논문 22-47-08-12 The Journal of Korean Institute of Communications and Information Sciences '22-08 Vol.47 No.08
https://doi.org/10.7840/kics.2022.47.8.1144

Light-CAM: 임베디드 단말의 Weakly Supervised Object Localization을 위한 경량화 모델

김 용 호•, 김 지 하*, 박 현 희°

Light-CAM: A Lightweight Model for Weakly Supervised Object Localization of Embedded Devices

Yongho Kim•, Jiha Kim*, Hyunhee Park°

요 약

하드웨어 성능의 지속적 발전으로 고성능 PC가 아니더라도 모바일 및 임베디드 단말에서 딥러닝을 활용하는 애플리케이션들이 많이 등장하였다. 하지만 하드웨어 성능의 발전함에도 모바일 및 임베디드 단말에서 파라미터 수가 많은 무거운 모델을 사용하기에는 한계가 존재한다. 본 논문에서는 임베디드 단말에서 Class Activation Map을 활용해 Weakly Supervised Object Localization을 수행할 때 Class Activation Map에 사용할 네트워크를 임베디드 단말에서 사용할 수 있도록 설계한다. 제안하는 모델인 Light-CAM은 Class Activation Map 네트워크의 계층을 얕게 설계하여 모델의 파라미터 수를 줄이면서 localization 성능 감소를 최소화한다. 실험을 통해 Bird and Dog 데이터 세트에서 여러 모델과 여러 CAM 기법을 사용하여 localization accuracy를 비교했을 때 Light-CAM+BR-AvgCAM 조합이 세 번째로 높은 성능을 보였다. 가장 높은 성능을 보인 VGG16+BR-AvgCAM 조합과 비교하면 localization accuracy는 5.9% 낮지만 파라미터 수는 9.34배가 줄어든 것을 확인할 수 있다. 이는 임베디드 단말과 같이 최소한의 컴퓨팅으로 연산을 하는 단말을 위해 적합한 모델임을 알 수 있다.

Key Words : Weakly Supervised Object Localization, Class Activation Map, Object Localization, Model Compression, explainable AI

ABSTRACT

With the continuous development of hardware performance, many applications that utilize deep learning in mobile and embedded devices have emerged, even if they are not high-performance PCs. However, despite advances in hardware performance, there are limitations in using heavy models with a large number of parameters on mobile and embedded devices. In this paper, we design a network to be used for embedded devices when performing weak supervised object localization using a class activation map. The proposed model, Light-CAM, designs the layer of the Class Activation Map network shallowly, reducing the number of parameters of the model and minimizing the reduction in localization performance. Experiments show that the

※ 본 연구는 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2021-0-00990, 설명가능한 인공지능 기반 무선랜 네트워크 시스템 고도화 핵심 기술 연구)

• First Author : Myoungji University Department of Information and Communication Engineering, yhkim98@mju.ac.kr, 학생회원
° Corresponding Author : Myoungji University Department of Information and Communication Engineering, hhpark@mju.ac.kr, 종신회원
* Myoungji University Department of Information and Communication Engineering, yaki5896@mju.ac.kr, 학생회원
논문번호 : 202203-029-C-RE, Received February 28, 2022; Revised May 27, 2022; Accepted June 9, 2022

Light-CAM+BR-AvgCAM combination showed the third-highest performance when comparing localization accuracy using multiple models and multiple CAM methods on Bird and Dog datasets. Compared to the VGG16+BR-AvgCAM combination with the highest performance, the localization accuracy is 5.9% lower, but the number of parameters is reduced by 9.34 times. It can be seen that the proposed Light-CAM model is a suitable model for small embedded devices with minimal computing.

Ⅰ. 서 론

Weakly Supervised Object Localization(WSOL)은 기존의 레이블링보다 약한 레이블링으로 학습을 시키는 문제이다. 예를 들면 하나의 이미지가 주어진 상황에서 Object Localization을 할 때, 이미지의 위치 정보를 제공하지 않고 찾으려는 객체 클래스만을 제공하여 해당 클래스의 객체 위치를 찾는 것이다. Class Activation Map(CAM)[1]은 WSOL 문제를 해결하기 위해 고안된 기본적인 기법이다. Object Localization이나 Object Detection을 학습하기 위해서는 객체의 이미지, 객체의 클래스 정보 및 이미지 내 객체의 위치 정보가 필요하다. 이미지 내 객체의 위치 정보를 제공하기 위해서는 이미지 내 객체를 바운딩 박스로 레이블링하는 작업을 거쳐야 한다. 여기서 바운딩 박스는 이미지 내 객체의 위치를 사각형으로 감싼 형태의 도형을 정의하고 꼭짓점의 좌표를 표현한 것이다.

