# 딥러닝 기반의 표 경계선 히트맵 회귀를 이용한 표의 구조 인식

이은지 [1], 박재우 [1], 구형일 [2], *조남익 [1]
[1] 서울대학교 전기정보공학부, [1] 뉴미디어통신공동연구소, [2] 아주대학교 전자공학과
jane0119@snu.ac.kr, bjw0611@snu.ac.kr, hikoo@ajou.ac.kr, *nicho@snu.ac.kr

# Table Structure Recognition using Borderline Heatmap Regression

EunJi Lee[1], Jaewoo Park[1], Hyung Il Koo[2], *Nam Ik Cho[1]
[1]Department of ECE, INMC, Seoul National University, [2]Department of ECE, Ajou Univiersity

## 요 약

본 논문에서는 딥러닝을 기반으로 문서영상에서 표 안의 셀 경계선을 히트맵 회귀(heatmap regression)로 추정함으로써 표의 구조를 인식하는 방법을 제안한다. 표는 기본적으로 행과 열로 이루어져 있기 때문에, 제안하는 방법에서는 먼저 1 차원 벡터 형태로 세로/가로 방향의 행/열 경계선 위치를 찾고, 이에 병합된 셀을 처리하기 위해 경계선이 그어져야 할 위치를 2 차원으로 추정한 결과를 적용하여 온전한 표의 경계선을 구한다. 이러한 구조를 통해 제안하는 방법은 표의 행과 열에 대한 정보를 효과적으로 이용함과 동시에, 복잡한 후처리 없이 병합된 셀을 처리할 수 있는 이점을 보인다. 실험은 1 차원의 행/열 경계선 위치를 반영하는 두 가지 방식에 대해 PubTabNet[11]에 대해 진행하여 결과를 보였다.

## 1. 서론

문서 이미지를 디지털화하는 작업은 대량의 문서를 다루는 산업 분야뿐 아니라 개개인의 일상에서도 중요한 역할을 차지한다. 문서의 디지털화 과정에는 문서 이미지 전처리, 레이아웃 분석, OCR 등의 여러 가지 하위 작업들이 있는데[8,12], 그 중에서도 문서 내 표의 구조를 인식하는 작업은 중요한 역할을 한다. 표는 자료를 직관적으로 파악하기 쉽게 정렬된 구조를 가지기 때문에, 표에 핵심적인 정보를 표기하는 문서가 많다. 그러므로 표의 구조를 인식하여 디지털화하는 작업은 문서를 이해하는 데 있어 빠질 수 없는 기능이다.

문서 내 표의 구조 인식 문제는 표 영상을 입력으로 받아 표의 구조를 기계가 이해할 수 있는 형식으로 출력하는 것인데, 보통 html 형태로 표현된다. 표는 기본적으로 행과 열로 나눌 수 있는 격자 형태로 이루어져 있지만, 많은 표에는 병합된 셀이 존재하기 때문에, 표의 구조를 단순한 격자 형태로 바라볼 수 없다. 이러한 표의 복잡한 특성을 고려하면서도 표의 격자성을 활용하기 위해, 본 논문에서는 표의 구조 인식 문제를 셀을 나누는 경계선을 긋는 문제로 바라보았다.

다른 컴퓨터 비전 분야와 마찬가지로, 최근 표 구조 인식 분야에서도 딥러닝 기반의 접근이 활발하게 연구되고 있다[2,4-7,9-11]. 기존 딥러닝 기반의 표 인식 방법은 대부분 1) 표의 구성 요소 검출 및 2) 구성 요소 간의 관계 파악을 통한 구조 인식의 두 단계로 이루어진다. 2)의 구조 인식은 대부분 규칙 기반의 후처리 기법으로 진행되었다. 1)의 구성 요소를 기준으로 기존 방법론은 크게 두 갈래로 나뉜다. 초기 딥러닝 기반 방법들은 대개 표의 구성 요소를 행/열로 삼아 진행하였다[2,4,6-7,9]. 해당 계열의 방법들은 객체 검출 혹은 영상 분할 네트워크를 이용하여 행(열)의 영역 또는 행(열)의 경계선을 찾아 표를 구성 요소 단위로 나눈 이후 표의 구조에 대한 규칙을 정의하여 휴리스틱하게 구조 파악을 진행하였다. 그러나 구성 요소를 행(열)으로 삼는 대부분의 방법들은 병합된

셀을 고려하지 않거나[2,7] 혹은 병합된 셀 처리를 위해 복잡한 후처리 기법을 적용하였다[4].

