

## 한국차세대컴퓨팅학회 논문지 Vol.16 No.3

ISSN : 1975-681X(Print)

### 실시간 손끝 탐지를 위한 VGGNet 기반 객체 탐지 네트워크

노대철, 김태영

**To cite this article :** 노대철, 김태영 (2020) 실시간 손끝 탐지를 위한 VGGNet 기반 객체 탐지 네트워크, 한국차세대컴퓨팅학회 논문지, 16:3, 16-26

① earticle에서 제공하는 모든 저작물의 저작권은 원저작자에게 있으며, 학술교육원은 각 저작물의 내용을 보증하거나 책임을 지지 않습니다.

② earticle에서 제공하는 콘텐츠를 무단 복제, 전송, 배포, 기타 저작권법에 위반되는 방법으로 이용할 경우, 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

[www.earticle.net](http://www.earticle.net)

# 실시간 손끝 탐지를 위한 VGGNet 기반 객체 탐지 네트워크

VGGNet-based Object Detection Network for Real-time Fingertip Detection

노대철, 김태영<sup>1)</sup>

Dae-Cheol Noh, Tae-Young Kim

(02713) 서울특별시 성북구 서경로 124 서경대학교 컴퓨터공학과  
{sheocjf1025, tykim}@skuniv.ac.kr

## 요약

최근 급속도로 발전된 딥러닝 기술을 적용하여 가상현실 및 증강현실 응용에서 사용자 친화적 인터페이스를 제공하기 위한 연구가 활발히 이루어지고 있다. 본 논문은 사용자의 손을 이용한 인터페이스를 제공하기 위하여 실시간 손끝을 탐지하는 딥러닝 기반 손끝 탐지 방법을 제안한다. 본 방법은 기존 객체 탐지 네트워크에서 필요한 주석 전처리 과정 없이 VGG-19 네트워크에 DenseNet의 연결 방식을 도입하여 총 파라미터 수와 소요 시간을 줄이고 Atrous Convolution과 Grad-CAM을 이용하여 손끝을 탐지한다. 본 방법을 다양한 환경에서 실험한 결과 기존 방법(SSD 네트워크)보다 평균 5% 높은 인식률로 34.4 ms의 실시간 처리가 가능함을 알 수 있었다. 본 연구 결과로 사용자의 손끝을 이용하여 실시간 에어 라이팅을 하는 응용을 제작함으로써 사용자 인터페이스의 활용 가능성을 보였다.

## Abstract

Recently, research is being actively carried out to provide a user-friendly interface in virtual reality and augmented reality applications by applying rapidly developed deep learning technology. This paper proposes a deep learning-based fingertip detection method that detects real-time fingertips in order to provide the interface using the user's hand. This method introduces the DenseNet Connectivity to the VGG-19 network without the required annotation preprocessing process in the existing object detection network, reducing the total number of parameters and the time required, and detecting the fingertips using the Atrous Convolution and the Grad-CAM. As a result of experimenting with this method in various environments, it was found that real-time processing of 34.4 ms is possible with an average recognition rate of 5% higher than the existing method (SSD network). As a result of this study, the application for real-time air-writing using the user's fingertips was developed, showing the usability of the user interface.

1) 교신저자

키워드: 딥러닝, 손끝 탐지, VGGNet, DenseNet, Grad-CAM, Atrous 컨볼루션

Keyword: Deep Learning, Fingertip Detection, VGGNet, DenseNet, Grad-CAM, Atrous Convolution

## 1. 서론

최근 가상현실 및 증강현실 응용에서 보다 자연스러운 사용자 인터페이스를 제공하기 위한 연구가 활발히 진행되고 있다[1-4]. 기존의 물리적인 공간에서는 사용자가 직접 키보드, 마우스, 터치패드와 같은 기계적 장치를 사용하였지만 가상현실과 증강 현실 공간 안에서는 사용자가 좀 더 몰입할 수 있는 직관적인 인터페이스가 필요하다.

사용자 친화적인 인터페이스 중 가장 접근성이 높은 도구는 인간의 손이라고 할 수 있다. 과거에 사용자의 손을 탐지하기 위해 색상 기반의 손 탐지[5], 장치 기반의 손 탐지[6-7] 등의 연구가 활발하게 진행되었지만, 대부분 다양한 배경과 조명 등의 외부 조건에 유연하게 대처하지 못하고, 특정한 장비나 센서에 의존한다는 단점이 있었다. 하지만 최근 고성능의 그래픽 처리 장치를 사용하는 딥러닝 기술이 발전함에 따라 기존 연구의 단점을 해결하면서 사용자의 손 탐지가 가능해졌다.

최근 사용자의 손을 탐지하기 위해 적합한 다양한 객체 탐지 네트워크[8-9]가 공개되었지만, 학습 데이터에 대해 정답 값(Ground Truth)에 맞는 주석(Annotation) 전처리 과정을 거쳐야 하고, 입력 영상에 대하여 Selective Search, Sliding Window, 그리드(Grid)로 분할하여 확률 점수(Probability Score)를 계산하는 등 복잡한 알고리즘을 거쳐야 하는 단점이 있다. 또한 특정 플랫폼에서 제공하는 객체 탐지 네트워크를 사용할 경우, 개발자의 목적에 맞게 개량하기 어렵고 유지보수가 어려운 단점이 있다.