이와 같은 바운딩 박스 레이블링은 인력에 의해 진행되기 때문에 모든 레이블링이 완벽하게 이루어지기에 어려움이 있고, 심지어 잘못된 레이블링으로 인해 모델의 성능이 감소하는 경우도 존재한다. 이러한 경우를 방지하기 위해 레이블링을 정교하게 하고자 하지만, 이때 많은 시간적 경제적 비용이 발생할 수 있다. 하지만 CAM을 활용하여 객체를 정확하게 지역화할 수 있다면, 바운딩 박스 레이블이 아닌 이미지 레이블만으로 모델을 학습하여 Object Localization과 Object Detection을 수행할 수 있다. 이로 인해 레이블링에 필요한 시간적 경제적 비용 또한 줄일 수 있다.

하드웨어의 성능이 발전함에 따라 고성능 PC뿐 아니라 모바일 및 임베디드 기기들에서도 딥러닝을 사용할 수 있게 되었다. 이에 따라 위와 같은 WSOL 문제를 모바일 및 임베디드 환경에서도 CAM을 활용하여 해결할 수 있다. 하지만 모바일 및 임베디드 단말들의 성능이 발전했음에도 불구하고 딥러닝 계층이 깊고 파라미터 수가 많은 모델을 사용하기에는 한계가 존재한다.

본 논문에서는 이와 같은 문제를 해결하기 위해 모바일 및 임베디드 환경을 위한 딥러닝 계층이 얕고 파라미터 수를 줄여 경량화된 CAM 모델인 Light-CAM을 제안한다.

CAM은 WSOL 문제 해결을 위해 사용될 뿐 아니라, 블랙박스 형태로 구성된 딥러닝 모델이 어떠한 이유로 클래스를 예측했는지에 대한 정보를 얻을 수 있다.

Ⅱ. 본 론

2.1 관련 연구

2.1.1 Class Activation Map

CAM은 기존 Convolutional Neural Network (CNN)[2] 구조에서 가장 마지막에 완전 연결 계층(Fully Connected Layer) 대신에 Global Average Pooling(GAP) 계층으로 바꾸어 이미지 레이블 외에 추가적인 정보 없이 Object Localization을 수행할 수 있다. 완전 연결 계층을 GAP 계층으로 변경하는 이유는 CNN에서 다양한 계층의 convolution unit들이 object detector의 역할을 한다[1]. 하지만 이러한 정보들이 완전 연결 계층을 지나면 object를 localize 하는 특징을 잃게 된다. 이와 같은 이유로 완전 연결 계층 대신 GAP 계층을 사용함으로써 object를 localize 하는 특징을 유지하여 Object Localization이 가능하게 된다.

다음은 CAM을 계산하는 방법이다.

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k w_k^c f_k(x,y) \quad (1)$$

식 (1)에서 k는 GAP를 통과하기 전 convolution 계층의 채널 수, w^{c_k}는 클래스별 가중치, $\sum_{x,y} f_k(x,y)$는 k번째 feature map의 GAP를 나타낸다. 위 수식을 계산하면 S_c라는 클래스 c에 대한 softmax 입력값을 구할 수 있다.

$$M_c(x,y) = \sum_k w^{c_k} f_k(x,y) \quad (2)$$

$$S_c = \sum_{(x,y)} M_c(x,y) \tag{3}$$

식 (1)에서 클래스 c에 대한 activation map을 식 (2)의 $M_c(x,y)$로 정의하면 식 (3)과 같이 정리를 할 수 있다. 결과적으로 $M_c(x,y)$는 클래스 c에 (x,y)지점이 미치는 영향을 나타내게 되면서 Class Activation Map을 생성하게 된다.

CAM은 단순히 CNN의 마지막 계층을 바꾸고 단순한 연산을 수행함으로 Object Localization 문제를 수행할 수 있어 WSOL 문제에서 주로 사용한다. 또한, 추가적인 바운딩 박스 레이블 없이 객체를 탐지할 수 있으므로 레이블링에 필요한 시간적 경제적 비용 또한 아낄 수 있다. 하지만 CAM의 고질적인 문제로 탐지해야 할 객체 주변, 즉 배경의 노이즈로 인해 localization이 부정확하게 될 수 있다는 문제가 존재한다.

