AI학습 맞춤형 이미지 데이터셋 구성에 대한 연구

이조순[O], 고병국[*], 강은수[*], 최하진[*], 김준오[*], 이병권[**]
[O]동국대학교 멀티미디어공학과,
[*]동국대학교 멀티미디어공학과,
[**]서원대학교 멀티미디어과
e-mail: sgs1159@naver.com[O]

AI Learning Cookie Image Data Set Construction

JoSun Lee[*], Ko Byeongguk[*], Eunsu Kang[O], Choi Hajin[*], Jun O Kim[*], Byongkwon Lee[**]
[O]Dept. of Multimedia Engineering, Dongguk University,
[*]Dept. of Multimedia Engineering, Dongguk University,
**Dept. of Multimedia Seowon University

● 요 약 ●

본 논문에서는 컴퓨팅 이미지 객체인식 시스템인 YOLO 성능 향상을 위한 효율적인 이미지 마킹 정책을 제안한다. 이 정책은 이미지 데이터를 통한 객체인식 학습 YOLO의 객체인식을 높이고 다른 객체와의 구분을 최대화하여 학습 모델의 성능을 높인다. YOLO의 성능을 최대화하기 위하여 YOLO의 학습을 몇 번 시킬 것인지 무엇을 객체로 인식시킬지 동적으로 할당한다. 이때 학습 싸이클에 따라 객체의 인식이 달라지며 어느 싸이클에서 가장 효율적인지, 왜 다른 객체를 같이 학습시켜야 하는지 증명한다. 본 논문에서는 YOLO의 싸이클과 다른 객체 학습에 있어서 최적의 객체인식 싸이클과 학습 성능 향상 면에서 우수함을 보인다.

키워드: YOLO, 싸이클(cycle)

I. Introduction

YOLO의 객체 인식 발전 방안을 위한 서론을 기술하고 있으며, 본 방안을 수행하기 위한 객체는 대한민국에 존재하는 탑으로 구성하였습니다. 발전은 탑을 YOLO에 학습시킴에 따라 YOLO의 학습 싸이클에 따른 문제점, 탑이 아닌 다른 객체 인식에 문제를 잡고 있으며 향후 적절한 학습 싸이클 및 다른 객체 학습을 통해 발전 방안을 제시할 수 있습니다.

II. Preliminaries

1. Related works

1.1 YOLO

YOLO는 네트워크의 최종 출력부분에서 경계박스를 위치 찾기와 클래스 분류가 동시에 이뤄집니다.

단 하나의 네트워크가 한 번에 특징도 추출하고, 경계박스도 만들고, 클래스도 분류합니다. 이런 방식이 다른 객체 인식보다 간단하고 빠른 장점을 가집니다.

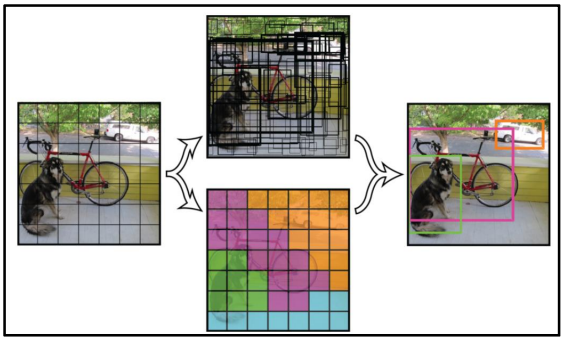

Fig. 1. YOLO Bounding box search and classification

Fig.1 안에는 수많은 경계 박스들과 영상을 7x7그리드로 나눴을 때 해당 그리드 셀 안에는 어떤 클래스가 있는지에 대한 정보(Fig.1. 중앙 2개)가 인코딩되어 있습니다. 가운데 위쪽은 경계 박스에 대한 정보 입니다. 서로 다른 크기의 상당히 많은 경계박스들이 그려져 있습니다[1][2][3]. 네트워크는 영상을 7x7 그리드로 나눕니다. 각 그리드에서 중심을 그리드 안쪽으로 하면서 크기가 일정하지 않은

경계박스를 2개씩 생성합니다. 이 중 경계 박스 안쪽에 어떤 오브젝트가 있을 것 같다고 확신(confidence score)할수록 박스를 굵게 그려줍니다. 굵은 경계 박스들만 남기고 얇은 경계박스(어떤 오브젝트도 없는 것 같다고 생각되는 것들) 을 지운다. 남은 후보 경계 박스들을 NMS(Nom-maximal suppression 비-최대값 억제) 알고리즘를 이용해 선별하면 우측의 이미지처럼 3개만 남게 됩니다.