병합된 셀을 쉽게 처리하기 위해 제안된 최근의 딥러닝 기반 방법들은 대부분 셀을 구성 요소로 삼아 객체 검출 네트워크를 이용하여 바로 셀을 검출하고 이를 바탕으로 표 구조 파악을 진행하였다[5,10-11]. 그러나 셀을 직접적으로 검출하는 방법들은 표의 격자성을 충분히 이용하지 못하는 한계를 가진다.

기존 방법들 중 하나인 SPLERGE[9]에서는 위 두 가지 한계점을 보완하여 행(열) 정보를 기반으로 하면서 셀 병합을 직관적으로 처리하였다. SPLERGE 의 구성은 먼저 표의 행(열) 경계선을 구해 표를 격자 단위로 분할(분할 네트워크)하고 이후 인접한 격자들을 병합해야 할 지를 판별(병합 네트워크)하는 두 단계로 이루어져있다. 그러나 SPLERGE 는 분할 네트워크와 병합 네트워크를 각각 학습함으로 각 네트워크가 서로 영향을 줄 수 없어 성능 향상에 한계가 있었다.

본 논문에서는 위의 행(열) 기반 방법과 셀 기반 방법의 장점을 접목하여 행(열) 구성 정보를 반영한 구조로 네트워크를 설계함과 동시에 히트맵을 이용한 간단한 방식으로 병합된 셀을 처리하는 표 구조 인식 방법을 제안한다. 제안하는 방식의 전체적인 구성은 SPLERGE[9]를 따른다. 제안하는 방식은 먼저 표의 행(열) 경계선을 1 차원 형태로 구하는 분할 디코더와 이 정보를 활용하여 병합된 셀이 반영된 최종적인 행(열) 경계선 히트맵을 출력하는 병합 디코더 두 단계로 구성된다. 기존 SPLERGE 에서는 분할 네트워크와 병합 네트워크를 각각 따로 학습한 반면, 본 논문에서는 분할 디코더와 병합 디코더를 같이 학습할 수 있는 구조를 제안한다. 실험은 공개 데이터셋 중 가장 큰 데이터셋인 PubTabNet[11]에 대해 진행하였고 이를 통해 제안하는 방법의 성능을 확인하였다.

## 2. 경계선 검출을 통한 표 구조 인식

### 2.1. 네트워크 구조

제안하는 표 구조 인식 네트워크는 문서 내의 표 영상을 입력으로 받아 행(열) 방향의 경계선 히트맵을 출력한다. 이 때, 두 단계로 나누어 히트맵을 출력하는 네트워크를 설계하였다. 두 단계는 다음과 같다: 1) 세로/가로 위치에 행(열) 경계선이 존재하는지 나타내는 1 차원 벡터를 구하고, 이후 2) 각 좌표가 행(열) 경계선에 포함되는지를 나타내는 2 차원 히트맵을 구해 최종 출력으로 삼는다. 행과 열 경계선 히트맵은 각각 독립된 네트워크로 구하며, 각 네트워크는 동일한 구조를 가진다. 설명의 단순화를 위해 행 경계선 출력 네트워크에 대해 기술한다. 행 경계 네트워크의 전체 구조는 Figure 1 에 나타나있다.

