

Knowledge Distillation 계층 변화에 따른 Anchor Free 물체 검출 Continual Learning

강수명[†], 정대원^{**}, 이준재^{***}

Anchor Free Object Detection Continual Learning According to Knowledge Distillation Layer Changes

Sumyung Gang[†], Daewon Chung^{**}, Joon Jae Lee^{***}

ABSTRACT

In supervised learning, labeling of all data is essential, and in particular, in the case of object detection, all objects belonging to the image and to be learned have to be labeled. Due to this problem, continual learning has recently attracted attention, which is a way to accumulate previous learned knowledge and minimize catastrophic forgetting. In this study, a continual learning model is proposed that accumulates previously learned knowledge and enables learning about new objects. The proposed method is applied to CenterNet, which is a object detection model of anchor-free manner. In our study, the model is applied the knowledge distillation algorithm to be enabled continual learning. In particular, it is assumed that all output layers of the model have to be distilled in order to be most effective. Compared to LWF, the proposed method is increased by 23.3%p mAP in 19+1 scenarios, and also rised by 28.8%p in 15+5 scenarios.

Key words: Continual Learning, Object Detection, Knowledge Distillation, Anchor Free.

1. 서 론

딥러닝의 다양한 학습 방법 중 학습 대상과 정답지를 함께 사용하는 지도 학습(supervised learning)은 제한적이고 폐쇄적인 환경에서 더욱더 좋은 결과를 보인다. 이때 제한적이고 폐쇄적인 환경은 클래스가 사전에 정의되어 변경되지 않고, 데이터가 충분히 수집되어 삭제되지 않는 환경을 말한다. 점차 데이터 증가 속도가 빨라지고 있기 때문에 딥러닝의 폐쇄적인 학습은 비효율적이라고 할 수 있다. 또한 분류 대

상도 점차 세분화되고 있어, Continual Learning(CL, 지속 학습)에서 이러한 문제를 해결하고자 하는 연구가 수행되고 있다[1,2].

CL은 점차 증가하는 데이터와 클래스의 문제를 해결하기 위한 연구 방법으로서, 이 방법에서는 학습된 모델에 태스크(task)나 클래스(class)를 점차 증가 시켜 재학습 시키더라도 이전 학습의 망각을 최소화하고 다음 학습 대상 정확도를 최대화할 수 있어야 한다. 인간의 경우 새로운 학습을 계속하면서도 기존의 학습 내용을 계속 누적시킨다. 또한 장기기억과

※ Corresponding Author : Joon Jae Lee, Address: (42601) 1095, Dalgubeol-daero, Dalseo-gu, Daegu, Republic of Korea, TEL : +82-53-580-6682, FAX : +82-53-580-5165, E-mail : joonlee@kmu.ac.kr

Receipt date : Mar. 7, 2022, Revision date : Apr. 13, 2022
Approval date : Apr. 14, 2022

[†] Dept. of Computer Engineering, Graduate School, Keimyung University (E-mail : smgang.kmu@gmail.com)

^{**} Mathematics Major, Faculty of Basic Sciences, Keimyung University
(E-mail : dwchung@kmu.ac.kr)

^{***} Faculty of Computer Engineering, Keimyung University
※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2020R111A3074667)*

단기기억을 별도로 저장하여 오래전에 학습한 내용이더라도 잊어버리는 문제가 발생하지 않는다. 딥러닝이 인간의 학습 방법을 모방하긴 하나, 학습을 누적한다는 측면에서는 아직 한계가 있으며, 인간을 완벽하게 모방하기 위해서는 기존 학습이 누적 지속적인 학습이 가능해야 한다.

CL은 아직까지 분류(classification) 문제에 집중되어 있으며, 점차 물체 검출(object detection) 및 의미 분할(semantic segmentation) 모델에도 확장되고 있다. 본 논문에서는 물체 검출 모델 중에서도 anchor를 사용하지 않는 anchor-free 모델을 CL이 가능하도록 모델 구조를 변형한다. anchor의 경우 모델의 학습 과정에서 사전 설정해 주어야 하는 파라미터로서, 물체 검출 모델이 학습 과정에서 물체를 예측할 때 물체의 크기와 비율이 사전적으로 정의되는 것이다. 연구실 환경에서 CL의 경우 신규 학습 대상이 어떤 물체인지 알 수 있지만, 실제 환경에서의 CL 학습에서는 어떠한 물체가 새로 학습될 것인지 모를 것으로 판단하여 anchor가 연속되는 모델 학습과는 적합하지 않을 것으로 판단하였다. 따라서 anchor-free 모델을 선정하여 연구에 사용하였다.

또한, 1차 학습에서 사용한 데이터를 2차 학습에서는 재사용하지 않고, 또한 새로운 학습 데이터 내에서 레이블 재사용을 하지 않더라도 기존 학습 내용의 망각을 최소화 할 수 있도록 한다. 본 논문에서 제안한 모델은 Knowledge Distillation(KD, 지식 증류)[3]방법을 활용하여, 1차 학습한 모델을 teacher 모델(t-모델), 2차 학습할 모델을 student 모델(s-모델)로 두고 t-모델로부터 지식을 전달받는 방법이다 [1,2,4].

