https://doi.org/10.7840/kics.2023.48.9.1135

# 딥러닝을 이용한 국부 영역 기반 손 분할

전 준 현◆, 김 태 훈*, 정 유 수**, 박 길 흠°

# Local Region-Based Hand Segmentation Using Deep Learning

Junhyun Jeon◆, Tae-Hun Kim*, Yoosoo Jeong**, Kil-Houm Park°

요 약

디지털 이미지 프로세싱의 분할(Segmentation) 기법을 통해 객체를 분할하는 것은 복잡한 배경에서 높은 정확도를 보이기 어렵다. 본 논문에서는 복잡한 환경에서 객체를 분할함에 있어 전처리 과정으로 딥러닝 객체 인식 기법을 통해 이미지의 관심 영역(ROI: Region of Interest)을 추출하고 지정한 뒤, 딥러닝 분할 기법을 사용하고, 후처리 과정에 오츠 이진화(Otsu's Binarization)를 이용해 임계 처리(Thresholding)를 적용 하는 방법을 제안한다. 제안한 방법은 YOLOv4 모델을 활용한 딥러닝 기반의 객체 인식 알고리즘을 사용하여 복잡한 이미지 속 손의 위치를 검출하고 ROI를 확장하여 U-Net 모델을 활용한 딥러닝 기반의 분할 기법이 전체 이미지에서 적용되는 것이 아닌 국부적으로 적용될 수 있도록 하고 오츠 이진화를 통해 임계처리 하는 것이다. 또한 관심 영역의 적용 여부에 따른 U-Net 분할의 효율성을 확인하고, 복잡한 배경을 가진 손 이미지에 제안하는 알고리즘을 적용한 뒤 정답과 결과 마스크의 IoU(Intersection over Union) 수치를 측정하여 제안하는 알고리즘의 유효성을 검증하였다.

**키워드 :** 손, 딥러닝, 객체 인식, 객체 분할
**Key Words :** Hand, Deep Learning, Object Detection, Object Segmentation

ABSTRACT

Segmenting objects through the Digital Image Processing makes it difficult to draw high accuracy in a complex background. In this paper, we propose a method of using deep-learning segmentation to segment objects in complex environments, providing ROI(Region of Interest) of images as a pre-processing, and Thresholding in the post-processing. The proposed method is to detect the position of the hand in a complex image using a deep learning-based object recognition algorithm employing the YOLOv4 model; to expand the ROI so that the deep learning-based segmentation techniques using the U-Net model can be applied locally, not on the entire image; and to process by Thresholding through Otsu's Binarization method. We applied the proposed algorithm to hand images with complex backgrounds and verified the effectiveness of the algorithm by measuring the IoU values of the masks of correct answers and results.

◆ First Author : Department of Electronic and Electrical Engineering, Kyungpook National University, danielj9203@gmail.com, 정회원
° Corresponding Author : Department of Electronic and Electrical Engineering, Kyungpook National University, khpark@ee.knu.ac.kr, 종신회원
* DIPVISION, dipvision.ceo@gmail.com
** Electronics and Telecommunications Research Institute(ETRI), yoosoojeong@etri.re.kr
논문번호 : 202304-067-C-RE, Received April 3, 2023; Revised May 30, 2023; Accepted June 8, 2023

## Ⅰ. 서 론

최근 이미지 속 객체를 검출, 추적, 인식하는 기법이 활발히 연구되고 있으며, 특히 이미지 속에서 객체의 정확한 구분과 위치를 찾는 것에 대한 연구는 지금도 계속해서 발전하고 있다. 또한 YOLO와 같이 딥러닝을 활용한 연구를 통해 여러 가지 객체 인식하는 방식은 높은 수준에 있다[1]. 더욱이 여러 가지 방식의 이미지 분석 및 처리 기술이 발전함과 동시에 계산 속도 또한 많이 향상 되었다[2].

객체의 인식을 위한 기법 중 이미지 분할(Segmentation)은 이미지 속 각 픽셀의 특징들에 기초하여, 색상이나 모양을 기준으로 영역을 구분하거나 유의미한 정보를 바탕으로 전경과 배경을 구분하는 등의 처리를 이야기한다. 이와 관련된 연구는 무수하게 많으나 어느 이미지에나 적용하여도 될 정도로 완벽한 알고리즘은 존재하지 않기에 이미지 픽셀들을 분할하거나 점 집합 사이의 관계성을 파악하는 등 다양한 분야에서 객체를 분할하기 위한 연구가 진행되고 있다[3]. 요즘과 같이 사람이 직접 착용하는 웨어러블 기술이 빠르게 성장하고 있는 추세에 컴퓨터가 인간의 손을 이해하게 된다는 것은 인간과 컴퓨터의 상호작용, 손 동작과 수화 인식 그리고 VR(Virtual Reality)과 AR(Augmented Reality) 등 많은 분야에 강점을 가진다. 결론적으로 손의 검출과 분할은 손의 자세(모양) 추정이나 손 동작 인식[4]의 초석이 된다.

이미지에서 손 분할은 색상 기반의 HSV 분할[5]과 회색 계열(Grayscale) 이미지의 픽셀 값을 통해 가정한 지형도를 활용하여 경계를 찾는 Watershed 분할[6], 이미지에서 픽셀들을 확인하여 전경과 배경으로 분할하는 Grab Cut 분할[7] 등 다양한 방식을 적용할 수 있다. 하지만 복잡한 환경에서의 손 분할은 광범위한 색 공간, 피부색과 질감의 차이, 배경의 노이즈, 그림자, 객체의 속도 등 갖가지 요인들에 의해 매우 어려운 상황이다. 또한 VR 및 AR 등 많은 분야에서 다양한 활용으로 이어지기 위해 손을 분할하는 것이 아닌 손을 포함한 팔 전체를 분할하고자 하였을 때, 옷 소매나, 악세사리 등이 같이 분할되지 않는 불완전한 결과를 보였다.

