한국컴퓨터그래픽스학회
Korea Computer Graphics Society Vol. 25, No. 5, P. 11~20

Atrous Convolution과 Grad-CAM을 통한 손 끝 탐지

노대철, 김태영
서경대학교 컴퓨터공학과
{sheocjf1025, tykim}@skuniv.ac.kr

Fingertip Detection through Atrous Convolution and Grad-CAM

Dae-Cheol Noh, Tae-Young Kim
School of Computer Engineering, Seokyeong University

요 약

딥러닝 기술의 발전으로 가상 현실이나 증강 현실 응용에서 사용하기 적절한 사용자 친화적 인터페이스에 관한 연구가 활발히 이뤄지고 있다. 본 논문은 사용자의 손을 이용한 인터페이스를 지원하기 위하여 손 끝 좌표를 추적하여 가상의 객체를 선택하거나, 공중에 글씨나 그림을 작성하는 행위가 가능하도록 딥러닝 기반 손 끝 객체 탐지 방법을 제안한다. 입력 영상에서 Grad-CAM으로 해당 손 끝 객체의 대략적인 부분을 잘라낸 후, 잘라낸 영상에 대하여 Atrous Convolution을 이용한 합성곱 신경망을 수행하여 손 끝의 위치를 찾는다. 본 방법은 객체의 주석 전처리 과정을 별도로 요구하지 않으면서 기존 객체 탐지 알고리즘 보다 간단하고 구현하기에 쉽다. 본 방법을 검증하기 위하여 Air-Writing 응용을 구현한 결과 평균 81%의 인식률과 76 ms 속도로 허공에서 지연 시간 없이 부드럽게 글씨 작성이 가능하여 실시간으로 활용 가능함을 알 수 있었다.

Abstract

With the development of deep learning technology, research is being actively carried out on user-friendly interfaces that are suitable for use in virtual reality or augmented reality applications. To support the interface using the user's hands, this paper proposes a deep learning-based fingertip detection method to enable the tracking of fingertip coordinates to select virtual objects, or to write or draw in the air. After cutting the approximate part of the corresponding fingertip object from the input image with the Grad-CAM, and perform the convolution neural network with Atrous Convolution for the cut image to detect fingertip location. This method is simpler and easier to implement than existing object detection algorithms without requiring a pre-processing for annotating objects. To verify this method we implemented an air writing application and showed that the recognition rate of 81% and the speed of 76 ms were able to write smoothly without delay in the air, making it possible to utilize the application in real time.

키워드: 딥러닝, Atrous Convolution, Grad-CAM, 객체 탐지
Keywords: Deep Learning, Atrous Convolution, Grad-CAM, Object Detection

*corresponding author: Tae-Young Kim/Seokyeong University(tykim@skuniv.ac.kr)

Received : 2019.11.05./ Review completed : 1st 2019.11.22. / Accepted : 2019.11.29.
DOI : 10.15701/kcgs.2019.25.5.11
ISSN : 1975-7883(Print)/2383-529X(Online)

1. 서론

최근 빅 데이터 처리가 용이해지고 컴퓨팅 디바이스 기술의 발전으로 고성능 컴퓨팅이 가능해짐에 따라 많은 연산을 요구하는 딥러닝 기술이 급속도로 발전하고 있다. 컴퓨터 비전 분야에서 영상 데이터에 대해 인간의 시각적인 처리를 모방하기 위해 중요점(Keypoint)을 사용하는 SIFT(Scale Invariant Feature Transform)[1], 하르 특징(Haar Feature)을 계산하여 분류하는 Haar Cascades[2], 영상 데이터에서 특징을 추출하여 히스토그램(Histogram) 기법으로 객체를 탐지하는 BoW(Bag of Word)[3] 등과 같은 디지털 영상처리(Digital Image Processing) 알고리즘을 이용한 연구가 활발하게 진행되었지만, 대부분 다양한 배경, 조명 등의 외부 조건에 유연하게 대처하지 못한다는 단점이 있다. 딥러닝 기술의 발전으로 이러한 문제점들이 해소되고 있는데, 대표적인 기술로 합성곱 신경망(Convolution Neural Network)을 사용한 객체 분류(Classification)[4-5], 객체 탐지(Object Detection)[6-7] 그리고 시맨틱 세그먼테이션(Semantic Segmentation)[8-9] 등이 있다.