III. The Proposed Scheme

Fig. 2. Recognize other objects as towers even if they are not towers

기존 탑을 YOLO를 통하여 학습시켰을 경우 탑을 인식하는데 있어서 문제가 없었지만 Fig 2 첫 번째 이미지 같이 다른 이미지를 넣었을 경우에도 객체를 탑을 인식시키는 경우가 발생하였습니다. 그래서 기존 탑 데이터셋에서 배경을 제거한 탑 데이터셋으로 데이터를 수정하여 다시 학습을 진행시켰습니다.

학습을 진행한 결과 1250싸이클에서는 탑을 인식하지 못하였고 1500 싸이클에 있어서 탑을 부분적으로 인식하여 수 많은 경계박스가 도출되었습니다. 1750 싸이클의 경우 탑이 아닌 객체에서 탑을 인식하는 문제가 발생하였습니다. 1750 싸이클 이후에서는 1750 싸이클과 동일하게 다른 이미지도 tower로 인식 하였습니다[4][5].

학습이미지가 대부분 탑이라 전체를 인식하는 것이 아닌가하는 생각을 하여 크기를 다양하게 준 탑 데이터셋 100개를 추가로 학습시켜 보았습니다. 600싸이클의 경우 타워를 부분 인식하기 시작하였고 800싸이클에서부터 타워를 제대로 인식하기 시작하였습니다. 하지만 1000싸이클에서부터 타워인식률이 다시 떨어졌고 1200싸이클이후에서부터는 다른 이미지도 타워를 인식하기 시작하였습니다.

따라서 저희는 타워가 인식된 800싸이클부터 부분 재학습을 시작하였습니다. 800~1000싸이클 사이에 있어서는 탑을 제외한 다른 이미지는 인식되지 않았지만 싸이클 회수에 따라 %가 오르락내리락하며, Bounding Box 또한 달라졌습니다[6][7].

Fig. 3. Learning 100 new top images 600 cycles on the top left, 800 cycles on the top right, 1000 cycles on the bottom left, and 1200 cycles on the left and right.

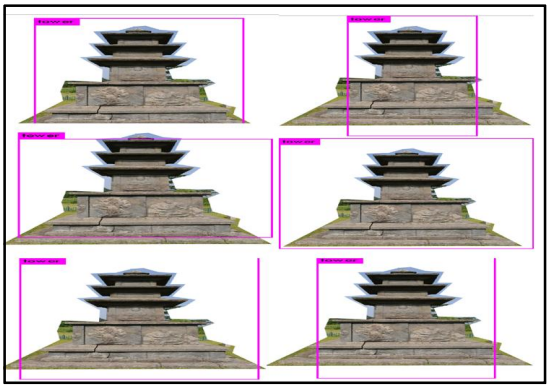

Fig. 4. 960, 940, 920, 900, 880, 860 Cycle (counterclockwise)

IV. Conclusions

YOLO의 학습데이터(탑) 인식에 있어서 일정 범위 학습(800싸이클)에서부터 탑을 제대로 인식하기 시작하였으며 일정 학습 이후(1200싸이클)부터는 새로운 객체가 인식되는 등 문제점이 있었습니다. 그 원인으로 학습 라벨이 tower 하나이기 때문에 대척점이 될 라벨이 없어서 학습이 중첩될수록 전반적인 이미지들이 tower에 대한 가중치가 쌓이는 것을 알 수 있어서 새로운 데이터에 대한 학습 라벨 학습이 필요함을 제안하는 바입니다.

ACKNOWLEDGEMENT

이 (성과)는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2020R1A2C100766811).

REFERENCES

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. ICCV, 2015. 2, 7, 8

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman . Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics, 2009. 3, 6

[3] Y. Bengio. Learning deep architectures for ai. Foundation s and trends in Machine Learning, 2009. 1, 2

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In Computer graphics and interactive techniques, 2000. 3

[5] R. Collobert and J.Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In ICML, 2008. 3

[6] C. Doersch, A. Gupta, and A. A. Efros. Context as superv isory signal: Discovering objects with predictable context. In ECCV, 2014. 2

[7] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. ICCV, 2015.