따라서 본 논문에서는 손을 포함한 전체 팔을 분할하는 것을 목표로 하고 복잡한 환경에서 손 분할의 정확도 향상을 위하여 딥러닝 분할을 적용하고, 분할 알고리즘의 입력값으로 이용될 이미지 속 객체의 질적인 향상을 위하여 관심 영역(ROI: Region of Interest)을 지정하여 객체에 대해 국부적으로 분할 기법을 적용할 수 있는 알고리즘을 제안한다. 본 논문에서는 분할의 입력값으로 이용될 이미지의 손 위치를 딥러닝 객체 인식 모델인 YOLOv4[8]를 통해 검출하여 이미지 전체가 아닌 국부적인 입력이 되도록 하였고, 딥러닝 분할 모델인 U-Net[9]을 이용하여 손을 포함한 전체 팔과 함께 옷 소매나 악세사리가 포함되도록 하고, 실험을 통하여 제안한 알고리즘의 유효성을 검증하였다.

본 논문의 구성은 다음과 같다. Ⅲ장에서 딥러닝 U-Net 분할과 전체 이미지가 아닌 국부적인 입력값의 유효성에 대해 설명하고, Ⅳ장에서는 실험을 통해 결과 수치를 제시하여 제안 알고리즘의 성능을 평가하고, 마지막으로 Ⅴ장에서 결론을 맺고 추후 개선될 사항에 대해 언급한다.

## Ⅱ. 딥러닝을 이용한 국부 영역 기반 손 분할

### 2.1 HSV 손 분할

색을 이용한 손 분할에서 객체의 Hue 값을 기준으로 분할 해야하기 때문에 기존의 RGB 기반의 이미지를 HSV로 변환해야 한다. HSV는 색상(Hue), 채도(Saturation), 명도(Value)를 말하며, Red, Green, Blue 세 가지 속성을 모두 참고 해야하는 RGB 기반의 이미지에 비해 더욱 다양하고 순수한 색 정보를 가지고 있다. 색상 성분은 붉은 계열 영역의 이어지는 색상 영역들과 채도를 위해 측정된 방사상의 거리(Radial distance)를 정의한다. 채도가 낮을 때에는 회색 계열의 값들로 색상의 강도를 추정할 수 있고, 채도가 높을 때에는 색상으로 근사할 수 있다[5].

그림 1은 HSV 분할에 대한 실험 결과이다. 이와 같이 HSV 분할은 복잡한 환경, 특히나 비슷한 색감을 가진 환경에서 매우 취약하다. 그림 1(b), (c)에서 볼

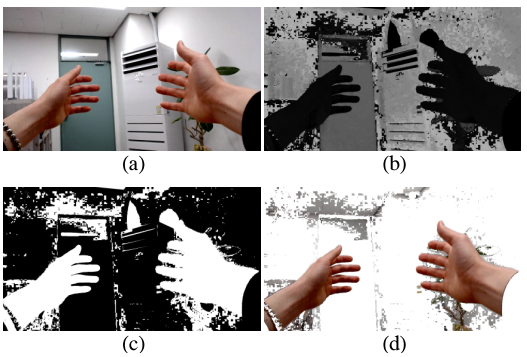



그림 1. HSV 분할 결과: (a) 원본 이미지, (b) HSV 변환 후 Hue 이미지, (c) HSV 분할 마스크, (d) 원본 이미지에서 마스크 부분 추출

Fig. 1. HSV segmentation result: (a) original image, (b) hue image after HSV transformation, (c) HSV segmentation mask, (d) Mask extraction from original image

수 있듯 손과 비슷한 Hue 값을 가진 요소들이 같이 분할 되었다. 또한, 손을 포함한 팔 전체의 분할에 대해서는 옷의 소매나 팔에 착용하는 악세사리가 포함되지 않고 분할되는 것을 볼 수 있다.

### 2.2 GrabCut 손 분할

색상을 기반으로 한 손 분할의 취약점을 보완하고자 GrabCut을 적용하였다. GrabCut은 Graph Cut을 바탕으로 한 영역 기반(Region-Based) 이미지 분할 기법이다. GrabCut의 기본 원리는 이미지의 픽셀을 노드로 생각하고, GMM(Gaussian Mixture Model)이 생성한 픽셀 분포에 대한 그래프가 작성되며, 픽셀 간의 엣지 정보나 유사도를 따져가며 각 픽셀을 새롭게 생성된 소스 노드나 싱크 노드에 추가하고 min cut 알고리즘을 통해 그래프를 분할 하여 픽셀들이 전경(Foreground)과 배경(Background), 두 개의 집합으로 분할되는 최적의 컷을 찾는 것이다[7].