본 논문은 기존 연구의 단점을 개선하면서 가상현실 및 증강현실에서 몰입도 있는 사용자 친화적 인터페이스로 사용 가능한 사용자의 손끝 객체를 탐지하는 방법을 제안한다. 객체 탐지의 소요시간 단축을 위해 구현 및 유지 보수가 쉬운 VGGNet[10]

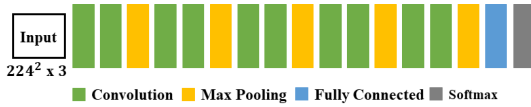
합성곱 신경망에 DenseNet[11] 합성곱 신경망의 연결 방식을 적용하였고 시맨틱 세그먼테이션 분야에서 사용되는 Atrous Convolution[12]과 주로 합성곱 신경망의 시각화(Visualization) 분야에서 사용되는 Grad-CAM[13] 기술을 적용하였다. 입력 손 영상에 대해 본 연구에서 제안한 합성곱 신경망을 수행한 다음 Grad-CAM을 사용하여 찾고자 하는 손끝 객체의 대략적인 부분을 원본 영상으로부터 잘라낸다. 잘라낸 데이터는 다시 합성곱 신경망의 입력 영상 데이터로 사용되어 합성곱 연산과 Atrous Convolution 연산을 거쳐 손끝의 위치를 알 수 있는 특징 맵을 제작한다. 이 특징 맵에 특정한 임계값(Threshold)을 기준으로 이진화한 후, 윤곽선 처리를 통해 최종적인 손끝 객체를 탐지한다. 본 방법은 기존의 정답 값에 맞는 주석 전처리 과정 대신 탐지하고자 하는 손끝 영상 데이터를 학습시켜 손 영역을 구분하였고, 실제로 탐지하고자 하는 객체가 존재하지 않는 위치도 검사하는 기존 객체 탐지 알고리즘과 달리 Grad-CAM을 사용하여 손 영역에 대해서만 처리하여 성능을 높였으며 Atrous Convolution을 통해 정확한 손끝 좌표를 찾도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구로 VGGNet과 DenseNet, Atrous Convolution과 Grad-CAM을 소개한다. 3장에서 본 논문에서 제안하는 손끝 객체 탐지 방법을 설명한다. 4장에서 본 방법의 학습 및 평가를 위해 자체 제작한 데이터 세트 소개 및 다양한 조건에서의 손끝 객체 탐지 결과, 기존 SSD 객체 탐지 네트워크와 성능 비교 결과, 기존 VGG-19와 성능 비교 결과 및 본 방법을 사용한 프로그램을 기술한 후 5장에서 결론을 맺는다.

## 2. 관련 연구

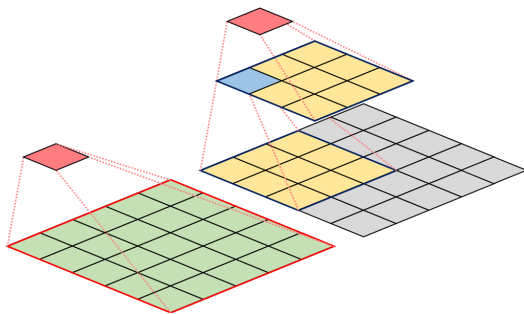
### 2.1 VGGNet과 DenseNet

VGGNet과 DenseNet은 각각 2014년과 2017년에 공개된 분류(Classification) 문제를 해결하기 위한 합성곱 신경망이다. VGGNet은 기존 합성곱 신경망에서 사용한 합성곱 층 - 최대 풀링 층 - 완전 연결 층의 구조를 간단하고 깊게 쌓은 네트워크 구조로, 단순한 구조에 반해 높은 성능을 보여 현재까지도 많이 사용되는 합성곱 신경망 구조이다(그림 1).



(그림 1) 단순한 VGGNet의 구조(10)

VGGNet은 기존의 합성곱 신경망의 합성곱 연산에 사용한 5x5 필터, 7x7 필터와 같이 다양한 크기의 필터 대신 3x3 크기의 필터 한 종류만 사용한다. (그림 2)와 같이 영상 데이터를 대상으로 5x5 크기의 필터로 1단계 합성곱 연산을 한 결과와 3x3 크기의 필터로 2단계에 걸쳐 합성곱 연산을 한 결과는 동일하다는 것을 알 수 있다.



