ETRI

2018
Electronics and
Telecommunications
Trends

Deep Neural Network &
Object Recognition 특집



# 딥러닝 기반 객체 인식 기술 동향

## Trends on Object Detection Techniques Based on Deep Learning

**이진수** (J.S. Lee, jinsulee@ust.ac.kr) 지식이러닝연구그룹/UST 석사과정
**이상광** (S.K. Lee, sklee@etri.re.kr) 지식이러닝연구그룹 책임연구원
**김대욱** (D.W. Kim, dooroomie@etri.re.kr) 지식이러닝연구그룹 연구원
**홍승진** (S.J. Hong, hsj9649@gmail.com) 홍익대학교 게임학부 석사과정
**양성일** (S.I. Yang, siyang@etri.re.kr) 지식이러닝연구그룹 책임연구원/PL

Object detection is a challenging field in the visual understanding research area, detecting objects in visual scenes, and the location of such objects. It has recently been applied in various fields such as autonomous driving, image surveillance, and face recognition. In traditional methods of object detection, handcrafted features have been designed for overcoming various visual environments; however, they have a trade-off issue between accuracy and computational efficiency. Deep learning is a revolutionary paradigm in the machine-learning field. In addition, because deep-learning-based methods, particularly convolutional neural networks (CNNs), have outperformed conventional methods in terms of object detection, they have been studied in recent years. In this article, we provide a brief descriptive summary of several recent deep-learning methods for object detection and deep learning architectures. We also compare the performance of these methods and present a research guide of the object detection field.

* DOI: 10.22648/ETRI.2018.J.330403

*본 연구는 2018년도 과학기술정보통신부의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임[R0118-16-1005, 디지털콘텐츠 In-House R&D].

## Ⅰ. 서론

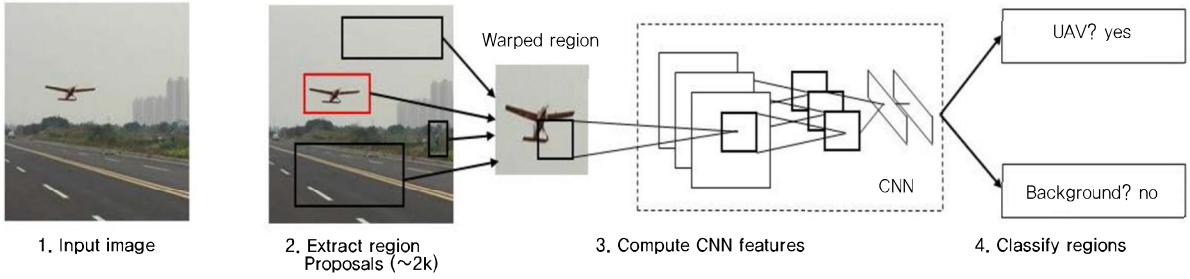
사람은 해야 할 일을 컴퓨터를 이용하여 처리함으로써 많은 편의를 얻고자 해왔다. 특히 객체 인식은 사람이 가장 많은 정보를 받아들이는 시각 정보를 컴퓨터가 대신하여 분석하고 해석할 수 있도록 하는 연구 분야이다. 이는 영상 감시, 얼굴 인식, 로봇 제어, IoT, 자율 주행, 제조업, 보안 등에 활용됨으로써 산업 전반에서 빠질 수 없는 핵심기술로 사용되고 있다. 이러한 중요성을 인지하여 객체 인식 분야의 연구자들은 PASCAL[1], ImageNet[2], 그리고 최근 MS COCO[3]에 이르기까지 객체 인식 관련 대회를 개최하여 이 분야의 발전을 도모하고 있다.

과거의 객체 인식 연구는 SIFT(Scale Invariant Feature Transform)[4], SURF(Speeded-Up Robust Features)[5], Haar[6], HOG(Histogram of Oriented Gradients)[7] 등과 같이 객체가 가지는 특징을 설계하고 검출함으로써 객체를 찾아내는 방식으로 진행되었다. 예를 들면 책의 경우, 영상에서 사다리꼴 형태로 나타나고 꼭짓점에서는 각이 생기며 두께가 어느 정도 있다는 특징이 있으므로, 이러한 정보를 어떻게 설계 및 검출할 것인가에 다양한 방법들이 연구된 것이다. 그 후, DPM(Deformable Part-based Model)[8]에서는 물체를 여러 부분으로 나누어 특징 정보를 구성하고, 각 부분의 유동적인 구조를 SVM(Support Vector Machine)과 같은 기계학습 방법으로 연결함으로써 객체 인식 성능을 높였다.

