

Tongue Segmentation Using the Receptive Field Diversification of U-net

Yu-Jie Li*, Sung-Tae Jung**

*Lecturer, School of Intelligent Manufacturing, Weifang University of Science and Technology, Shandong, China

Student, Dept. of Computer and Software Engineering, Wonkwang University, Jeonbuk, Korea

**Professor, Dept. of Computer and Software Engineering, Wonkwang University, Jeonbuk, Korea

[Abstract]

In this paper, we propose a new deep learning model for tongue segmentation with improved accuracy compared to the existing model by diversifying the receptive field in the U-net. Methods such as parallel convolution, dilated convolution, and constant channel increase were used to diversify the receptive field. For the proposed deep learning model, a tongue region segmentation experiment was performed on two test datasets. The training image and the test image are similar in TestSet1 and they are not in TestSet2. Experimental results show that segmentation performance improved as the receptive field was diversified. The mIoU value of the proposed method was 98.14% for TestSet1 and 91.90% for TestSet2 which was higher than the result of existing models such as U-net, DeepTongue, and TongueNet.

▶ **Key words:** Tongue, Segmentation, Deep Learning, U-net, Receptive Field

[요 약]

본 논문에서는 U-네트에서 수용 영역을 다양화하여 기존의 모델보다 정확도가 개선된 새로운 혀 영역 분할을 위한 딥러닝 모델을 제안한다. 수용 영역 다양화를 위하여 병렬 컨볼루션, 팽창된 컨볼루션, 상수 채널 증가 등의 방법을 사용하였다. 제안된 딥러닝 모델에 대하여, 학습 영상과 테스트 영상이 유사한 TestSet1과 그렇지 않은 TestSet2의 두 가지 테스트 데이터에 대해 혀 영역 검출 실험을 진행하였다. 수용 영역이 다양화됨에 따라 혀 영역 분할 성능이 향상되는 것을 실험 결과에서 확인할 수 있었다. 제안한 방법의 mIoU 값은 TestSet1의 경우 98.14%, TestSet2의 경우 91.90%로 U-net, DeepTongue, TongueNet 등 기존 모델의 결과보다 높았다.

▶ **주제어:** 혀, 분할, 딥러닝, U-네트, 수용 영역

-
- First Author: Yu-Jie Li, Corresponding Author: Sung-Tae Jung
 - *Yu-Jie Li (trista0211@wku.ac.kr), School of Intelligent Manufacturing, Weifang University of Science and Technology; Dept. of Computer and Software Engineering, Wonkwang University
 - **Sung-Tae Jung (stjung@wku.ac.kr), Dept. of Computer and Software Engineering, Wonkwang University
 - Received: 2021. 08. 24, Revised: 2021. 09. 23, Accepted: 2021. 09. 24.

I. Introduction

혀 상태를 이용한 진단은 비 침습성 및 편리성으로 인해 한의학에서 널리 사용되고 있다. 전통적인 혀 진단은 의사의 주관적인 관찰을 통해서 진행되므로 개인적인 경험, 환경 변화, 조명 변화 등에 의해 편향될 수 있다. 따라서 객관적이고 표준화된 진단 결과를 얻기 위하여 컴퓨터 및 기타 관련 기술을 사용하여 다양한 유형의 컴퓨터 보조 설진 시스템이 개발되고 있다[1-3]. 설진 시스템에서 혀 영역 분할은 진단의 첫 번째 단계이며 그 정확도는 후속 진단에 직접적인 영향을 미친다. 혀 몸체와 입술 색상이 유사하여 구별하기 어렵고, 혀의 모양과 크기는 개인마다 다르며 혀 표면에는 많은 병리학적인 세부 사항이 포함될 수 있고 촬영 각도와 촬영 환경이 다양하게 설정될 수 있기 때문에 영상에서 혀의 위치도 다르고 혀의 질감과 혀 코팅의 색상이 상당히 다를 수 있는 등 혀 영역 분할은 상당히 복잡한 작업이다.

얼굴 영상에서 혀 영역을 효과적으로 분할하기 위해 많은 방법이 개발되었다. 전통적인 이미지 처리 기술을 기반으로 하는 방법으로 적응형 임계값, 기울기 벡터 흐름, 적응형 능동 윤곽 모델, 방사형 경계선 검출 등 다양한 방법들이 제안되었다[4-8]. 최근에 들어, 컨볼루션 신경망과 같은 딥러닝 기술의 부상으로 영상 분류, 객체 감지 및 영상 분할과 같은 컴퓨터 비전 분야에서 다양한 작업의 성능이 크게 향상되었다. 일부 연구자들은 혀 영역 분할에 컨볼루션 신경망을 적용하려고 시도했다[9-11]. 이러한 딥러닝 기반 혀 영역 분할 방법은 기존의 이미지 처리 기술 기반 혀 분할 방법보다 더 나은 결과를 얻을 수 있었다.

본 논문에서는 기존의 영상 분할을 위한 딥러닝 모델인 U-네트[12]를 확장하여 성능이 개선된 새로운 딥러닝 모델을 제안한다. U-네트에서는 수용 영역이 3x3인 컨볼루션 레이어를 직렬로 연결하여 사용하는데, 본 논문에서는 하나의 컨볼루션 레이어를 여러 개의 컨볼루션이 병렬로 연결된 레이어로 바꾸고 각 컨볼루션의 수용 영역을 다르게 함으로써 수용 영역을 다양화하는 방법을 제안한다. 딥러닝 모델의 학습을 빠르게 하기 위해서는 GPU를 사용하는데, 모델이 커지게 되면 GPU의 메모리가 부족하게 되어 학습이 불가능해질 수 있다. 본 논문에서는 네트워크의 크기가 커지는 것으로 줄이기 위한 방법으로 병렬 컨볼루션을 적용하기 전에 채널수를 축소하고 병렬 컨볼루션을 적용한 다음에 채널수를 원래대로 확대하는 방법을 사용한다. 또한 일반 컨볼루션 대신에 수용 영역은 크게 하면서도 네트워크의 크기는 같은 팽창된 컨볼루션을 사용한다. 그리고 채널수를 상수만큼 증가시키는 방법을 사용한다.

본 논문에서는 BioHit에 의해 공개된 영상을 사용하였다(<https://github.com/BioHit/TongelImageDataset>). 이 영상에는 혀 영역을 안정적으로 촬영할 수 있도록 만들어진 전용 영상 획득 장치에서 촬영한 300개의 영상이 포함되어 있다. 300개의 영상 중 180개의 영상을 학습 영상으로 사용하였다. 테스트 영상으로는 학습 영상과 같은 환경에서 촬영된 영상으로 구성된 TestSet1과 학습 영상과 다른 환경에서 촬영된 영상을 포함한 TestSet2를 사용하였다. 혀 영역 분할의 성능을 측정하는 지표로는 정확도(Precision), Dice 계수, ME(Misclassification Error), mIoU(mean Intersection over Union)의 4개 지표를 측정하였다.