(그림 2) 5x5 필터의 1단계 합성곱 연산 형태와 3x3 필터의 2단계 합성곱 연산 형태

합성곱 연산 시 입력 데이터의 가로를  $R$ , 세로를  $C$ , 깊이를  $Ch$ 로 가정하고, 필터의 가로를  $K_r$ , 세

로를  $K_c$ , 깊이를  $K_{ch}$ 로 가정할 때, 파라미터의 수  $P$ 는 식 1과 같다.

$$P = R * C * Ch * K_r * K_c * K_{ch} \quad (1)$$

식 2에 의하면, 5x5 필터를 사용한 합성곱 연산 1회 결과의 파라미터 수보다 3x3 필터를 사용한 합성곱 연산 2회 결과의 파라미터 수가 더 적음을 알 수 있다.

$$P_1 = R * C * Ch * 5 * 5 * K_{ch}$$

$$P_2 = R * C * Ch * 3 * 3 * K_{ch} + (R-2) * (C-2) * Ch * 3 * 3 * K_{ch}$$

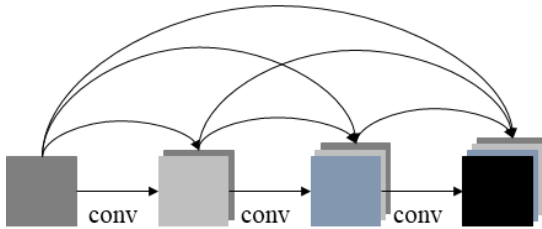
$$\frac{P_2}{P_1} = \frac{R * C * Ch * 3 * 3 * K_{ch}}{R * C * Ch * 5 * 5 * K_{ch}} \quad (2)$$

$$+ \frac{(R-2) * (C-2) * Ch * 3 * 3 * K_{ch}}{R * C * Ch * 5 * 5 * K_{ch}}$$

$$= \frac{9}{25} + \frac{(R-2) * (C-2) * 9}{R * C * 25} < 1$$

$$(R, C \geq 5)$$

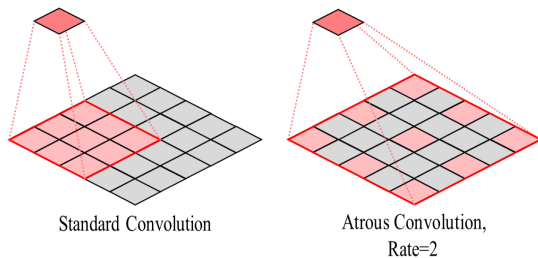
DenseNet은 각 합성곱 층의 출력을 이후 연산의 출력에 덧붙이는 연결(Concatenation) 방식을 사용한 고밀도 연결 구조(Dense Connectivity) 네트워크이다(그림 3). 합성곱 연산은 네트워크를 통과하는 동안 영상 데이터 값을 잃어버리는 단점을 가지고 있는데, DenseNet에서 사용한 연결 방식은 영상 데이터 값을 가공하지 않은 채 깊이 방향으로 덧붙이기 때문에 기존의 ResNet[14]의 합산(Summation) 방식보다 영상 데이터 값을 효율적으로 보존하는 장점이 있다. 또한 DenseNet의 깊은 신경망 구조는 연결 방식으로 인해 특징 맵의 깊이가 커지면서 파라미터 수가 증가하는 단점이 있는데, 각각의 합성곱 층에서 사용하는 필터의 채널 값을 작게 사용함으로써 출력되는 특징 맵의 채널이 폭발적으로 증가하지 않도록 하였다.



(그림 3) DenseNet의 연결 방식

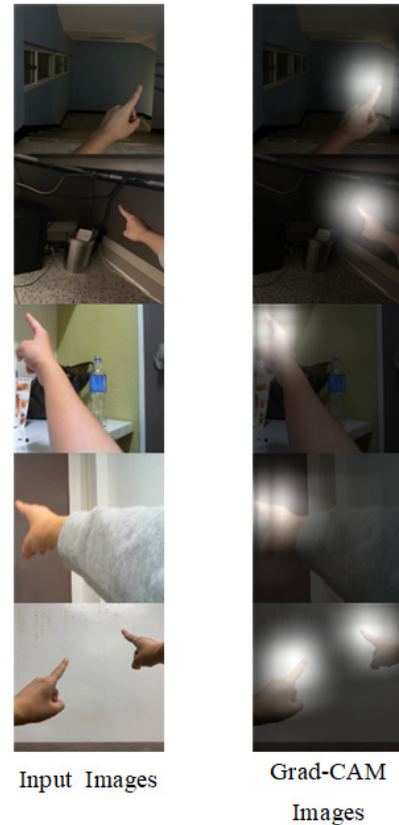
## 2.2 Atrous Convolution과 Grad-CAM

Atrous Convolution은 (그림 4)와 같이 일반적인 합성곱 형태와 동일한 연산량을 가지지만 인접한 픽셀 간에 일정한 공백을 둬으로써 필터가 한 번에 볼 수 있는 수용 영역(Receptive Field)을 확장한 형태의 합성곱이다. Atrous Convolution은 찾고자 하는 객체에 대해 픽셀 단위의 세밀한 부분을 찾아야 하는 시맨틱 세그먼테이션에서 주로 사용하는 합성곱 형태인데, 일반적인 합성곱 연산과 달리 수용 영역이 넓기 때문에 배경과 객체 사이의 차이를 구분하여 객체를 쉽게 찾을 수 있다는 장점이 있다.