하지만, 합성곱 신경망(CNN: Convolutional Neural Network)이 ImageNet 2012 대회에서 기존 방식의 성능을 압도적으로 뛰어넘는 결과를 보여주면서[9], 딥러닝을 이용한 객체 인식 방법이 학계의 주목을 받고 주류가 되었다. CNN은 LeCun 교수의 필기체 숫자 인식[10]에서 처음 등장하였는데, 기존의 신경망에서는 픽셀 주위의 지역적인 정보를 표현하지 못했으나 합성곱 연산을 도입함으로써 이를 극복하였다. 그 후, CNN의 인식률을 향상시키기 위해 네트워크를 더 깊게 구성하는 방식으로 연구가 진행되었으며, ZFNet[11], VGG[12], ResNet[13], GoogLeNet[14], DenseNet[15] 등이 등장하였다. 본고에서는 이러한 네트워크의 발전 동향을 다룰 것이다.

한편, 영상에서 객체가 무엇인지 인식하는 문제는 CNN을 통해 어느 정도 성공을 거뒀으나, 영상에서 객체가 어디에 존재하는지를 찾아내는 것은 또 다른 문제였다. 따라서 근래에는 객체의 위치를 검출하는 방법에 대한 연구가 등장하였다.

R-CNN(Region-based Convolutional Neural Networks)[16]은 이 문제를 딥러닝 회귀(Regression) 방법으로 해결한 초기 연구다. R-CNN의 단점이었던 느린 검출 속도를 보완하기 위하여 Fast R-CNN[17]이 개발되었으나, 여전히 객체의 후보 영역을 찾는 데에는 딥러닝을 이용할 수 없다는 단점이 있었다. Faster R-CNN[18]에서는 이를 해결함으로써 검출 속도를 향상시키는 것뿐만 아니라, 딥러닝만을 이용하여 객체 인식을 구현할 수 있게 되었다. 이후에는 Faster R-CNN이 영상의 지역적 정보에 의존적이라는 단점을 보완하기 위해 R-FCN[19]가 등장하였다. 이러한 발전으로부터 객체 인식 속도가 크게 개선되었으나 실시간에 가까운 처리 속도를 필요로 하는 로봇, 자율주행 등의 응용분야에 적용하기에는 충분하지 못했다. YOLO(You Only Look Once)[20]는 이러한 속도 문제를 해결하기 위해, 객체 인식의 모든 과정을 하나의 딥러닝 네트워크로 구성하는 방법을 제안하였다. 최근에는 SSD(Single Shot MultiBox Detector)[21]와 같이 모바일에서도 동작 가능한 정도의 빠른 검출 속도를 보이는 방법들이 제안되고 있다. 또한, 객체의 영역 박스를 찾는 수준에서 한 단계 더 나아가, 객체의 픽셀 영역을 찾는 영상 분할(Image Segmentation) 분야에 대한 연구도 활발히 진




(그림 1) R-CNN의 객체 인식 시스템

[출처] Y. Yang et al., "Aerial Target Tracking Algorithm Based on Faster -CNN Combined with Frame Differencing," *Aerospace*, vol. 4, no. 2, 2017, pp. 32:1-32:17, doi:10.3390/aerospace4020032, CC BY 4.0.

행되고 있다.

본고에서는 앞서 언급했듯이, CNN 네트워크를 기반으로 하는 객체 인식 모델의 발전 동향뿐만 아니라, R-CNN에서 비롯되는 객체 인식 방법의 발전 동향 또한 다룰 것이다.

## II. 객체 인식 기술 동향

### 1. 객체 인식 방법

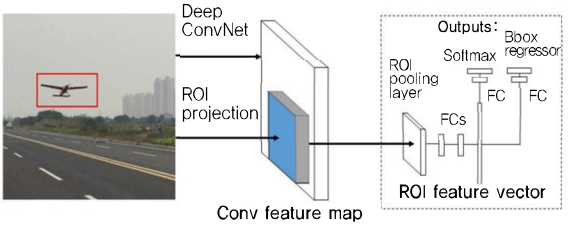
1절에서는 CNN을 기반으로 한 객체 인식 방법의 발전 동향에 대해 다루고자 한다. 객체 인식에 CNN을 적용하기 시작한 R-CNN부터 최근 높은 검출 속도를 보인 SSD까지 발전 과정과 각 방법의 장·단점을 소개한다.

#### 가. R-CNN

R-CNN[16]은 후보 영역(Region Proposal)을 생성하고 이를 기반으로 CNN을 학습시켜 영상 내 객체의 위치를 찾아낸다. R-CNN의 객체 인식 과정은 (그림 1)과 같이 크게 세 단계로 이루어진다. 첫째, 입력된 영상에서 선택적 탐색(Selective Search) 알고리즘[22]을 이용하여 후보 영역들을 생성한다. 둘째, 생성된 각 후보 영역들을 동일한 크기로 변환하고, CNN을 통해 특징(feature)을 추출한다. 셋째, 추출된 특징을 이용하여 후보 영역 내의 객체를 SVM(Support Vector Machine)을 이용하여 분류한다. 첫 번째 단계에서 생성된 후보 영역의 위치는 정확하지 않기 때문에, 최종적으로 회귀 학습을 통해 객체의 영역 박스 위치를 더 정확히 보정한다. R-CNN을 PASCAL VOC 2010 데이터셋에 적용한 결과, 약 53.7%의 mAP(Mean Average Precision)를 기록하여 기존의 객체 검출 방법들에 비해 큰 폭의 성능 향상을 보였다.

