



# FU-net: Multi-class Image Segmentation Using Feedback Weighted U-net

≡ 분류	모델 세그멘테이션로스
≡ 링크	<a href="https://arxiv.org/pdf/2004.13470">https://arxiv.org/pdf/2004.13470</a>
≡ Summary	BRU-net을 기반으로 동적으로 가중된 교차 엔트로피 손실 함수 분할 정확도가 낮은 픽셀에 더 높은 가중치를 할당하면 네트워크가 잘못 예측된 이미지 영역에서 더 많이 학습
👤 발표자	유림 유림 오

- 이 논문을 적합한 이유
  - 픽셀별 예측 정확도에 따라 가중치를 동적으로 조정하는 가중치 교차 엔트로피 손실 함수를 도입하여, 잘못 예측된 이미지 영역에서 더 많은 학습을 가능하게 함

FU-net은 T1 가중 뇌 MRI를 기반으로 종뇌와 피각의 분할을 평가하였으며, 훈련 예제가 적을 때에도 BRU-net과 U-net보다 통계적으로 유의미한 성능 향상을 보임

GitHub에 코드 공개

1. **제안된 방법 - FU-net:** 이 논문에서는 U-net 방법을 개선하여 동적으로 가중치가 부여된 교차 엔트로피 손실 함수를 도입합니다. 각 픽셀의 가중치는 예측 정확도에 따라 계산되며, 높은 정확도를 가진 픽셀에는 낮은 가중치가 부여되고, 반대로 낮은 정확도를 가진 픽셀에는 높은 가중치가 부여됩니다. 이를 통해 네트워크는 잘못 예측된 이미지 영역에서 더 많이 학습할 수 있습니다.

이러한 내용을 통해 FU-net이 의료 이미지 분할에서 효과적인 방법임을 입증하고자 합니다.

## 2.1 네트워크 아키텍처

- **U-net 구조:** U-net 아키텍처는 두 가지 주요 경로로 구성됩니다: 수축 경로와 확장 경로.
  - **수축 경로:** 이 경로는 입력 이미지의 공간 차원을 줄이면서 특징 채널의 수를 증가시킵니다. 이 경로의 각 레이어는 두 개의  $3 \times 3$  합성곱 연산과 ReLU 활성화 함수를 포함합니다. 각 레이어의 출력은 2의 보폭을 가진 최대 풀링을 사용하여 다운샘플링 되어 추상화된 정보만 남깁니다.
  - **확장 경로:** 이 경로는 특징 맵의 공간 차원을 재구성하는 것을 목표로 합니다.  $2 \times 2$  업-합성곱을 사용하여 특징 맵의 크기를 증가시키고, 이전 레이어의 특징 채널 수를 절반으로 줄입니다. 수축 경로와 마찬가지로 각 레이어는 두 개의  $3 \times 3$  합성곱과 ReLU를 포함합니다. 또한, 수축 경로의 특징 맵과 확장 경로의 특징 맵을 결합하는 연결 작업이 수행됩니다. 마지막 레이어는  $1 \times 1$  합성곱을 사용하여 출력 차원을 클래스 수에 맞추고, 소프트맥스 함수를 적용하여 출력 값을  $[0, 1]$  범위로 정규화합니다.
- **손실 함수:** 원래의 U-net은 최적화를 위해 가중치가 부여된 교차 엔트로피 손실 함수를 사용하며, 이 손실 함수의 가중치는 특정 픽셀, 특히 경계 근처의 픽셀 분류의 어려움에 따라 미리 계산됩니다.
- **개선 사항:** 제안된 FU-net은 배치 정규화(BN)와 잔차 블록(RB)을 통합하여 U-net을 개선하여 훈련 효율성과 수렴성을 높입니다. 손실 함수의 가중치는 훈련 중 예측 정확도에 따라 동적으로 계산되어 모델이 어려운 예측 영역에 집중할 수 있도록 합니다.