2.1.2 Background Remove-Average Class Activation Map

Background Remove-Average Class Activation Map(BR-AvgCAM)[3]은 기존 CAM에서 배경으로 인한 노이즈를 제거하여 localization 성능을 개선한 기법이다. BR-AvgCAM은 모델을 학습할 때 분류하고자 하는 클래스 외에 배경 클래스를 추가해서 학습한다. 그림 1은 BR-AvgCAM의 생성과정을 그림으로 나타낸 것이다.

우선, 학습된 모델로 분류할 클래스의 수만큼 CAM을 생성한다. 그 후 배경 클래스의 CAM을 제외한 나머지 클래스들의 CAM 평균값을 계산하여 AvgCAM을 구한다. 마지막으로 AvgCAM에서 배경 CAM을 빼면 배경 노이즈가 제거된 BR-AvgCAM이 생성된다.

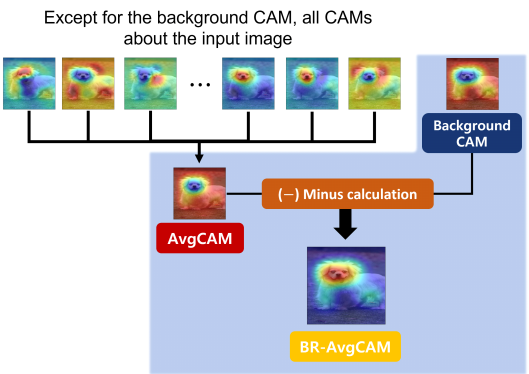

그림 1. BR-AvgCAM 생성과정
Fig. 1. The process of generating BR-AvgCAM.

2.2 제안하는 모델

Light-CAM은 계산 능력이 비교적 낮은 스마트폰이나 임베디드 단말에서 CAM을 사용해 Object Localization을 하기 위해 제안하는 경량화 된 CAM 모델이다. 기존 CAM 모델과 비교했을 때 파라미터 수와 모델의 용량을 낮추어 계산 능력이 낮은 단말에서도 CAM을 사용하여 Object Localization을 수행할 수 있다. 제안하는 모델의 구조는 그림 2와 같다. 입력

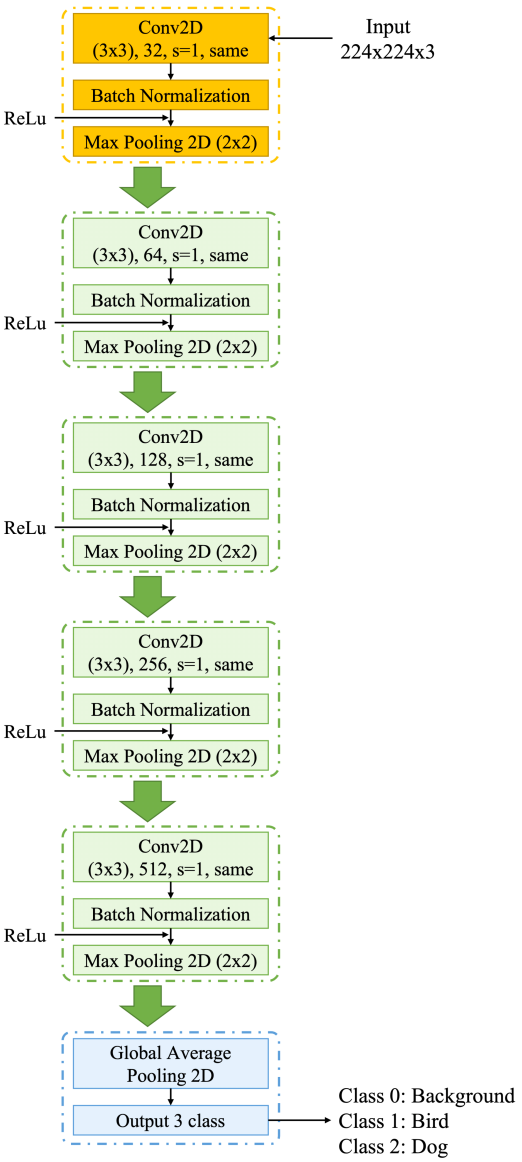

그림 2. Light-CAM 모델 구조
Fig. 2. Light-CAM model structure.