기존 다양한 연구에서 KD 기법을 사용하며, 물체 검출 모델에서도 KD 기법을 모델에 적합하게 수정하여 사용하고 있다. 하지만 anchor-free 모델에서는 anchor를 사용하는 YOLOv3[5]와 같은 물체 검출 모델과는 달리 클래스 정보와 물체의 위치 및 크기 정보를 담고 있는 최종 출력 계층이 모두 맵 형식으로 출력되므로 어떤 출력층을 KD 해야 할 것인가에 대한 호호성이 있다[4]. 본 논문에서는 모든 출력층을 사용하는 것이 가장 효과적일 것으로 보고, 다양한 출력층의 변화를 통해 어떤 조합이 가장 좋은 결과를 제시하는지 비교검증 한다.

Anchor-free 모델 중에서도 CenterNet [6]을 사용

하여 해당 모델의 출력층인 히트 맵(heatmap), 크기 맵(WH map), 오프셋 맵(Offset Map) 중 가장 중요한 클래스 및 중심 정보를 담고 있는 히트 맵을 기본으로 다른 맵을 조합하여 KD를 수행한 후 결과를 비교한다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문과 관련된 CL 및 KD에 관한 연구를 살펴본다. 3장에서는 제안한 방법에 대해 살펴보고, 4장에서는 제안한 방법을 활용한 실험 및 고찰을 수행하여, 객관적인 지표로 비교 및 성능을 평가한다. 마지막으로 5장에서 결론을 맺는다.

2. 기존 연구

2.1 Continual Learning과 Knowledge distillation

CL은 실제 환경에서 인간이 학습하는 것과 마찬가지로 연속적으로 분류 및 검지해야 하는 대상이 점차 증가한다. 인간은 지속해서 지식을 습득한다고 하더라도 이전에 습득된 지식을 잊어버리지 않는다. 지속 학습에서는 인간이 지식을 누적하는 것처럼 연속적인 학습이 가능해야 한다. 지속학습이 가져야 하는 기본적인 특성은 연속적인 학습, 기존 지식 보유 및 축적, 새로운 대상에 대한 점진적인 학습 등과 같은 능력이다[2].

이미 학습된 모델은 재학습되는 과정 즉, 새로운 지식을 받아들이는 과정에서 기존 지식을 상당 부분 잊게 되고, 기존 지식을 과도하게 고정하는 경우 새로운 지식을 받아들이기 힘들다. 이 문제를 안정성-가소성 딜레마(stability-plasticity dilemma)라고 한다. 또한 앞서 언급한 기존 지식을 상당 부분 잊는 문제를 catastrophic forgetting(치명적 망각)이라고 한다. CL의 목적은 catastrophic forgetting을 최소화 하고 지속적인 학습 과정에서 이상적인 결정 경계를 찾는 것이다[2,7].

정규화 기반 접근 방법에서 데이터 정규화의 한 방법인 KD 기반의 CL 방법은 Hinton 등 [3]에 의해서 제안된 모델 압축 기법에서 시작되어 그 적용이 CL로 확장되었다. KD는 사전 학습된 모델이 기존의 클래스 검출을 위해 이전에 학습한 가중치 정보와 노드의 활성화 유무 등을 새로 학습하는 모델을 위해 전달하는 기법이다. 해당 기법은 원래 모델의 압축 등을 위해 제안된 기법이나, 최근 지속학습을 위한

방법으로 주로 사용되고 있다. KD는 기존 학습된 모델을 동결시킨 후 t-모델로 사용하고, 새롭게 학습해야 하는 모델을 s-모델로 두어 teacher로부터 기존에 저장된 지식을 student에 전달하는 것이다. 이때 student 모델은 새로운 데이터를 학습하면서 기존 정보를 전달받아 이전 태스크와 새로운 태스크를 모두 알 수 있게 한다. 지속학습에서는 student 모델은 기존 모델과 최종 출력층을 제외하고 동일한 계층으로 구성되나, 최종 출력층은 태스크 혹은 클래스에 맞게 증가한다. 이 방법을 가장 처음으로 CL에 사용한 연구가 LwF(Learning without Forgetting)[8]이다. LwF를 사례로 하는 CL 적용은 분류 문제에 대하여 제안된 사례로 대부분의 CL 연구가 아직까지 분류 문제에 초점이 맞춰져 있으나 점차 객체 검출과 의미 분할에 대해 연구 대상이 확장되고 있다.

본 논문에서는 클래스 점진 학습을 대상으로 하며, 객체 검출에 관한 연구를 수행하므로 데이터의 구성과 시나리오가 분류 학습과는 다르다. 분류 모델에 CL을 적용하는 경우 데이터 재사용에 대해서만 고려하면 되지만, 객체 검출과 의미 분할 모델에서는 하나의 영상 안에 여러 가지 물체가 있다. 영상 내 물체가 1차 학습에 관계된 것인지, 2차 학습에 관계된 것인지에 따라 여러가지 시나리오가 존재한다. 이에 대해 Michieli 및 Zanuttigh [9]의 연구에서는 기존 데이터의 사용 유무 및 기존 클래스의 레이블 유무에 따라 총 3가지로 구성을 달리할 수 있다고 하였다. 즉, 1) 순차(sequential), 2) 분리(disjoint), 3) 중첩(overlapped)으로 나뉜다[9]. 순차의 경우 이전 데이터는 사용하지 않으나 새로운 데이터에서 이전 레이블도 사용한다. 분리의 경우 이전 데이터도 사용하지 않고, 새로운 데이터에서 이전 레이블이 있다고 하더라도 사용하지 않는다. 중첩의 경우 모든 데이터 누적하여 사용하지만, 이전 레이블은 사용하지 않는다[9].