## 2.2 가중치가 부여된 교차 엔트로피 비용 함수

- **동적 가중치 계산:** 전통적인 방법이 클래스 픽셀 수에 따라 고정된 가중치를 사용하는 것과 달리, FU-net은 각 훈련 반복에서 동적으로 가중치를 계산합니다. 이 접근 방식은 예측 오류가 더 큰 픽셀에 더 높은 가중치를 부여하여 모델이 이러한 영역에서 더 효과적으로 학습할 수 있도록 합니다.
- **가중치 매핑:** 픽셀별 가중치 맵은 예측 확률에 따라 생성됩니다. 예측 정확도가 낮은 픽셀은 더 높은 가중치를 받아, 네트워크가 이러한 영역에 집중하도록 유도합니다. 가중치 함수는 가중치가 특정 범위( $[0.01, 1]$ ) 내에 있도록 정의되어, 잘 예측된 픽셀을 완전히 무시하는 것을 방지합니다.
- **배치 훈련:** 동일한 가중치 매핑 함수가 모든 훈련 이미지에 적용되며, 최적화를 위해 미니 배치 방법이 사용됩니다. 이는 더 큰 잘못된 예측된 영역을 가진 이미지가 훈련 과정에 더 큰 영향을 미치게 하여, 쉬운 예제와 어려운 예제 간의 훈련 균형을 효과적으로 맞추는 것을 목표로 합니다.

## 1. 모델 아키텍처

## 1.1 U-net 구조



Fig. 1. (a) Layer of the original U-net. (b) Layer by adding batch normalization (BN) and residual block (RB).

- 수축 경로 (Contracting Path):

- 이 경로는 입력 이미지를 점진적으로 다운샘플링하여 특징 맵을 생성합니다. 각 레이어는 두 개의 3x3 합성곱(Conv)과 ReLU 활성화 함수를 포함합니다.
- 각 레이어의 출력은 최대 풀링(max pooling)을 통해 다운샘플링됩니다.

- 확장 경로 (Expansive Path):

- 수축 경로에서 얻은 특징 맵을 기반으로 이미지를 복원하는 과정입니다. 각 레이어는 2x2 업컨볼루션(up-convolution)을 사용하여 업샘플링하며, 이전 레이어의 특징 맵과 연결(concatenation)됩니다.
- 이 경로에서도 두 개의 3x3 합성곱과 ReLU가 사용됩니다.

- 최종 레이어:

- 마지막 레이어에서는 1x1 합성곱을 사용하여 출력의 차원을 클래스 수로 변환하고, 소프트맥스(softmax) 함수를 적용하여 각 픽셀의 클래스 확률을 계산합니다.

## 1.2 개선 사항

- 배치 정규화 (Batch Normalization):

- 각 레이어에 배치 정규화를 추가하여 훈련 속도를 높이고, 더 깊은 네트워크의 수렴을 촉진합니다.

- 잔차 블록 (Residual Block):

- 잔차 학습을 통해 더 깊은 네트워크를 효과적으로 훈련할 수 있도록 합니다.

## 2. 손실 함수

FU-net에서 사용되는 손실 함수는 동적으로 가중치가 조정되는 교차 엔트로피 손실 함수입니다. 이 손실 함수는 다음과 같은 방식으로 작동합니다:

- 픽셀별 예측 정확도:

- 각 훈련 반복(iteration)에서 모델의 예측 정확도를 기반으로 픽셀의 가중치를 계산합니다. 잘못 예측된 픽셀에 더 높은 가중치를 부여하여, 네트워크가 이러한 픽셀에서 더 많은 학습을 하도록 유도합니다.

- **가중치 계산:**

- 가중치는 픽셀의 예측 정확도에 따라 동적으로 업데이트되며, 이는 훈련 과정에서 지속적으로 조정됩니다.

- **손실 함수의 수식:**

- 손실 함수는 다음과 같이 표현됩니다:

$$E = \sum_{x \in \Omega} w(x) \log(p_{l(x)}(x))$$

여기서  $p_l(x)$ 는 픽셀  $x$ 의 실제 클래스에 대한 예측 확률이며,  $w(x)$ 는 해당 픽셀의 가중치입니다. 이 방식은 특히 클래스 불균형이 심한 데이터셋에서 효과적입니다.

### 웨이티드 로스 예시

## 결론

FU-net은 U-net의 기본 구조를 유지하면서도, 배치 정규화와 잔차 블록을 추가하고, 동적으로 조정되는 가중치 교차 엔트로피 손실 함수를 통해 성능을 크게 향상시킵니다. 이러한 개선 사항들은 특히 의료 이미지 분할과 같은 도전적인 작업에서 유용하게 작용합니다.

## 유넷 과의 차이점

### 1. 동적 가중치 손실 함수:

- **FU-net:** 픽셀의 예측 정확도에 따라 동적으로 가중치를 조정하여 손실 함수를 개선합니다. 잘못 예측된 픽셀에 더 높은 가중치를 부여하여 네트워크가 이러한 픽셀에서 더 많은 학습을 하도록 유도합니다.
- **U-net:** 고정된 가중치를 사용하여 손실 함수를 정의하며, 가중치는 사전에 결정되어 있습니다.

