

Joint Table and Column Segmentation for Automated Document Information Extraction Using a Group-Normalized ResNet-Transformer U-Net Architecture

Ngabirano Buregeya Hubert

Faculty of Computing and Information Sciences
University of Lay Adventists of Kigali (UNILAK)
Kigali, Rwanda

Enan Muhire Nyesheja

Faculty of Computing and Information Sciences
University of Lay Adventists of Kigali (UNILAK)
Kigali, Rwanda

Mungwarakarama Irene

Faculty of Computing and Information Sciences
University of Lay Adventists of Kigali (UNILAK)
Kigali, Rwanda

Ntivuguruzwa Jean De La Croix

Faculty of Computing and Information Sciences,
University of Lay Adventists of Kigali (UNILAK), Kigali, Rwanda
Department of Business Information Technology,
University of Rwanda, Kigali, Rwanda

Corresponding Author: jc.ntivuguruzwa@ur.ac.rw

Abstract- The growing adoption of automated information extraction in modern enterprises has increased the importance of accurately segmenting tables and their columns from document images. This task remains challenging due to the coexistence of tables with textual and graphical content, as well as substantial variations in table dimensions, layouts, and boundary structures. To address these challenges, we introduce ResForTable, a U-Net-based architecture featuring a pretrained group-normalized ResNet-50 encoder and two transformer-enhanced decoder branches dedicated to table and column segmentation. Evaluation on the Marmot dataset yields DICE scores of 0.9352 for table segmentation and 0.9045 for column segmentation. ResForTable consistently surpasses TransTab across the evaluated metrics while reducing computational requirements by 42.7% in terms of FLOPs compared with a comparable VGG-19-based architecture. Further analysis identifies a persistent difference between overlap-based DICE and boundary-based ASSD measurements across both segmentation branches. Based on Weyl's Tube Formula, the DICE deficit is modeled as a function of ASSD and the mask perimeter-to-area ratio (P/A). The statistically significant relationship between DICE deficit and P/A indicates that DICE can penalize geometric shape even when segmentation boundaries are accurately positioned. These findings emphasize the need to employ complementary evaluation metrics for more reliable assessment of table and column segmentation.

Keywords: *Table Segmentation; Column Segmentation; Document Image Analysis; ResNet-Transformer; Deep Learning*

1. INTRODUCTION

Enterprise operations increasingly depend on extracting information from large volumes of documents. However, much of this extraction is still performed manually, resulting in

inefficient use of human resources and considerable time consumption [1]. These inefficiencies can subsequently increase the time required to deliver services [2], [3]. Automating and digitizing document information extraction offers an effective

means of addressing these challenges by reducing manual effort and improving processing efficiency [1]–[3]. Tables constitute an important component of enterprise documents because they provide a compact and structured representation of information [4]. Consequently, automated extraction of information from document tables requires reliable identification and segmentation of both table regions and their constituent columns. This task is challenging because tables can occur alongside text blocks and graphical elements within the same document page. Moreover, tables exhibit considerable variation in dimensions, structural organization, layouts, and boundary delineation, further increasing the difficulty of accurate segmentation.

A variety of machine learning and deep learning models have been developed for document segmentation tasks. Their performance is generally assessed by comparing predicted segmentation regions with corresponding ground-truth regions using specialized evaluation metrics. These metrics can broadly be categorized into overlap-based and boundary-based measures. Overlap metrics quantify the degree of spatial agreement between predicted and reference regions, with the Dice Similarity Coefficient (DICE) being one of the most widely used measures. DICE ranges from 0, indicating no overlap, to 1, representing complete agreement between the predicted and ground-truth regions [5]. Boundary-based metrics, in contrast, assess the spatial alignment between the boundaries of predicted and reference regions. The Average Symmetric Surface Distance (ASSD) is a commonly adopted boundary metric that calculates the average distance between points on the predicted boundary and the corresponding ground-truth boundary in both directions [6]. Although overlap metrics are effective for evaluating segmentation quality, they tend to penalize smaller and thinner structures more strongly

than larger regions [7]. This characteristic is particularly relevant to document analysis because table columns generally occupy smaller and thinner regions than complete tables.