이 중 객체 탐지 알고리즘은 어떠한 영상 데이터에서 특정한 객체를 탐지하는 알고리즘이다. 최근 다양한 객체 탐지 네트워크[10-11]가 공개되었지만, 학습 데이터에 대해 정답 값(Ground Truth)에 맞는 주석(Annotation) 전처리 과정을 거쳐야 하고, Selective Search, Sliding Window, 영상 데이터를 일정 크기의 그리드(Grid)로 분할하여 확률 점수(Probability Score)를 계산하는 등 복잡한 알고리즘을 거쳐야 한다. 위의 알고리즘은 구현 시 어려움을 제공할 수 있고, 실제로 탐지하고자 하는 객체가 없는 부분까지 모두 검사하는 등의 불필요한 과정이 포함된다.

본 논문은 위의 단점들을 개선하기 위해 합성곱 신경망의 시각화에 주로 사용되는 Grad-CAM[12] 기술과 시맨틱 세그먼테이션 분야에서 사용되는 Atrous Convolution[13]을 사용하여 손 끝 객체 탐지 방법을 제안한다. 입력 손 영상 데이터에 대하여 합성곱 신경망을 수행한 다음 기울기 값을 이용하여 Grad-CAM 알고리즘을 수행하여 일차적으로 손 끝이라고 할 수 있는 대략적인 범위 영역을 원본 영상 데이터에서 잘라낸다. 이 데이터는 다시 합성곱 신경망의 입력 영상 데이터로 사용되어 일반적인 합성곱 연산과 Atrous Convolution 연산을 거쳐 손 끝의 위치를 알 수 있는 특징 맵을 제작한다. 이 특징 맵에 특정한 임계 값(Threshold)을 기준으로 이진 처리를 한 후, 윤곽처리를 통해 최종적으로 손 끝 객체를 탐지한다. 본 방법은 기존의 정답 값에 맞는 주석 전처리 과정을 탐지하고자 하는 손 끝 영상 데이터를 학습시킴으로써 해결하였고, 실제로 탐지하고자 하는 객체가 존재하지 않는 위치까지 검사하는 기존 객체 탐지 알고리즘과 달리 Grad-CAM을 통해 그 영역을 한정시켜 속도를 높였고 Atrous Convolution을 통해 정확한 손 끝 좌표를 찾는다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구로 Atrous Convolution과 Grad-CAM을 소개한다. 3장에서 본 논문에서 제안하는 손 끝 객체 탐지 방법을 설명한다. 4장에서 본 방법의 학습 및 평가를 위해 자체 제작한 데이터 세트 소개 및 일반적인 합성곱 연산과 Atrous Convolution의 비교, CIFAR-10 데이터 세트에 대한 객체 탐지 실험 결과를 기술한 후 5장에서 결론을 맺는다.

2. 관련 연구

2.1 Atrous Convolution

Atrous Convolution은 2017년에 발표된 DeepLab v2 논문에서 공개한 합성곱 연산의 형태로써, 같은 크기의 필터를 사용하는 일반적인 합성곱 형태와 동일한 연산량을 가지지만 수용 영역(Receptive Field)을 늘려준 합성곱 형태이다. 기존의 일반적인 합성곱 연산은 영상 내부에서 국소적인 부분에 대해서만 연산을 수행하지만, Atrous Convolution은 같은 크기의 일반적인 합성곱 연산과 같은 연산량을 가지면서 전역적이고 배경과 객체 간의 차이점을 잡아낼 수 있다는 장점이 있다.

Atrous Convolution은 그림 1과 같이 인접한 픽셀 간에 일정한 공백을 둠으로써 영상에 적용되는 합성곱의 수용 영역을 확장한다.

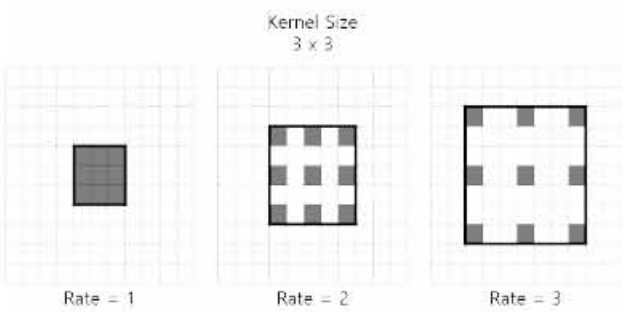


Figure 1. Atrous convolution with different rates

Atrous Convolution은 찾고자 하는 객체에 대해 픽셀 단위의 세밀한 부분을 찾아야 하는 시맨틱 세그먼테이션(Semantic Segmentation) 분야에서 주로 사용되는 합성곱 형태이다. 대부분의 합성곱 신경망은 합성곱 이후 각각의 풀링(Pooling) 단계에서 특징 맵(Feature Map)의 크기가 이전 특징 맵 크기의 1/4로 줄어들기 때문에 세밀한 정보를 찾기 힘들다는 단점이 있다. Atrous

Convolution은 출력된 특징 맵에서 배경과 분류하고자 하는 객체 사이의 차이점을 찾기에 적합한 합성곱이므로 원본 크기인 224x224로 업샘플링(Upsampling) 했을 경우 원본 영상의 객체 위치에 대한 복원율이 높다.