그림 2는 전체적인 GrabCut 알고리즘에 대한 실험

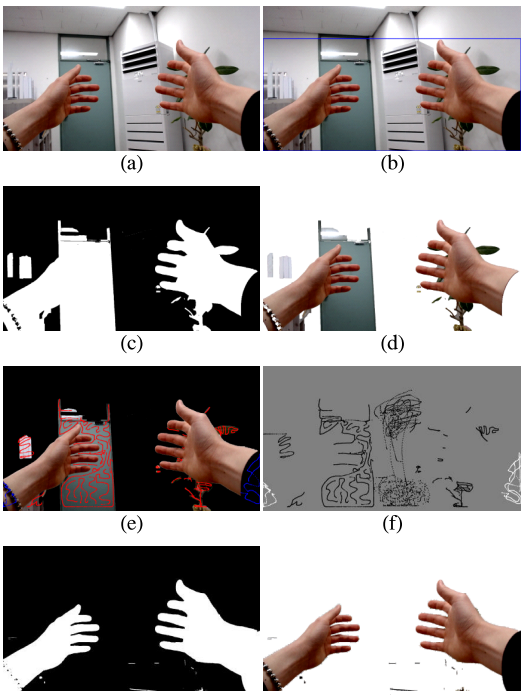



그림 2. GrabCut 결과: (a) 원본 이미지, (b) 전경 설정, (c) GrabCut 마스크, (d) 원본 이미지에서 마스크 부분 추출, (e) 전경 배경 마킹, (f) 마킹 반복 결과, (g) 마킹 후 GrabCut 마스크, (h) 마킹 후 원본 이미지에서 마스크 부분 추출
Fig. 2. GrabCut result: (a) original image, (b) setting foreground, (c) GrabCut mask, (d) mask extraction from original image, (e) marking foreground and background, (f) repetition marking, (g) GrabCut mask after marking, (h) mask extraction from original image after marking

결과이다. 그림 2(a)는 원본 이미지, (b)는 GrabCut 알고리즘 적용을 위한 사용자가 지정한 전경이다. 그림 2(c)는 GrabCut의 결과로 마스크화 했고, 그림 2(d)는 원본 이미지에서 GrabCut 결과 마스크 부분을 추출한 결과이다. 그림 2(e)는 사용자가 전경과 배경에 대한 마킹을 한 것이고 (f)는 반복적인 마킹을 통한 결과 마스크이다. 그림 2(g), (h)는 최종 결과 마스크와 원본 이미지에 대한 마스크 부분 추출이다.

이와 같이 GrabCut은 객체의 색상 이상의 관점에서 분할이 적용 되었다. 하지만 그림 2(c), (d)와 같이 전경과 배경의 모호한 결과를 보이며, 일부 영역에 대해서는 객체를 특정하지 못하는 것을 볼 수 있다. 이에 더 나은 결과를 위해 그림 2(e)와 같이 사용자가 직접 전경과 배경에 대한 마킹을 반복적으로 추가하여 (f)와 같은 마스크가 형성되고, 그림 2(g), (h)와 같은 결과를 얻게 된다.

최종적으로 얻게 되는 결과는 좋지만, 사용자가 직접 전경과 배경에 대한 반복적인 마킹을 해야하는 불편함을 감수 해야한다.

### 2.3 제안 방식

기존 방식의 분할 기법으로 복잡한 환경의 손 분할을 하였을 때 색상, 엣지, 수동 마킹 등 여러 가지 한계점을 보였다. 또한 손을 포함한 전체 팔까지 분할하고자 하였을 때, 옷 소매나, 악세사리 등이 같이 분할되지 않는 불완전한 결과를 보였다. 이에 본 논문에서는 복잡한 배경을 가진 이미지 속 손 분할 방식으로 국부 영역 딥러닝 분할을 제안한다.

그림 3은 제안하는 알고리즘의 블록도이다. 복잡한 배경을 가진 이미지를 입력으로 받고, 이미지의 관심 영역을 자동적으로 제공하고자 YOLOv4를 이용한 손 객체 인식(Object Detection) 기법을 사용한다. 그 후에 앞선 기존 분할 기법들을 보완하기 위한 U-Net 딥러닝 분할 기법을 적용하고 분할의 결과에 오츠 이진화를 이용해 임계 처리를 하여 결과 마스크를 만든다.

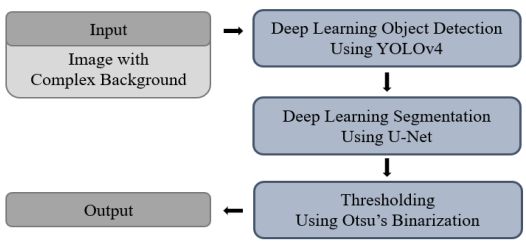



그림 3. 제안하는 알고리즘의 블록도
Fig. 3. Block diagram of proposed algorithm

2.3.1 YOLOv4를 이용한 관심 영역 검출

본 논문에서는 복잡한 배경을 가진 이미지의 손 분할이 좀 더 효과적이고 높은 정확도를 보이도록 개선하고자 전처리 단계에 관심 영역(ROI) 제공을 제안 한다.

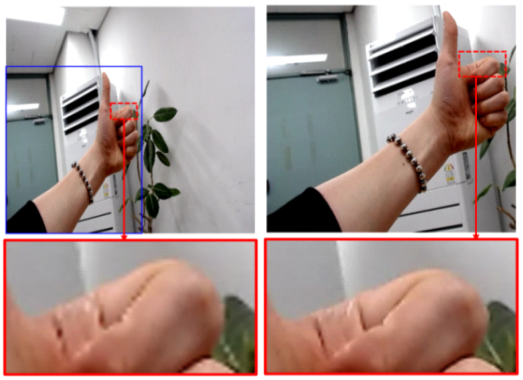
그림 4(a)와 같이 손 분할을 방해할 요인들이 많은 복잡한 배경의 이미지를 분할 하는 것 보다 그림 4(b)와 같이 손의 위치를 특정하고 분할 결과에 영향을 미칠 수 있는 복잡한 배경을 근본적으로 줄인 이미지를 이용하여 분할의 정확도를 올린다.