단계별 채널 증가량과 병렬로 연결되는 컨볼루션 개수를 변화시켜서 학습을 진행하였다. 실험 결과 단계별 채널 증가량을 줄여도 성능이 크게 저하되지 않았으며, 단계별 채널 증가량을 줄이고 병렬로 연결되는 컨볼루션을 더 확대한 경우에 성능이 개선되었다. TestSet1의 경우 mIoU 값이 Deeptongue[10]은 94.17%이었고 Tonguenet[11]은 97.74%인 반면에 본 논문에서 제안된 방법은 98.14%로 향상되었으며 나머지 지표 값도 향상되었다. TestSet2의 경우 mIoU 값이 DilatedNet[13]은 79.54% ENet[14]은 90.75%인 반면에 본 논문에서 제안된 방법은 91.90%로 보다 개선된 성능을 보였다.

II. Related Work

2.1 Image segmentation with convolution network

영상이 어떤 부류에 속하는지를 판별하는 영상 분류에 컨볼루션 신경망이 좋은 성능을 보이면서 다양한 컨볼루션 신경망이 제안되었다. 영상 분류에 사용된 전통적인 컨볼루션 신경망은 컨볼루션 레이어와 풀링 레이어로 구성된 단계들을 통하여 영상의 특징을 추출하고 그다음에는 완전히 연결된 레이어에서 영상 특징을 바탕으로 영상이 어느 부류에 속하는지를 판별하는 구조로 되어있다.

영상 분류에 사용되던 전통적인 컨볼루션 신경망의 구조를 일부 변경함으로써 각 픽셀이 어떤 부류에 속하는지 판별하는 영상 분할 모델도 만들 수 있다는 것이 발견되었고 다양한 형태의 영상 분할 모델들이 제안되었다. FCN(Fully Convolutional Network)[15]은 영상 분류에 사용되던 컨볼루션 신경망에서 완전히 연결된 레이어를 제거하고 컨볼루션 레이어만 사용하였다. FCN은 Fig. 1과 같이 풀링 레이어를 거치면서 줄어들었던 영상의 크기만큼 업샘플링을

통하여 영상을 확대하여 원래 크기로 공간 정보를 복원하여 영상 분할을 수행하였다. 그림에서 가장 왼쪽의 세로 막대는 입력 영상, 가장 오른쪽의 세로 막대는 분할 결과 맵, 중간에 세로 또는 가로 막대들은 특징 맵을 나타낸다. 세로 또는 가로 막대 위에 표기된 숫자는 데이터 채널수를 나타내고 막대 아래에 표기된 숫자는 특징 맵의 가로와 세로 크기를 나타낸다. 막대들은 채널수가 커짐에 따라 가로로 넓어지도록 그렸으며 특징 맵의 가로와 세로 크기가 커짐에 따라 세로 방향으로 길어지도록 그렸다.

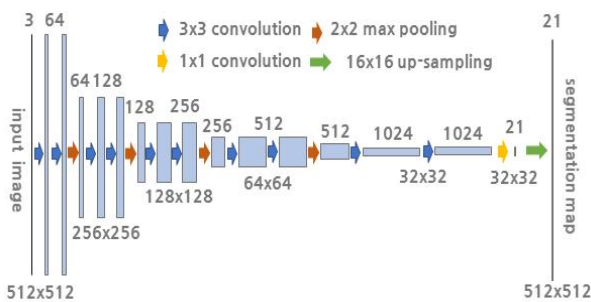


Fig. 1. FCN for Twenty-one Objects Segmentation

2.2 Encoder-decoder architecture

U-네트는 Fig. 2와 같이 알파벳 “U”자 형태의 구조를 가지고 있으며 인코더와 디코더로 구성된다. 인코더에서는 전통적인 컨볼루션 신경망과 같이 특징 맵을 추출하고 공간 정보를 감소시킨다. 디코더에서는 업샘플링과 컨볼루션을 통하여 공간 정보를 복원해나가며 최종적으로 입력 영상에 대한 분할 맵을 생성하게 된다. 인코더에서 손실된 공간 정보를 더 잘 복구하기 위해 인코더의 각 단계에서 구한 특징 맵과 디코더에서 생성되는 높은 수준의 거친 의미 정보인 특징 맵이 수평 연결을 통해 병렬로 결합되어 사용된다. LinkNet[16]은 U-네트와 유사하게 U자 형태의 인코더-디코더 구조를 가지고 있으며 네트워크의 크기를 줄이기 위하여 인코더에 ResNet18[17]을 사용하였다. U-네트와 같이 인코더에서 구한 특징 맵이 디코더에 수평 연결을 통하여 연결되는데, U-네트에서는 두 특징 맵을 병렬로 결합하여 사용하는 것과 달리 LinkNet에서는 두 특징 맵의 값을 더해서 사용함으로써 네트워크의 크기를 줄였다. U-네트와 LinkNet은 네트워크 구조가 중앙을 중심으로 양쪽이 대칭 구조를 가지며 인코더에서 디코더로 특징 맵이 전달된다는 공통점이 있다.

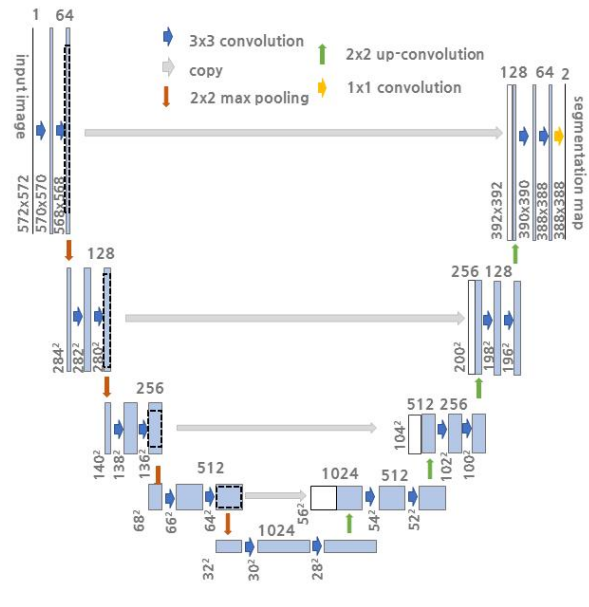


Fig. 2. Architecture of U-net

2.3 Global context based architecture

ParseNet[18]에서는 Fig. 3에 나타나 있듯이 입력 특징 맵에 대하여 글로벌 풀링 레이어를 통하여 단일 전역 특징을 생성하고 정규화 과정을 거친 다음에 역 풀링을 통하여 원래 크기로 복원한다. 이에 병행하여 입력 특징 맵에 정규화를 적용하고 특징을 구한다. 이렇게 구한 두 가지 특징을 결합하여 사용함으로써 전역 정보를 유지하는 네트워크이다.

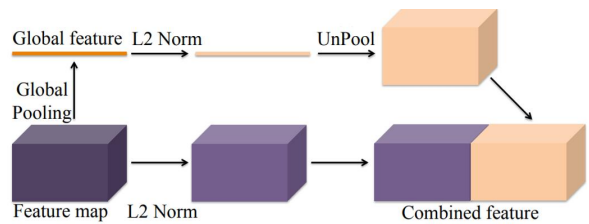


Fig. 3. ParseNet Contexture Overview[18]

PSPNet[19]에서는 개선된 영상 분할을 위해 먼저 특징 맵을 추출한 다음에 피라미드 풀링 모듈을 적용하였다. GCN[20]에서는 지역 및 전역 특징의 이점을 모두 적용하기 위해 대형 커널을 사용하는 전역 컨볼루션 개념을 이용하였다. ParseNet, PSPNet, GCN은 개선된 영상 분할을 위해 지역 특징과 함께 전역 컨텍스트 정보를 사용한다는 공통점을 가지고 있다.