으로는 224x224x3 크기로 resizing한 이미지를 받는다. 이미지를 받은 후 3x3 크기의 convolution 계층의 입력으로 들어간다. 2x2 크기의 max pooling으로 feature map의 크기가 줄어들기 때문에 padding을 주어 convolution 연산 이후에는 feature map의 크기가 변하지 않도록 했다. Convolution 연산 후에는 batch normalization, ReLu, max pooling을 차례대로 통과한다. 이와 같은 일련의 과정을 블록이라 부른다. 이러한 블록을 총 5개만 사용하여 얕은 계층의 적은 파라미터 수를 가지는 Light-CAM의 구조를 구성했다.

CAM은 특징 추출이 중요하므로 계층이 깊어질수록 추상적인 특징을 많이 추출할 수 있어야 하며, 이를 위해 convolution 계층이 깊어 질수록 더 많은 필터 개수를 갖도록 설계하였다. 또한, 계층이 깊어질수록 다음 계층의 입력으로 들어가는 값의 크기가 작아지기에 깊은 계층에서 필터 개수를 늘리면 연산량이 줄어든다.

모델의 계층이 얕은 만큼 과적합에 빠질 위험이 있으므로, 이를 방지하기 위해 각 convolution 계층마다 batch normalization을 추가했다. 학습 시 수렴이 빨리 되는 효과 또한 존재한다[4].

Light-CAM 모델이 기존 CAM에 사용하는 모델들과 비교하면 모델의 계층이 얕고 파라미터 수가 적으면서 단순한 구조를 가지기에 경량화된 CAM 모델로 볼 수 있다.

Light-CAM의 구조는 Limited Discriminator GAN[5]의 판별자 모델을 참조하여 설계했다. 이 모델의 구조를 참조한 이유는 CAM과 같은 설명가능한 인공지능을 위해 설계된 모델이기에 참조했다.

Ⅲ. 실 험

3.1 데이터 세트

본 논문에서는 실험을 위한 데이터 세트로 Stanford Dogs[6], CUB-200-2011[7], Dog and Cat Detection[8], Landscape Pictures[9] 데이터 세트를 사용했다.

Stanford Dogs 데이터 세트는 120개의 강아지 품종을 분류하기 위한 데이터 세트이다. 본 실험에서는 120개의 클래스 중 10개의 클래스만 추출한 Stanford Dogs 10, 전체 클래스를 다 쓰는 Stanford Dogs, 120개의 클래스로 나누어진 강아지 데이터를 하나의 클래스로 묶어서 사용하는 Dog 데이터 세트로 나누어 사용했다.

CUB-200-2011 데이터 세트는 200개의 새 품종을 분류하는 데이터 세트이다. 이 데이터 세트는 200개의 클래스로 나누어진 데이터를 하나의 클래스로 묶어서 사용했다.

Dog and Cat Detection 데이터 세트는 강아지와 고양이를 분류하는 데이터 세트이다.

위 세 데이터 세트를 사용한 이유는 이미지 내 객체에 대한 바운딩 박스 레이블이 제공되어 CAM의 localization 성능 평가를 할 수 있어 사용했다. 또한, 다양한 데이터 세트에서 모델의 성능을 비교하기 위해 여러 가지 데이터 세트를 사용했다.

Landscape Pictures 데이터 세트는 kaggle에서 제공하는 다양한 풍경과 배경들을 모아놓은 데이터 세트이다. Localization 하고자 하는 클래스들과 배경 클래스를 같이 학습하여 CAM의 배경 노이즈를 제거하는 기법인 BR-AvgCAM을 위해 사용했다.

정리하면 학습에는 Stanford Dogs 10 데이터 세트, Stanford Dogs 데이터 세트, Stanford Dogs의 전체 클래스를 dog 클래스로 합치고 CUB-200-2011 데이터 세트 전체 클래스를 bird 클래스로 합쳐 강아지와 새를 분류하는 Bird and Dog 데이터 세트, 고양이와 강아지를 분류하는 Cat and Dog 데이터 세트를 사용한다. Landscape Pictures 데이터 세트는 각 데이터 세트에 background 클래스로 추가되어 같이 학습한다. 위와 같은 데이터 세트들로 여러 학습모델과 CAM 기법을 활용해 localization 성능을 비교하는 실험을 진행했다. 실험과 모델 학습은 표 1과 같은 환경에서 진행했다.