또한, 손 분할을 하기 위해 딥러닝 모델을 사용한다면, 입력으로 원본 이미지가 아닌 네트워크 크기로 리사이징 후 사용된다. 그림 5는 이미지 리사이징에 대한 해상도 비교이다. 그림 5(a) 객체의 엣지 부분이 (b) 객체의 엣지 부분보다 더 훼손된 이미지임을 확인할 수 있다. 이와 같이 관심 영역을 제공한다면 분할 단계에서 각 픽셀들이 가진 특징들이 보다 원본에 가깝게 되므로 높은 질의 입력 이미지를 이용한 분할이 가능하다.

하지만 관심 영역을 수동적으로 설정하여 분할을 한다면 유효한 결과가 나오더라도 다양한 분야에 활용하기엔 어려움이 있다. 이에 딥러닝 기술을 이용하여 객체 인식을 한 뒤 그 결과를 활용하여 관심 영역을 자동적으로 설정하고자 한다.

(a) (b)

그림 4. 관심 영역 제공을 통한 복잡한 배경 제거
Fig. 4. Removing complex background by providing ROI



(a) (b)

그림 5. 리사이징 해상도 비교: (a) 2MP의 원본 이미지를 512×512로 리사이징한 결과, (b) 원본 이미지에서 객체 부분을 잘라낸 후 512×512로 리사이징한 결과
Fig. 5. Comparison resizing resolution: (a) 512×512 resizing result of 2MP original image, (b) 512×512 resizing result of cropped object image from original image

본 논문에서는 U-Net Segmentation과 함께 딥러닝 기반 객체 인식 기법인 YOLOv4 모델을 관심 영역을 제공하기 위한 전처리 단계로 활용하고자 한다. YOLOv4는 YOLOv3[10] 이후에 나왔으며, 새로운 방법론을 제시하지 않고 기존의 다양한 방법들을 적용해 Single GPU 환경에 최적화 시키고, 딥러닝의 정확도를 개선하고 나아가 전체적인 성능을 극대화 하였다.

그림 6과 같이 YOLOv4는 BOF(Bag of Freebies)와 BOS(Bag of Specials) 방법론들을 Ablation Study를 통해 가장 좋은 성능의 방법들을 적용하고, YOLOv3에서 Backbone과 Neck에 변형을 준 것이다.

Backbone은 기능을 추출하는 역할을 담당하며, 분류 모델이다. YOLOv4의 Backbone으로 CNN (Convolutional Neural Network)의 성능을 향상 시키는 CSPDarknet53(Cross Partial Network)이 사용되었다. CSPDarknet53의 활성화 함수(Activation Function)은 Mish[11]가 사용되었다. Mish 함수의 출력 값 범위는 $[-0.31, \infty]$로 음의 값을 0으로 만들어 정보가 손실되는 ReLu 함수와 달리 Mish 함수는 작은 음의 값을 허용하여 그레디언트가 더 잘 흐르게 한다.

Neck은 정확한 추론을 위한 넓은 범위의 특징을 추출하기 위해 Backbone에서 사용된다. 즉, Backbone의 여러 단계에서 특징 맵(Feature Map) 수집을 담당한다. YOLOv4의 Neck에 적용된 SPP(Spatial Pyramid Pooling)는 Backbone을 통해 추출된 특징 맵을 고정 크기로 Pooling하는 역할을 맡아 이미지 픽셀 간의 정보(Context, 이미지 문맥)를 구분하는 수용 필드(Reception Field)의 급격한 증가를 도우며, PAN(Path Aggregation Network)은 낮은 레이어의 정보를 꼭대기로 쉽게 전파하기 위한 Short-cut Path가 추가된 네트워크로 다양한 Detector 클래스들을 수집한다[8].

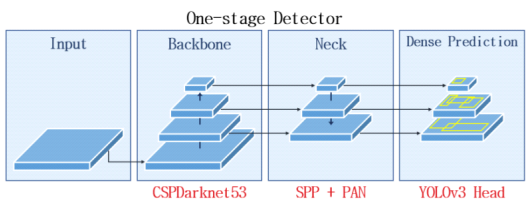



그림 6. YOLOv4의 구조
Fig. 6. Architecture of YOLOv4

### 2.3.2 U-Net 손 분할

1) U-Net

본 논문에서는 복잡한 배경을 가진 이미지의 객체 분할 방식으로 먼저 딥러닝을 활용한 분할 모델인 U-Net을 제안한다. U-Net 모델은 생물 의학 관련 이미지 처리를 위해 개발된 모델로써, FCN(Fully-Convolutional Network) 기반 모델이며, End-to-End 방식으로 설계되었다. U-Net 특징은 인코딩 레이어와 디코딩 레이어를 직접 연결하고 합치는 방식(Concatenated Skip Connection)으로 저차원의 이미지 특징만을 추출하는 것이 아닌 고차원의 이미지 특징 또한 추출하여 높은 정확도의 픽셀 단위 지역화(Localization)가 가능하다. 객체 위치 정보를 잃지 않고, 실제 데이터의 수 보다 많은 학습 데이터를 활용할 수 있기에 적은 수의 학습 데이터를 이용하더라도 높은 정확도의 이미지 분할 성능을 보여준다.