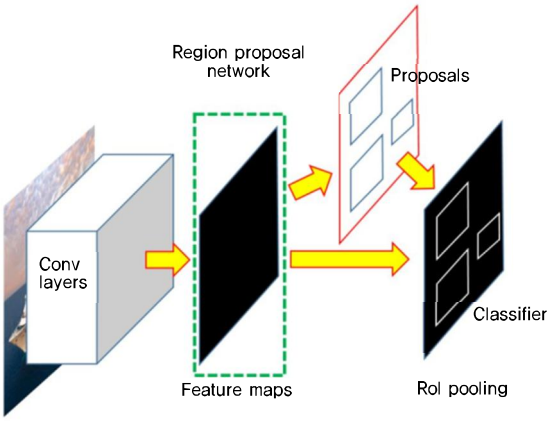
#### 나. Fast R-CNN

R-CNN은 CNN, SVM, 회귀의 학습 단계가 모두 분리되어 있고, 수천 개의 후보 영역에서 각각의 CNN을 학습해야 하므로 훈련하는 데에 많은 시간이 소요된다. 이를 보완하기 위하여 Fast R-CNN[17]은 (그림 2)와 같이 하나의 입력 영상에 대해 하나의 CNN을 학습한




(그림 2) Fast R-CNN의 객체 인식 시스템

[출처] Y. Yang et al., "Aerial Target Tracking Algorithm Based on Faster -CNN Combined with Frame Differencing," *Aerospace*, vol. 4, no. 2, 2017, pp. 32:1-32:17, doi:10.3390/aerospace4020032, CC BY 4.0.




(그림 3) Faster R-CNN의 객체 인식 시스템

[출처] K. Kim et al., "Probabilistic Ship Detection and Classification Using Deep Learning," *Appl. Sci.*, vol. 8, no. 6, 2018, 936:1-936:17, doi:10.3390/app8060936, CC BY 4.0.

다. 학습된 CNN을 통해 생성된 Feature map을 통합(Pooling)하여 특징을 추출한다. 또한, 분류기의 손실과 영역 박스 회귀의 손실을 합하여 동시에 훈련시킴으로써 훈련 단계를 단순화하였다. 분류기로는 기존의 SVM 대신 Softmax를 사용하였는데, Fast R-CNN에서는 Softmax를 적용하였을 때 성능이 더 우수하다는 것을 보여주었다. 이러한 개선을 통해 Fast R-CNN은 R-CNN보다 더 높은 mAP를 보이며 훈련에 소요되는 시간을 크게 감소시킬 수 있었음을 확인하였다.

### 다. Faster R-CNN

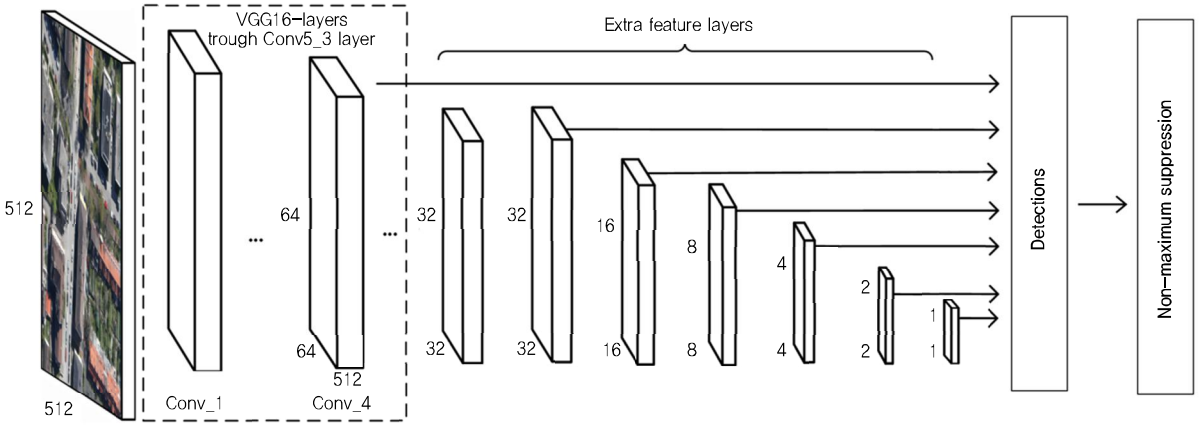
Fast R-CNN에서는 후보 영역을 생성하는 알고리즘이 CNN 외부에서 수행된다. 하지만, 이러한 구조는 속도 측면에서 비효율적이며, 알고리즘을 학습을 시킬 수 없다는 단점이 있다. Faster R-CNN[18]은 후보 영역을 생성하는 데에 선택적 탐색 알고리즘을 이용하지 않고, (그림 3)과 같이 Feature Map을 추출하는 CNN의 마지막 층(Layer)에 후보 영역을 생성하는 별도의 CNN인 영역 제안 네트워크(RPN: Region Proposal Network)을 적용하였다. RPN은 Fast R-CNN에서 CNN의 출력인 Feature Map을 입력으로 받아 객체의 위치를 추정하여 후보 영역을 출력하는 네트워크다. CNN에서 추출된 Feature Map을 RPN에서 추정된 후보 영역으로 잘라내어 객체를 인식한다. 이와 같이 Feature Map을 추출하는 CNN 과정과 후보 영역을 생성하는 과정을 일련의 네트워크로 구성함으로써, 같은 조건에서 Fast R-CNN보다 훈련 시간을 10배 정도 감소시키면서 mAP 또한 향상시켰다.