표 1. 실험 환경 & 모델 학습 파라미터
Table 1. The experimental environment & Model training parameters.

OS	Windows
Python	3.8.8
GPU count	1
GPU type	GeForce GTX 1660 Super
Frame work	Tensorflow 2.3.0
Optimizer	Adam
Learning rate	0.001
Epochs	50

3.2 성능 평가

각 모델과 CAM 기법들 조합의 localization 성능 비교를 위한 지표로 localization accuracy를 사용했다. Localization accuracy는 다음과 같은 식으로 표현할 수 있다.

$$Localization\ Accuracy(\%) = \frac{\sum_{i=1}^{T} C_i}{T} \times 100 \quad (4)$$

식 (4)에서 T는 전체 테스트 데이터의 수, $\sum_{i=1}^{T} C_i$는 Localization을 성공한 테스트 데이터의 수이다.

Localization accuracy 계산을 위해 localization 성공 기준을 식 (5)와 같이 정의했다.

$$C_i = \begin{cases} 1 & where\ A_i : \frac{O_i}{U_i} \geq 0.5 \\ & \text{and } B_i : i^{th} predict = i^{th} true \\ 0 & otherwise \end{cases} \quad (5)$$

C_i는 조건 A_i와 조건 B_i를 만족할 때 1을 한 조건이라도 만족하지 않으면 0이 된다.

조건 A_i에서 $\frac{O_i}{U_i}$는 Intersection over Union(IoU)으로 O_i는 테스트 데이터 세트에 대한 Ground Truth 바운딩 박스와 CAM으로 생성한 바운딩 박스의 overlap 영역 넓이 U_i는 두 바운딩 박스의 union 영역 넓이를 의미한다. IoU가 0.5 이상이면 조건 A_i를 만족한다.

조건 B_i에서 $i^{th} predict$는 모델이 i번째 테스트 데이터에 대해 예측한 클래스의 값이고 $i^{th} true$는 i번째 테스트 데이터의 정답 클래스 값이다. 따라서 두 값이 일치하면 조건 B_i를 만족한다.

3.2.1 바운딩 박스 생성 방법

바운딩 박스는 CAM 최댓값의 20%가 넘는 부분들이 먼저 선택되고, 이 부분들을 가장 많이 포함할 수 있는 바운딩 박스를 선택하여 생성한다[1].

3.2.2 Intersection over Union(IoU)

Intersection over Union(IoU)는 두 바운딩 박스가 얼마나 일치하는지 측정하기 위한 지표이다. 그림 3에서 보이듯이 두 바운딩 박스의 overlap 영역 넓이에서 union 영역 넓이를 나누어서 계산한다.

3.3 실험 방법

실험은 앞서 소개한 데이터 세트들을 사용하여 각 데이터 세트별 클래스를 분류하는 모델을 학습한다. 분류 모델 학습에는 VGG16[10], MobileNet[11], ResNet50[12], InceptionV3[13] 그리고 제안하는 모델인 Light-CAM을 사용하여 학습했다. 학습한 모델들을 CAM과 BR-AvgCAM을 활용하여 각 데이터 세트별 클래스를 localization 한 후 localization accuracy를 각각 측정하여 성능을 비교했다.

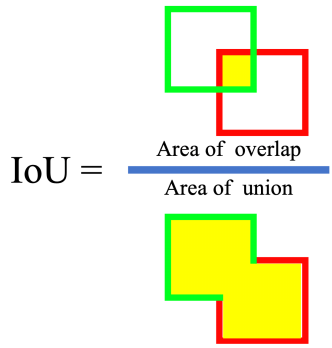

그림 3. IoU 예시.
Fig. 3. Example of IoU.

3.4 실험 결과

3.4.1 데이터 세트별 Light-CAM의 localization accuracy 결과

첫 번째 실험은 제안하는 모델인 Light-CAM으로 여러 가지 데이터 세트에 적용하여 CAM과 BR-AvgCAM을 활용해 Object Localization을 수행했을 때 localization accuracy를 비교했다. 표 2를 보면 Stanford Dog 10, Stanford Dog, Bird and Dog 데이터 세트에서 BR-AvgCAM을 사용할 때 localization accuracy가 기존 CAM을 사용할 때보다 개선된 것을 확인할 수 있다. 하지만, Cat and Dog 데이터 세트에서는 오히려 성능이 감소하였다. Cat and Dog 데이터 세트의 레이블링을 확인한 결과 그림 4의 (a)와 같이 바운딩 박스 레이블이 객체 전체를 감싸지 않아 잘못된 레이블링이 된 것을 확인할 수 있었다.