우선 KD를 적용한 객체 검출 사례 중 하나인 Shmelkov 등[10]은 객체 검출 모델인 Fast RCNN를 활용하여 KD 기반의 CL 모델을 제안하였다. 해당 연구에서는 태스크를 별도로 지정하지 않고 클래스를 점차 증가시키는 학습 시나리오를 따랐다. 앞서 기술한 LwF 기법을 적용하여, 다음 차례에 학습된 모델이 이전 차례에 학습한 정보를 전달받도록 하였다. 해당 연구에서는 데이터 사용 방법을 연속형 방법으로 사용하였다.

Peng 등[4]은 anchor 상자가 없는 객체 검출 모델인 CenterNet과 FCOS(Fully Convolutional One-Stage Object Detection)[11]에 증류 기법을 적용한 객체 검출 모델을 제안하였다. 이 연구는 anchor free 모델에서는 최초로 CL 모델을 제안하였으며, 데이터 사용을 분리형으로 진행하였다. 해당 연구에서는 두 가지 anchor free 모델에서 회귀형식이 아닌 출력층에 대해 KD를 진행하였으며, 영상 간의 관계를 증류하는 방법인 inter-relation을 제안하였다. 이때의 계층은 디코더(decoder)의 가장 끝 계층에서 나온 특징맵을 활용하였다. 한 배치 차례 트레이닝이 끝나면 기존 학습된 모델의 최종 출력부 중 클래스와 관련된 출력층을 다음 차례의 모델에 복구(restore)함으로써 물리적으로 계층이 가진 가중치 정보를 원상 복구한다. 즉, 이전 학습 과정에서 잊어버린 정보를 다시 회복시키는 것에 목적이 있다.

Michieli 및 Zanuttigh [9]는 의미 분할 모델인 DeepLabv3+ 에 증류 기법을 활용하여 여러 차례에 걸쳐 몇 가지 방법의 CL 모델을 제안하였고 다양한 데이터시나리오에 대해 검증하였다[9,12,13]. [9]에서는 증류 방법 외에 3가지의 손실함수를 추가하여 기존 연구 대비 좋은 성과를 제시하였다. 즉, CL 과정에서 생기는 catastrophic forgetting을 최소화하기 위하여 t-모델의 예측과 s-모델의 예측을 KD 기법을 활용하여 증류하였다. 해당 연구 역시 이전 연구에서 KD만으로 효과가 충분하지 않다는 것을 확인하여 1) 특징값의 움직임 줄이도록 하는 프로토타입 매칭(prototypes matching) 손실, 2) 같은 클래스 내의 특징은 서로 당기고 다른 특징은 서로 밀도록 유도하는 대조적(contrastive) 손실, 3) 학습 과정에서 기존 특징들의 밀도를 줄여주어 새로운 특징 분포를 구성할 수 있도록 하는 특징 희소(feature sparsity) 손실을 추가로 제안하였다.

본 논문에서는 현실 세계의 데이터 수집 및 적용에 가장 유사하도록 데이터 시나리오를 구성한다. 즉, 분리(disjoint) 방법의 데이터 시나리오를 사용하며, 1차와 2차 학습의 데이터를 별도로 분리하되, 각 순서에서 관련 없는 레이블은 삭제한다.

2.2 CenterNet

이 절에서는 Zhou 등[6]의 연구에서 설명하는 CenterNet에 대해 살펴본다. CenterNet의 구조에서

는 입력 영상을 압축하는 인코더(Encoder) 부와, 물체의 중심 및 클래스, 크기를 예측하기 위해 다시 출력층을 위해 크기가 늘어나는 디코더부로 구성되어 있다. 즉, ResNet50을 백본(backbone)으로 사용하며 입력영상이 $512 \times 512 \times 3$ 이라고 가정할 때, ResNet-50 백본을 거치게 되면 $16 \times 16 \times 2048$ 으로 줄어들며, 이를 $128 \times 128 \times 32$ 로 크기를 업스케일링(up-scaling) 하도록 디컨볼루션(deconvolution)한다. 최종 계층은 출력을 준비하는 단계로, 가장 최종 출력된 크기인 $128 \times 128 \times 32$ 이 채널 수만 64 채널(channel)로 확장된다. 그 이후 히트 맵($128 \times 128 \times n$), 크기 맵 ($128 \times 128 \times 2$), 오프셋 맵($128 \times 128 \times 2$)의 크기로 출력된다. 여기서 n 은 클래스 개수를 의미한다.

히트 맵의 클래스 개수 만큼의 채널에서 물체의 클래스를 예측할 수 있고, 128×128 의 영상 내 위치에서 물체의 중심을 알 수 있다. 크기 맵에서는 해당 물체의 중심이 있는 동일 위치에서 너비와 높이의 값을 확인할 수 있고, 오프셋에서 원래 사이즈로 복원할 때의 보간 값을 예측한다. 즉, 오프셋 맵은 입력 크기인 512×512 에 비해 출력 계층에서 128×128 로 1/4배가 되므로 다시 크기를 복원할 때 원래 값을 되찾기 위한 소수점 오차 값을 예측한다.

CenterNet의 최종 손실함수는 식 (1) [6]과 같다. CenterNet에서는 최종 출력이 3가지이므로 각 출력에 대한 손실함수를 별도로 계산하여 총 3개의 항을 가지고 있다[6].

$$L_{det} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off} \quad (1)$$

3. 제안한 방법

3.1 Knowledge distillation을 적용한 CenterNet

CenterNet에 KD를 적용하여 CL 모델로 사용할 때 t-모델과 s-모델은 모두 같은 형태이며, 출력층만 일부 변경된다. 즉 클래스와 관계된 히트 맵의 채널 개수만 변경된다. 두 모델 모두 백본으로 모두 ResNet 50을 사용하며, 3번의 디컨볼루션을 통해 입력의 1/4배 크기로 최종 출력 크기가 정해진다. 이 출력이 64채널 특징 맵으로 2차원 컨볼루션을 거쳐, 히트 맵, 크기 맵, 오프셋 맵으로 출력된다.