### 라. R-FCN

R-FCN[19]은 위치 정보를 포함하고 있는 Score Map을 이용하여 물체의 위치를 정확하고 효율적으로 찾아낸다. 이 Score Map은 CNN을 통해 추출된 Feature Map으로부터 얻어지며, 각 Score Map은 입력된 영상 내 특정 위치의 정보를 포함한다. 이를 이용하여 특정 위치마다 분류 결과를 얻어내고, 이 결과를 종합하여 최종적으로 특정 위치 내의 객체를 분류한다. 특정 위치가 찾고자 하는 객체를 포함할 경우 Score Map의 반응이 커지는 반면, 그렇지 않은 경우 그 반응이 작아진다. R-FCN에서의 Score Map은 위치 정보를 포함하고 있으면서 훈련이 요구되지 않는다는 장점이 있다. R-FCN에 PASCAL VOC 2007 데이터셋을 적용하여 실험한 결과, Faster R-CNN보다 훈련 시간이 3배 정도 빠르며, mAP가 약 0.2% 더 높은 것으로 나타났다.

### 마. YOLO

YOLO[20]는 객체 인식 문제를 하나의 회귀 문제로 접근하여 전체적인 구조를 간소화함으로써 훈련 및 검출 속도를 크게 향상시켰다. 입력된 영상은 CNN을 거쳐 텐서(Tensor) 형태로 출력된다. 이 텐서는 영상을 격자 형태로 나누어 각 구역을 표현하게 되며, 이를 통해 해당 구역의 객체를 인식한다. 이와 같이 YOLO는 관심 영역을 추출하기 위한 별도의 네트워크가 필요하지




(그림 4) SSD의 객체 인식 시스템

[출처] Modified from T. Tang et al., "Arbitrary-Oriented Vehicle Detection in Aerial Imagery with Single Convolutional Neural Networks," *Remote Sens.*, vol. 9, no. 11, 2017, pp. 1-17, doi:10.3390/rs9111170, CC BY 4.0.

않으며, 일련의 추론 과정을 통해 객체를 인식한다. PASCAL VOC 2007과 2012 데이터셋으로 실험한 결과, 초당 155장의 영상을 훈련하여 Faster R-CNN보다 월등히 빠른 훈련 속도를 보였다. 하지만 인식 정확도는 Faster R-CNN보다 다소 떨어지며, 특히 작은 물체를 인식하는 데에 어려움을 보였다.

### 바. SSD

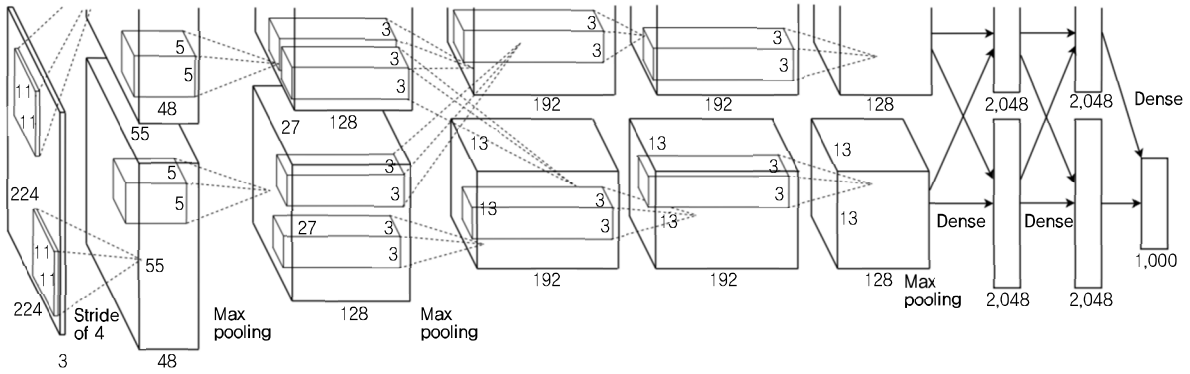
SSD[21]는 후보 영역을 생성하기 위한 RPN을 따로 훈련시키지 않고 다양한 크기의 Feature Map을 이용하여 객체를 인식한다. CNN 모델로부터 얻은 Feature Map은 (그림 4)와 같이 합성곱 층(Convolution Layer)이 진행됨에 따라 크기가 줄어들게 된다. SSD는 이 과정에서 추출된 모든 Feature Map들을 추론 과정에 사용하여 객체를 인식한다. 얕은 깊이에서 추출되어 크기가 큰 Feature Map은 작은 물체들을 검출할 수 있고, 깊은 깊이에서 추출되어 크기가 작은 Feature Map은 큰 물체들은 검출할 수 있다. SSD는 RPN을 제거함으로써 Faster R-CNN보다 훈련 속도를 향상시켰으며, 다양한 크기의 Feature Map을 이용하여 YOLO보다 정확하게 객체를 인식할 수 있다. PASCAL VOC 2007 데이터셋으로 실험한 결과, Faster R-CNN보다 약 3% 높은 mAP를 보였고, 초당 22장의 영상을 처리하여 YOLO보다 빠른 검출 속도를 보였다.