2.4 Realtime architecture

위에서 논의한 네트워크를 포함하여 기존의 많은 영상 분할 네트워크는 실시간 및 효율성보다 분할의 정확성에 초점을 맞추고 있다. 심층 컨볼루션 신경망을 기반으로 하

는 이러한 영상 분할 네트워크들은 분할 성능을 크게 향상시켰지만 종종 메모리 및 전력 소비를 포함하여 많은 컴퓨팅 자원을 필요로 한다. 따라서, 실시간 영상 분할을 위해 가볍고 효율적인 네트워크 모델이 제안되었다. ENet은 Fig. 4와 같이 네트워크 시작 부분에서 입력 영상의 크기를 줄여서 작은 특징 맵을 사용하고 채널을 축소하여 매개변수의 수를 크게 줄임으로써 매우 작은 크기의 네트워크를 구성하였으며 정확도 향상을 위해 팽창된 컨볼루션을 사용하여 수용 영역을 확장하였다.

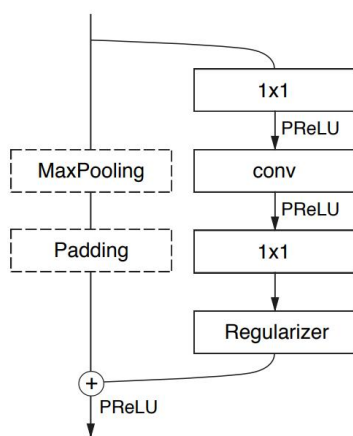


Fig. 4. ENet Bottleneck Module[14]

ERFNet[21]은 1 차원 필터 컨볼루션과 잔차 연결을 사용하고 있다. $N \times N$ 2차원 컨볼루션 커널은 2개의 연속된 $1 \times N$ 및 $N \times 1$ 의 1 차원 컨볼루션 커널로 분할되어 계산될 수 있는데, 이렇게 함으로써 매개변수의 양을 크게 줄이면서 2 차원 컨볼루션과 유사한 분할 효과를 유지하도록 하였다. ThunderNet[22]은 소형 ResNet18을 축소된 작은 네트워크를 인코더로 사용하고 피라미드 풀링 모듈을 디코더와 통합하여 사용함으로써 속도와 성능의 균형을 이루었다.

2.5 Dataset

위에서 설명된 영상 분할 모델 중에서 U-네트는 현미경 사진에서 세포 영역을 분할하는데 사용되었으며 나머지 모델들은 Pascal VOC[23], Microsoft COCO[24], Cityscapes[25] 등의 데이터 세트에 대해 영상 분할을 수행하였다. Pascal VOC에는 사람, 개, 고양이, 자전거, 자동차, 의자, 테이블 등의 21개 부류의 객체를 포함하고 있다. Microsoft COCO는 일반 생활에서 볼 수 있는 91개 부류의 객체를 포함한다. Cityscapes는 시내 도로에서 볼 수 있는 도로, 사람, 차량, 자연, 하늘 등 30개 부류의 객체를 포함한다.

2.6 Architecture for tongue segmentation

기존의 영역 분할 모델들을 혀 영상으로 구성된 학습 영상을 사용하여 다시 학습을 진행하면 혀 영역 분할에 사용할 수 있는데, Deeptongue, Tonguenet와 같이 혀 영역 분할을 위한 전용 모델들도 제안되었다. Deeptongue 모델은 ResNet을 바탕으로 특징을 추출한 다음에 56×56 크기의 분할 마스크를 이용하여 혀 영역을 분할하였다. Tonguenet은 특징 추출 단계, 혀 영역 제안 단계, 예측 단계의 3단계로 구성된다. 특징 추출 단계에서는 팽창된 컨볼루션을 이용하여 다중 스케일 특징을 추출한다. 혀 영역 제안 단계에서는 앞 단계에서 추출된 특징 맵으로부터 혀를 포함하고 있을 확률이 높은 일련의 영역에 대한 제안을 생성한다. 예측 단계에서는 분할 마스크를 이용하여 혀 영역을 분할한다.

III. The Proposed Method

본 장에서는 기존의 U-네트의 수용 영역을 다양화하여 혀 영상 분할 성능을 향상하는 네트워크를 기술한다.

3.1 Modified U-net

Fig. 5에는 본 논문에서 사용한 수정된 U-네트가 나타나 있다. 사용한 입력 영상 크기는 576×768 이고 3개 채널을 가진 컬러 영상이다. 인코더에서는 맨 위 단계에서는 컨볼루션에서 64 채널을 사용하였고 아래 단계로 한 단계 내려 갈 때마다 채널수는 2배가 되도록 하였다. 1024 채널까지 증가한 채널수는 디코더에서 위 단계로 올라가면서 2배씩 축소하여 맨 위 단계에서는 다시 64 채널로 복원된 다음에 최종적으로 분할 맵에서는 1채널로 축소된다.

왼쪽의 인코더에서는 각 단계마다 3×3 컨볼루션을 2번 연속하여 적용한 다음에 2×2 최댓값 풀링을 적용하여 특징 맵의 크기를 줄여나간다. 디코더의 막대 중에 흰색으로 칠해진 막대와 파랑색으로 칠해진 막대가 연결되어 있는데, 흰색 막대는 왼쪽의 인코더의 특징 맵을 복사한 것을 나타낸다. 이와 같이 인코더의 각 단계에서 구해진 특징 맵을 다시 디코더에서 사용함으로써 인코더에서 손실된 공간 정보를 보완하게 되고 이를 통하여 분할 정확도가 향상되게 된다.

수정된 U-네트에서는 컨볼루션 수행시 특징 맵의 크기가 그대로 유지되도록 패딩을 적용함으로써 인코더에서의 4번의 최댓값 풀링과 디코더에서의 4번의 업컨볼루션을 거치고 나면 입력 영상과 분할 맵의 크기가 같게 되어 분할 결과를 바로 사용할 수 있다.