그림 7은 U-Net 구조이다. 그림 7의 각 푸른색 상자는 다중 채널 특징 맵(Multi-Channel Feature Map)을 나타내며, 흰색 상자는 복사된 특징 맵(Copied Feature Map)을 나타낸다. 가로 방향 숫자는 채널 수를 나타내며, 세로 방향 숫자는 맵의 차원(x-y-size)을 나타낸다. 그림 7에서 볼 수 있듯 U-Net은 'U'자형 구조로 축소 네트워크와 확장 네트워크가 서로 대칭을 이루고 있다. 축소 경로(Contracting Path)로 불리는 인코더(Encoder)는 이미지의 전반적인 픽셀 간의 정보(Context, 이미지 문맥)를 얻기 위해 구성된 네트워크이며 컨볼루션 블록과 맥스 풀링으로 이루어져 있고, 확장 경로(Expanding Path)로 불리는 디코더(Decoder)는 정

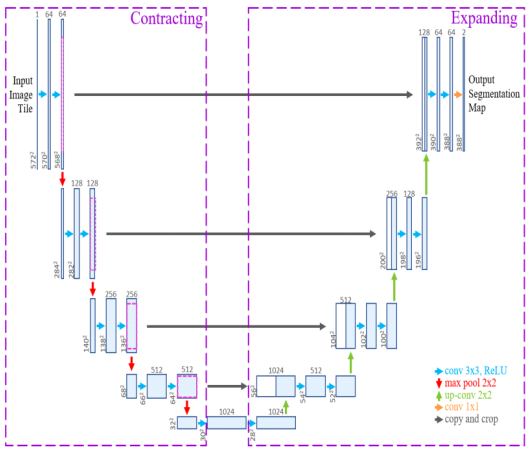
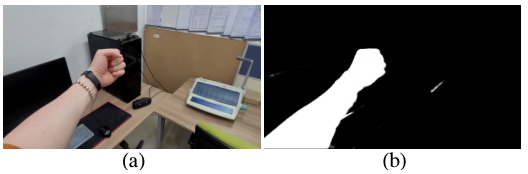



그림 7. U-Net 구조
Fig. 7. Architecture of U-Net



(a) (b)

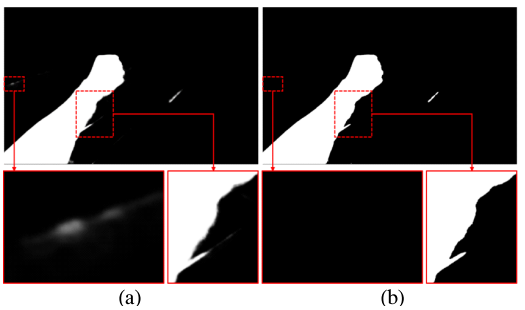
그림 8. U-Net 결과: (a) 원본 이미지, (b) U-Net 결과 마스크
Fig. 8. U-Net result: (a) original image, (b) mask of U-Net result

확하고 세밀한 지역화를 위해 구성된 네트워크이며 이미지를 확장시키고 출력 이미지 채널 수를 조정하는 Convtranpose와 컨볼루션 블록으로 이루어져 있다. 컨볼루션 블록은 3x3 컨볼루션과 정규화(Batch Normalization) 그리고 ReLu 활성화 함수가 두 번 반복 배치되어 있다[9].

그림 8은 U-Net의 결과이다. 딥러닝을 활용하지 않은 분할의 결과와 달리 색상이나 엣지 등이 정보를 모두 포함하고, 학습된 데이터 기준으로 판단하여 높은 정확도의 분할 결과를 보인다. 또한 전체 팔과 악세사리와 옷 소매까지 포함되어 있다. 하지만 배경 일부에서도 같은 객체로 분할되어 포함되어 있다.

2) 오츠 이진화를 이용한 임계처리

본 논문에서는 YOLOv4 모델을 통해 제공된 관심 영역에 U-Net 분할 모델을 적용하고 후처리 단계에 자동으로 임계값을 찾는 오츠 이진화를 통한 임계 처리된 마스크 생성을 제안한다. 오츠 이진화는 먼저 임계값 $T$를 임의로 정하고 히스토그램 이용하여 전체 이미지의 픽셀들을 두 그룹으로 분할하고, 두 그룹의 명암 분포를 구하는 작업을 반복하여, 모든 경우의 수 중에서 두 그룹의 명암 분포가 균등할 때의 값 $T$를 선택하는 것이다. 이를 통해 복잡한 배경에 맞게 적절한 임계값을 적용할 수 있다[12].



(a) (b)

그림 9. U-Net 결과에 오츠 이진화 적용 결과: (a) U-Net 결과, (b) 오츠 이진화 적용 결과
Fig. 9. Otsu's Binarization result on U-Net result: (a) U-Net result, (b) Otsu's Binarization result

그림 9는 U-Net 분할 결과에 오츠 이진화를 적용한 결과로 객체 이외의 불필요한 부분들이 제거되었고, 마스크의 픽셀 값이 0과 1로 구분되어, 결과를 활용하기 용이해졌다.

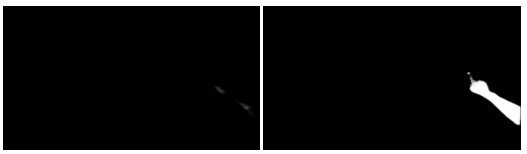
## Ⅲ. 실험 및 결과

본 논문에서 제안한 복잡한 환경에서의 손 분할 정확도 향상을 위한 국부 영역 자동 검출 알고리즘의 성능 평가를 위해 2MP(Megapixels) 해상도의 USB 카메라와 12MP 해상도의 카메라를 통해 여러 가지 배경을 가진 손 이미지들을 획득하여 실험 데이터로 사용하였다. 제안한 알고리즘의 유효성 검증을 위해 정답 마스크를 만들어 결과 마스크와 $IoU$(Intersection over Union) 수치를 확인하였다. $IoU=1$이면 완전한 분할, $IoU=0$이면 완전한 오분할이다.