1차 학습에서는 단일 모델만 활용하며 이때 학습된 모델이 2차 학습부터 t-모델로 사용된다. 차수가 점차 증가할 때, 실험 시나리오에 따라 1차 t-모델을

계속 쓰는 경우도 있고, t-모델을 계속 변경해가며 이전 차수의 모델을 t-모델로 사용할 수도 있다.

CenterNet 모델 구조상에서 가장 정보를 많이 담고 있는 계층은 ResNet의 최종 출력층인 인코더 계층의 가장 끝 레이어와, 디컨볼루션 계층인 디코더 계층의 가장 끝 계층인 64채널 특징 맵 계층과, 마지막으로 출력 계층인 히트 맵, 크기, 오프셋 계층으로 볼 수 있다. 본 논문에서 제안하는 방법은 입력 영상에 대해 예측 내용을 담고 있는 출력 계층 3개를 모두 KD를 수행하는 것이 가장 좋을 것으로 가정한다. 단, 히트 맵의 경우 시그모이드(sigmoid) 함수 적용 전의 특징 맵을 증류하여야 하는데, 그 이유는 시그모이드 과정에서 일부 정보가 사라지기 때문이다.

증류 대상은 Fig. 1과 같다. 점선으로 t-모델과 s-모델을 연결한 부분이 증류 대상이다. 본 논문에서 모든 출력 맵이 증류되어야 한다고 가정 한 것은, 신규 학습되는 물체의 크기에 따라 크기 맵과, 오프셋 맵의 가중치가 달라질 것으로 예상했기 때문이다. 즉, Anchor free 모델은 경계 상자(bounding box)가 없기 때문에 학습 물체에 따라 크기 맵의 변동을 예측할 수 있다.

본 논문에서는 3개 최종 출력 계층에 대해 모두 KD를 적용하며, 각 KD를 위한 손실은 다음과 같이 정의한다. 1차 학습에 사용된 클래스 개수가 n 개이고, 2차 학습 클래스의 총 개수가 $n+m$ 이라고 가정한다. \hat{Y} 은 히트 맵이며 \hat{Y}_t 는 t-모델의 예측 결과, \hat{Y}_s 는 s-모델의 예측 결과이다. Y 는 c^o 개수의 히트 맵과, c^n 개수의 히트 맵이 결합된 $[Y^o \ Y^n]$ 형태이며, $Y^o \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times n}$, $Y^n \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times m}$ 이다. 즉,

$$\hat{Y}_t \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times n} \text{ 이고 } \hat{Y}_s \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times n+m} \text{ 이다.}$$

s-모델의 히트 맵은 t-모델의 히트 맵보다 클래스가 개수가 증가하였으므로 맵 개수가 많기 때문에 이전 학습된 클래스 개수의 채널까지만 잘라서 사용한다. 따라서 클래스 C^o 에 대한 히트 맵만 사용하는 $\hat{Y}_{s_c^o}$ 을 사용하여 L_1 손실을 계산한다. \hat{Y}_t 와 $\hat{Y}_{s_c^o}$ 의 손실 함수 L_{kdHM} 은 아래 식 (2)와 같다. \hat{Y}_t 와 $\hat{Y}_{s_c^o}$ 의 손실은 요소 별 계산하여 평균을 구하는 것으로 행렬의 요소별 계산을 수행한다. 이 수식에서 다운샘플링된 출력층의 높이 $\frac{H}{R}$ 는 H' 라고 하고, $\frac{W}{R}$ 는 W' 라고 정의하였다. C 는 채널 개수 즉, 클래스 개수를 의미

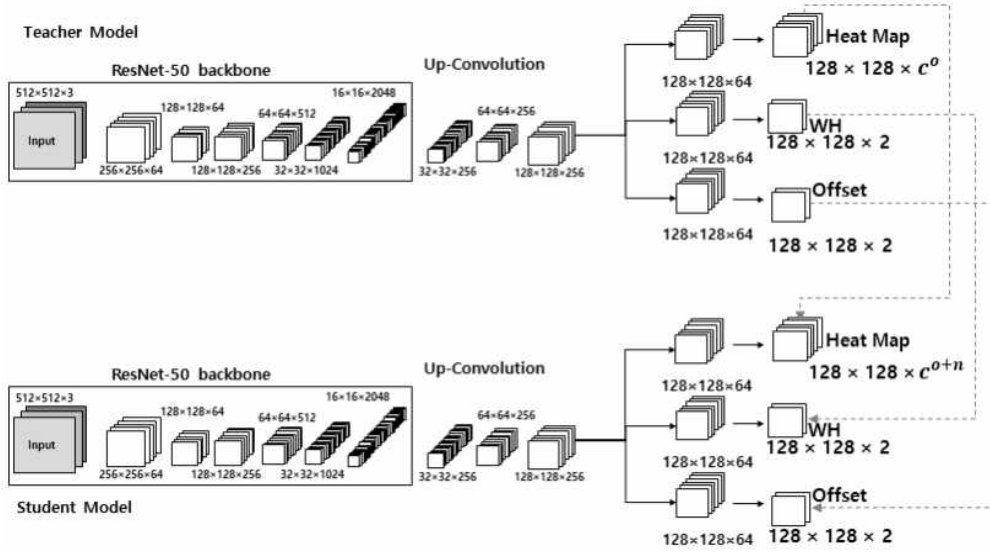


Fig. 1. Proposed method.