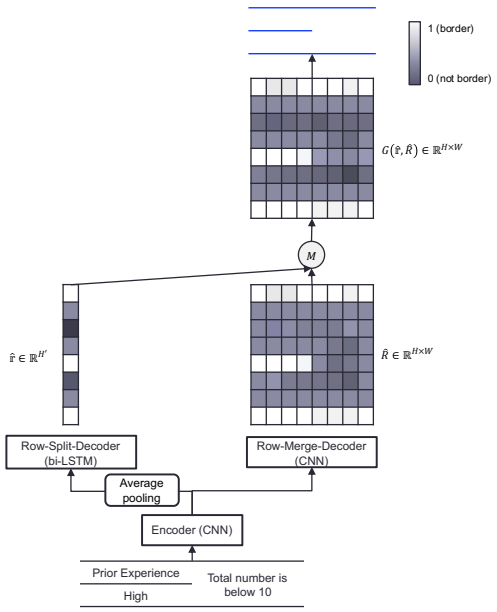


Figure 1. 행 경계선 네트워크 구조

먼저 표 영상 $I \in \mathbb{R}^{H\times W\times 3}$ 을 인코더 에 통과시켜 표 이미지에 대한 특성맵 $\hat{F} \in \mathbb{R}^{H'\times W'\times 256}$를 얻는다. 이후 $\hat{F}$를 세로축 방향으로 평균을 구하여 $\bar{F} \in \mathbb{R}^{H'\times 256}$ 을 추출하고, 이를 양방향 LSTM 기반의 행 분할 디코더(Row-Split-Decoder)에 통과시켜 행 경계선 벡터 $\hat{r} \in \mathbb{R}^{H'}$을 얻는다. 그리고 최종 출력인 2 차원 행 경계선 히트맵을 출력하기 위해 $\hat{F}$를 입력으로 하여 행 병합 디코더(Row-Merge-Decoder)를 통과시켜 히트맵 $\hat{R} \in \mathbb{R}^{H\times W\times 1}$ 을 얻는다. 여기서 $\hat{R}$ 에서 행 경계에 있다고 출력한 좌표들 중 $\hat{r}$에서 행 경계선이라고 판단한 부분만 활성화하기 위하여 마스킹 모듈 $M(\cdot,\cdot)$를 통해 최종 출력 행 경계선 히트맵 $M(\hat{r},\hat{R})$을 얻는다.

마스킹 모듈은 $M_{hard}$, $M_{soft}$의 2 가지 방식으로 실험하였다. 먼저 $M_{hard}$ 는 $\hat{r}$의 값 중 임계치 $t$ 이상인 값에 해당하는 가로축에 대해서만 $\hat{R}$값을 사용하는 방식이다. 반면 $M_{soft}$는 $\hat{r}$의 값 중 $t$이상인 값에 해당하지 않더라도 최종 출력에 반영하여 $\hat{R}$의 모든 영역을 경사도 학습에 반영한 방식이다. 이를 식으로 나타내면 다음과 같다.

$$M_{hard}(\mathrm{x},X) = tile\big(1(\mathrm{scale(x)} > t)\big) \otimes X \quad (1)$$

$$M_{soft}(\mathrm{x},X) = tile\big(\mathrm{scale(x)}\big) \otimes X \quad (2)$$

이 때 $tile(\mathrm{x})$은 x 벡터를 (행 단위의 경우) 가로 방향으로 복사하는 것을, $\mathrm{scale(x)}$는 x를 X의 크기에 맞게 조절하는 하는 과정을, $1(\cdot)$ 은 지시 함수를, $\otimes$ 는 아다마르(hadamard) 곱을 의미한다.

열 방향에 대해서도 위와 동일한 과정으로 1 차원 열 경계선 벡터 $\hat{c}$ 및 열 경계선 히트맵 $\hat{C}$를 출력한다.

이와 같이 세로 방향에 대해 행 경계선을 구별하고 이를 2 차원의 행 경계선 히트맵에 반영해줌으로써 표가 행으로 구성되어 있음을 충분히 활용함과 동시에 병합된 셀도 효율적으로 처리하였다.

### 2.2. 학습 과정

표 구조 인식을 위해 사용한 네트워크는 인코더, (행, 열) 분할 디코더, 및 (행, 열) 병합 디코더의 5 개이다. 먼저 인코더 및 분할 디코더를 학습하기 위해 손실 함수를 설계하였다. 행(열)의 경계 부분을 구별하는 문제에서는, 경계인 영역이 경계가 아닌 영역보다 적게 존재하기 때문에 클래스 불균형 문제가 발생한다. 이를 해결하기 위해 focal loss[3]를 사용하였다.

$$\mathcal{L}_{row_{1d}} = E\left(FL\left(r_j, \hat{r_j}\right)\right), \mathcal{L}_{col_{1d}} = E\left(FL(c_i, \hat{c_i})\right) \quad (3)$$

$$FL(p, \hat{p}) = \begin{cases} -\alpha(1-\hat{p})^{\gamma}\log(\hat{p}) & p = 1 \\ -\alpha\hat{p}^{\gamma}\log(1-\hat{p}) & p = 0 \end{cases} \quad (4)$$

이 때 $\mathbb{r}, \mathbb{c}$는 각각 참값 행, 열 경계 벡터를, $r_j, \hat{r_j}, c_i, \hat{c_i}$ 는 각각 $\mathbb{r}, \hat{\mathbb{r}}, \mathbb{c}, \hat{\mathbb{c}}$의 원소를, $FL$은 Focal Loss 를 나타내며, $\alpha$와 $\gamma$는 각각 0.25 와 2 로 사용하였다.