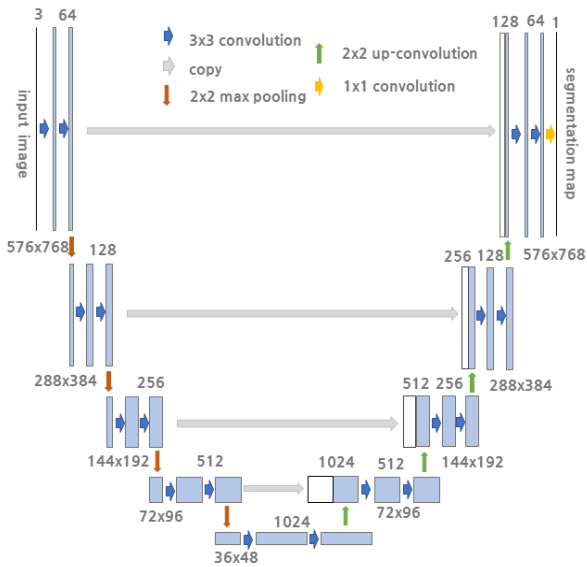


Fig. 5. Modified U-net

3.2 Receptive field diversification in modified U-net

수정된 U-네트의 인코더의 각 단계에서는 3x3 컨볼루션을 2번 연속하여 적용한다. 3x3 컨볼루션의 수용 영역은 3x3이고 3x3 컨볼루션을 2번 연속 적용하면 수용 영역은 5x5가 된다. 따라서, U-네트의 각 단계에서 모두 5x5 크기의 동일한 수용 영역을 사용하는데, 본 논문에서는 다양화된 수용 영역을 사용하여 정확도를 향상시키는 방법을 제안한다. 수용 영역을 다양하게 하는 한 가지 방법은 여러 가지 크기의 수용 영역을 갖는 컨볼루션을 병렬로 사용한 다음에 그 결과를 연결해서 사용하는 것이다. Fig. 6에는 3x3 컨볼루션과 5x5 컨볼루션을 병렬로 적용한 다음에 그 결과를 연결한 예가 나타나 있다. 이와 같은 병렬 컨볼루션을 2번 연속하여 적용하면 3x3 컨볼루션을 두 번 연속으로 적용하는 경우, 3x3 컨볼루션과 5x5 컨볼루션을 연속으로 적용하는 경우, 5x5 컨볼루션을 두 번 연속으로 적용하는 경우 이렇게 세 가지 경우의 컨볼루션이 적용된다. 3x3 컨볼루션을 두 번 연속으로 적용하면 수용 영역이 5x5가 되고, 3x3 컨볼루션과 5x5 컨볼루션을 연속으로 적용하면 수용 영역이 7x7이 되고, 5x5 컨볼루션을 두 번 연속으로 적용하면 수용 영역이 9x9가 되므로, Fig. 6의 병렬 컨볼루션을 두 번 연속하여 적용하면 수용 영역이 5x5, 7x7, 9x9 이렇게 세 가지가 적용되므로 수용 영역이 다양해지고 분할 정확도를 향상시킬 수 있게 된다.

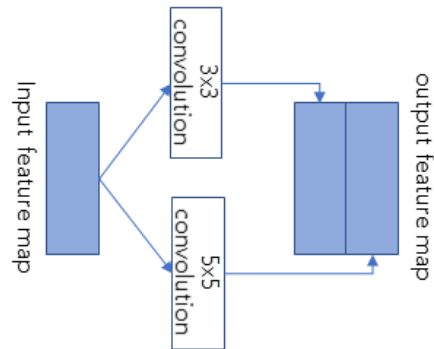


Fig. 6. Parallel Convolution

Fig. 6과 같이 컨볼루션을 추가하면 수용 영역이 다양해지는 장점이 있으나 네트워크의 크기가 커져서 학습 실행에 문제가 발생할 수 있다. 학습을 빠르게 하기 위해 GPU(Graphic Processing Unit)를 이용한 병렬처리가 필수적이다. 학습에는 많은 양의 GPU 메모리가 필요로 하는데 네트워크가 커지면 GPU 메모리가 부족해져서 학습이 불가능해 질 수 있다. 학습하는 동안에 GPU 메모리에는 네트워크의 가중치와 편향값 등의 매개변수, 최적화기에서 사용하는 변수값, 훈련의 전방향 처리에서 계산된 변수값, 컨볼루션 커널 구현에 필요한 변수 값 등이 저장된다. 필요한 GPU 메모리의 양에 영향을 미치는 요인은 여러 가지가 있는데, 네트워크의 크기, 입력 영상의 크기, 배치 크기 등이 주요 요인이다. 본 논문에서는 입력 영상의 크기는 576x768의 고정된 값을 사용하고 배치 크기 8의 고정된 값으로 사용하기로 한다. 네트워크가 커질수록 매개변수 값이 커지는데, 수정된 U-네트의 경우에 매개변수가 31.06백만 개이다.

수정된 U-네트의 경우에 입력 영상의 크기는 576x768이고 배치 크기가 8일 때에 33GB의 GPU 메모리를 사용해서는 메모리 부족으로 학습이 불가능하였고 44GB의 GPU 메모리를 모두 사용해야 학습이 가능하였다. 그런데 Fig. 6과 같이 5x5 컨볼루션을 추가하여 네트워크를 확장하면 매개변수가 159.80백만 개로 늘어나고 44GB의 GPU 메모리를 사용해서도 학습이 불가능하였다.

이와 같이 네트워크 크기가 커지면 제한된 자원 환경 하에서 학습이 불가능해질 수 있기 때문에 네트워크 크기를 크게 하지 않으면서도 수용 영역을 다양화할 수 있는 방법이 필요하다. 한 가지 방법은 팽창된 컨볼루션을 사용하는 것이다. Fig. 7에 3x3 크기의 팽창된 컨볼루션의 예가 나타나 있다. 팽창된 컨볼루션에서는 팽창 비율을 사용한다. Fig. 7(a), (b), (c)의 팽창 비율은 각각 1, 2, 3이다. Fig. 7에서 색이 칠해진 점들은 컨볼루션 수행시 값을 취하는

점을 나타낸다. Fig. 7(a)에서는 바로 이웃한 점들의 값을 취하고 이웃한 점 사이의 거리가 1이므로 팽창 비율이 1이 된다. 팽창 비율이 1인 컨볼루션은 일반 컨볼루션과 동일하다. Fig. 7(b)에서는 값을 2개점을 건너서 취하므로 팽창 비율이 2이고 Fig. 7(c)에서는 3개점을 건너가면서 값을 취하므로 팽창 비율이 3이다. 팽창 비율만큼 건너가면서 값을 취하므로 수용 영역이 더 커지게 되는데, 팽창 비율 2일 때 수용 영역은 5x5가 되고 팽창 비율이 3일 때는 수용 영역은 7x7이 된다. 이것을 일반화시키면 팽창 비율이 n 일 때 수용 영역은 $(2n+1) \times (2n+1)$ 이 된다.

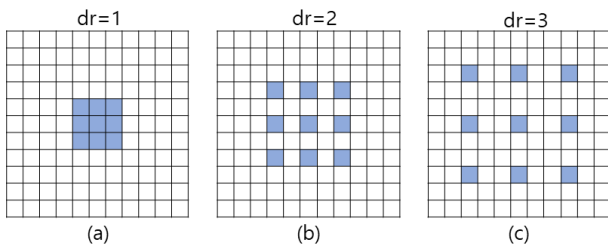


Fig. 7. Dilated Convolution

Fig. 6의 5x5 컨볼루션을 팽창 비율이 2인 3x3 팽창된 컨볼루션으로 바꾸면 매개변수는 92.82백만 개가 된다. 팽창된 컨볼루션을 사용함으로써 수용 영역은 동일하게 하면서 매개변수를 159.80백만 개에서 92.82백만 개로 줄일 수 있었지만, 이렇게 수정된 네트워크도 44GB 보다 큰 메모리를 필요로 해서 학습이 불가능하다.