2.2 Grad-CAM

Grad-CAM은 네트워크의 내부를 해석하기 위한 시각화 분야에서 유용하게 사용 중인 기술로 클래스 활성화 맵(Class Activation Map)[14]에서 발전된 것이다. 클래스 활성화 맵은 전역 평균 풀링(Global Average Pooling)[15] 층이 반드시 들어가기 때문에 전역 평균 풀링 층이 없는 합성곱 신경망은 재학습을 실시해야 하고 전역 평균 풀링 이전의 합성곱 층에서 출력되는 특징 맵만 사용할 수 있는 단점이 있었다.

대부분의 객체 분류 네트워크의 학습 방법은 임의의 가중치(Weight)를 시작으로 순방향으로 합성곱 연산을 수행한 후, 출력층에서 오차를 계산하고 역전파(Backpropagation) 알고리즘을 사용하여 역방향으로 진행하며 가중치를 갱신한다. 이와 같은 과정은 오차가 더 이상 줄어들지 않을 때까지 반복 진행된다. 갱신되는 가중치 값은 현재의 가중치 값에서 가중치의 기울기(Gradient) 값을 뺌으로써 구할 수 있는데, 기울기가 0에 가까워질수록 네트워크가 객체 분류에 있어서 높은 성능을 보여준다는 것을 의미한다. Grad-CAM은 대부분의 객체 분류 네트워크가 사용하는 기울기 정보를 사용하기 때문에 기존의 클래스 활성화 맵의 단점을 보완하면서 보고자 하는 합성곱 층의 특징 맵에 가장 적합하다고 판단되는 부분을 활성화시켜 컴퓨터가 해당 영상의 어느 부분을 바라보고 있는지 시각적으로 볼 수 있다.

Grad-CAM은 우선 역전파 방법을 통한 기울기 값을 전역 평균 풀링 방식($\frac{1}{Z}\sum_i\sum_j$)으로 곱해 뉴런 중요도 가중치(Neuron Importance Weights) α_k^c를 구한다(식 1). 여기서 기울기 $\frac{\partial y^c}{\partial A_{ij}^k}$는 소프트맥스 함수 이후에 출력되는 분류 범주 값 c에 대한 소프트맥스 이전의 레이어의 값 y를 보고자 하는 합성곱 층의 특징 맵 A^k으로 미분한 값이다.

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (1)$$

이 뉴런 중요도 가중치로 찾고자 하는 분류 범주에 대한 특징 맵의 중요도를 찾아낼 수 있다. 뉴런 중요도 가중치 α_k^c와 보고자 하는 층의 특징 맵 A^k를 곱한 후에 특징 맵의 채널 수 k만큼 모두 더해주고 ReLU[16] 활성화 함수(Activation Function)를 적용함으로써 Grad-CAM을 구할 수 있다(식 2).

$$Grad-CAM = ReLU(\sum_k \alpha_k^c A^k) \quad (2)$$

Grad-CAM은 클래스 활성화 맵과 달리 전역 평균 풀링이 없는 네트워크의 재학습이 필요하지 않기 때문에 더욱 포괄적이고, 마지막 합성곱 층이 아닌 신경망 중간에 위치한 합성곱 층의 특징 맵에 대해서도 객체 분류의 중요한 부분을 볼 수 있기 때문에 더욱 일반적으로 사용될 수 있다. 결론적으로, Grad-CAM을 사용하면 그림 2와 같이 일반적인 영상에서 컴퓨터가 영상의 어느 부분을 보고 분류 결과를 도출하였는지 시각적으로 확인할 수 있다.

Figure 2. Grad-CAM visualization results; (a) original images; (b) Grad-CAM images; (c) original + Grad-CAM images; (d) original + contoured Grad-CAM images