그리고 인코더와 병합 디코더를 학습하기 위해 mean squared error 손실 함수를 사용하였다.

$$\mathcal{L}_{row_{2d}} = MSE\left(R, \hat{R}\right), \mathcal{L}_{col_{2d}} = MSE\left(C, \hat{C}\right) \quad (5)$$

이 때 $R, C$ 는 각각 참값 행(열) 경계 히트맵을 나타내며, 학습에 사용된 최종 손실 함수는 다음과 같다.

$$\mathcal{L} = \mathcal{L}_{row_{1d}} + \mathcal{L}_{col_{1d}} + \mathcal{L}_{row_{2d}} + \mathcal{L}_{col_{2d}} \quad (6)$$

## 3. 실험 결과 및 분석

### 3.1. 데이터셋

제안하는 방법의 성능 검증을 위해 과학 논문 내의 표로 구성된 PubTabNet[11] 데이터에 대해 표 구조 인식 실험을 진행하였다. 구성 표들은 생의학 문서를 제공하는 사이트인 PMCOA[1]의 문서에서 추출한 것이다. 데이터는 이미지와 이에 대응되는 html 토큰으로 구성되어 있고, html 토큰이 부정확한 샘플은 제외하고 사용하였다. 따라서, 제공된 데이터의 학습/검증/실험 셋 각각 표 500,777/9,115/9,138 개 중 표의 구조가 정확하지 않은 것들을 제외한 456,101 개를 학습에 사용하였고, 실험 데이터의 정답이 제공되지 않아 실험 데이터로는 검증 데이터인 표 9,115 개를 사용하였다.

### 3.2. 구현 세부 사항

학습과 실험 모두 입력 영상은 너비와 높이 중 긴 수치가 512 가 되도록 조정한 후 제로 패딩하여 512x512 크기의 영상을 사용하였다. 인코더는 ResNet-101[1]의 conv2_x 부분까지 사용하였고 ImageNet 으로 사전 훈련된 모델로 초기화하였다. 인코더의 출력 크기( $H', W'$ )는 입력 크기의 1/4 배인 128x128 이고 채널은 256 이다. Split-Decoder 는 2 층의 양방향 LSTM 과 1 층의 완전 연결 레이어로 구성되어 있으며, LSTM 의 출력 벡터는 128 채널, 완전 연결 레이어의 출력 값은 1 채널이며 시그모이드 함수를 거친 값이다. 병합 디코더는 3 개의 블록으로 이루어져 있고, 각 블록은 3x3 합성곱-배치 정규화-정류선형유닛-2 배 업샘플링-3x3 합성곱-배치 정규화-정류선형유닛으로 구성되어 있다. 단, 마지막 블록에서는 업샘플링 과정을 생략, 두번째 배치정규화-정류선형유닛 대신 시그모이드를 사용하였다. 합성곱 레이어는 팽창 합성곱 레이어를, 업샘플링은 최근접 보간법을 사용하였다. 학습 단계의 히트맵 마스킹 모듈은 $M_{hard}, M_{soft}$ 두 가지 모두 사용하였고, $M_{hard}$ 에서의 임계값은 0.5 로 설정하였다. 실험 단계에서의 히트맵 마스킹 모듈 $M(\cdot,\cdot)$ 는 임계값 0.1 의 $M_{hard}$ 방식을 사용하였다.

### 3.3. 실험 결과

실험은 경계선 히트맵 마스킹 연산 $M(\cdot,\cdot)$을 두 가지(hard, soft) 방식으로 학습한 모델에 대해 진행하여 비교하였다. 성능은 html 파일을 트리 구조로 변환하여 표 트리 간의 편집 거리를 기반으로 측정되는 TEDS[11]를 이용하였고, 셀 안의 내용을 제외한 표의 구조에 대한 성능을 측정하였다(TEDS-struc.). Table 1 에 나타나듯이, soft 방식이 hard 방식보다 높은 성능을 보였다. 실험을 통해 병합 디코더의 출력 히트맵 좌표 중 (분할 디코더가) 양성으로 판단한 영역에만 손실 함수를 반영하는 것보다는 전체 영역을 경사도 학습에 사용하는 것이 성능향상에 기여하는 것을 확인하였다.