본 논문에서는 네트워크 크기를 줄이기 위한 두 번째 방법으로 Fig. 8에 나타나 있는 바와 같이 컨볼루션을 적용하기 전에 채널수를 줄이고 결과를 내보낼 때에 다시 채널수를 원래대로 복원하는 방법을 사용한다. 특징 맵의 채널수를 변경할 때에는 1x1 컨볼루션을 이용한다. Fig. 8에서는 채널 수가 n 인 입력 특징 맵에 대하여 병렬로 1x1 컨볼루션을 적용하여 특징 맵의 채널수를 $n/4$ 으로 줄이고 3x3 컨볼루션과 3x3 팽창된 컨볼루션을 적용하여 얻은 특징 맵을 연결하면 채널수가 $n/2$ 이 되고 1x1 컨볼루션을 적용하여 채널수가 n 인 출력 특징 맵을 구한다. Fig. 8과 같은 병렬 컨볼루션을 사용하면 수정된 U-네트의 매개변수는 17.28백만 개로 줄어들고 44GB의 GPU 메모리를 사용하여 학습이 가능하다.

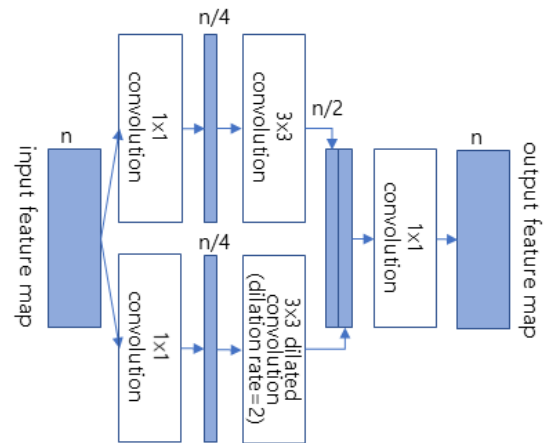


Fig. 8. Convolution with Channel Reduction

Fig. 8의 병렬 컨볼루션에서는 2가지의 컨볼루션을 사용하였지만, Fig. 9와 같이 팽창 비율을 다양하게 하여 병렬 경로를 늘리면 수용 영역을 더 다양하게 할 수 있을 것이다. 그러나 병렬 경로를 늘리면 필요로 하는 GPU 메모리도 같이 커지는데, 지금까지 기술된 방법으로는 병렬 경로가 세 개인 경우에도 44GB 보다 더 많은 GPU 메모리를 필요로 하여 학습이 불가능하였다.

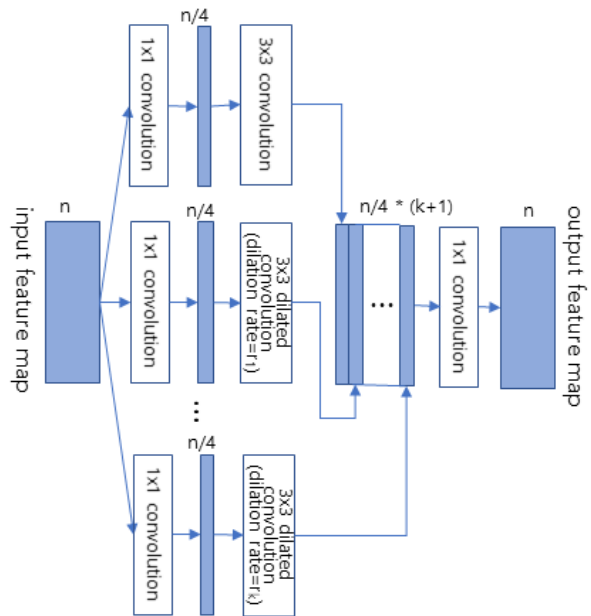


Fig. 9. Expansion of Parallel Convolution

U-네트에서는 수축 경로의 1단계에서 64채널로 시작하고 단계마다 채널수가 2배씩 선형 증가하여 5단계에서 1024 채널을 사용하고 있다. 본 논문에서는 단계마다 상수로 채널을 증가시켜 네트워크 크기를 줄이는 방법을 사용한다. 단계별 채널수를 상수로 증가시키면서 Fig. 9의

병렬 컨볼루션을 사용할 때에 44GB 크기의 GPU 메모리 환경 하에서 사용 가능한 병렬 경로의 수가 Table 1에 나타나 있다.

Table 1. Maximum Parallel Paths per Channel Increment.

Channel Increment	Channel Number of 5 th Level	Maximum Parallel Paths
16	128	5
32	192	4
48	256	3
64	320	3

본 논문에서는 위에서 기술한 바와 같이 병렬 컨볼루션, 팽창된 컨볼루션, 채널 축소 후 컨볼루션 적용, 단계별 채널수의 상수 증가의 방법을 기존의 영상 분할 네트워크인 U-네트에 접목하여 수용 영역을 다양화하였다.

IV. Experiment

4.1 Dataset

지금까지 혀 분할 및 성능 평가에 사용할 수 있는 공개 표준 혀 이미지 데이터 세트는 소수이며 그 크기도 매우 작은 편이다. 본 논문에서는 BioHit에 의해 공개된 혀 영상 데이터 세트를 이용하여 딥러닝 학습을 진행한다. 이 데이터 세트는 균일한 조명 조건에서 동일한 획득 장치에 의해 수집된 300개의 혀 영상으로 구성된다. Fig. 10에는 BioHit의 혀 영상의 예가 나타나 있다.



Fig. 10. Tongue Images of BioHit

300개의 영상을 6:2:2 비율로 나누어 180개 영상은 학습용으로, 60개는 검증용으로, 60개는 테스트용으로 사용하였다. 이렇게 구성된 테스트 데이터를 TestSet1이라 하자. TestSet1의 경우에는 학습 영상과 테스트 영상이 같은 환경에서 촬영되어 영상 분할이 용이할 수 있다. 그러

나 학습 모델이 보다 견고하게 동작하기 위해서는 학습한 영상과 다른 다양한 환경에서 촬영된 영상도 처리할 수 있어야 한다. 다양한 환경에서 촬영된 영상으로 구성된 테스트 세트를 만들기 위해서 참고문헌 [26]에 제안된 5가지 방법으로 획득된 영상을 각각 40개씩 모았다. Fig. 11에는 5가지 방법으로 촬영된 영상 예가 나타나 있다. Fig. 11(a)의 영상들은 구글 이미지 중에서 혀를 검색어로 하여 검출된 영상 중에서 선택된 영상들이다. 이들 영상은 다양한 환경에서 다양한 카메라로 촬영된 영상들이다. Fig. 11(b)의 영상들은 BioHit의 영상들로 학습에 사용된 영상들과 같은 환경에서 획득된 영상들이다. Fig. 11(c)의 영상들은 참고문헌 [27]에서 제공한 영상들로 전용 촬영 장치를 이용하여 획득한 영상들이다. Fig. 11(d)의 영상들은 본 연구실에서 일반 카메라를 이용하여 실내에서 촬영한 영상들이다. Fig. 11(e)의 영상들은 한국한의학연구원에서 개발된 촬영 장치를 이용하여 획득한 영상들이다[28]. 이와 같이 구성된 200개의 테스트 혀 영상들은 TestSet2라고 하자. TestSet2는 서로 다른 장치와 다른 환경에서 촬영된 영상들이 혼합되어 있어서 TestSet1에 비하여 영상 분할이 더 어렵다.