3. Atrous Convolution과 Grad-CAM을 이용한 손 끝 객체 탐지 방법

본 논문은 Atrous Convolution과 Grad-CAM을 이용한 손 끝 객체 탐지 방법을 제안한다. 기존의 객체 탐지 네

트워크 모델은 학습을 위한 데이터 세트 준비 단계에서 모든 데이터에 대해 정답 값(Ground Truth)을 위한 주석(Annotation) 전처리 과정이 필요하다. 이는 학습 데이터의 수량이 적을 경우에는 별다른 문제가 되지 않지만, 많은 양의 데이터를 학습시켜야 하는 딥러닝에서 각 학습 데이터마다 객체의 위치를 찾아서 기록해야 한다는 것은 학습 이전 단계부터 많은 작업량을 요구한다. 또한, Faster-RCNN, YOLO와 같은 대표적인 객체 탐지 네트워크는 객체 탐지 중에 불필요한 과정이 많이 포함된다. Faster-RCNN 네트워크는 영상 전체에 걸쳐 일정 크기의 앵커 상자(Anchor Box)를 이동시키며 객체를 찾아내는 방법을 사용하고, YOLO 네트워크는 입력 영상을 일정 크기와 개수의 그리드로 분할하고 각 그리드마다 찾고자 하는 객체의 존재 여부를 검사한다. 이 두 방법 모두 입력 영상 내에서 탐지하고자 하는 객체가 존재하지 않는 부분까지 모두 검사를 해야 하는 단점이 있다. 또한, YOLO 네트워크의 경우 객체를 탐지하기 위해서 NMS(Non-Maximum Suppression)[17] 알고리즘을 통해 각 앵커 상자가 가지고 있는 확률 점수를 토대로 가장 높은 점수의 경계 상자를 선택해야 하고, 대상 객체의 위치에 많은 개수의 경계 상자를 그린 후 실제 정답 값과 IoU(Intersection Of Union)를 계산해야만 최종 경계 상자를 찾을 수 있다. 마지막으로, 개발자가 실제로 네트워크를 직접 구현하지 않고 특정 플랫폼에서 제공하는 것을 사용할 경우, 특수한 목적에 맞게 개량하기 힘들거나 유지보수가 어려운 단점이 있다.

이러한 단점을 개선하기 위해 본 논문은 VGGNet[18] 합성곱 신경망에 Atrous Convolution을 적용한 신경망을 제안한다(그림 3). 본 방법은 기존의 Faster-RCNN이나 YOLO 네트워크와 달리 Grad-CAM으로 탐지하고자 하는 객체의 대략적인 위치를 찾아낸 후 그 위치의 영상에 대해서만 합성곱 신경망으로 추출한 특징 맵을 사용하기 때문에 입력 영상의 모든 부분을 검사할 필요가 없다. 이로 인해 데이터 세트 주석 전처리 과정은 손 끝 영상을 학습시키는 것으로 생략할 수 있다.

손 끝 객체 탐지를 위한 입력 데이터는 일반적인 USB 카메라에서 초당 60프레임의 속도로 촬영한 사용자의 손끝 영상을 사용한다. 입력 데이터는 네트워크 구조에 맞게 224x224 크기로 리사이즈하여 신경망의 입력으로 사용하고, 3x3 필터를 사용하여 합성곱을 두 번 진행하고 최대 풀링을 수행한다. 세 번째 최대 풀링 층까지 총 7번의 합성곱을 수행한 후에, Atrous Convolution을 사용하여 6번의 합성곱을 수행한다. 이 때, 특징 맵의 데이터 손실을 최소화하기 위해 세 번째 최대 풀링 층에서 28x28의 크기로 줄인 이후 더 이상 최대 풀링 연산을 수행하지 않는다. 6번째 Atrous Convolution을 수행한 후에 28x28 필터를 사용하여 전역 평균 풀링을 수행하고, 완전 연결 층(Fully Connected Layer)을 거쳐 소프트맥스 함수를 통해 인식 결과를 출력한다.

본 논문에서 제안하는 손 끝 객체 인식 방법은 그림 4와 같다. 손 영상을 입력 데이터로 하여 합성곱 신경망을 거쳐 출력된 인식 결과와 소프트맥스 함수 이전의 완전 결합 층의 출력 값, 보고자 하는 특징 맵, 역전파법을 통한 가중치를 바탕으로 Grad-CAM을 계산하여 일차적으로 보고자 하는 손 끝 객체의 대략적인 범위를 찾는다. 이후 원본 영상에서 Grad-CAM으로 찾은 부분을 잘라낸 후, 이 영상을 합성곱 신경망의 입력으로 사용한다. 합성곱 신경망 이후, 2번째, 3번째 최대 풀링 층