Table 1. 마스킹 기법에 따른 성능 차이

| | TEDS-struc. (%) |
|---|---|
| Hard 마스킹 | 76.6 |
| Soft 마스킹 | 90.4 |

또한 표 이미지에서 구조를 추정한 결과를 Figure 2 에

[1] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

나타내었다. 두번째 행과 같이 셀 내부에 여러 줄의 텍스트 라인이 있는 경우에 대해서도 soft 마스킹이 hard 마스킹에 비해 잘 동작함을 확인하였다. 하지만 마지막 행에서 보이듯, hard 마스킹과 soft 마스킹 방식 모두 행을 과잉 분할하는 한계가 존재한다. 향후 표의 시각 정보와 더불어 언어 정보를 이용함으로써 과잉 분할과 같은 문제를 해결할 것이다.

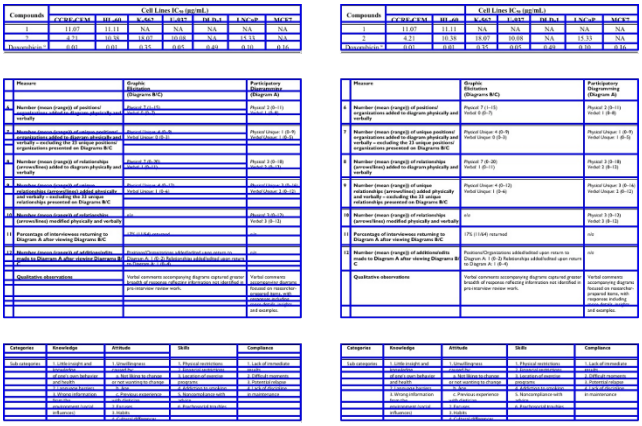

Figure 2. 표 구조 인식 결과
왼쪽: hard 마스킹 방식, 오른쪽: soft 마스킹 방식

## 4. 결론

본 논문에서는 표의 행(열) 단위의 정보를 효과적으로 활용하면서도 병합된 셀을 간단하게 처리할 수 있는 셀 경계선 히트맵 회귀 기반의 표 구조 인식 방법을 제안하였다. 이는 기존의 행/열 기반의 방법의 단점과 셀 기반 방법의 단점을 보완하기 위해 각 방법의 장점을 합쳐 구성한 것이다. 기존 방법 중에도 분할 모듈과 병합 모듈로 구성된 비슷한 시도가 있었으나 각 모듈을 각자 학습한다는 한계가 존재했고, 제안하는 기법에서는 두 모듈을 한번에 학습할 수 있는 구조를 제안하였다. 또한 분할 모듈의 결과를 반영 모듈에 반영하는 두 가지 방식을 새롭게 제안하여 벤치마크 데이터셋에 대한 실험을 통해 성능을 확인하였다.

## 감사의 글

이 논문은 2021 년도 BK21 FOUR 정보기술 미래인재 교육연구단 및 LG AI 연구원의 지원을 받아 이루어졌음.

## 참고문헌

[1] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[2] Khan, Saqib Ali, et al. "Table structure extraction with bi-directional gated recurrent unit networks." *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019.

[3] Lin, Tsung-Yi, et al. "Focal loss for dense object detection." *Proceedings of the IEEE international conference on computer vision*. 2017.

[4] Paliwal, Shubham Singh, et al. "Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images." *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019.

[5] Raja, Sachin, Ajoy Mondal, and C. V. Jawahar. "Table structure recognition using top-down and bottom-up cues." *European Conference on Computer Vision*. Springer, Cham, 2020.

[6] Schreiber, Sebastian, et al. "Deepdesrt: Deep learning for detection and structure recognition of tables in document images." *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*. Vol. 1. IEEE, 2017.

[7] Siddiqui, Shoaib Ahmed, et al. "Rethinking semantic segmentation for table structure recognition in documents." *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019.

[8] Smith, Ray. "An overview of the Tesseract OCR engine." *Ninth international conference on document analysis and recognition (ICDAR 2007)*. Vol. 2. IEEE, 2007.

[9] Tensmeyer, Chris, et al. "Deep splitting and merging for table structure decomposition." 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.

[10] Zheng, Xinyi, et al. "Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021.

[11] Zhong, Xu, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. "Image-based table recognition: data, model, and evaluation." *European Conference on Computer Vision*. Springer, Cham, 2020.

[12] Zhong, Xu, Jianbin Tang, and Antonio Jimeno Yepes. "Publaynet: largest dataset ever for document layout analysis." *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019.