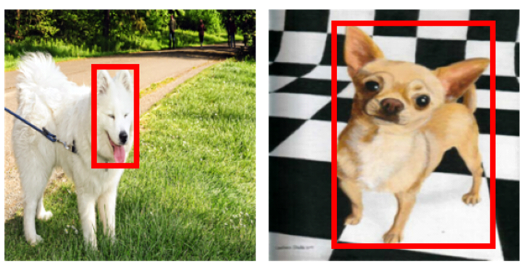

그림 4. 바운딩 박스 레이블링 된 데이터 예시
Fig. 4. Example of data labeled with a bounding box.

표 2. 데이터 세트별 Light-CAM의 localization accuracy 결과
Table 2. The results of the localization accuracy of Light-CAM by dataset.

Method	Dataset	Localization Accuracy(%)	Number of train data
CAM	Cat and Dog	8.10	2,810
BR-AvgCAM		7.43	
CAM	Stanford Dog 10	13.52	1,248
BR-AvgCAM		15.03	
CAM	Stanford Dog	20.60	12,410
BR-AvgCAM		23.60	
CAM	Bird and Dog	38.58	21,759
BR-AvgCAM		40.02	

표 3. Bird and Dog 데이터 세트에서 각 모델과 CAM 기법별 localization accuracy 결과
Table 3. Localization accuracy results for each model and CAM method in the Bird and Dog dataset

Method	Model	Localization Accuracy(%)	parameter
CAM	ResNet50	34.61	23,593,859
BR-AvgCAM	ResNet50	32.94	
CAM	InceptionV3	40.90	21,808,931
BR-AvgCAM	InceptionV3	32.94	
CAM	VGG16	36.79	14,716,227
BR-AvgCAM	VGG16	45.93	
CAM	MobileNet	39.48	3,231,939
BR-AvgCAM	MobileNet	38.80	
CAM	Light-CAM(ours)	38.58	1,574,083
BR-AvgCAM	Light-CAM(ours)	40.02	

반면에 그림 4의 (b)는 Stanford Dog 데이터 세트의 바운딩 박스 레이블이다. 객체 전체를 잘 감싸서 레이블링이 잘 된 것을 확인할 수 있다. 이와 같은 잘못된 레이블링으로 인해 Cat and Dog 데이터 세트에서는 제대로 된 성능 평가가 되지 않은 것으로 보인다.

Cat and Dog 외 데이터 세트에서의 localization accuracy를 볼 때 Light-CAM은 학습 데이터가 많을수록 성능이 높아지는 것을 확인할 수 있다. 이는 Light-CAM의 얕은 모델 구조와 적은 파라미터 수의 영향으로 보인다. 모델의 깊이가 얕기 때문에 이미지에서 특징 추출이 부족할 수 있고 파라미터 수가 적기 때문에 이미지 특징을 충분히 학습하기 어려울 수 있다. 이러한 이유로 Light-CAM은 많은 데이터에서 다양한 특징을 뽑아냄으로 더 좋은 성능을 달성할 수 있는 것으로 보인다.

3.4.2 Bird and Dog 데이터 세트에서 모델별 localization accuracy 결과

두 번째 실험은 Bird and Dog 데이터 세트에서 여러 가지 모델로 CAM과 BR-AvgCAM을 사용하여 Object Localization을 수행했을 때 localization accuracy를 비교했다. 표 3을 보면 BR-AvgCAM을 사용했다 해서 모든 모델의 localization accuracy가 개선되지 않았지만, 가장 높은 성능은 BR-AvgCAM으로 성능이 개선된 VGG16이 45.93%의 localization accuracy를 보였다. Light-CAM 또한 BR-AvgCAM으로 인해 성능이 개선되어 세 번째로 높은 40.02%의 localization accuracy를 보였다.

Light-CAM이 가장 높은 localization accuracy를 달성하지는 못했지만 다른 모델들의 파라미터 수와 비교해 볼 때 파라미터 수에 비해서 높은 성능을 보인다는 것을 알 수 있다.

