 대한 낮은 정확도의 단점이 있어 이후에 기존 YOLO 의 성능을 향상시키지 위해서 여러가지의 작은 변화들을 적용하여 YOLOv2, YOLO9000[2] 등 성능 개선하였으며, 현재 YOLOv3 [3]에서 기존 YOLO 의 작은 객체에 대한 낮은 검출 성능을 개선하였다. 그러나 실제 산업 분야에서는 다소 낮은 정확도 성능 대신에 빠른 속도와 단일 프로세싱 보다는 멀티 프로세싱을 통해 비용을 절감 할 수 있는 성능이 필요하다. 따라서 YOLOv3 의 보다는 빠르고, YOLOv3-tiny 보다는 정확한 균형적인 성능을 가진 딥러닝 모델이 필요하다.

본 논문에서는 YOLOv3 와 YOLOv3-tiny 의 각 높은 정확도와 빠른 속도의 장점을 가진 수정된 YOLOv3-tiny 를 제안한다. 제안한 모델은 YOLOv3 처럼 작은 객체에서도 검출 성능이 높고, YOLOv3-tiny 와 같이 빠른 속도를 가진 성능 위해 YOLOv3-tiny 아키텍처에서 max-pooling 과정을 대신에 YOLOv3 아키텍처처럼 FCN(Fully Convolutional Network) 적용하였다. 이러한 모델 아키텍처를 통해 실제 산업 분야에서도 적용 가능한 모델 아키텍처를 제안한다.

2. 본론

YOLOv3 는 작은 객체를 검출하기 위해 FCN 를 통한

마지막 layer 를 2 배 Upsample 을 두 번하여 그전 layer 와 합친 Feature 를 이용하여 예측을 추가로 하는데 그림 1 과 같이 총 3 가지 scale 로 예측하여 작은 객체에 대한 검출 정확도를 높였다.

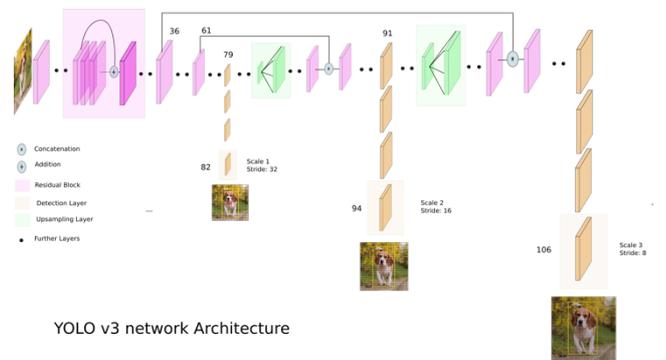


그림 1. YOLOv3 network Architecture

또한 YOLOv3-tiny 는 YOLOv3 의 정확도보다 속도를 높인 아키텍처로 FCN 대신에 Convolutional 과 Pooling 과정으로

연산 리소스를 줄여 속도를 높였으며, 한 번의 Upsample 을 통해 작은 객체에 대한 정확도 성능 또한 높였으나 YOLOv3 보다는 현저히 낮다. 제안하는 방법은 그림 2 처럼 YOLOv3-tiny 아키텍처를 수정한 검출기로서 기존 YOLOv3-tiny 의 max-pooling 의 단점인 위치나 공간에 대한 정보 손실로 단순한 Upsample 과정에서 객체에 대한 검출 정확도가 낮은 문제점을 FCN 을 활용하여 위치 정보를 유지와 YOLOv3

보다 적은 layer 를 쌓는 방법으로 연산 속도를 높였다. 또한 낮은 객체에 대한 검출 정확도 성능을 높이기 위해 YOLOv3 처럼 두 번의 Upsample 과정을 통한 검출을 하도록 구성하였다.