### 2. 배치 정규화 (Batch Normalization):

- **FU-net:** 각 레이어에 배치 정규화를 추가하여 훈련 속도를 높이고, 더 깊은 네트워크의 수렴을 촉진합니다.
- **U-net:** 배치 정규화가 포함되어 있지 않습니다.

### 3. 잔차 블록 (Residual Block):

- **FU-net:** 잔차 블록을 추가하여 더 깊은 네트워크를 효과적으로 훈련할 수 있도록 합니다.
- **U-net:** 잔차 블록이 포함되어 있지 않습니다.

### 4. 성능 향상:

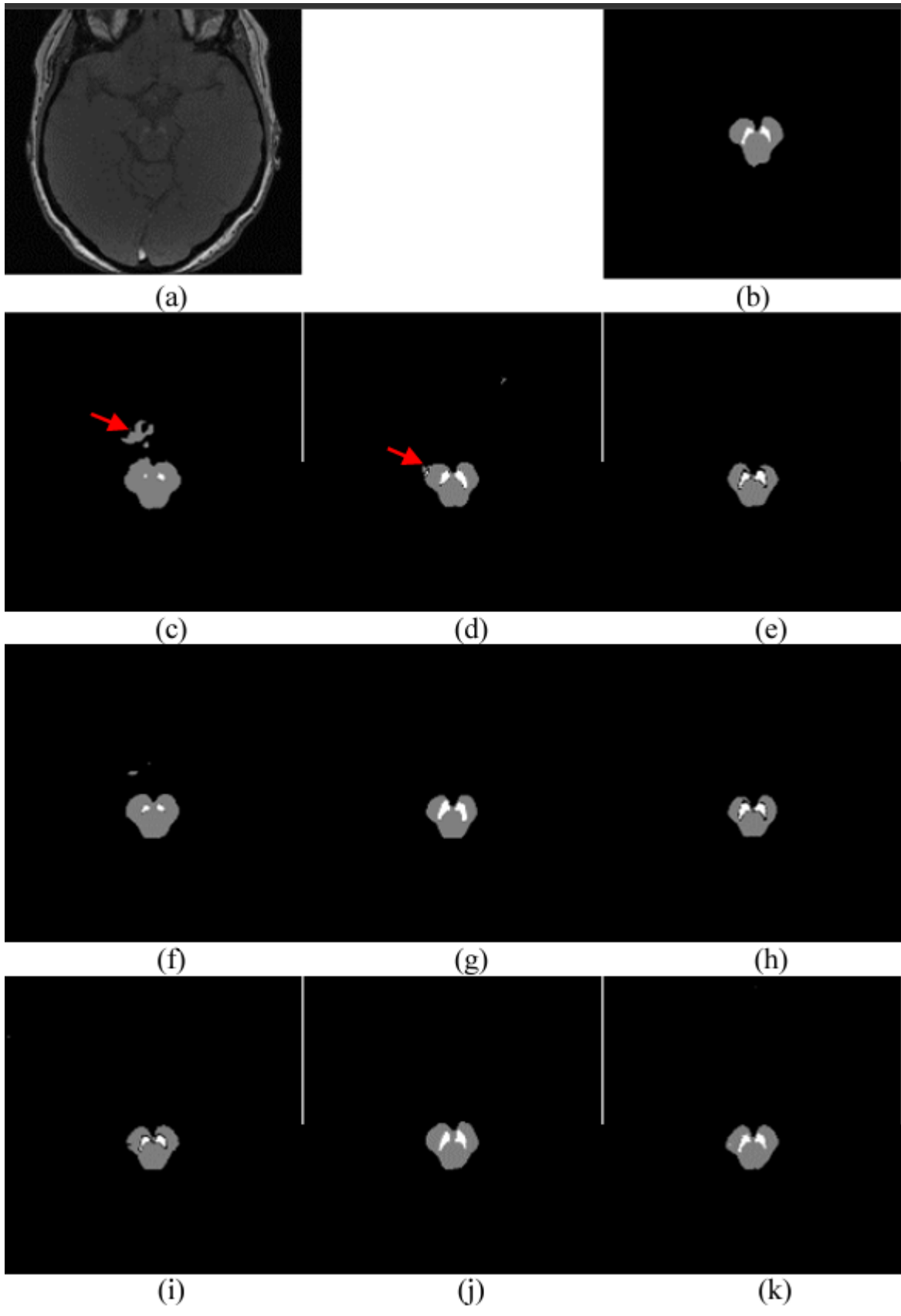
- **FU-net:** 클래스 불균형이 심한 데이터셋에서 성능이 크게 향상되며, 더 적은 수의 훈련 샘플로도 효과적으로 학습할 수 있습니다.
- **U-net:** 클래스 불균형 문제에 대한 특별한 대처가 없습니다.