하며 이때 개수는 이전 학습에서 사용된 클래스의 개수이다.

$$L_{kdHM} = \sum_{W'H'C^o} |\hat{Y}_t - \hat{Y}_s^e| \quad (2)$$

L_{kdWH} 은 식 (3)과 같고 L_{kdOff} 은 식 (4)와 같다 두 손실함수는 t-모델과 s-모델의 계층 구조가 완전히 동일하기 때문에 H' , W' 에 x , y 에 해당하는 채널 2개에 대해 요소별 계산을 수행한다. 이 손실 역시 L_1 손실 함수를 사용한다.

$$L_{kdWH} = \sum |\hat{S}_t - \hat{S}_s| \quad (3)$$

$$L_{kdOff} = \sum |\hat{O}_t - \hat{O}_s| \quad (4)$$

CenterNet은 히트 맵 연산을 위해 negative의 과적합(overfitting)을 방지하는 focal 손실을 사용하며, 이 연산을 위해 히트 맵에 대해 시그모이드를 적용한 후 손실을 계산한다. 또한, 크기 맵과 오프셋 맵을 위해서 L_1 손실로 연산한다. 앞서 정의한 것처럼 히트 맵을 위한 focal 손실을 L_k 로 보고, 크기 맵 L_1 손실을 L_{size} , 오프셋 맵의 L_1 손실을 L_{off} 라고 할 때, 기존 CenterNet의 손실함수 L_{det} 는 앞서 정의한 식 (1)과 같다. 증류 정보를 위한 손실함수는 L_1 손실을 사용하며 증류 계층은 히트 맵, 크기 맵, 오프셋 맵이다. 최종 손실함수는 식 (5)와 같다.

$$L_{tot} = L_{det} + \lambda_{kdHM}L_{kdHM} + \lambda_{kdWH}L_{kdWH} + \lambda_{kdOff}L_{kdOff} \quad (5)$$

3.2 시나리오 구성

지속 학습 모델을 검증하기 위해서는 일반적으로 분류하고자 하는 클래스가 증가한 상황을 가정한다. 즉, PascalVOC 2007 데이터의 경우 19+1, 15+5 두 가지 시나리오에 대해서 제안한 모델을 검증하며, 이전에 사용한 데이터와 레이블은 새로운 데이터에서 사용할 수 없다. PascalVOC는 레이블 순서가 알파벳의 순서로 되어있으므로 19+1의 경우에는 tv/monitor가 증가하는 상황, 15+5에서는 pottedplant, sheep, sofa, train, tv/monitor가 증가하는 상황을 가정한다.

4. 실험 결과 및 고찰

학습과 테스트를 위한 학습 환경은 Intel(R) Core (TM) i9-10900F CPU @ 2.80GHz, 64GB RAM, GeForce RTX 3090 그래픽카드를 사용하였으며 딥러닝 프레임워크는 python 3.6, Anaconda 3, pytorch 1.7.1와 같이 구성하였다 사용한 CentetNet의 코드는 논문 저자가 게시한 공식 코드[14]를 사용하였다.

학습을 위해서 사용한 PascalVOC 2007과 2012 데이터의 경우 공식 사이트에서 제공하는 자료인 'trainval'을 다운로드 받아, 학습으로 해당 폴더에 들어있는 trainval.txt 데이터를 사용하였으며 PascaIVOC 2007 test 데이터를 테스트용으로 사용한다.

이때 CL 시나리오를 생성하기 위하여 ImageSets 폴더 내부에 있는 Main 폴더에 위치한 클래스별 텍스트 데이터를 기준으로 시나리오별 사용 데이터 집합을 정의하였다. 이때 trainval로 파일명이 끝나는 파일을 학습용으로 사용한다. 사전적으로 t-모델을 학습한 후, s-모델은 증가된 상황을 가정하여 히트 맵의 개수를 신규 클래스만큼 증가시킨다. 또한 teacher 모델의 사전 학습된 데이터를 학습 전 초기화 student 모델에 불러온다. 하지만 딥러닝 연산 프레임워크가 제공하는 가중치 불러오기 기능을 사용할 경우 그 크기가 다를 때는 불러오지 않으므로, student 모델의 히트 맵을 이전 학습 클래스의 개수만큼 잘라 강제로 가중치를 불러온다. 바이어스(bias)에 대해서는 초기화만 진행한다. 이후 학습을 수행한다. KD 증류 연산 시 손실함수 전에 연산을 수행하며 이때 Pytorch 함수 중 clone() 함수를 활용한다. 이 점이 detach()함수 대비 좋은 결과를 보였다.

제안한 모델과의 비교를 위하여 다음과 같은 비교 대상을 설정하였다.

- Fine-tuning : 전이학습(Fine tuning)을 그대로 사용하는 경우
- LWF : LwF(Learning without Forgetting) [8] 기법
- KD-HM : 히트 맵만 증류한 경우
- KD-HM,WH : 히트 맵, 크기 맵을 증류한 경우
- KD-HM,OFF : 히트 맵, 오프셋 맵을 증류한 경우
- KD-ALL : 히트 맵, 크기 맵, 오프셋 맵을 모두 증류 및 한 경우

우선 Table 1에서 일반적인 환경의 학습 결과를 보인다. 이때 테스트는 nms 옵션과 flip test의 옵션을 켜서 실행시킨 결과이다. 해당 결과는 CenterNet의 백본을 ResNet 50으로 사용하여 실험한 결과이므로 최근 제안된 물체 검출 모델 대비 좋은 결과라고 할 수 없다. 단, CL 과정에서 누적된 지식을 어느 정도 있었는지 알 수 있는 비교 자료로 활용할 수 있다.