To address the challenges associated with simultaneous table and column segmentation, we propose ResForTable, an end-to-end joint segmentation architecture inspired by the U-Net framework [8]. The proposed model is designed to learn representative features for both tables and columns while capturing global relationships among different elements within document images. ResForTable employs a pretrained group-normalized ResNet-50 backbone [9], [10] as its encoder, followed by two dedicated decoder branches. The first branch performs table segmentation, whereas the second performs column segmentation. Each decoder incorporates a Vision Transformer (ViT) encoder [11], [12] to capture global contextual relationships within the extracted representations. The resulting feature representations are progressively upsampled using transposed convolutions and subsequently transformed into segmentation masks through a sigmoid activation function. The proposed architecture is evaluated using the Marmot dataset, where it demonstrates competitive table segmentation performance relative to other U-Net-based architectures while providing improved computational efficiency.

In addition to evaluating segmentation performance, this study investigates the relationship between overlap-based and boundary-based evaluation measures. Specifically, a first-order approximation of Weyl's Tube Formula for planar curves [13] is employed to characterize the Dice deficit, defined as Δ , as a function of the Average Symmetric Surface Distance (ASSD) and the perimeter-to-area ratio ρ of the ground-truth region. The relationship between these quantities provides a geometric basis for

examining whether DICE is influenced by the shape of a segmented region independently of boundary-placement accuracy. To empirically investigate this relationship, partial correlation is calculated between Dice deficit and P/A while controlling for ASSD. The resulting statistically significant relationship provides evidence that DICE is sensitive to the geometric shape of segmentation regions, even when differences in boundary placement are accounted for. The principal contributions of this study are summarized as follows:

1. Joint Table and Column Segmentation Model: We propose ResForTable, an end-to-end dual-branch architecture for jointly segmenting tables and columns, achieving strong segmentation performance on the Marmot dataset while maintaining computational efficiency.
2. Analysis of Dual-Branch Architecture Design: We demonstrate that a canonical fully convolutional module configuration for dual-branch segmentation does not provide an overlap-based performance advantage.
3. Computationally Efficient Encoder: We demonstrate that replacing the conventional VGG-19 backbone with a group-normalized ResNet-50 reduces computational requirements by 42.7% without compromising segmentation accuracy.
4. Geometric Analysis of DICE: We provide empirical evidence that the Dice coefficient exhibits an inherent geometric bias against regions with high perimeter-to-area ratios, such as document columns, independently of boundary-placement accuracy.

The remainder of this paper is organized as follows. Section 2 reviews existing research on table and column segmentation and discusses the identified research gaps. Section 3 presents the proposed ResForTable

architecture, loss function, and implementation details. Section 4 describes the dataset, evaluation metrics, geometric analysis, and experimental configuration. Section 5 reports the experimental results. Section 6 discusses the findings and limitations of the study, while Section 7 concludes the paper.

2. RELATED WORKS

2.1 Heuristic and early learning-based table detection

Before data inferred approaches, practitioners used metadata and heuristic methods to detect tables. Pyredd and Croft [14] proposed a textual tabular data extraction framework using structural information from tables and their fields. Cesarini et al. [15] were the first to model table detection as a supervised learning problem incorporating hierarchical feature extraction, handling variation and complexities across tables.

Kasar et al. [16] used horizontal and vertical line delineators that split an image into regions which a Support Vector Machine (SVM) [17] classified as either containing a table or not. Costa e Silva [18] tackled table detection using a probabilistic graph model encompassing a joint probability distribution of observed table elements and hidden Markov models. These methods however, only worked for somewhat predictable documents. They were effective for documents with ruling lines, consistent whitespace, or accessible metadata. The methods' performance plummeted when faced with borderless tables or table layouts that drifted from assumptions and rules.