Light-CAM+BR-AvgCAM과 가장 높은 성능을 달성한 VGG16+BR-AvgCAM을 비교하면 localization accuracy는 5.9% 낮지만 파라미터 수는 9.34배가 줄어들었다. 두 번째로 성능이 높은 InceptionV3+CAM과 비교해보면 localization accuracy는 0.88%의 크지 않은 성능 저하가 있었지만, 파라미터 수는 13.85배나 줄었다. 마지막으로 Light-CAM과 비슷한 경량화 모델인 MobileNet+CAM과 비교했을 때 파라미터 수가

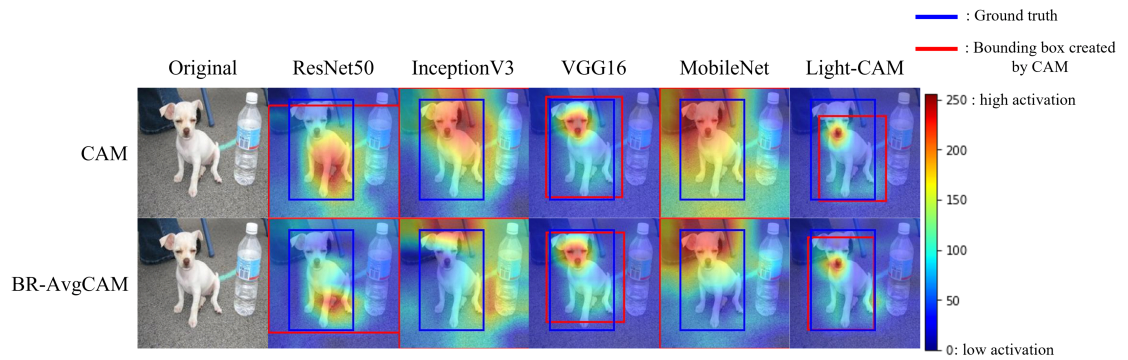

그림 5. Bird and Dog 데이터 세트에서 여러 모델로 학습된 CAM과 BR-AvgCAM으로 생성한 바운딩 박스
Fig. 5. Bounding boxes created by CAM and BR-AvgCAM learned from multiple models in Bird and Dog datasets.

2.05배 낮으면서도 localization accuracy를 0.54% 더 높아진 것을 확인할 수 있다.

그림 5는 Bird and Dog 데이터 세트의 한 이미지를 여러 모델의 입력으로 넣어 CAM과 BR-AvgCAM을 사용하여 생성한 바운딩 박스 결과이다. Light-CAM으로 생성한 바운딩 박스가 성능이 가장 높은 VGG16으로 생성한 바운딩 박스와 유사하게 생성되는 것을 확인할 수 있다.

위와 같은 결과로 Light-CAM의 파라미터 수를 줄여 모델 경량화를 하였음에도 준수한 성능의 localization accuracy를 얻었음을 알 수 있다.

3.4.3 각 모델들과 CAM 기법들 조합의 information density 비교

Information density는 학습 파라미터 수 대비 accuracy의 비율로 최근 딥러닝 모델 설계의 효율성을 측정하는 지표로 사용한다[14]. Accuracy density라고도 부른다. Information density는 식 (6)처럼 계산할 수 있다.

$$D(N) = \frac{a(N)}{p(N)} \tag{6}$$

식 (6)에서 $D(N)$은 information density이고 $a(N)$은 모델의 Top-1 accuracy, $p(N)$은 모델의 학습 파라미터 수이다. Information density가 클수록 모델의 효율이 더 높다는 뜻이다.

본 논문에서는 localization accuracy와 파라미터 수의 information density를 비교하기 위해 $a(N)$에 Top-1 accuracy 대신 식 (4)로 계산한 Top-1 localization accuracy를 대입해 계산했다.

표 3 결과를 기반으로 제안하는 모델과 다른 모델들의 CAM 기법별 information density를 비교했다. 그림 6을 보면 본 논문에서 제안하는 모델인 Light-CAM이 CAM과 BR-AvgCAM을 사용했을 때 각각 24.5095, 25.4243의 가장 높은 information density를 보임으로 이 실험에서 Light-CAM의 모델 구조가 가장 효율적인 것을 알 수 있다.

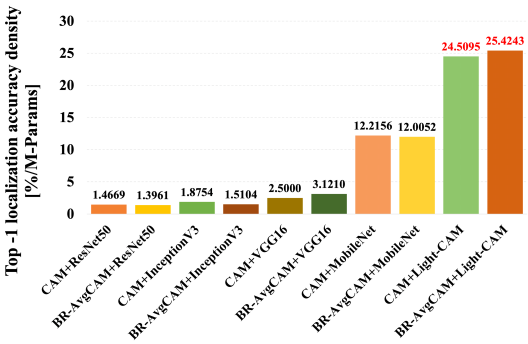

그림 6. 각 모델들의 CAM 기법별 information density
Fig. 6. Information density by CAM method for each model.