## 2. 객체 인식 모델

2절에서는 객체 인식을 위한 CNN 모델의 발전 동향에 대해 알아보고자 한다. 처음으로 깊은 CNN 구조를 시도했던 AlexNet부터 더 깊은 구조로의 발전 과정을 살펴본다.

### 가. AlexNet

AlexNet[9]은 8개의 층으로 이루어진 CNN 구조로, 이전보다 규모가 크고 깊은 CNN을 사용하였다. 이로 인해 발생되는 과적합(Over-Fitting) 문제를 방지하기 위한 방법을 제시하였고, 빠른 연산을 위해 2개의 그래픽처리장치(GPU: Graphics Processing Unit)를 사용하였다. AlexNet은 (그림 5)와 같이 5개의 합성곱 층과 3개의 완전하게 연결된 층(Fully-Connected Layer)으로 이루어진 2개의 CNN이 병렬적으로 구성되어 있어, 2개의 GPU을 이용하여 각 CNN을 학습한다. 활성 함수(Activation Function)로는 기존에 사용되던 쌍곡 탄




(그림 5) AlexNet의 구조

[출처] HE Haiwei et al., "Interchange Recognition Method Based on CNN," Acta Geodaetica et Cartographica Sinica, 2018;47(3):385-395 DOI: 10.11947/j.AGCS.2018.20170265 CC BY-NC-ND.

젠트(Hyperbolic Tangent)나 로지스틱 회귀(Logistic Regression)가 아닌 ReLU(Rectified Linear Unit)을 사용하여 학습 속도를 약 6배 향상시켰다. 과적합을 방지하기 위해 입력 영상을 임의로 잘라내거나 픽셀의 밝기를 조정하는 데이터 가공 방법을 이용하였으며, 완전연결 층에 Dropout[23]을 적용하였다. AlexNet은 ILSVRC-2012에서 높은 인식 정확도를 기록하며 1위를 차지했다.

### 나. ZFNet

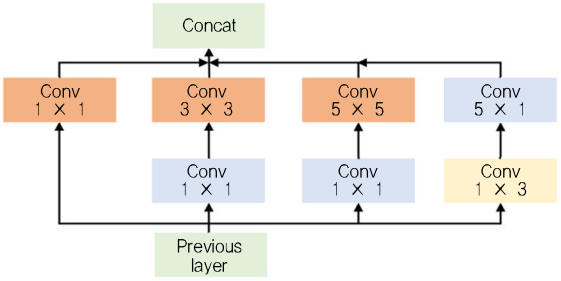
AlexNet 이후로 영상 분류의 정확도를 향상시키기 위한 깊은 구조의 CNN 모델들이 등장하였다. 하지만 이 모델들의 성능이 왜 향상되었는지, 그리고 어떻게 더 향상시킬 수 있는지에 대한 이해가 부족했다. ZFNet[11]의 저자는 이 문제점을 지적하며 CNN 모델 내부의 층들을 시각화하는 방법을 제시하였으며, 이를 이용하여 AlexNet을 개선한 모델인 ZFNet을 소개하였다. CNN 내부의 한 층을 시각화하기 위해, 층의 반응을 입력 영상의 크기로 매핑(Mapping)하여야 한다. 하나의 층은 합성곱, 활성, 통합의 세 가지 과정으로 이루어져 있기 때문에, 이를 각 층마다 역으로 수행하여, 입력 영상의 크기로 매핑시킨다. ZFNet의 5개의 합성곱 층 중 얕은 층에서는 선이나 모양 등의 단순한 특징들이 추출되고, 깊은 층에서는 객체의 형태와 가까운 특징들이 추출된다. 합성곱 층을 시각화하여 얻어낸 정보를 이용하여 AlexNet의 초기 층을 수정하여 ZFNet을 구성하였고, 이는 AlexNet보다 높은 분류 정확도를 보였다.