2.2 Deep learning for table detection and segmentation

Deep learning replaced heuristic and rule-based methods with learned representations. Deep learning approaches fall into two main categories. The first category of approaches

uses bounding boxes. Tables are seen as objects to be localised. Schreiber et al. [19] proposed DeepDeSRT that applied a region-based detector to table detection and structure recognition. Siddiqui et al. [20] proposed DeCNT which used deformable convolutions for the detector as opposed to standard convolutions. This way the receptive field adapts to extensive variation in table aspect ratio. Prasad et al. [21] proposed an instance segmentation model called CascadeTabNet. This approach coupled a cascade detector with transfer learning. Agarwal et al. [22] argued that a model trained at a single 0.5 intersection over union (IoU) produces loose localisation. For this reason, they introduced CDeC-Net, a composite deformable cascade network which trains a sequence of detectors at increasing IoU thresholds. Smock et al. [23] released PubTables-1M and a transformer-based detector for table extraction. The second category of approaches assigns a label to every pixel and formulates table detection as a semantic segmentation task. Paliwal et al. [24] proposed TableNet which uses a VGG-19 backbone beneath two decoder branches. One branch predicts the tables mask, and the other predicts the column mask. TableNet established the dual-branch design for table and column segmentation that later work inherited. Wang et al. [25] proposed TransTab, which retains TableNet’s topology and inserts transformer encoders into each branch, so that the branches can model long range dependencies between the page elements. Sakuhara et al. [26] reevaluated TableNet on the Marmot benchmark using an automated LaTeX pipeline that compares document pages to their aligned ground truth masks. The authors reported performance in terms of the pixel-wise XOR error rate. This paper compares ResForTable segmentation performance to other segmentation methods that score pixel level agreement.

2.3 Segmentation metrics and their geometric behaviour

Segmentation performance is normally evaluated using overlap and boundary metrics of which DICE and ASSD are representatives respectively. Taha and Handbury [27] catalogued these metrics and reported that overlap measure behaves differently according to the size and shape of the target structure. Maier-Hein et al. [28] and Reinke et al. [7] convey that overlap metrics optimise for region overlap but are weak in shape awareness, and penalise small and thin structures, and as a result should be reported alongside a boundary metric. Sahukara et al. [26] arrived at an associated conclusion from a different angle, observing that region metrics summarized at the level of a bounding box are insensitive to the thin borders that delineate table cells. They continue that slightly shifts in the borders can leave the region metric nearly unchanged while noticeably increasing visual error.

3. METHODOLOGY

3.1 ResForTable architecture

ResForTable is overall made of four components: the ResNet-50 backbone, the fully convolutional module, the ViT encoder, and the decoder. Figure 1 visually presents the components and data flow amongst them. A document image () is the input into the ResNet-50 encoder. We choose a batch size of two to conserve computational resources. The ResNet-50 consists of a stem and four stages. The stem of the model comprises of a (7x7) convolution layer followed by a GroupNorm (GN), a ReLu, and a stride-2 (3x3) max pooling layer. We use a GN instead of a BatchNorm because a batch size of 2 would result in noisy batch statistics. The GN is independent of batch since it normalizes across channels [10]. Stages 1, 2, 3, and 4 consist of 3, 4, 6, and 3 residual blocks respectively. The first residual block of each stage doubles the number of channels

except for the one in stage 1. Each residual block consists of three convolution layers: a (1x1) convolution, a (3x3) convolution, and a (1x1) convolution.

Each convolution is followed by group normalization and ReLu activation.