(그림 4) 일반적인 합성곱과 Atrous Convolution의 수용 영역 비교[12]

Grad-CAM은 딥러닝의 시각화(Visualization) 분야에서 주로 사용되는 기술이다. Grad-CAM을 사용하면 분류 문제에서 임의의 입력 영상 데이터에 대하여 합성곱 신경망을 거친 후 분류 값이 나오기까지 컴퓨터가 영상 데이터의 어느 부분을 보고 그 분류 값을 도출하였는지 알 수 있는 장점이 있다 (그림 5).



(그림 5) 입력 영상에 대한 Grad-CAM 영상

Grad-CAM은 클래스 활성화 맵(Class Activation Map) [15] 에서 발전된 것으로, 클래스 활성화 맵 기술은 전역 평균 풀링(Global Activation Pooling) 층이 반드시 존재해야 한다는 단점이 있었다. 이 단점은 전역 평균 풀링 층이 없는 네트워크에는 사용할 수 없기 때문에 네트워크에 전역 평균 풀링 층을 추가하여 재학습(Retraining)을 시켜야 하는 문제가 있었다. 또한, 전역 평균 풀링 층 이전의 특징 맵만을 추출하여 사용할 수 있는 단점이 존재했다. 합성곱 신경망의 학습 방법은 임의의 값의 가중치(Weight)를 시작으로 순방향으로 합성곱 연산을 진행한 후, 출력 층의 손실 함수(Loss Function)에서 오차를 계산하여 역전파(Backpropagation) 알고리즘을 사용하여 역방향으로 가중치를 갱신하는 방법을 사용한다. 위의 과정은 오차가 더 이상 줄어들지 않을 때까지 반복 진행한다. 갱신되는 가중치

의 값은 현재 가중치 값에서 가중치의 기울기(Gradient) 값을 뺄으로써 구할 수 있는데, 기울기가 0에 가까워질수록 네트워크가 객체 분류에 있어서 높은 성능을 보여준다는 것을 의미한다. 클래스 활성화 맵은 합성곱 신경망에 실제로 존재하는 전역 평균 풀링 층에 의존적이지만, Grad-CAM은 대부분의 합성곱 신경망이 가지는 기울기 값을 사용하여 전역 평균 풀링 방식으로 연산한다. 이로 인해 합성곱 신경망에 반드시 전역 평균 풀링 층이 존재해야 할 필요가 없고, 모든 합성곱 층의 특징 맵을 시각화할 수 있다.

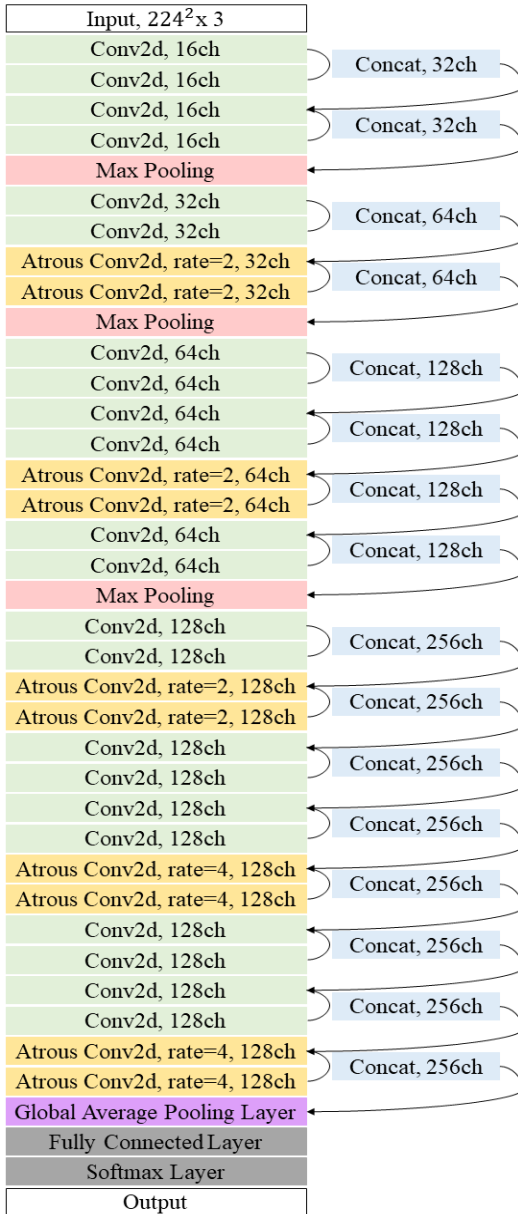
### 3. 손끝 객체 탐지 방법

대중적으로 사용되는 Fast-RCNN, YOLO 등의 객체 탐지 네트워크는 학습을 위한 데이터 준비 과정에서 모든 학습 데이터의 정답 값(Ground Truth)에 대해 주석(Annotation) 전처리 과정이 필요하다. 컴퓨터가 학습하고자 하는 객체의 위치를 개발자가 직접 입력해주어야 하므로 학습을 위한 입력 데이터가 많은 경우는 이에 따른 작업시간이 많이 소요된다. 또한, 기존의 객체 탐지 네트워크는 입력 데이터에 대해 불필요한 부분까지 모두 검사해야 한다. 예를 들어, YOLO 네트워크의 경우는 입력 데이터를 일정 크기의 격자(Grid)로 나눈 후, 각각의 모든 격자에 대해 탐지하고자 하는 객체가 존재하는지 검사를 한다. 이러한 방식은 실제 탐지하고자 하는 객체가 아주 작은 영역을 차지하는 경우, 객체 탐지를 위해 실시하는 대부분의 검사가 불필요한 과정이 된다. 또한, 특정 플랫폼에서 제공하는 객체 탐지 네트워크를 사용할 경우, 개발자의 목적에 맞게 개량하기 힘들고 유지보수가 어려운 단점을 가진다.