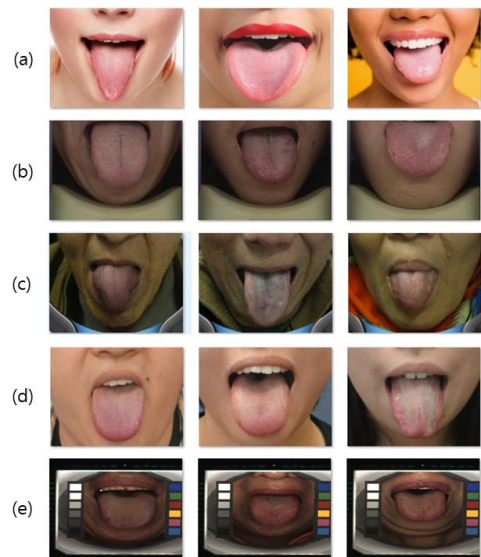


Fig. 11. Examples of Test Images Filmed in Different Environments

BioHit의 180개 영상을 학습에 이용하기로 하였지만 180개 영상만을 학습에 이용하기에는 충분하지 않다. 학습 데이터의 부족 문제를 해결하기 위하여 참고문헌 [26]에서 제안된 데이터 증강 방법을 이용하여 각각의 영상에 대하여 회전, 확대/축소, 밝기 조절, 명암대비 조절 등의 변환을 조합하여 10개의 영상을 생성하였다. 이렇게 함으

로써 새로이 생성된 영상 1,800개와 원래의 영상 180개를 합쳐서 총 1,980개의 영상을 학습에 이용하였다. Fig. 12(a)에는 하나의 원본 영상이 나타나 있고 Fig. 12(b)에서 Fig. 12(j)에는 데이터 증강에 의해 생성된 영상의 예가 나타나 있다.

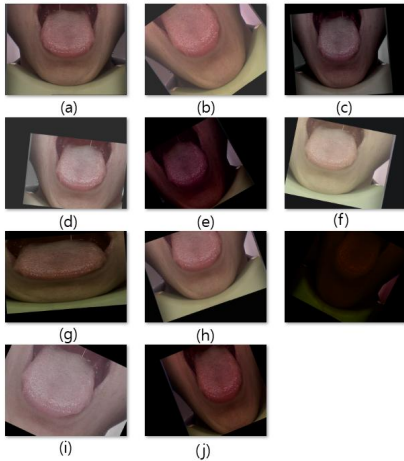


Fig. 12. Examples of Data Augmentation

4.2 Evaluation metrics

모델 성능을 평가하기 위해 기존에 영상 분할에 널리 사용되는 네 가지 지표, 즉 정확도(Precision), mIoU(mean Intersection over Union), Dice 계수, ME(Misclassification Error)를 사용한다. 이들 지표는 다음과 같이 정의된다.

$$Precision = \frac{card(T_G \cap T_P)}{card(T_P)} \quad (1)$$

$$Dice = \frac{2 \times card(T_G \cap T_P)}{card(T_G) + card(T_P)} \quad (2)$$

$$ME = 1 - \frac{card(T_G \cap T_P) + card(B_G \cap B_P)}{card(T_G) + card(B_G)} \quad (3)$$

$$mIoU = \frac{1}{2} \left(\frac{card(T_G \cap T_P)}{card(T_G \cup T_P)} + \frac{card(B_G \cap B_P)}{card(B_G \cup B_P)} \right) \quad (4)$$

식에서 T_P 와 B_P 는 각각 예측 결과에서 혀 영역과 및 혀가 아닌 영역을 나타내고, T_G 와 B_G 는 각각 실제의 혀 영역과 혀가 아닌 영역을 나타낸다. $card()$ 는 영역의 넓이 즉 영역 내의 픽셀 수를 나타낸다. 정확도는 혀로 분할된 픽셀 중에 실제로 혀인 픽셀의 비율을 나타낸다. Dice 계수는 L.R. Dice에 의해 제안된 계수로서 두 샘플의 유사도를 측정하는 데에 사용된다. 여기에서는 혀로 예측된 영역과 실제로 혀인 영역의 유사도를 측정하는 지표이다. ME는

실제로 혀이거나 혀가 아닌 픽셀 중에서 잘못 분할된 픽셀의 비율을 나타낸다. IoU는 두 영역의 교차영역의 넓이를 합영역의 값으로 나눈 값을 뜻하고 분할하는 각 부류별로 IoU를 구하게 되고 mIoU는 이들 값의 평균값이다. 영상 분할에서는 mIoU값이 가장 널리 사용되는 지표이다.

4.3 Results

본 논문에서는 Geforce RTX 2080ti GPU 보드 4개가 장착된 시스템을 사용하였다. 이 GPU 보드 한 개에 11GB의 메모리가 장착되어 있으므로, 총 44GB의 GPU 메모리를 사용 가능하다. 본 논문에서는 Ubuntu 18.04 운영 체제에서 Keras 및 TensorFlow 딥러닝 프레임 워크를 사용하여 학습과 테스트를 진행하였다. 네트워크는 배치 크기 8을 사용하여 10 epoch 동안 훈련하였다. 손실을 계산하기 위한 목적 함수로는 이진 교차 엔트로피 손실 함수를 사용하였다. 네트워크의 학습 과정에서는 매개 변수 초기화와 데이터 선택 등에서 무작위 선택을 사용하는데, 이로 인하여 동일한 데이터를 이용해서 네트워크를 학습하더라도 학습할 때마다 네트워크 매개 변수가 달라지고 네트워크 성능도 약간씩은 달라진다. 따라서 본 논문에서는 학습과 테스트 과정을 30번 반복한 다음에 지표의 평균값을 구하여 지표 값으로 사용하였다.

Table 2에는 TestSet1에 대한 실험 결과가 나타났다. 본 논문에서 팽창된 컨볼루션 사용, 채널 축소 후 컨볼루션 적용, 단계별 채널수의 상수 증가의 방법을 사용하였는데, U의 첨자에서 D는 채널 축소 후 컨볼루션을 적용한 것을 의미하고, C16, C32, C48, C64는 각각 단계별로 채널수를 16, 32, 48, 64개씩 증가시킨 것을 나타낸다. 또한 P()는 괄호 안의 컨볼루션들이 병렬로 수행된다는 것을 나타내며, 3x3 크기의 일반 컨볼루션은 3x3과 같이 나타내고 3x3 크기의 팽창된 컨볼루션은 팽창 비율이 2인 컨볼루션은 r2, 팽창 비율이 3인 컨볼루션은 r3과 같이 나타낸다. 예를 들어 P(3x3,r2)는 일반 3x3 컨볼루션과 팽창 비율이 2인 3x3 팽창된 컨볼루션을 병렬로 적용한다는 것을 나타낸다. Table 2의 TestSet1에 대한 혀 영역 분할 결과를 보면, 채널 축소 후 컨볼루션 적용, 단계별 채널수의 상수 증가, 팽창된 컨볼루션, 병렬 컨볼루션 방법을 적용하여 만든 여러 가지 네트워크의 혀 영역 분할 성능 지표 값이 수정된 U-네트의 값보다 모두 향상된 것을 볼 수 있다. 또한 채널 증가량이 고정되어 있을 때 병렬 컨볼루션의 수가 증가하여 수용 영역이 다양해질수록 성능 지표 값들이 대부분 향상되는 것을 볼 수 있으며 5개의 병렬 컨볼루션을 사용한 $U_{D,C16,P(3 \times 3,r2,r3,r4,r5)}$ 네트워크가 최고 성능을 보였다.