layer	filters	size	input	output
0 conv	16	3 x 3 / 1	416 x 416 x 3	416 x 416 x 16
1 conv	32	3 x 3 / 2	416 x 416 x 16	208 x 208 x 32
2 conv	16	1 x 1 / 1	208 x 208 x 32	208 x 208 x 16
3 conv	32	3 x 3 / 1	208 x 208 x 16	208 x 208 x 32
4 res	1		208 x 208 x 32	208 x 208 x 32
5 conv	128	3 x 3 / 2	208 x 208 x 32	104 x 104 x 128
6 conv	64	1 x 1 / 1	104 x 104 x 128	104 x 104 x 64
7 conv	128	3 x 3 / 1	104 x 104 x 64	104 x 104 x 128
8 res	5		104 x 104 x 128	104 x 104 x 128
9 conv	128	3 x 3 / 2	104 x 104 x 128	52 x 52 x 128
10 conv	256	1 x 1 / 1	52 x 52 x 128	52 x 52 x 256
11 conv	128	3 x 3 / 1	52 x 52 x 256	52 x 52 x 128
12 res	9		52 x 52 x 128	52 x 52 x 128
13 conv	256	3 x 3 / 2	52 x 52 x 128	26 x 26 x 256
14 conv	512	1 x 1 / 1	26 x 26 x 256	26 x 26 x 512
15 conv	256	3 x 3 / 1	26 x 26 x 512	26 x 26 x 256
16 res	13		26 x 26 x 256	26 x 26 x 256
17 conv	512	3 x 3 / 2	26 x 26 x 256	13 x 13 x 512
18 conv	256	1 x 1 / 1	13 x 13 x 512	13 x 13 x 256
19 conv	512	3 x 3 / 1	13 x 13 x 256	13 x 13 x 512
20 res	17		13 x 13 x 512	13 x 13 x 512
21 conv	1024	3 x 3 / 1	13 x 13 x 512	13 x 13 x 1024
22 conv	256	1 x 1 / 1	13 x 13 x 1024	13 x 13 x 256
23 conv	512	3 x 3 / 1	13 x 13 x 256	13 x 13 x 512
24 conv	27	1 x 1 / 1	13 x 13 x 512	13 x 13 x 27
25 yolo				
26 route	22			
27 conv	128	1 x 1 / 1	13 x 13 x 256	13 x 13 x 128
28 upsample		2x	13 x 13 x 128	26 x 26 x 128
29 route	28 15			
30 conv	256	3 x 3 / 1	26 x 26 x 384	26 x 26 x 256
31 conv	27	1 x 1 / 1	26 x 26 x 256	26 x 26 x 27
32 yolo				
33 route	29			
34 conv	64	1 x 1 / 1	26 x 26 x 384	26 x 26 x 64
35 upsample		2x	26 x 26 x 64	52 x 52 x 64
36 route	35 12			
37 conv	128	3 x 3 / 1	52 x 52 x 192	52 x 52 x 128
38 conv	27	1 x 1 / 1	52 x 52 x 128	52 x 52 x 27
39 yolo				

그림 2. The proposed modified YOLOv3-tiny detection layers

3. 실험

실험은 WIDER FACE[4], 자체 수집한 데이터베이스에서 YOLOv2, YOLOv3, YOLOv3-tiny 와 제안한 YOLOv3-tiny 의 정확도와 속도를 비교하였다. 검출 객체는 총 4 가지로 얼굴, 사람, 번호판, 신분증 이미지 데이터를 사용하였으며, 학습 데이터 17,056 장, 검증 데이터 1,895 장으로 총 18,951 장의 이미지 데이터를 200,000 번 학습하였다. 그 결과 표 1 과 같이

표 1 Deep learning model comparison result

Model	mAP	FPS
YOLOv2	75.41%	39.5
YOLOv3	88.99%	30.3
YOLOv3-tiny	67.93%	155.2
Proposed YOLOv3-tiny	87.48%	100.5

정확도는 각 75.41%, 88.99%, 6 7.93%, 87.48% 로 YOLOv2 와 YOLOv3-tiny 보다는 12.07%, 19.55% 향상되었으며, 속도에서는 YOLOv2, YOLOv3 보다는 61 FPS, 70.2 FPS 빠르다.

3. 결론

본 논문에서는 작은 객체를 검출하기 위한 균형적인 속도와 정확도 가진 수정된 YOLOv3-tiny 를 제안하였다. 제안한 모델은 실제 산업 분야에서 적용 가능하도록 YOLOv3 의 높은 정확도를 잃지 않고 YOLOv3-tiny 의 속도를 유지 할 수 있도록 설계하였다. 설계된 YOLOv3-tiny 는 기존 YOLOv3-tiny 의 pooling 과정을 대신하여 FCN 을 사용하여 연산 리소스를 줄임과 동시에 검출 객체의 위치와 공간의 정보를 유지하였다. 또한 한 번의 Upsample 과정을 두 번으로 확장하여 작은 객체를 검출할 수 있도록 설계하여 속도와 정확도를 균형적인 성능을 보이는 장점을 가질 수 있었다. 본 논문에서 실험은 WIDER FACE 와 자체 수집한 데이터베이스에서 실험하였으며, 그 결과 정확도는 각 75.41%, 88.99%, 6 7.93%, 87.48% 로 YOLOv2 와 YOLOv3-tiny 보다는 12.07%, 19.55% 향상되었으며, 속도에서는 YOLOv2, YOLOv3 보다는 61 FPS, 70.2 FPS 빠른 성능 얻을 수 있었다. YOLOv3 보다는 낮은 정확도 성능 대신에 속도를 높였으며, YOLOv3-tiny 보다는 느린 속도를 대신에 높은 정확도 성능을 보여 실제 산업 분야에서 적용 가능한 딥러닝 모델을 증명할 수 있었다.

Acknowledgement

본 논문은 2017 년도 서울시 도시문제 해결형 기술개발 지원사업(과제번호 2016-시정-04) 의 지원을 받아 수행한 결과입니다.

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015. 5, 6.

[2] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," arXiv preprint arXiv:1612.08242, 2016

[3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018

[4] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang, " WIDER FACE: A Face Detection Benchmark", arXiv preprint arXiv:1511.06523, 2015.