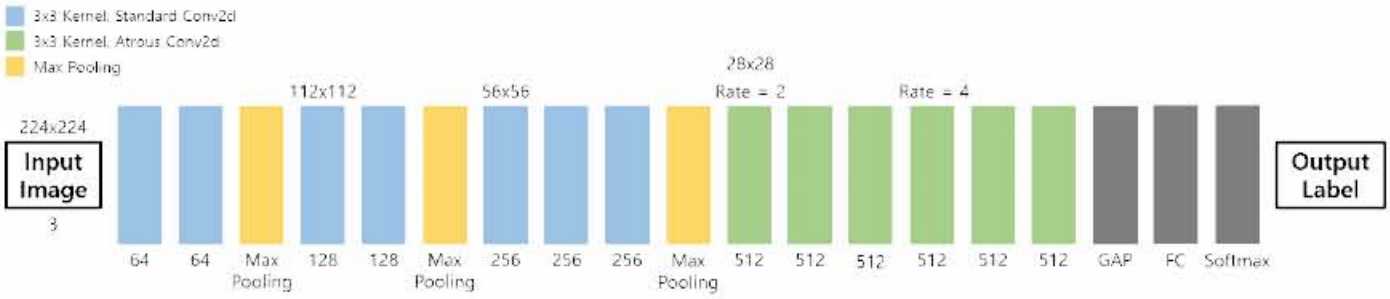


Figure 3. VGGNet network using Atrous convolution

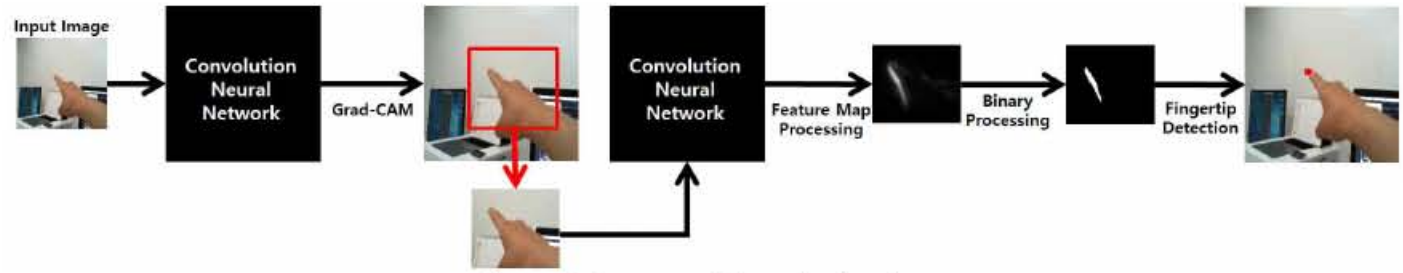


Figure 4. Process of fingertip detection

이전의 합성곱 층인 4번째, 7번째 층과 Atrous Convolution의 비율 값이 각각 2와 4에서 3번씩 합성곱을 진행한 10번째, 13번째 층의 특징 맵을 얻는다(그림 5).

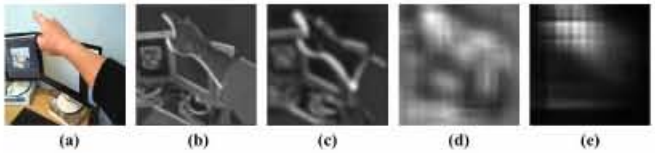


Figure 5. Feature map results; (a) original image; (b) feature map of the 4th layer; (c) feature map of the 7th layer; (d) feature map of the 10th layer; (e) feature map of the 13th layer

그림 5의 특징 맵들은 크기가 서로 다르기 때문에 224x224 크기로 이진 선형 보간(Bilinear Interpolation)을 해준 후에 모두 곱해줌으로써 분류 범주에 대해 중요도가 높은 부분만을 남겨줄 수 있다. 이후 특정한 임계 값을 바탕으로 이진화한 다음 윤곽(Contouring) 처리를 해줌으로써 최종적으로 손 끝 객체를 탐지할 수 있다(그림 6).

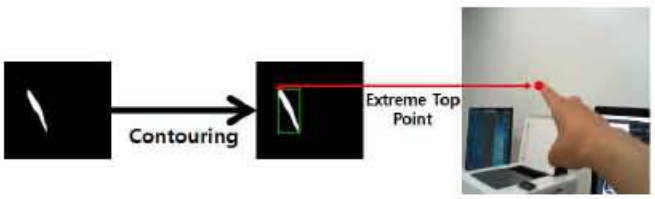


Figure 6. Final process of fingertip detection

4. 실험

본 실험은 프로세서 Intel Core i5 7500, 그래픽카드 GeForce GTX 1060, RAM 8GB 등으로 구성된 장비와 Python 3.6 기반의 Tensorflow 1.14.0 GPU 버전을 개발도구로 사용한 환경에서 진행되었다.