### 다. VGG

VGG[12]는 CNN의 층 깊이에 따른 성능의 변화를 연구를 통해 제안되었으며, 모델의 구조에서 층의 깊이를 제외한 모든 조건을 공평하게 하기 위해 5번의 통합을 사용하고 모든 필터의 크기를 3으로 설정하는 등 각 모델의 설정을 동일하게 사용하였다. 모델의 깊이는 11개부터 19개 사이에서 총 5개의 모델을 사용하여 실험하였다. VGG 모델은 필터의 크기를 3으로 설정하고 반복적으로 사용하면 더 큰 크기의 필터의 역할을 할 수 있다는 점을 강조하였으며, 이후에 개발된 많은 모델에서 필터의 크기를 3으로 사용하게 되었다. 비록 ILSVRC-2014에서 2위를 달성했지만, 매우 간단한 구조에 비해 우수한 성능을 보여 여전히 주목받는 모델이다.

### 라. GoogLeNet

CNN의 성능을 향상시키기 위한 가장 직관적인 방법




(그림 6) GoogLeNet의 Inception 모듈

[출처] Modified from M. Peng et al., "NIRFaceNet: A Convolutional Neural Network for Near-Infrared Face Identification," *Inform.*, vol. 7, no. 4, 2016, pp. 1-14, doi:10.3390/info7040061, CC BY 4.0.

은 구조를 깊게 하는 것이다. 하지만, 깊은 CNN 구조에서는 과적합이나 기울기 값의 소실(Vanishing Gradient) 문제가 일어날 가능성이 높아지고, 요구되는 연산량이 급격하게 증가한다. 이러한 문제는 CNN의 밀도를 낮게(Sparse) 함으로써 해결할 수 있지만, 연산량 측면에서는 밀도가 높은(Dense) 구조가 효과적이다. GoogLeNet[14]의 저자는 위의 두 가지 구조의 장점을 적절히 포함하는 Inception 모듈을 개발하여 적용하였다. Inception 모듈은 (그림 6)과 같이 여러 크기의 합성곱 층과 통합 층(Pooling Layer)이 병행하게 수행되고 그 결과를 합침으로써 다양한 특징을 추출할 수 있다. 이 구조는 여러 크기의 합성곱을 수행하기 때문에 연산량이 많이 요구된다. 이를 해결하기 위해 크기가 1인 합성곱을 이용하여 차원을 감소시키는 병목(Bottleneck) 구조를 구성하였다. GoogLeNet은 Inception 모듈 9개로 이루어진 CNN 모델로, ILSVRC-2014에서 VGG를 제치고 1위를 차지하였다. 이후 GoogLeNet 팀은 초기 모델을 개선하여 Inception V2, V3[24]를 발표했으며, GoogLeNet에 ResNet을 결합한 Inception ResNet[25]을 선보이며 지속적으로 개선해왔다.

### 마. ResNet

영상 객체 인식을 위한 CNN은 더 깊은 네트워크 구조를 구현함으로써 성능이 향상되어왔다. 하지만, 깊이가 수백, 수천 개로 증가하게 되면 오히려 정확도가 떨어지는 문제가 발생한다. ResNet[13]은 이 문제를 해결하기 위하여 잔차 학습(Residual Learning)이라는 방법을 적용하였다. 잔차 학습은 특정 층이 단순히 출력을 학습하는 것이 아니라 입력과 출력의 차이를 학습하여 작은 변화에 대해 민감하게 반응할 수 있도록 훈련하는 방법이다. 입력과 출력의 차이를 학습하게 하는 것은 덧셈으로만 구현되어 추가적인 매개변수도 요구되지 않아 계산상의 효율성도 유지시킬 수 있다. ResNet은 잔차 학습의 개념을 34개의 층으로 구성된 VGG에 적용시킨 모델이다. 잔차 학습이 적용되지 않은 동일한 VGG 모델의 경우 층수를 18개에서 34개로 증가시켰을 때 정확도가 감소했지만, 잔차 학습을 적용한 VGG 모델, 즉 ResNet 모델의 경우는 층이 증가함에 따라 정확도 또한 증가하는 것을 실험을 통해 확인하였다.