그림 10은 U-Net 모델의 입력 이미지 크기에 따른 결과를 비교한 것이다. 그림 10(a)는 원본 이미지, (b)는 입력 이미지의 크기가 256×256일 때의 결과 마스크, $IoU$ 수치와 분할 시간이고 (c)는 입력 이미지의 크기가 512×512일 때의 결과 마스크, $IoU$ 수치와 분할 시간이다. 그림 10과 같이 분할의 입력 이미지의 크기가 작은 경우에 빠른 속도를 보이나 리사이징 단계에서 원본 이미지로부터 소실되는 픽셀들이 많이 발생하여 $IoU$ 수치가 매우 낮은 걸 확인할 수 있다.

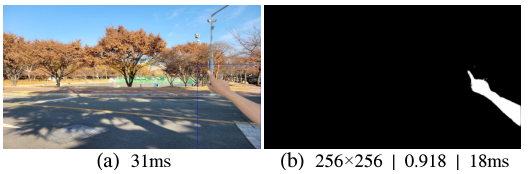
그림 11은 U-Net 모델의 입력 이미지로 전체 이미지가 아닌 관심 영역만을 적용하였을 때의 결과이다. 그림 11(a)는 원본 이미지에서 YOLOv4를 통한 관심 영역 추출 결과와 속도이고, (b)는 전체 이미지가 아닌 추출된 관심 영역만을 입력으로 하고, 크기가 256×256일 때의 결과 마스크, $IoU$ 수치와 분할 시간이다. 그림 11과 같이 분할의 입력 이미지의 크기가 작은 경우에도

(a)



(b) 256×256 | 0.348 | 20ms (c) 512×512 | 0.924 | 58ms

그림 10. U-Net 모델 입력 이미지 크기에 따른 결과 비교
Fig. 10. Result comparison of U-Net input size



(a) 31ms (b) 256×256 | 0.918 | 18ms

그림 11. U-Net 모델 입력 이미지 관심 영역 적용에 따른 결과
Fig. 11. Result of U-Net input image ROI application

관심 영역만을 입력으로 했을 때, 빠른 속도의 장점은 유지되고 리사이징 단계에서 원본 이미지로부터 소실되는 픽셀들이 적게 발생하여 높은 $IoU$ 수치를 보인다. 이를 근거로 분할의 입력 이미지로 전체 이미지가 아닌 관심 영역이 주어졌을 때 더 효율적임을 확인할 수 있다.

딥러닝 기반 객체 인식 기법인 YOLOv4를 이용한 검출 결과에서 그림 12와 같이 관심 영역에 객체의 일부분이 포함되지 않는 경우가 있다. 이러한 결과에 분할을 적용하게 되면 객체의 일부를 배경으로 인식하는 문제가 발생할 수 있다. 따라서 검출 결과를 일정 비율 확장하여 원하는 객체가 사각형 내에 모두 포함될 수 있도록 한다.

그림 13은 딥러닝 객체 인식 모델인 YOLOv4를 통해 검출된 관심 영역의 확장 여부에 따른 U-Net 결과

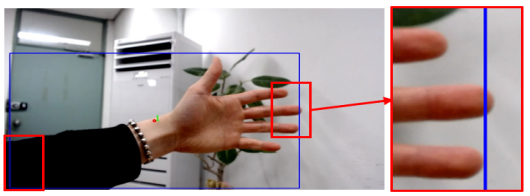
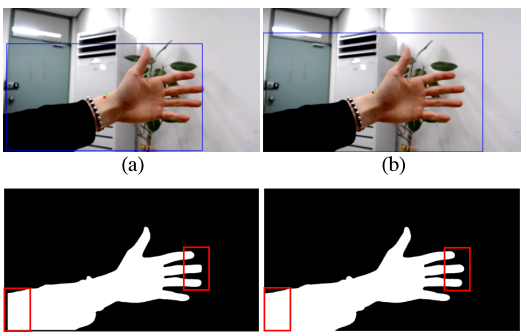


그림 12. YOLOv4의 결과가 객체 일부를 미포함 하는 경우
Fig. 12. YOLOv4 result with missing part of object



(a) (b) (c) 0.957 (d) 0.983

그림 13. YOLOv4로 검출된 결과의 확장 여부에 따른 U-Net 결과 비교
Fig. 13. Comparison of U-Net result of YOLOv4 result expanding

마스크와 $IoU$ 수치이다. 그림 13(a)는 YOLOv4의 손 검출 결과이며, (b)는 검출 결과를 20% 확장한 결과이다. 그림 13(c)는 (a)를 입력으로 한 U-Net의 결과이며, (d)는 (b)를 입력으로 한 U-Net의 결과이다. 그림 13과 같이 YOLOv4의 결과로 객체가 일부 소실되었을 때 분할을 한 결과 보다 YOLOv4의 결과를 일부 확장하여 분할을 한 결과가 더 좋은 $IoU$ 수치를 보인다. 따라서 본 논문에서는 YOLOv4의 결과를 20% 확장하여 분할의 입력으로 사용하도록 하였다.