4.1 데이터 세트

본 실험에 사용된 데이터 세트는 640x480 해상도의 카메라로 다양한 배경, 각도, 거리, 위치 등을 기준으로 촬영한 손 데이터 영상을 사용하였다. 분류 범주는 총 3개로 구성되는데, 각각 손 끝 영상과 연관성이 전혀 없는 영상(Nothing), 객체 탐지가 가능한 손 끝 영상(Pointing), 그리고 누적된 평가 기록을 초기화시키는 정적 제스처 영상(Spread Palm)으로 구성된다 (표 1). 각 분류 범주마다 학습 데이터 4,000장과 평가 데이터 400장을 수집 및 촬영한 후, 잘라내기(Cropping) 기법을 사용하여 5배 증대시켜 각각 20,000장과 2,000장을 제작하였다. 또한, 조명에 강인한 합성곱 신경망 학습을 위해 제작된 손 데이터 영상은 영상 전체에 0.5에서 1.2 사이의 무작위 값을 곱해줌으로써 주변의 밝기로 인한 영향을 줄였다.

Table 1. Examples of dataset

Label	Examples
Nothing	
Pointing	
Spread Palm	

4.2 실험 결과

본 논문의 방법을 검증하기 위해 Atrous Convolution을 사용하지 않고 일반적인 합성곱 연산만으로 실험한 결과 Atrous Convolution을 사용할 때보다 비교 영상 데이터 안에서 분류 범주에 강인한 부분을 제대로 찾지 못해 부정확한 객체 탐지 결과가 발생함을 알 수 있었다. 또한, Grad-CAM의 유무에 따라 손 끝 객체 탐지에 영향이 있다는 것을 확인하였고, 본 논문의 성능 검증을 위해 기존의 SSD[19] 객체 탐지 네트워크와 동일한 평가 데이터 세트를 사용하여 인식률과 속도를 비교하였다. 마지막으로 자체 제작한 손 데이터 뿐 아니라 대중적으로 사용 중인 10개의 분류 범주를 가진 CIFAR-10[20] 데이터 세트를 학습시켜 객체 탐지를 실험한 결과, 10개 분류 범주에 대해 모두 성공적으로 객체 탐지가 가능함을 확인하였다.

4.2.1 일반적인 합성곱과 Atrous Convolution의 비교

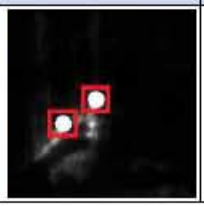
그림 7은 Atrous Convolution을 사용한 경우와 그렇지 않은 경우를 그림으로 비교한 것이다. 결과 영상에서 알 수 있듯이 Atrous Convolution을 사용했을 경우 더 효과가 좋음을 알 수 있다. 즉 Atrous Convolution은 손 끝 부분을 가장 중요도가 높다고 판단하지만, 일반적인 합성곱은 어느 부분이 중요한지 표현하지 못함을 알 수 있다.

Figure 7. Original image, feature map with Atrous convolution and feature map with standard convolution

4.2.2 Grad-CAM의 필요성

표 2는 손끝 영상에 대하여 Grad-CAM을 사용한 경우와 사용하지 않은 경우를 그림으로 비교한 것이다. 표에서 보는 바와 같이, Grad-CAM을 사용하지 않은 경우, 손 끝이라고 판단할 수 있는 위치가 두 곳으로 분류되었고, Grad-CAM을 사용한 경우, 손 끝 부분을 정확하게 분류하였다. 이 결과를 통해 Grad-CAM으로 원본 영상을 한 번 잘라준 후에 재평가하는 것이 성능이 더 좋음을 알 수 있다.

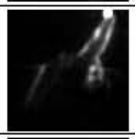
Table 2. Comparison between using and not using the Grad-CAM

Original Image	No Grad-CAM	Grad-CAM
		

4.2.3 여러 환경에서의 손 끝 탐지

본 논문에서는 배경, 조명, 위치 등 다양한 환경에 대해 강인한 분류를 보이도록 학습 데이터를 제작하였다. 실험결과, 배경, 조명, 위치 등에 영향을 받지 않고 손 끝 좌표를 정확히 찾아냄을 알 수 있었다 (표 3).

Table 3. Fingertip detection in various environments

Condition	Original image	Result image	Feature map image
Normal			
Dark			
Complex Background			
Two Hands			

4.2.4 SSD 객체 탐지 네트워크와 비교

본 논문에서 사용한 네트워크와 알고리즘의 성능을 검증하기 위해 100장의 손 끝 평가 데이터 세트를 통해 기존의 SSD 객체 탐지 네트워크와 성능 비교를 하였다. 표 4는 SSD 객체 탐지 네트워크와 본 방법의 성공 사례이고, 표 5는 SSD 객체 탐지 네트워크의 실패 사례,

표 6은 본 방법의 실패 사례이다.