Table 2. Experimental Results for TestSet1

Model	Precision	DICE	ME	mIoU
Modified U-net	97.59	97.53	1.11	96.89
U _{D_C64_P(3x3,r2)}	98.55	98.14	0.83	97.64
U _{D_C64_P(3x3,r2,r3)}	98.59	98.24	0.78	97.77
U _{D_C48_P(3x3,r2)}	98.00	97.90	0.94	97.35
U _{D_C48_P(3x3,r2,r3)}	98.89	98.36	0.73	97.92
U _{D_C32_P(3x3,r2)}	98.74	98.29	0.76	97.83
U _{D_C32_P(3x3,r2,r3)}	98.70	98.37	0.73	97.93
U _{D_C32_P(3x3,r2,r3,r4)}	98.68	98.42	0.70	98.00
U _{D_C16_P(3x3,r2)}	97.74	97.84	0.97	97.27
U _{D_C16_P(3x3,r2,r3)}	98.38	98.22	0.79	97.75
U _{D_C16_P(3x3,r2,r3,r4)}	98.77	98.52	0.66	98.12
U _{D_C16_P(3x3,r2,r3,r4,r5)}	98.91	98.54	0.65	98.14

Table 3에는 TestSet2에 대한 실험 결과가 나타나 있다. TestSet2에는 학습 영상과 다른 환경에서 획득된 영상이 75%가 포함되어 있어서 학습 영상과 테스트 영상이 같은 환경에서 만들어진 TestSet1에 비하여 성능 지표 값이 많이 낮아진 것을 볼 수 있다. 전반적으로 성능 지표 값이 낮아지기는 하였지만 병렬 컨볼루션이 증가하여 수용 영역이 다양화됨에 따라 지표 값들이 대부분 향상되었으며 5개의 병렬 컨볼루션을 사용한 U_{D_C16_P(3x3,r2,r3,r4,r5)} 네트워크가 최고 성능을 보였다.

Table 3. Experimental Results for TestSet2

Model	Precision	DICE	ME	mIoU
Modified U-net	89.46	71.95	7.55	74.04
U _{D_C64_P(3x3,r2)}	92.77	79.77	5.69	80.10
U _{D_C64_P(3x3,r2,r3)}	93.25	84.00	4.64	83.74
U _{D_C48_P(3x3,r2)}	90.13	78.32	6.21	78.85
U _{D_C48_P(3x3,r2,r3)}	94.49	84.60	4.45	84.36
U _{D_C32_P(3x3,r2)}	92.77	80.37	5.54	80.65
U _{D_C32_P(3x3,r2,r3)}	91.80	86.30	4.19	85.65
U _{D_C32_P(3x3,r2,r3,r4)}	94.54	89.65	3.14	88.95
U _{D_C16_P(3x3,r2)}	93.19	82.83	4.90	82.81
U _{D_C16_P(3x3,r2,r3)}	90.84	87.00	4.08	86.23
U _{D_C16_P(3x3,r2,r3,r4)}	95.45	89.60	3.08	89.04
U _{D_C16_P(3x3,r2,r3,r4,r5)}	95.98	92.62	2.27	91.90

U_{D_C16_P(3x3,r2,r3,r4,r5)} 네트워크와 기존 네트워크의 허 영역 분할 결과를 비교한 결과가 Table 4와 Table 5에 나타나 있다. FCN, DilatedNet 등 기존의 영상 분할 모델에 대하여 허 영상을 입력으로 하여 학습을 진행한 다음에 TestSet1과 TestSet2에 대해 영상 분할을 수행하였다. FCN에 비하여 U_{D_C16_P(3x3,r2,r3,r4,r5)} 네트워크는 mIoU 값이 TestSet1에 대해서는 1.7% 향상되었으나 TestSet2에 대해서는 11.1% 향상된 것을 볼 수 있다. 이러한 실험 결과는 본 논문에서 제안된 방법이 학습 영상과 테스트 영상에 차이가 많을 때에도 안정적으로 영상 분할을 수행할 수 있음을 보이고 있다. Deeptongue과 Tonguenet의 지표값

은 참고문헌 [11]에서 같은 데이터에 대해 제시한 결과 값이다. U_{D_C16_P(3x3, dr2, dr3, dr4, dr5)} 네트워크는 mIoU 값이 Deeptongue에 비하여 4.2% 향상되었고 Tonguenet에 비하여 0.4% 향상되었다.

Table 4. Comparison with the Existing Model for TestSet1

Model	Precision	DICE	ME	mIoU
FCN	97.50	97.21	1.24	96.50
DilatedNet	98.41	97.74	1.00	97.16
LinkNet	97.39	97.91	0.95	97.37
PSPNet	98.21	97.96	0.91	97.43
ENet	98.63	98.51	0.67	98.10
Deeptongue	92.78	95.22	1.69	94.17
Tonguenet	98.63	97.96	0.85	97.74
U _{D_C16_P(3x3,dr2,dr3,dr4,dr5)}	98.91	98.54	0.65	98.14

Table 5. Comparison with the Existing Model for TestSet2

Model	Precision	DICE	ME	mIoU
FCN	89.81	82.96	5.08	82.71
DilatedNet	89.42	79.04	6.01	79.54
LinkNet	89.19	90.52	3.11	89.75
PSPNet	81.72	79.98	6.41	79.93
ENet	90.35	91.67	2.78	90.75
U _{D_C16_P(3x3,dr2,dr3,dr4,dr5)}	95.98	92.62	2.27	91.90

V. Conclusions

본 논문에서는 U-네트를 기반으로 여러 개의 컨볼루션을 병렬로 연결하여 수용 영역을 다양하게 함으로써 허 영역 분할 성능을 향상시킨 딥러닝 모델을 제안하였다. 컨볼루션을 병렬로 연결함에 따라 네트워크의 크기가 커져서 컴퓨팅 자원의 부족으로 학습이 불가능해지는 문제를 해결하기 위하여, 컨볼루션을 적용하기 전에 채널수를 축소하였다가 컨볼루션을 적용한 다음에 채널수를 원래대로 복원하는 방법을 사용하였다. 또한 수용 영역은 크게 하면서 네트워크 크기는 증가시키지 않는 팽창된 컨볼루션을 사용하였으며 단계마다 증가하는 채널수를 상수로 하는 방법을 사용하였다. 단계마다 증가하는 채널수가 작아지면 병렬로 연결할 수 있는 컨볼루션의 수가 증가하였는데, 실험 결과 병렬로 연결하는 컨볼루션 개수에 따른 수용 영역의 다양성이 성능에 많은 영향을 주는 것을 확인할 수 있었으며 단계마다 채널이 16씩 증가하게하고 병렬로 연결하는 컨볼루션을 5개로 하였을 때 지표 값이 가장 우수하였다. 기존의 영상 분할 모델인 FCN, DilatedNet, LinkNet, PSPNet, ENet을 허 영상에 대해 학습한 모델에 비하여 본

논문에서 제안된 모델의 성능이 우수하였으며, 기존의 혀 영역 분할 모델인 Deeptongue과 Tonguenet에 비하여도 본 논문에서 제안된 모델의 성능이 우수하였다.