그림 14는 U-Net 분할과 제안한 알고리즘이 각각 적용된 결과이다. 그림 14(a)는 사람이 직접 작성한 객체에 대한 정답 마스크이다. 그림 14(b)는 U-Net 분할의 결과 마스크와 $IoU$ 수치이며, (c)는 제안된 알고리즘이 적용된 U-Net 분할의 결과 마스크와 $IoU$ 수치이다. 제안하는 알고리즘의 적용 유무에 따라 분할된 마스크의 차이를 확인할 수 있다. 또한 $IoU$ 수치를 통해 83%와 97%의 분할률을 확인할 수 있으며, 육안으로

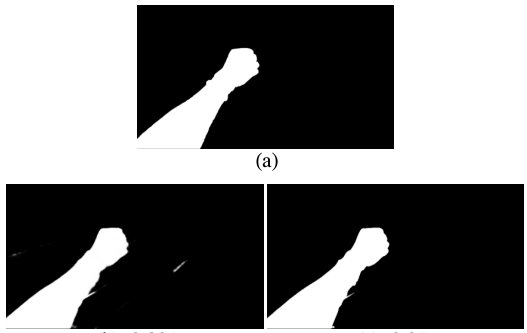



그림 14. U-Net과 제안 알고리즘이 적용된 U-Net 결과 비교: (a) 정답 마스크(Ground Truth), (b) U-Net 결과 마스크와 IoU, (c) 제안된 알고리즘이 적용된 U-Net 결과 마스크와 IoU
Fig. 14. Comparison of U-Net and the proposed method result: (a) ground truth, (b) U-Net result mask and IoU, (c) the propose method result mask and IoU

보여지는 차이는 크지 않으나, $IoU$ 수치의 차이를 통한 분석으로 확인해보았을 때 마스크 내에 노이즈나 블러


그림 15. 제안하는 알고리즘을 적용한 결과: (a) 원본 이미지, (b) GrabCut 적용 결과와 IoU, (c) U-Net 적용 결과 IoU와 분할 시간, (d) YOLOv4를 적용 결과 시간, (e) ROI Expanding 적용 결과, (f) 제안하는 알고리즘 적용 결과 IoU와 분할 시간
Fig. 15. Result of the proposed method: (a) original image, (b) GrabCut result and IoU, (c) U-Net result IoU and segmentation time, (d) YOLOv4 result, (e) ROI Expanding result, (f) proposed method result IoU and segmentation time

등 픽셀 단위의 잘못된 분할 결과가 존재한다는 것을 알 수 있다.

그림 15는 학습에 포함되지 않은 복잡한 배경을 가진 다양한 이미지에서 제안하는 알고리즘을 각기 다른 이미지에 적용한 결과들이다. YOLOv4를 이용한 손 검출 및 인식을 통해 전체 이미지에서 객체의 위치를 특정하고, 적당한 크기의 ROI 확장으로 일부 소실되었던 객체를 포함시키고, U-Net 분할을 통한 손 분할을 통해 배경과 분리시킨 후 오츠 이진화를 적용하여 일부 불완전한 마스크를 적절히 정리해준 모습을 보인다. 하지만 이미지에서 구분해야 할 객체가 배경과의 색상 및 밝기에 대해 유사도가 높은 경우에 딥러닝 분할 단계에서 일부 배경들까지 한 객체로 인식하는 경우가 발생 하였다.

표 1은 그림 15의 $IoU$ 수치에 대한 결과들을 정리한 표로 GrabCut과 전체 이미지를 입력으로 하는 U-Net 그리고 제안하는 알고리즘을 차례로 적용한 결과들이다. 제안한 알고리즘에서 각각의 이미지들과 전체 평균치에서 가장 높은 $IoU$ 수치를 보였다.

표 2는 그림 15의 분할 속도에 대한 결과들을 정리한 표로 GrabCut은 여러차례 반복을 통한 결과로 제외되었고, 전체 이미지를 입력으로 하는 U-Net과 제안하는 알고리즘의 관심 영역 검출을 위한 YOLOv4 모델과 검출된 관심 영역을 입력으로 하는 U-Net에 대한 결과이다. 전체 이미지를 입력으로 U-Net 손 분할을 하였을 때 평균 66.4ms 이며, 약 15FPS의 처리가 가능하다. 제안하는 알고리즘을 적용했을 때 YOLOv4를 통한 손 검출 단계에서 평균 27.4ms, 검출된 관심 영역을 입력으로 한 U-Net 손 분할은 평균 25.8ms 이므로, 제안하는 알고리즘의 분할 시간은 평균 53.2ms 이며, 약 19FPS의 처리가 가능하다. 표 1과 표 2에서 볼 수 있듯 제안하는 알고리즘이 전체 이미지를 입력으로 넣었을 때 보다 높은 $IoU$ 수치와 처리 속도를 가지며 유효성을 검증 하였다. 추후 효과적인 관심 영역의 확장과 딥러닝 분할 네트워크 설계를 연구하여 주어진 관심 영역의 효과를 극대화하는 것, 옷과 악세사리를 포함하고 손을 포함한 전체 팔 분할로써 학습 데이터 증강(Data Augmentation) 연구를 통해 알고리즘 성능을 개선 및 향상 시킬 수 있을 것으로 기대된다.