Table 4. Comparison between SSD network and ours (in case the SSD and our method are correct)

Original image	SSD	Ours	Our feature map

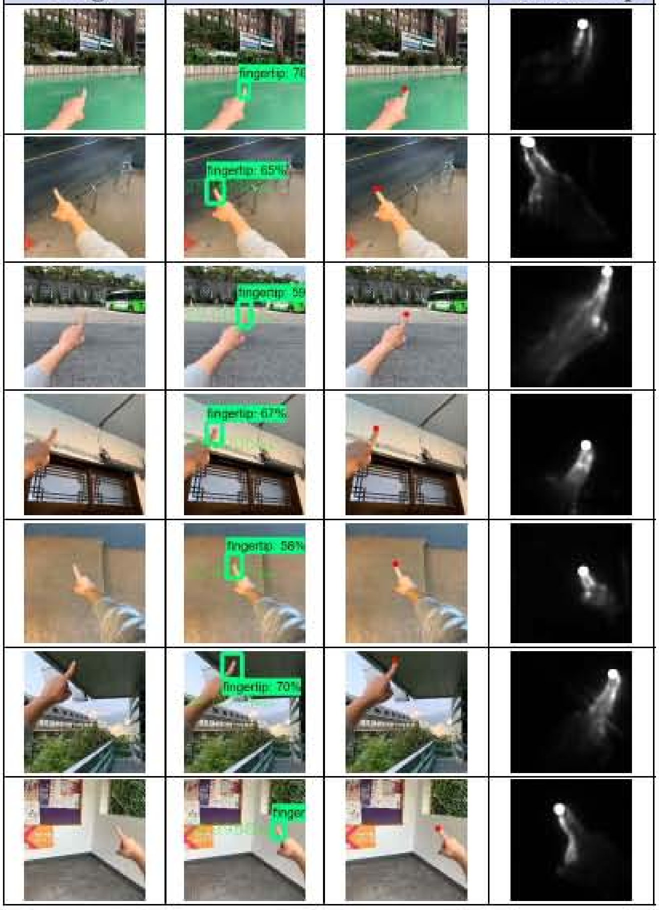


Table 5. Comparison between SSD network and ours (in case the SSD is wrong)

Original image	SSD	Ours	Our feature map

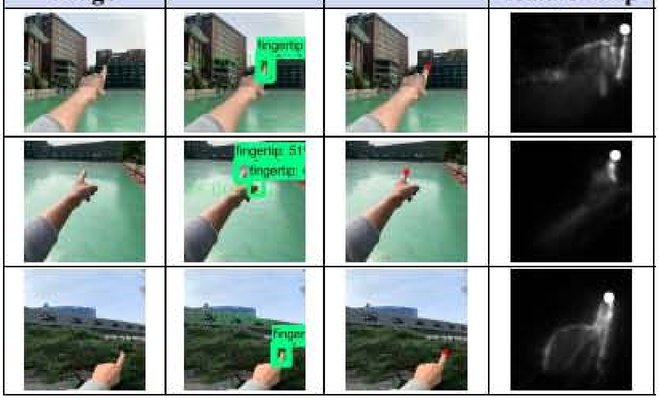
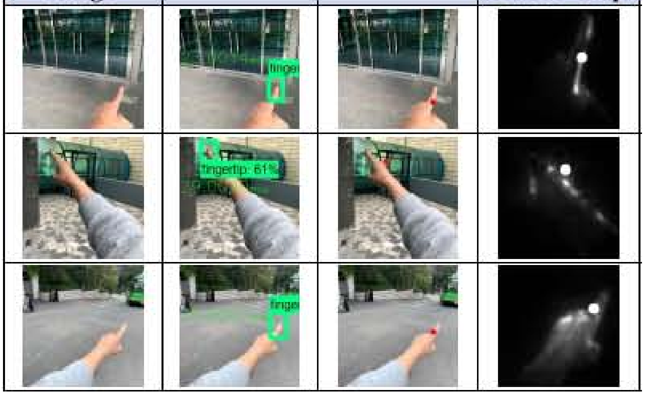


Table 6. Comparison between SSD network and ours (in case our method is wrong)

Original image	SSD	Ours	Our feature map



성능 비교 결과, SSD 객체 탐지 네트워크는 평균 31.34ms의 속도, 인식률 79%의 성능을 보였고, 본 논문에서 제안한 방법은 76.02ms의 속도, 인식률 81%의 성능을 보였다 (표 7). 본 방법의 속도가 SSD보다 다소 낮지만, 본 방법은 높은 인식률로 초당 13장의 이미지를 분류할 수 있는 속도를 가지므로 실시간으로 활용 가능함을 알 수 있었다.