ACKNOWLEDGEMENT

This paper was supported by Wonkwang University in 2020.

REFERENCES

- [1] M.H. Tania, K.Lwin, M.A. Hossain, "Advances in automated tongue diagnosis techniques," *Integrative Medicine Research*, Vol. 8, no. 1, pp. 42-56, March 2019. DOI: 10.1016/j.imr.2018.03.001
- [2] M.H. Subandia, N.D. Kamarudinb, M.A. Yusof, A.A. Bakar, "Prototyping digital tongue diagnosis system on roborealm and raspberry-pi," *ZULFAQAR Journal of Defence Science, Engineering & Technology*, Vol. 2, no. 1, pp. 24-33, June 2019.
- [3] M.H. Tania, K. Lwin, and M.A. Hossain, "Advances in automated tongue diagnosis techniques," *Integrative Medicine Research*, Vol. 8, no. 1, pp. 42-56, March 2019. DOI: 10.1016/j.imr.2018.03.001
- [4] Z. Li, Z. Yu, W. Liu, J. Hu, Y. Lin, and Z. Zhang, "Tongue image segmentation via thresholding and clustering," *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1-5, Oct. 2017. DOI: 10.1109/CISP-BMEI.2017.8302207
- [5] J. Ning, D. Zhang, C. Wu, et al., "Automatic Tongue Image Segmentation Based on Gradient Vector Flow and Region Merging," *Neural Computing and Applications*, Vol. 21, no. 8, pp. 1819-1826, Nov. 2012. DOI: 10.1007/s00521-010-0484-3
- [6] J. Guo, Y. Yang, Q. Wu, J. Su, and F. Ma, "Adaptive active contour model based automatic tongue image segmentation," *International Congress on Image and Signal Processing, Biomedical Engineering and Informatics*, pp. 1386-1390, Oct. 2016. DOI: 10.1109/CISP-BMEI.2016.7852933
- [7] Saparudin, Erwin, and M. Fachrurrozi, "Tongue Segmentation Using Active Contour Model," *International Conference on Electrical Engineering, Computer Science and Informatics*, pp. 1-5, Nov. 2016. DOI: 10.1088/1757-899X/190/1/012041
- [8] W. Liu, J. Hu, Z. Li, Z. Zhang, Z. Ma, and D. Zhang, "Tongue Image Segmentation via Thresholding and Gray Projection," *KSII Transactions on Internet and Information Systems*, Vol. 13, no. 2, pp. 945-961, Feb. 2019. DOI : 10.3837/tiis.2019.02.025
- [9] P. Qu, Z. Hui, Z. Li, Z. Jing, and G. Chen, "Automatic tongue image segmentation for traditional chinese medicine using deep neural network," *Intelligent Computing Theories and Application*, pp. 247-259, July 2017. DOI: 10.1007/978-3-319-63309-1_23
- [10] B. Lin, J. Xle, C. Li, and Y. Qu, "Deeptongue: tongue segmentation via resnet," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1035-1039, April 2018. DOI: 10.1109/ICASSP.2018.8462650
- [11] C. Zhou, H. Fan and Z. Li, "Tonguenet: accurate localization and segmentation for tongue images using deep neural networks," *IEEE Access* Vol. 7, pp. 148779-148789, Oct. 2019. DOI: 10.1109/ACCESS.2019.2946681
- [12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional Networks for Biomedical Image Segmentation," *International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241, Nov. 2015. DOI: 10.1007/978-3-319-24574-4_28
- [13] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *International Conference on Learning Representations*, May. 2016. arXiv preprint arXiv:1511.07122v1
- [14] A. Paszke, A. Chaurasia, S. Kim, et al., "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," arXiv preprint arXiv:1606.02147, June 2016.
- [15] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 39, no. 4, pp. 640-651, April 2017. DOI: 10.1109/TPAMI.2016.2572683
- [16] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," *IEEE Visual Communications and Image Processing*, pp. 1-4, Dec. 2017. DOI: 10.1109/VCIP.2017.8305148
- [17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE conference on computer vision and pattern recognition*, pp. 770-778, June 2016. DOI: 10.1109/CVPR.2016.90
- [18] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," arXiv preprint arXiv:1506.04579v2, Nov. 2015.
- [19] H. Zhao, J. Shi, X. Qi, et al., "Pyramid scene parsing network," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881-2890. July 2017. DOI: 10.1109/CVPR.2017.660
- [20] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters-improve semantic segmentation by global convolutional network," *IEEE conference on computer vision and pattern recognition*, pp. 4353-4361, July 2017. DOI: 10.1109/CVPR.2017.189
- [21] E. Romera, J.M. Álvarez, L.M. Bergasa, and R. Arroyo, "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, no. 1, pp. 263-272, Jan. 2018. DOI: 10.1109/ITS.2017.2750080
- [22] W. Xiang, H. Mao, and V. Athitsos, "ThunderNet: A Turbo

- Unified Network for Real-Time Semantic Segmentation," IEEE Winter Conference on Applications of Computer Vision, pp. 1789-1796, Jan. 2019. DOI: 10.1109/WACV.2019.00195
- [23] M. Everingham, L.V. Gool, C.K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," International journal of computer vision, Vol. 88, no. 2, pp. 303-338, June 2010. DOI: 10.1007/s11263-009-0275-4
- [24] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C.L. Zitnick, "Microsoft COCO: Common objects in context," European conference on computer vision, pp. 740-755, Sep. 2014. DOI: 10.1007/978-3-319-10602-1_48
- [25] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213-3223, June 2016. DOI: 10.1109/CVPR.2016.350
- [26] H. Chen, S.T. Jung "Enhancement of Tongue Segmentation by Using Data Augmentation," The Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 13. no. 5, pp. 313-322, Oct. 2020. DOI: 10.17661/jkiict.2020.13.5.313
- [27] D. Shi, C. Tang, S.V. Blackley, L. Wang, J. Yang, Y. He, S.I. Bennett, Y. Xiong, X. Shi, L. Zhou, and D. W. Bates, "An annotated dataset of tongue images supporting geriatric disease diagnosis," Data in Brief, Vol. 32, 106153, ISSN 2352-3409, Oct. 2020. DOI: 10.1016/j.dib.2020.106153
- [28] G.H. Kim and D.H. Nam, "The development trends of tongue diagnosis system," The Magazine of the IEIE, Vol. 43, no. 12, 35-43, Dec. 2016.

Authors



Yu-Jie Li received the B.S. degree in Information Engineering from Ludong University, China, in 2008 and received the B.S. and M.S. degrees in Computer Engineering from Wonkwang University,

Korea, in 2008 and 2010, respectively. Yu-Jie Li is currently pursuing the Ph.D. degree in Computer Engineering from Wonkwang University, Korea. She is currently a Lecturer in the Department of Computer Science and Engineering, Weifang University of Science and Technology, China. She is interested in image processing, deep learning and pattern recognition.



Sung-Tae Jung received the B.S., M.S. and Ph.D. degrees in Computer Engineering from Seoul National University, Korea, in 1987, 1989 and 1994, respectively. Dr. Jung is currently a Professor in the Department of

Computer and Software Engineering, Wonkwang University. He is interested in computer vision, artificial intelligence, image processing and computer graphics.