Ⅳ. 결 론

본 논문에서는 고성능 컴퓨터에 비해 계산 능력이 낮은 모바일 및 임베디드 단말에서 CAM을 통해 Object Localization을 수행하기 위한 모델 경량화 연구를 진행했다. 실험 결과를 통해 모델을 얕게 쌓아 파라미터 수를 줄여도 localization 성능이 크게 떨어지지 않는 것을 확인할 수 있다. 이로 인해, 모바일 및 임베디드 단말에서도 CAM을 통한 Object Localization을 수행할 수 있다.

향후 연구에서는 다양한 벤치마크 데이터 세트로

성능 향상을 검증하고 Light-CAM의 성능 또한 더 개선할 방법을 연구하고자 한다.

References

[1] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. CVPR*, pp. 2921-2929, 2016. (https://doi.org/10.1109/CVPR.2016.319)

[2] Y. LeCun, et al., "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998. (https://doi.org/10.1109/5.726791)

[3] Y. Kim and H. Park, "Background remove-average class activation map for improving bounding box IoU," in *Proc. Symp. KICS*, 350-351, Nov. 2021.

[4] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Int. Conf. Mach. Learn.*, PMLR, pp. 448-456, 2015.

[5] J. Kim and H. Park, "Limited discriminator GAN using explainable AI model for overfitting problem," *ICT Express*, 2022. (https://doi.org/10.1016/j.icte.2021.12.014)

[6] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization," in *Proc. CVPR Wkshp. Fine-grained Visual Categorization (FGVC)*, vol. 2, no. 1, 2011.

[7] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "*The Caltech-UCSD Birds-200-2011 dataset*," Computation & Neural Systems Technical Report, CNS-TR-2011-001.

[8] Kaggle[Website], (2022, Feb. 14), https://www.kaggle.com/andrewmvd/dog-and-cat-detection.

[9] Kaggle[Website], (2022, Feb. 14), https://www.kaggle.com/arnaud58/landscape-pictures.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv: 1409.1556*, 2014. (https://doi.org/10.48550/arXiv.1409.1556)

[11] A. G. Howard, et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv: 1704.04861*, 2017. (https://doi.org/10.48550/arXiv.1704.04861)

[12] K. He, et al., "Deep residual learning for image recognition," in *Proc. IEEE Conf. CVPR*, pp. 770-778, 2016. (https://doi.org/10.1109/CVPR.2016.90)

[13] C. Szegedy, et al., "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. CVPR*, pp. 2818-2826, 2016. (https://doi.org/10.1109/CVPR.2016.308)

[14] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *arXiv preprint arXiv: 1605.07678*, 2016. (https://doi.org/10.48550/arXiv.1605.07678)

김 용 호 (Yongho Kim)

2021년 : 한국성서대학교 컴퓨터소프트웨어학과 졸업

2021년~현재 : 명지대학교 정보통신공학과 석박사통합과정

<관심분야> AI/ML 모델링, 지능형 시스템 개발, 컴퓨터 비전, 연합학습

[ORCID:0000-0001-7099-4336]

김 지 하 (Jiha Kim)

2020년 : 한국성서대학교 컴퓨터소프트웨어학과 졸업

2020년~현재 : 명지대학교 정보통신공학과 석박사통합과정

<관심분야> 딥러닝, GAN, Wireless LAN, WiFi 7

[ORCID:0000-0002-6711-6812]

박 현 희 (Hyunhee Park)

2011년 : 고려대학교 전자컴퓨터공학과 공학박사

2011년~2012년 : 고려대학교 정보기술사업단 연구교수

2012년~2014년 : 프랑스 INRIA Research Center Postdoctoral researcher

2014년~2017년 : LG전자 차세대표준연구소 선임연구원

2017년~2020년 : 한국성서대학교 컴퓨터소프트웨어학과 조교수

2020년~현재 : 명지대학교 정보통신공학과 부교수

<관심분야> 무선통신 표준화, 통신시스템, 데이터 분석 및 알고리즘

[ORCID:0000-0003-3810-7367]