Table 2는 19+1 시나리오에서의 학습 결과를 확인할 수 있다. 일반적인 학습에서 주로 사용하는 fine-tuning의 방법은 CL에는 전혀 알맞지 않음을 알 수 있다. mAP 기준으로 가장 낮은 결과를 보이는 것을 알 수 있다. 또한 1-19 클래스에 대한 정확도가 가장

Table 1. CenterNet Experiment results in Normal model. (%)

class	20 class	19 class	15 class
aeroplane	73.7	76.6	74.9
bicycle	79.1	81.2	81.3
bird	70.6	72.1	70.1
boat	62.6	60.1	60.3
bottle	56.9	54.5	54.7
bus	79.1	77.7	76.6
car	83.7	84.3	84.1
cat	81.2	80.5	79.6
chair	58.4	58.6	56.9
cow	78.8	79.5	70.5
diningtable	70.5	67.5	63.1
dog	79.3	78.5	76.1
horse	81.6	84.0	82.9
motorbike	79.3	81.9	81.0
person	78.7	79.4	79.2
pottedplant	47.7	45.5	-
sheep	77.9	77.6	-
sofa	70.1	69.7	-
train	79.1	79.5	-
tvmonitor	73.4	-	-
mAP	73.1	73.1	72.8

19.1%로 가장 낮은 것을 알 수 있다. LwF 경우 단순히 fine-tuning을 하는 것에 비해 결과가 일부 개선되었고, fine-tuning을 제외하고는 신규 학습 대상인 tv/monitor 결과가 가장 좋은 것을 알 수 있다. 하지만 전체적인 평균 mAP가 실제 환경에 적용하기에는 무리가 있어 보인다. 19+1시나리오에서, 히트 맵만 증류한 경우 LWF보다 전체 클래스에 대한 mAP가 약 20%p 높아지는 것을 확인할 수 있다. LWF의 방법은 분류 모델에서는 효과적이었으나, 본 논문에서 활용한 anchor-free 모델에서는 효과적이지 않음을 알 수 있다. 이는 anchor-free 모델의 히트 맵 결과는 어떠한 벡터 형태의 결과가 아니라, 텐서 모양의 결과이므로 히트 맵의 출력층이 영상 전체의 정보를 포함하고 있기 때문으로 보인다.

히트 맵을 기본으로 증류하고, 크기 맵과 오프셋 맵을 비교한 경우에는 mAP 기준 크기 맵이 기준 1-19 클래스 정보에 대해 더욱 중요한 정보를 담고 있는 것을 알 수 있다. 하지만 t-모델의 정보를 증류

Table 2. Experiment results in 19+1 scenario.

(%)

Class	Fine-tuning	LWF	KD-HM	KD-HM,WH	KD-HM,OFF	KD-ALL
aeroplane	10.7	41.9	67.8	68.0	67.7	69.1
bicycle	11.7	33.5	68.8	74.4	69.3	73.7
bird	26.1	34	64.8	66.2	65.9	65.6
boat	11.7	26.1	45.4	46.7	46.1	48.6
bottle	8.5	21.5	40.4	43.1	41.7	46.4
bus	12.9	38.9	58.1	66.8	57.3	67.4
car	14.4	45.5	25.0	25.8	32.8	28.5
cat	49	58.4	77.0	78.0	78.0	77.2
chair	9.1	21.4	28.1	43.0	31.4	42.2
cow	30.4	60.6	74.1	73.9	74.4	72.2
diningtable	11.6	22.3	60.5	61.4	61.1	60.3
dog	27.3	39.5	69.7	72.1	70.3	73.6
horse	14.4	40.9	71.1	79.1	74.4	79.6
motorbike	0.5	10.9	60.2	66.8	60.8	62.0
person	27.5	57.1	72.3	71.8	72.6	72.9
pottedplant	9.1	17.8	33.0	30.2	33.3	31.7
sheep	40.5	45	75.3	71.9	75.3	71.0
sofa	19.5	33.1	58.0	61.5	58.5	61.3
train	27.2	54.3	65.9	73.4	67.5	74.9
tvmonitor	42.4	22.6	20.7	13.4	19.2	13.5
mAP	20.2	36.3	56.8	59.4	57.9	59.6
1-19 mAP	19.1	37.0	58.7	61.8	59.9	62.0
20 mAP	42.4	22.6	20.65	13.38	19.18	13.52

하는 과정은 중요 정보를 전달할수록 새로운 지식을 배울 수 없도록 만드는 일종의 trade-off 문제가 발생하므로 KD-HM, WH에서 KD-HM, OFF 보다 신규 클래스의 학습이 저하되는 것을 확인할 수 있다. mAP 기준 가장 결과가 좋은 것은 본 논문에서 제안한 KD-ALL일 뿐만 아니라 기존 학습 결과도 가장 좋은 62.0%를 보였다. 신규 학습은 LWF 대비 결과가 약 9%p 낮은 결과이긴 하지만, 전체 클래스 평균이나, 기존 클래스 기록을 볼 때 실제 현장에 적용할 수 있을 것으로 판단되는 모델은 제안한 KD-ALL이라고 할 수 있다. 특히 KD-HM, WH보다도 전체 클래스 mAP와, 기존 클래스 mAP, 신규 클래스 mAP에서 모두 좋은 결과를 보인다는 걸 알 수 있다. 즉, 증류 시 특정 출력층을 사용하는 것 보다, 모든 출력층을 함께 사용하는 것이 더욱 좋은 결과를 보인다는 것을 알 수 있다. Fig. 2는 19+1 시나리오에 대한

KD-ALL의 결과 영상이며, 이전에 학습된 물체를 잊지 않고 인식하는 것과, 새로운 물체의 인식이 된 것을 확인할 수 있다. 하지만 어느 정도 치명적 망각이 발생했으므로 검출에 대한 확신 능력이 떨어진 것을 알 수 있다.