표 1. 제안 알고리즘 적용을 통한 IoU 비교
Table 1. IoU comparison of segmentation algorithms

| | Each IoU | | | | | Avg IoU |
|---|---|---|---|---|---|---|
| | (1) | (2) | (3) | (4) | (5) | |
| GrabCut | 0.753 | 0.303 | 0.395 | 0.418 | 0.879 | 0.550 |
| U-Net | 0.810 | 0.924 | 0.926 | 0.944 | 0.871 | 0.895 |
| **Proposed Method** | **0.942** | **0.961** | **0.976** | **0.957** | **0.959** | **0.959** |

표 2. 제안 알고리즘 적용을 통한 분할 시간 비교
Table 2. Time comparison of segmentation algorithms

| | | Each Segmentation Time | | | | | Avg Time |
|---|---|---|---|---|---|---|---|
| | | (1) | (2) | (3) | (4) | (5) | |
| U-Net | | 71ms | 59ms | 66ms | 68ms | 68ms | 66.4ms |
| **Proposed Method** | **YOLOv4** | **25 ms** | **31 ms** | **24 ms** | **28 ms** | **29 ms** | **27.4ms** |
| | **U-Net** | **26 ms** | **20 ms** | **28 ms** | **2 7ms** | **28 ms** | **25.8ms** |

## Ⅳ. 결 론

본 논문에서는 다양하고 복잡한 배경을 가진 이미지에서의 손을 포함한 전체 팔 분할을 목적으로 하여 딥러닝 분할 기법인 U-Net 모델을 제안함에 있어 전체 이미지에서 적용되는 것이 아닌 국부적으로 분할 기법이 적용될 수 있도록 YOLOv4 모델을 활용하여 복잡한 배경을 가진 이미지 속 손의 위치를 특정하고, 검출된 손의 영역을 확장하여 분할의 입력으로 관심 영역을 제공하고 분할 결과에 오츠 이진화를 이용해 임계 처리를 하는 알고리즘을 제안하였다.

또한, HSV 색상기반 손 분할과 Watershed 알고리즘, GrabCut 알고리즘 기법을 적용해보는 실험을 통하여 딥러닝 분할의 필요성을 확인하고, 제안하는 알고리즘에 대해 실험을 하여, 결과로써 알고리즘이 불필요한 배경을 제거하고 더 나은 결과를 보임을 확인하고 유효성을 증명하였다. 하지만 구분해야 할 객체가 높은 유사도를 보이는 배경과 이어져 있거나, 빛의 밝기가 너무 어둡거나 밝은 상황 등 낮은 결과치를 보이는 상황들이 있었다.

향후 실험 결과의 수치적인 해석을 통해 더욱 명확하고 자세한 정보를 얻고, 이를 바탕으로 더 정밀한 분할 결과를 얻기 위한 알고리즘에 대해 연구를 진행하여 개선하고자 한다.

## References

[1] J. Redmon, et al., "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. CVPR,* pp. 779-788, 2016. (https://doi.org/10.48550/arXiv.1506.02640)

[2] P. Dayan, S. Kakade, and P. R. Montague,

"Learning and selective attention," *Nature neurosci.*, vol. 3, no. 11, pp. 1218-1223, 2000. (https://doi.org/10.1038/81504)

[3] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision,* vol. 59, no. 2, pp. 167-181, 2004. (https://doi.org/10.1023/B:VISI.0000022288.19776.77)

[4] A. Urooj and A. Borji, "Analysis of hand segmentation in the wild," in *Proc. IEEE Conf. CVPR*, 2018. (https://doi.org/10.48550/arXiv.1803.03317)

[5] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," in *Proc. Int. Conf. Image Process.,* vol. 2, 2002. (https://doi.org/10.1109/ICIP.2002.1040019)

[6] Y. Wu and Q. Li, "The algorithm of watershed color image segmentation based on morphological gradient," *Sensors*, vol. 22, no. 21, 8202, 2022. (https://doi.org/10.3390/s22218202)

[7] C. Rother, V. Kolmogorov, and A. Blake, "Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graphics (TOG),* vol. 23, no. 3, pp. 309-314, 2004. (https://doi.org/10.1145/1015706.1015720)

[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934,* 2020. (https://doi.org/10.48550/arXiv.2004.10934)

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Int. Conf. MICCAI,* Springer, Cham, 2015. (https://doi.org/10.1007/978-3-319-24574-4_28)

[10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767,* 2018. (https://doi.org/10.48550/arXiv.1804.02767)

[11] D. Misra, "Mish: A self regularized non-monotonic activation function," *arXiv preprint arXiv:1908.08681,* 2019. (https://doi.org/10.48550/arXiv.1908.08681)

[12] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 9, no. 1, pp. 62-66, 1979.

**전 준 현** (Junhyun Jeon)

2019년 2월 : 영남대학교 컴퓨터공학과 학사
2023년 2월 : 경북대학교 전자전기공학부 석사
<관심분야> 딥러닝, 영상인식, 영상처리
[ORCID:0009-0009-0809-7592]

**김 태 훈** (Tae-Hun Kim)

2004년 2월 : 부산외국어대학교 컴퓨터전자공학과 학사
2009년 2월 : 경북대학교 산업공학과 회로 및 시스템공학 석사
2014년 2월 : 경북대학교 IT대학 전자공학부 박사
<관심분야> 영상처리, 패턴인식, 생체신호처리
[ORCID:0009-0003-9472-4500]

**정 유 수** (Yoosoo Jeong)

2019년 8월 : 경북대학교 전자공학부 박사
2019년 9월~6월 : (재)대구경북첨단의료산업진흥재단 선임연구원
2023년 7월~현재 : 한국전자통신연구원 선임연구원
<관심분야> 영상처리, 머신러닝
[ORCID:0000-0001-8022-0975]

**박 길 흠** (Kil-Houm Park)

1982년 2월 : 경북대학교 전자공학과 공학사
1984년 2월 : 한국과학기술원 전기전자공학과 공학석사
1990년 2월 : 한국과학기술원 전기전자공학과 공학박사
1990년 2월~현재 : 경북대학교 전자전기컴퓨터공학부 교수
<관심분야> 영상신호처리, 패턴인식, 영상압축
[ORCID:000 0-0003-0180-5962]