이러한 단점을 개선하기 위해 본 논문은 VGG-19 구조에 DenseNet의 연결 방식, Atrous Convolution 그리고 Grad-CAM을 사용한 네트워크로 손끝 객체를 탐지하는 방법을 제안한다. 본 논문에서 제안하는 방법은 영상 데이터의 모든 부분을 검사할 필

요 없이 Grad-CAM을 사용하여 탐지하고자 하는 객체의 대략적인 위치를 찾아낸 후, 그 위치의 영상 데이터에 대해서만 합성곱 신경망으로 추출한 특징 맵을 사용한다. 특징 맵을 추출할 때는 Atrous Convolution 연산으로 출력된 특징 맵을 사용하는데, 일반적인 합성곱 연산과 달리 Atrous Convolution은 배경과 보고자 하는 객체의 차이점을 찾기에 적합한 합성곱 형태이다. 본 논문에서 제안하는 방법은 컴퓨터에게 정답 값을 가르쳐주지 않아도 탐지하고자 하는 객체의 분류 값(Label)만 알려주면 스스로 찾기 때문에 학습 데이터의 주석 전처리 과정을 손끝 영상 학습만으로 해결할 수 있고, 입력 데이터의 모든 부분에 대해 객체의 존재 여부를 검사할 필요가 없다. 또한, VGGNet 기본구조에 DenseNet 연결 방식을 도입하여 합성곱 연산으로 소실되는 영상 데이터의 값을 최대한 보존함으로써 더욱 효과적인 분류 및 객체 탐지가 가능하도록 설계하였다.

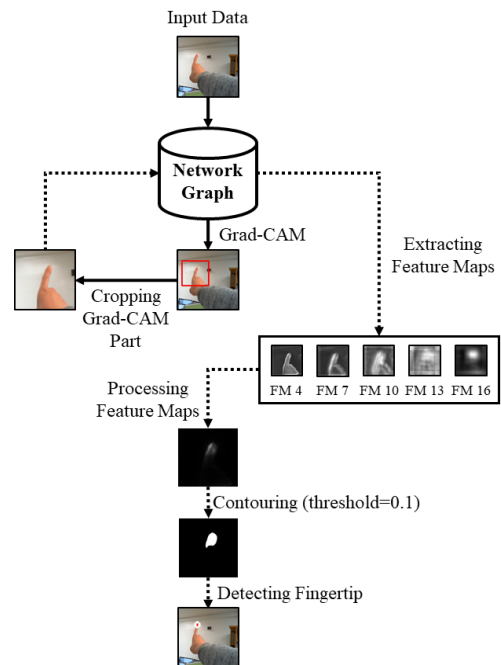
손끝 객체 탐지를 위한 입력 데이터는 USB 카메라에서 초당 60프레임의 속도로 촬영한 사용자의 손끝 영상을 사용한다. 입력 데이터는 네트워크 구조에 맞게 224x224 크기로 재조정(Resize)하여 본 연구에서 자체적으로 제작한 VGG-19(그림 6)의 입력으로 사용한다. 새롭게 제작한 VGG-19는 기존의 VGG-19의 각 합성곱 층에서 사용한 필터의 채널 크기의 절반만을 사용하여 전체적인 합성곱 신경망의 크기를 줄였다. 기존 VGG-19의 한 개 합성곱 층을 두 개로 나누어 합성곱 연산 후, 이 두 합성곱 층의 출력 특징 맵들을 연결 방식으로 합친다. 또한, 정확한 손끝 탐지를 위해 특정 위치의 합성곱 층은 일반적인 합성곱 연산 대신 Atrous Convolution을 사용하여 효과적으로 손끝 객체의 특징을 찾도록 하였다. 그리고 최대 풀링(Max Pooling) 층을 3개만 둬으로써 최종 특징 맵의 해상도 축소에 따른 영상 데이터 손실을 최소화하였다. 마지막 합성곱 연산 이후 28x28 필터를 사용하여 전역 평균 풀링을 수행하고, 완전 연결 층을 거쳐 소프트맥스 함수를 통해 인식 결과를 출력한다.