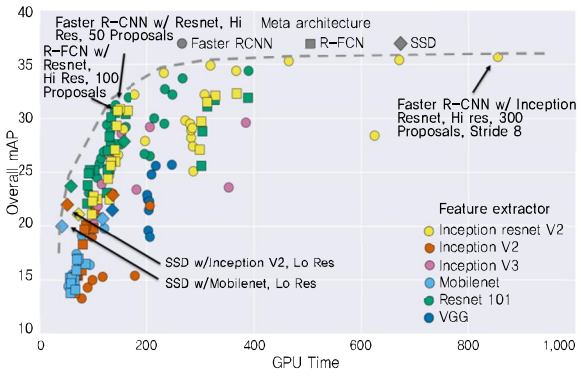
### 바. DenseNet

AlexNet의 등장 이후로 많은 모델이 성능 향상을 위해 깊이를 증가시켰고 ResNet의 등장으로 깊이가 증가하면서 발생하는 문제들을 해결하였다. 하지만, ResNet 이후로 모델의 구조 자체의 변화보다는 성능 향상만을 위한 연구가 진행되었다. 또한, ResNet과 이를 응용한 많은 모델은 특정 층들이 최종 출력에 기여하는 정도가 적거나 없어지는 문제를 가지고 있었다. DenseNet[15]은 ResNet 모델의 구조에 변화를 주어, ResNet의 층이 출력에 기여하지 않는 문제를 해결하여 성능을 향상시켰다. DenseNet 은 모든 층이 ResNet의 잔차 학습 방법으로 연결되어있는 구조다. 모든 층의 출력은 각각의 후속 층의 입력으로 들어간다. 또한, ResNet과는 다르게 합 연산을 하는 구조가 아닌 입력이 중첩되는 구조로, 예를 들어 $L$ 번째 층은 $L(L+1)/2$개의 입력을 갖는다. DenseNet 은 실험을 통해서 ResNet보다 더 적은 매개

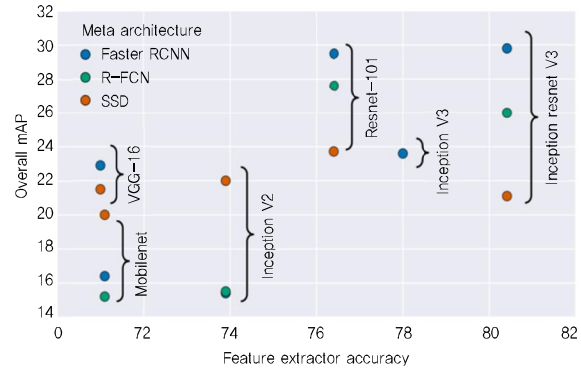
변수를 가지기 때문에 더 빠른 학습을 할 수 있고 향상된 성능을 나타낸다는 것을 증명하였다.

## Ⅲ. 객체 인식 기술 성능 비교

Ⅱ장에서는 다양한 CNN 모델과 이에 기반한 객체 인식 방법의 동향에 대해 살펴보았다. 각 모델과 방법은 정확도나 검출 속도 등의 측면에서 장단점을 가지고 있는데, Huang et al.[26]은 이를 동일한 조건에서의 실험을 통해 비교하였다. (그림 7)을 통해 위의 실험 결과를 훈련 시간과 정확도에 따라 확인할 수 있다. 대표적인 객체 인식 방법인 Faster R-CNN, R-FCN, SSD를 대상으로 실험하였으며, R-FCN과 SSD는 훈련에 소요되는 시간이 비교적 적지만 낮은 정확도를 보였고, Faster R-CNN은 훈련에 시간이 많이 소요되지만 비교적 높은 정확도를 보였다. CNN 모델에 따른 객체 인식 성능은




(그림 7) 객체 인식 방법의 비교[26]




(그림 8) CNN 모델의 객체 인식 성능 비교[26]

(그림 8)을 통해 확인할 수 있다. ResNet-101과 Inception ResNet V2의 정확도가 가장 높았으며, SSD의 경우 CNN 모델에 따른 정확도의 변동이 적은 편임을 확인할 수 있다.

## Ⅳ. 결론

현재의 객체 인식 기술은 특정 도메인에서의 객체 인식 정확도는 높은 편이지만, 일반적인 객체를 인식하는 데에는 아직 좋은 결과를 보이지 못한다. 이는 정확도를 높이기 위하여 더 깊은 CNN 모델을 구성하면, 과적합이나 기울기 소실 등의 문제가 발생하기 때문이다. 따라서 이를 해결하기 위한 연구가 계속되고 있으며, 객체를 상자 형태로만 검출하는 것이 아니라 정확한 픽셀 단위로 분할하는 연구도 이뤄지고 있다[27], [28].

또한, 현재의 객체 인식 기술은 작지 않은 규모의 하드웨어를 필요로 한다. 예를 들어, 위의 비교 실험을 하는 데에 32GB의 RAM(Random Access Memory)과 GeForce GTX Titan X GPU를 이용하였다. 즉, 모바일 환경이나 다른 임베디드 시스템에 고성능의 딥러닝 기반 객체 인식 기술을 적용하기에는 아직 어려움이 있다. 요구되는 하드웨어 자원을 감소시키기 위해, 보다 단순한 구조의 객체 인식 방법과 가벼운 CNN 모델에 대한 연구가 진행되어야 한다.