Table 7. Speed and recognition rate comparisons of SSD network and ours

Network	Speed	Recognition rate
SSD	31.34ms	79%
Ours	76.02ms	81%

4.2.5 CIFAR-10 Dataset

본 논문에서 사용한 손 데이터뿐만 아니라, 일반적으로 사용되는 CIFAR-10 데이터 세트를 통해 객체 탐지를 실시하였다. CIFAR-10 데이터 세트는 비행기, 자동차, 새, 고양이, 사슴, 개, 개구리, 말, 배, 트럭으로 이루어진 10개의 분류 범주를 가진 데이터 세트로, 각 분류 범주마다 5000장의 학습 데이터와 1000장의 평가 데이터를 가지고 있다. 실험 결과 표 8과 같이 주변에서 흔히 볼 수 있는 영상 데이터에 대해서도 객체 탐지가 가능함을

알 수 있었다.

Table 8. Experimental results for the CIFAR-10 dataset

Label	Original Image	Detected Image
Airplane		
Automobile		
Bird		
Cat		
Deer		
Dog		
Frog		
Horse		
Ship		
Truck		

4.3 응용 사례

본 논문의 손 끝 객체 탐지의 성능을 확인하기 위하여 Air-Writing 프로그램을 개발하였다(그림 8). 프로그램은 손 끝 객체를 실시간으로 탐지 및 추적하여 문자를 작성하는 프로그램으로, 본 논문에서 소개한 Atrous Convolution과 Grad-CAM을 사용한 합성곱 신경망 및 손 끝 객체 탐지 방법이 평균 76ms로 지연 시간 없이 높은 정확성을 가지고 문자를 작성할 수 있음을 알 수 있었다.

Figure 8. Air-Writing with fingertip detection

5. 결론

본 논문은 주석 전처리 과정과 반복적이고 복잡한 객체 탐지 알고리즘 없이 Atrous Convolution과 Grad-CAM을 이용한 손 끝 객체 탐지 방법을 제안하였다. 본 방법을 이용하여 사용자가 허공에 쓴 단어를 인식하는 Air-Writing 응용 프로그램을 구현한 결과 일반적인 USB 웹캠 카메라로 영상을 입력받아 지연 시간 없이 실시간으로 문자 작성이 가능하였다. 위의 결과를 바탕으로 가상 현실이나 증강 현실 응용에서 Air Writing 이외에도 사용자의 손을 이용한 서명, 메모 작성이나 그림을 그리는 등 다양한 손 인터페이스로 사용될 수 있음을 알 수 있다.

감사의 글

본 연구는 2019학년도 서경대학교 교내연구비 지원에 의하여 이루어졌음.

References

[1] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV,* 60(2): pp. 91-110, 2004.

[2] P. Viola, and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *CVPR,* pp. 511-518, 2004.

[3] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” *Workshop on statistical learning in computer vision, ECCV,* pp. 1-22, 2004.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ ImageNet classification with deep convolutional neural networks,” *Advances in neural information processing systems,* pp. 1097-1105, 2012.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 770-778, 2016.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp.580-587, 2014.

[7] R. Girshick, “Fast R-CNN,” *Proceedings of the IEEE international conference on computer vision,* pp.1440-1448, 2015.

[8] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence,* pp. 1915-1929, 2012.

[9] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *arXiv preprint arXiv: 1412.7062*, 2014.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems,* pp. 91-99, 2015.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788, 2016.

[12] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Proeceedings of the IEEE international conference on computer vision,* pp. 618-626, 2017.

[13] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE transactions on pattern analysis and machine intelligence,* 40(4): pp. 834-848, 2017.

[14] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," P*roceedings of the IEEE conference on computer vision and pattern recognition,* pp. 2921-2929, 2016.

[15] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[16] V. Nair, G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807-814, 2010.

[17] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 4507-4515, 2017.

[18] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv: 1409.1556*, 2014.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *European conference on computer vision*, pp. 21-37, 2016.

[20] A. Krizhevsky, and G. Hinton, "Learning multiple layers of features from tiny images," *Tech Report*, 2009.

〈저자소개〉

노 대 철

- 2014년-현재 서경대학교 컴퓨터공학과 학사 재학 중
- 관심분야: 가상현실, 게임프로그래밍, 컴퓨터 비전, 머신 러닝
- https://orcid.org/0000-0001-5516-5484

김 태 영

- 1991년 이화여자대학교 전자계산학과 학사
- 1993년 이화여자대학교 전자계산학과 석사
- 1993년-2002년 한국통신 멀티미디어연구소 선임연구원
- 2001년 서울대학교 전기컴퓨터공학부 박사
- 2002년-현재 서경대학교 컴퓨터공학과 교수
- 관심분야: 실시간 렌더링, 증강현실, 딥러닝, 영상처리, 모바일 3D
- https://orcid.org/0000-0002-4992-4197