(그림 6) 연결 방식을 추가한 VGG-19 네트워크의 구조

본 논문에서 제안하는 손끝 객체 인식 방법은 (그림 7)과 같다. 손 영상을 입력 데이터로 하여 합성곱 신경망을 거쳐 출력된 인식 결과와 소프트맥스 함수 이전의 완전 결합층의 출력 값, 보고자 하는 특징 맵, 역전파 법을 통한 가중치를 바탕으로 Grad-CAM을 계산하여 탐지하고자 하는 손끝 객

체의 대략적인 위치를 찾는다. 원본 영상 데이터에서 Grad-CAM으로 찾은 손끝 객체의 대략적인 위치에 대한 영상 데이터를 추출하여 합성곱 신경망을 거친다. 이 과정 중에 Atrous Convolution으로 연산한 4번째, 7번째, 10번째, 13번째, 16번째의 특징 맵을 추출한다. 해당 위치의 특징 맵들은 서로 다른 해상도를 가지기 때문에 224x224 크기로 이진 선형 보간(Bilinear Interpolation)을 거친 후, 손끝 객체 부분은 높은 값을 가지는 특징을 이용하여 해당 특징 맵들을 서로 곱해줌으로써 분류 범주에 대해 중요도가 높은 손끝 객체 부분만 남긴다. 이후 특정한 임계값을 바탕으로 이진화한 다음 윤곽(Contouring) 처리를 해줌으로써 최종적으로 손끝 객체를 탐지한다.



(그림 7) 손끝 객체 인식 방법

#### 4. 실험

본 실험은 프로세서 Intel Core i5 7500, 그래픽 카드 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 영상을 사용하였다. 분류 범주는 각각 손끝 객체와 연관성이 전혀 없는 영상(Nothing), 손끝 객체 영상(Pointing), 그리고 누적된 평가 기록을 초기화시키는 정적 제스처 영상(Spread Palm)으로 구성된다. 각 분류 범주마다 학습 데이터 8,000장과 평가 데이터 800장을 촬영 및 수집한 후, 잘라내기(Cropping) 기법을 사용하여 5배 증대시켜 각각 40,000장과 4,000장을 제작하였다<표 1>.

<표 1> 학습 데이터 세트의 예시

Label	Examples
Nothing	
Pointing	
Spread Palm	

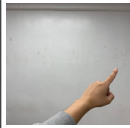
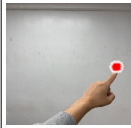


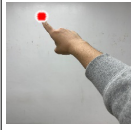
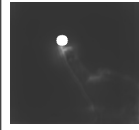

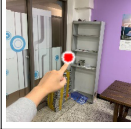
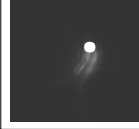


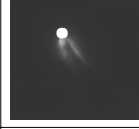
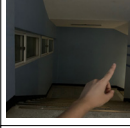
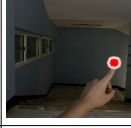
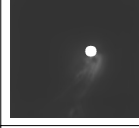


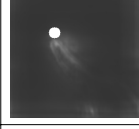

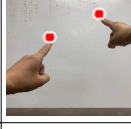
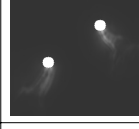

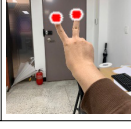

#### 4.2 실험 결과

본 논문의 성능 검증을 위해 SSD 객체 탐지 네트워크[16]와 동일한 평가 데이터 세트를 사용하여 인식률과 속도를 비교한다. 또한 기존 VGG-19 네트워크를 사용한 모델과 파라미터 수를 비교한다.

##### 4.2.1 본 연구의 손끝 객체 탐지 결과

본 논문에서 제안한 방법에 대하여 각각 일반적인 경우(Normal), 복잡한 배경(Complex Background), 어두운 배경(Dark Background), 양 손끝(Two Hands), 한 손에서 두 손가락 끝(Two Fingertips) 등 다양한 조건에서 평가한 결과, <표 2>와 같이 모두 성공적으로 탐지함을 알 수 있었다.

<표 2> 여러 조건에서의 손끝 객체 탐지

Condition	Original image	Result image	Feature map image
Normal (Left)			
Normal (Right)			
Complex Background (Left)			
Complex Background (Right)			
Dark Background (Left)			
Dark Background (Right)			
Two Hands			
Two Fingertips			



#### 4.2.2 SSD 객체 탐지 네트워크와 비교

다양한 장소, 배경, 조명 등을 기준으로 USB 웹캠으로 촬영한 100장의 손끝 평가 데이터 세트(그림 8)에 대해 기존의 SSD 객체 탐지 네트워크와 성능 비교를 하였다. <표 3>에서 보는 바와 같이 본 논문에서 제안한 방법은 속도상의 차이가 거의 없으면서 인식률이 SSD 네트워크보다 평균 5% 더 높음을 알 수 있었다.



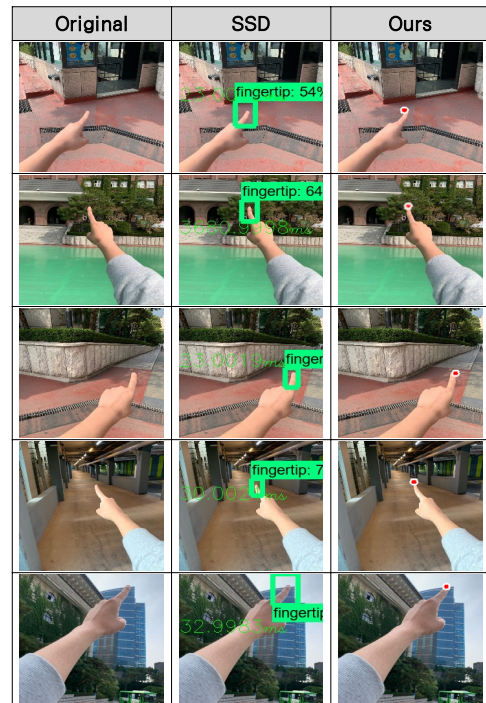
(그림 8) SSD 객체 탐지 네트워크와 비교를 위한 100장의 평가 데이터 예시