### 약어 정리

| | |
|---|---|
| CNN | Convolutional Neural Network |
| DPM | Deformable Part-based Model |
| HOG | Histogram of Oriented Gradients |
| IoT | Internet of Everything |
| mAP | Mean Average Precision |
| RAM | Random Access Memory |
| R-CNN | Region-based Convolutional Neural Network |
| ReLU | Rectified Linear Unit |
| RPN | Region Proposal Network |
| SIFT | Scale-Invariant Feature Transform |

| | |
|---|---|
| SSD | Single Shot Multibox Detector |
| SURF | Speed-Up Robust Features |
| SVM | Support Vector Machine |
| YOLO | You Only Look Once |

## 참고문헌


[1] M. Everingham et al., "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, June 2010, pp. 303-338.

[2] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, Dec. 2015, pp. 211-252.

[3] T. Lin et al., "Microsoft COCO: Common Objects in Context," *Eur. Conf. Comput. Vision(ECCV)*, Amsterdam, Netherlands, Oct. 8-16, 2014, pp. 740-755.

[4] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, 2004, pp. 91-110.

[5] H. Bay et al., "Speeded-Up Robust Features (SURF)," *Comput. Vision Image Understanding*, vol. 110, no. 3, 2008, pp. 346-359.

[6] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Proc. IEEE Conput. Soc. Conf. Comput. Vision Pattern Recogn.*, Kauai, HI, USA, Dec. 8-14, 2001, pp. I:511-I:518

[7] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn.*, San Diego, CA, USA, June 20-25, 2015, pp. 886-893.

[8] P.F. Felzenszwalb et al., "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, 2010, pp. 1627-1645.

[9] A. Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks," *Conf. Neural Inform. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 3-6, 2012, pp. 1097-1105.

[10] Y. Lecun et al., "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE*, vol. 86, no. 11, Nov. 1998, pp. 2278-2324.

[11] M.D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *Eur. Conf. Comput. Vision(ECCV)*, Amsterdam, Netherlands, Oct. 8-16, 2014, pp. 818-833.

[12] K. Simonyan et al., "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Int. Conf. Learning Representations*, San Diego, USA, May 7-9, 2015.

[13] K. He et al., "Deep Residual Learning for Image Recognition," *IEEE Conf. Comput. Vision Pattern Recogn.*, Las Vegas, NV, USA, June 27-30, 2016, pp. 770-778.

[14] C. Szegedy et al., "Going Deeper with Convolutions," *IEEE Conf. Comput. Vision Pattern Recogn.*, Boston, MA, USA, June 7-12, 2015, pp. 1-9.

[15] G. Huang et al., "Densely Connected Convolutional Networks," *IEEE Conf. Comput. Vision Pattern Recogn.*, Honolulu, HI, USA , July 21-26, 2017, pp. 2261-2269.

[16] R. Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *IEEE Conf. Comput. Vision Pattern Recogn.*, Columbus, OH, USA, June 23-28, 2014, pp. 580-587.

[17] R. Girshick, "Fast R-CNN," *IEEE Int. Conf. Comput. Vision*, Santiago, Chile, Dec. 7-13, 2015, pp. 1440-1448.

[18] S. Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, 2017, pp. 1137-1149.

[19] J. Dai et al., "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *Conf. Neural Inform. Process. Syst.*, Barcelona, Spain, Dec. 4-6, 2016, pp. 379-387.

[20] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," *IEEE, Conf. Comput. Vision Pattern Recogn.*, Las Vegas, NV, USA, June 27-30, pp.779-788.

[21] W. Liu et al., "SSD: Single Shot MultiBox Detector," *Eur. Conf. Comp. Vision*, Amsterdam, Netherlands, Oct. 8-16, 2016, pp. 21-37.

[22] J.R.R. Uijlings et al., "Selective Search for Object Recognition," *Int. J. Comput. Vision*, vol. 104, no. 2, 2013, pp. 154-171.

[23] N. Srivastava et al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learning Res.*, vol. 15, 2014, pp. 1929-1958.

[24] C. Szegedy et al., "Rethinking the Inception Architecture for Computer Vision," *Comput. Vision Pattern Recogn.*, Las Vegas, NV, USA, June 27-30, 2016, pp. 2818-2826.

[25] C. Szegedy et al., "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Feb. 4-9, 2017, pp. 4278-4284.

[26] J. Huang et al., “Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors,” *Comput. Vision Pattern Recogn.*, Honolulu, HI, USA, July 22-24, 2017, pp. 7310-7319.

[27] K. He et al., “Mask R-CNN,” *IEEE Int. Conf. Comput. Vision*, Venice, Italy, Oct. 22-29, 2017, pp. 2980-2988.

[28] J. Long et al., “Fully Convolutional Networks for Semantic Segmentation,” *IEEE Conf. Comput. Vision Pattern Recogn.*, Boston, MA, USA, june 7-12, pp. 3431-3440.