Table 3은 15+5 시나리오에서의 실험 결과를 보인다. 이 시나리오에서는 fine-tuning보다 오히려 LWF에서 좋지 않은 결과를 보이는 것을 확인하였다. Anchor-free 모델만의 고유한 방법이 필요한 것을 확인할 수 있다. 이 실험에서도 KD-ALL에서 전체 클래스와 기존 클래스에서 가장 좋은 결과를 보이는데, 신규 클래스의 경우 전체 결과 중에서는 가장 낮은 결과를 보이나 실제 환경에서 적용하기 위해서는 전체 결과도 함께 고려해야 하므로 모든 계층의 t-모델의 지식을 전달하는 것이 가장 중요할 것으로 판단된다. 이 실험에서도 KD-HM, WH와 KD-HM, OFF

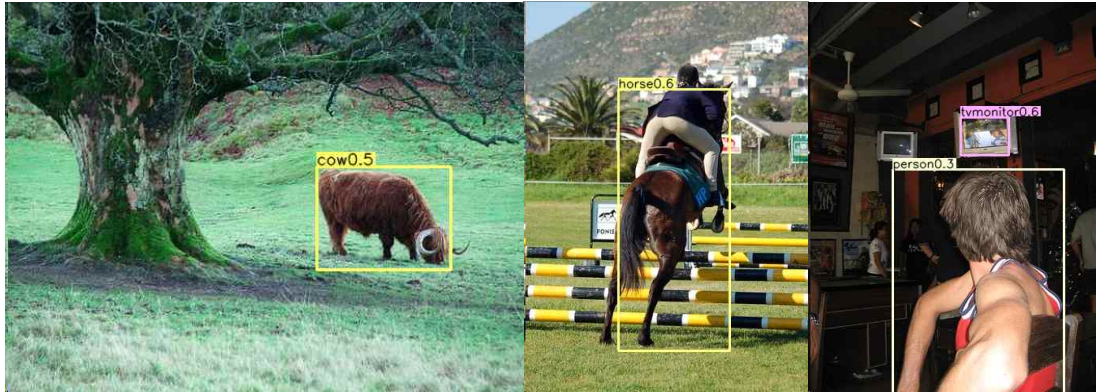


Fig. 2. KD-All results examples of 19+1 scenario experiment.

Table 3. Experiment results in 15+5 scenario. (%)

Class	Fine-tuning	LWF	KD-HM	KD-HM,WH	KD-HM,OFF	KD-ALL
aeroplane	26.4	9.1	33.23	55.9	31.97	61.9
bicycle	0	9.1	61.03	76.49	61.37	76.41
bird	9.1	9.1	4.19	52.33	4.07	56.86
boat	4.9	25.6	1.5	16.8	2.96	17.91
bottle	9.1	0	1.45	21.7	1.91	14.8
bus	0	6.1	2.18	52.92	3.83	52.69
car	34.8	16.2	19.57	8	21.58	7.64
cat	9.1	0	66.73	76.21	66.52	75.67
chair	9.1	0	22.39	6.31	23.02	7.29
cow	18.4	0	11.38	48.93	11.97	45.16
diningtable	0	0	30.92	50.34	32.49	51.69
dog	7.7	0	39.41	64.46	41.8	66.41
horse	27.2	0	65.45	77.87	66.95	77.9
motorbike	0	9.1	25.81	39.53	28.81	47.48
person	35.2	27	55.81	74.98	56.99	74.45
pottedplant	21.4	18.7	17.14	8.5	17.1	8.49
sheep	19.2	19.1	16	12.36	15.97	12.43
sofa	38	35.5	37.04	21.36	37.34	21.8
train	14.5	22.8	16.41	14.82	16.02	15.15
tvmonitor	41.3	32.5	37.86	24.81	38.42	24.08
mAP	16.3	12.0	28.3	40.2	29.1	40.8
1-15 mAP	12.7	7.4	29.4	48.2	30.4	49.0
16-20 mAP	26.9	25.7	24.9	16.4	25.0	16.4

를 비교해 볼 때, 크기 맵이 오프셋 맵보다 증류에서 중요한 역할을 한다는 것을 알 수 있었다.

5. 결 론

본 논문에서는 앵커를 사용하지 않는 CenterNet을 지식 증류 기반 모델로 변경하여 KD 손실 함수를

통해 새로 증가한 클래스에 대해 학습하면서 기존 클래스에 대해 치명적인 망각을 줄이는 지속 학습 모델을 제안하였다. 앞선 연구에서는 히트 맵 혹은 히트 맵 이전의 계층만 증류해도 좋은 결과를 보인다고 하였으나, 실제 실험 결과에서는 모든 출력 계층을 다 KD 하여 t-모델의 지식을 증류하는 것이 catastrophic forgetting에 가장 도움이 되는 것을 확인할 수 있었다.