<표 3> 본 방법과 SSD 네트워크의 성능 비교

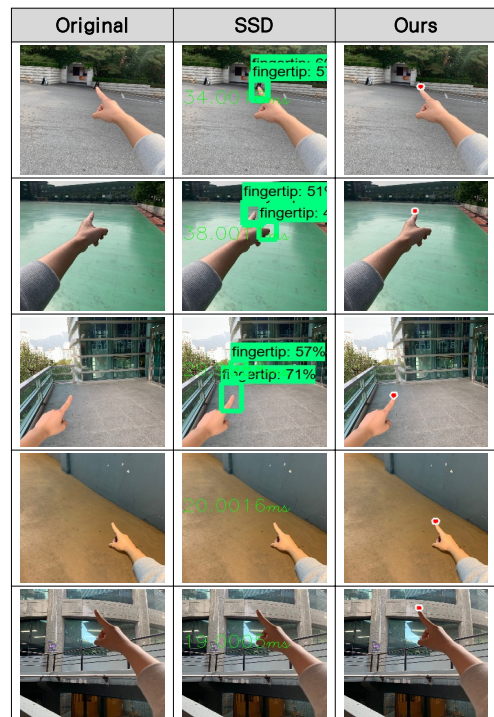
Network	Recognition rate	Time per Image
Ours	84%	31.34ms
SSD	79%	34.36ms

<표 4>는 SSD 네트워크와 본 방법 모두 정상적으로 손끝 객체를 탐지한 사례이다. <표 5>와 <표 6>은 각각 SSD 네트워크의 탐지 오류 사례와 본 방법의 탐지 오류 사례이다. 비교 결과, 두 네트워크 모두 대부분의 손끝 영상을 정상적으로 탐지하였지만, SSD 네트워크는 4개의 영상 데이터에 대해 미탐지 현상을 보였고, 17개의 영상 데이터에 대해 다중 탐지 현상을 보였다. 반면, 본 연구에서 사용한 방법은 5개의 영상 데이터에 대해 미탐지 현상을 보였고, 3개의 영상 데이터에 대해 다중 탐지 현상을 보였으며, 8개의 영상 데이터에 대해 잘못된 위치 탐지 현상을 보였다(<표 7>).

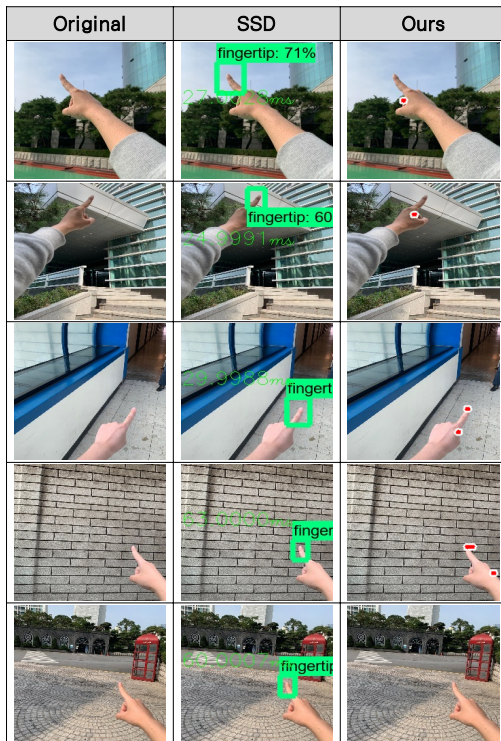
<표 4> SSD 네트워크와 본 방법의 정상 탐지 사례



<표 5> SSD 네트워크의 탐지 오류 사례



〈표 6〉 본 방법의 탐지 오류 사례



〈표 7〉 본 방법과 SSD 네트워크의 성능 차이

Network	Undetection	Multiple detection	False detection
Ours	5	3	8
SSD	4	17	0

#### 4.2.3 기존 VGG-19와 파라미터 수 및 속도 비교

본 연구에서는 기존 VGG-19 네트워크와 달리 DenseNet의 연결방식을 적용한 VGGNet 네트워크를 사용하였다. 〈표 8〉에서 보는 바와 같이 본 연구에서 제안한 VGGNet 네트워크는 기존 VGG-19 네트워크의 25% 수준의 파라미터 수를 가지고 있음을 알 수 있다. 두 네트워크의 파라미터 수 차이로 인한 영상 데이터 1장에 대한 처리 속도를 비교한 결과 본 방법은 평균 34.36ms의 시간을 소요했지만, 기존 방법은 본 방법의 약 1.49배 수준인 51.16ms의 시간을 소요하였다〈표 9〉.