Number of training/testing examples	Method	Mean of DC $\pm$ Std	
		MB	SN
200/100	U-net	0.9000 $\pm$ 0.03	0.7095 $\pm$ 0.17
	BRU-net	0.8775 $\pm$ 0.14	0.7164 $\pm$ 0.18
	FU-net	0.8929 $\pm$ 0.05	<b>0.7563<math>\pm</math>0.15</b>
100/200	U-net	0.8584 $\pm$ 0.18	0.7022 $\pm$ 0.18
	BRU-net	0.8550 $\pm$ 0.16	0.7005 $\pm$ 0.15
	FU-net	0.8710 $\pm$ 0.15	<b>0.7575<math>\pm</math>0.16</b>
50/250	U-net	0.8135 $\pm$ 0.19	0.4831 $\pm$ 0.26
	BRU-net	0.8088 $\pm$ 0.15	0.6387 $\pm$ 0.17
	FU-net	0.8182 $\pm$ 0.20	<b>0.7087<math>\pm</math>0.24</b>
		P values of paired t-test	
200/100	U-net/ BRU-net	0.0589	0.6086
	BRU-net/ FU-net	0.1260	0.0026
100/200	U-net/ BRU-net	0.6050	0.8706
	BRU-net/ FU-net	0.6062	<0.0001
50/250	U-net/ BRU-net	0.5725	<0.0001
	BRU-net/ FU-net	0.3138	<0.0001

50개의 이미지만 사용하여 훈련할 때 FU-net의 성능은 여전히 높았으며(DC=0.7087), 이는 BRU-net(DC=0.6387) 및 U-net(DC=0.4831)보다 훨씬 높은 수치입니다.

표 1의 결과에서 볼 수 있듯이, 세 가지 방법 모두 MB(중뇌) 분할에 대해 훈련 샘플 수에 관계없이 유사한 분할 성능을 달성했으며, 통계적 유의성은 없었습니다. 그러나 SN(흑질) 클래스의 경우, 픽셀 수가 MB 클래스보다 훨씬 적고 분할하기 더 어려운 특성을 가지고 있기 때문에, 제안된 FU-net은 모든 실험에서 U-net 및 BRU-net 방법보다 일관되게 우수한 성능을 보였습니다. 특히 50개의 이미지만 사용하여 훈련할 때 FU-net의 성능은 여전히 높았으며(주사위 계수  $DC=0.7087$ ), 이는 BRU-net( $DC=0.6387$ ) 및 U-net( $DC=0.4831$ )보다 훨씬 높은 수치입니다.

즉, FU-net은 적은 훈련 샘플로도 더 나은 성능을 발휘하며, 특히 어려운 분할 작업에서 효과적임을 보여줍니다.



우리는 제안한 방법의 장점을 보여주기 위해 몇 가지 시각적 예시를 제공합니다. 그림 3에서는 50, 100, 200개의 훈련 이미지를 기반으로 한 예시 이미지의 분할 결과를 제시합니다. 그림 3(a)와 (b)는 각각 원본 이미지와 정답 주석(ground truth annotation)입니다. 그림 3(b)에서 어두운 영역은 중뇌(MB)를 나타내고, 밝은 영역은 흑질(SN)을 나타냅니다. 그림 **3(c), (d) 및 (e)**는 각각 **U-net, BRU-net 및 FU-net의 분할 결과입니다.**

- (a) 원본 이미지
- (b) 정답 주석(ground truth)
- (c), (d), (e) 각각 50개의 훈련 샘플을 사용하여 U-net, BRU-net, FU-net에 의해 생성된 분할 결과
- (f), (g), (h) 각각 100개의 훈련 샘플을 사용하여 U-net, BRU-net, FU-net에 의해 생성된 분할 결과
- (i), (j), (k) 각각 200개의 훈련 샘플을 사용하여 U-net, BRU-net, FU-net에 의해 생성된 분할 결과