또한 클래스의 정보를 담고 있는 히트 맵을 기본 KD 계층으로 설정하고, 크기 맵과 오프셋 맵을 비교하였을 때는 크기 맵에서 더 많은 정보를 담고 있는 것을 알 수 있었다. 단 KD 과정은 기존 지식을 잊지 않게 하는 역할이 지나치게 강할 경우 신규 학습이 잘되지 않는다는 점에서 적절한 밸런스를 찾아야 할 것으로 판단된다.

본 논문에서는 이전 데이터와 이전 레이블은 전혀 사용하지 않는 가장 제한이 강한 연구를 수행하였기 때문에 catastrophic forgetting이 완전히 해결되지 않았다. 따라서 이러한 망각 문제를 해결하기 위하여 강한 KD를 적용하였을 때 신규 학습이 잘되지 않는 것을 확인할 수 있었다. 이러한 문제를 해결하기 위해, 기존 데이터를 어떠한 방법으로 남겨둘지에 대한 모델이 제안될 수 있으며, 초상권, 저작권등을 보호하기 위하여 특징 맵을 저장하는 방법 등을 고려할 수 있다.

제안한 모델은 기존 방법 대비 상대적으로 좋은 결과를 보이나, 향후 실제 CL이 필요한 현장에서 사용하기 위해서는 강한 KD에도 신규학습이 더 잘 이루어지는 모델의 제안이 필요할 것으로 판단된다.

REFERENCE

- [1] S. Gang, D. Chung, and J.J. Lee, "Knowledge Distillation Based Continual Learning for PCB Part Detection," *Journal of Korea Multimedia Society*, Vol. 24, No. 7, pp. 868-879, 2021.
- [2] Z. Chen and B. Liu, "Lifelong Machine Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Vol. 12, No. 3, pp. 1-207, 2018.
- [3] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network Geoffrey," *arXiv Preprint*, arXiv:1503.02531, 2015.
- [4] C. Peng, K. Zhao, S. Maksoud, M. Li, and B. C. Lovell, "SID: Incremental Learning for Anchor-free Object Detection via Selective and Inter-related Distillation," *Computer Vision and Image Understanding*, Vol. 210, 103229, pp. 1-8, 2021.
- [5] H. Yoon and J. Lee, "PCB Component Classification Algorithm Based on YOLO Network for PCB Inspection," *Journal of Korea Multimedia Society*, Vol. 24, No. 8, pp. 988-999, 2021.
- [6] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as Points," *arXiv Preprint*, arXiv:1904.07850, 2019.
- [7] M. Masana, X. Liu, B. Twardowski, M. Menta, A.D. Bagdanov, and J. van de Weijer, "Class-incremental Learning: Survey and Performance Evaluation on Image Classification," *arXiv Preprint*, arXiv:2010.15277, pp. 1-26, 2020.
- [8] Z. Li and D. Hoiem, "Learning without Forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 12, pp. 2935-2947, 2017.
- [9] U. Michieli and P. Zanuttigh, "Continual Semantic Segmentation via Repulsion-Attraction of Sparse and Disentangled Latent Representations," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1114-1124, 2021.
- [10] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental Learning of Object Detectors without Catastrophic Forgetting," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3420-3429, 2017.
- [11] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: A Simple and Strong Anchor-free Object Detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 44, Issue 4, pp. 1922-1933, 2020.
- [12] U. Michieli and P. Zanuttigh, "Incremental Learning Techniques for Semantic Segmentation," *Proceedings of the IEEE/CVF*

International Conference on Computer Vision Workshops, pp. 1-8, 2019.

- [13] U. Michieli and P. Zanuttigh, "Knowledge Distillation for Incremental Learning in Semantic Segmentation," *Computer Vision and Image Understanding*, Vol. 205, 103167, pp.1-16, 2021.
- [14] GitHub - xingyizhou/CenterNet: Object detection, 3D detection, and pose estimation using center point detection, <https://github.com/xingyizhou/CenterNet> (accessed Oct. 24, 2021).



강 수 명

2011년 계명대학교 게임모바일콘텐츠학과 졸업 (학사)
 2013년 계명대학교 미디어아트학과 게임모바일전공 졸업 (석사)
 2014년 ~ 2017년 (주)지오씨엔아이 공간정보기술연구소 선임연구원

2022년 계명대학교 컴퓨터공학과 졸업(박사)
 관심분야: 영상처리, 패턴인식, 컴퓨터비전, 딥 러닝



정 대 원

2002년 중앙대학교 수학과 졸업 (학사)
 2004년 중앙대학교 수학과 졸업 (석사)
 2010년 University of New Mexico 졸업(박사)

2010년 ~ 2012년 University of New Mexico 겸임강사
 2012년 ~ 2014년 인하대학교 응용수학연구소 박사 후 연구원
 2014년 ~ 현재 계명대학교 기초과학부 조교수
 관심분야: 조화해석학, 편미분방정식, 역문제, 인공지능



이 준 재

1986년 경북대학교 전자공학과 졸업(학사)
 1990년 경북대학교 전자공학과 졸업(석사)
 1994년 경북대학교 전자공학과 졸업(박사)

1995년 ~ 2007년 동서대학교 컴퓨터 정보공학부 부교수
 1998년 ~ 1999년 Georgia Institute of Technology 방문교수
 2000년 ~ 2001년 (주)파미 연구소장
 2007년 ~ 현재 계명대학교 컴퓨터공학부 교수
 관심분야: 영상처리, 3D 컴퓨터 비전, 딥 러닝, 게임