〈표 8〉 총 파라미터 수의 비교

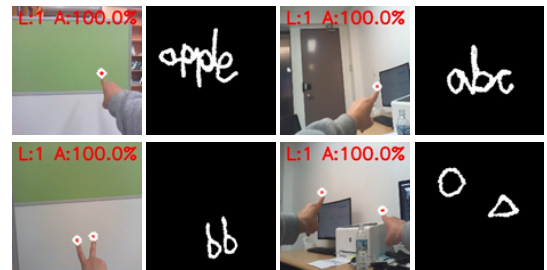
Network	Number of parameters
Ours	4,992,712,704
Common	19,508,428,800

〈표 9〉 영상 데이터 처리 속도의 비교

Network	Time per Image
Ours	34.36ms
Common	51.16ms

#### 4.3. 응용 사례

본 논문의 손끝 객체 탐지의 성능을 확인하기 위하여 허공에 글씨를 쓰는 Air Writing 프로그램을 제작하였다(그림 9). 실험 결과 본 논문에서 소개한 개선된 VGG-19 네트워크와 Grad-CAM, Atrous Convolution을 사용하여 평균 34ms의 속도로 자유롭게 문자 작성이 가능함을 알 수 있었다.



(그림 9) 여러 조건에서의 Air Writing 출력

#### 5. 결론

본 논문은 VGG-19 네트워크에 DenseNet의 연결 방식을 적용하여 개선한 합성곱 신경망에 Grad-CAM과 Atrous Convolution을 사용하여 사용자의 손끝을 탐지하는 방법을 제안하였다. 실험결과 본 방법은 기존의 SSD 객체 탐지 네트워크보다 5% 높은 인식률로 약 34ms의 처리 속도를 가져 실시간으로 손끝 탐지가 가능함을 알 수 있었다. 본 방법을 바탕으로 사용자의 손끝을 탐지하여 허공에 글씨를 쓰는 Air Writing 응용 프로그램을 구현한 결과 지연 시간 없이 실시간으로 문자 작성이

가능하였다. 위의 결과를 바탕으로 가상현실 혹은 증강현실에서 사용자의 손끝 객체를 탐지하여 사용자 친화적 인터페이스로 사용될 수 있다.

## ■ 감사의 글

이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임.(No. NRF-2017R1D1A1B03029834)

## ■ 참고문헌

- [1] 김현덕, 이상훈, 손명규, "실시간 머리 방향 인식 및 손 동작 인식 기반 NUI 프레임워크," 한국정보과학회 학술발표논문집, pp.1197-1199, 2015. 12.
- [2] 이새봄, 정일홍, "키넥트를 사용한 NUI 설계 및 구현," 한국디지털콘텐츠학회 논문지, 15(4), pp.473-480, 2014.08.
- [3] 오동한, 이병희, 김태영, "외부 환경에 강인한 딥러닝 기반 손 제스처 인식," 한국차세대컴퓨팅학회 논문지, 14(5), pp.31-39, 2018.10.
- [4] 이호준, 하규태, 이상호, 차재광, 김시호, "VREscape-Sim : 사용자 스스로 체득하는 가상현실 지하철 재난 탈출 기능성 게임," 한국차세대컴퓨팅학회 논문지, 12(4), pp.125-132, 2016.
- [5] 박양재, 강성관, "사용자 상호작용을 위한 색상 기반 손과 손가락 탐지 기술," 한국정보기술학회 논문지, 8(2), pp.51-58, 2010.02.
- [6] 김희애, 이창우, "키넥트를 이용한 손 영역 검출의 정확도 개선," 한국정보통신학회 논문지, 18(11), pp.2727-2732, 2014.11.
- [7] 김은아, 안인희, 김숙진, "프레젠테이션을 위한 NUI 기반 웨어러블 컴퓨터 디자인 연구," 한국차세대컴퓨팅학회 논문지, 12(6), pp.92-103, 2016.
- [8] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
- [9] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," In NIPS, pp.91-99, 2015.
- [10] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," In ICLR, pp.1-14, 2014.
- [11] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, "Densely Connected Convolutional Networks," In CVPR, pp.4700-4708, 2017.
- [12] 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.
- [13] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, "Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization," In ICCV, pp.618-626, 2017.
- [14] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," In CVPR, pp.770-778, 2016.
- [15] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, "Learning Deep Features for Discriminative Localization," In CVPR, pp.2921-2929, 2016.
- [16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," In ECCV, pp.21-37, 2016.

## ■ 저자소개

### ◆ 노대철



- 2014년 3월~현재 서경대학교 컴퓨터 공학과 학사 재학
- 관심 분야: 가상 현실, 게임 프로그래밍, 컴퓨터 비전, 머신 러닝

### ◆ 김태영



- 1991년 2월 이화여자대학교 전자계산학과 학사
- 1993년 2월 이화여자대학교 전자계산학과 석사
- 1993년 3월~2002년 2월 한국통신 멀티미디어연구소 선임연구원
- 2001년 8월 서울대학교 전기컴퓨터공학부 박사
- 2002년 3월~현재 서경대학교 컴퓨터 공학과 교수
- 관심 분야: 실시간 렌더링, 증강 현실, 영상처리, 모바일 3D