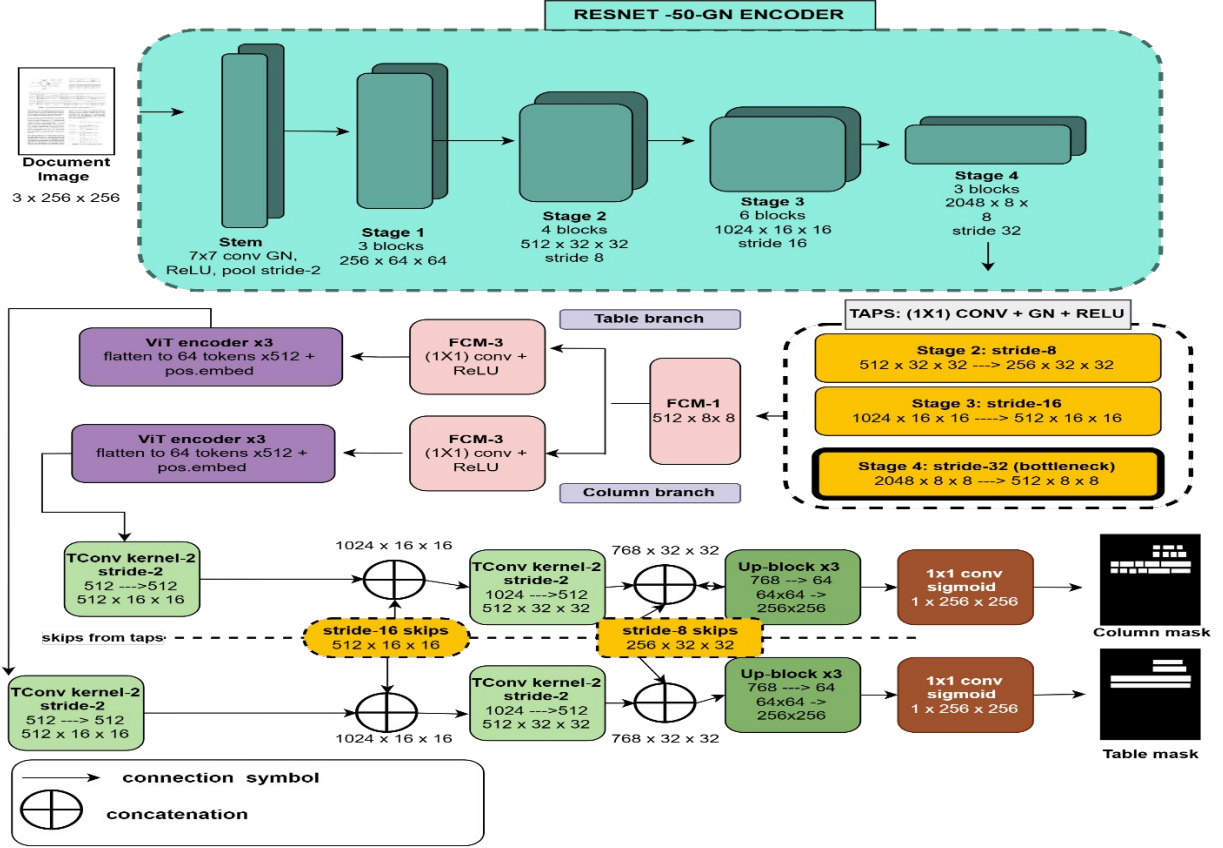


Figure 1: ResForTable components and data flow

The data flow in a residual block goes as follows: the input splits into two identical streams before it enters a convolution block. The first stream goes through a (1x1) convolution to shrink its channel width, then it goes through a (3x3) convolution that extracts features, after which it is restored to the original width by a (1x1) convolution. This limits the spatially expensive (3x3) convolution to a reduced width, keeping the depth cost manageable. The output of the convolution block is a residual. The second stream uses a skip connection to go straight to the point of addition with the residual. If there is need the second stream input to match dimensions of the residual, a (1x1) convolution is applied to it beforehand. After addition, the feature map is

passed through a ReLu activation. The backbone features are tapped at stages 2, 3, and 4. Stage 2 produces a feature map at stride 8 with spatial size of 32x32, and 512 channels. Stage 3 produces a feature map at stride 16 with spatial resolution of 16x16, and 1024 channels. Stage 4 constructs a feature map at stride 32 at a spatial size of 8x8, with 2048 channels. The tap at each stage passes through a (1x1) convolution followed by group normalization and a ReLu. This operation maps the channels of the tapped feature maps from (512, 1024, 2048) to (256, 512, 512) respectively. The stride-32 map is the bottleneck and feeds the rest of the network. The stride-16 and stride-8 maps are kept as skip connections that restore spatial detail casted off during downsampling. This

is important especially for the column branch since column boundaries are thin vertical structures whose spatial positioning cannot be inferred accurately from an 8x8 grid.

3.1.2. Fully convolutional module

The bottleneck first pass through a shared fully convolutional module, which we denote as FCM-1. FCM-1 consists of two (1x1) convolutions, each followed by a ReLu activation, with a two-dimensional dropout probability of 0.2 after each. The fully convolutional modules in this architecture preserve the 512-channel count and 8x8 spatial resolution of the bottleneck. The convolution operations mix information across channels at each spatial location without changing the feature map shape. The output of FCM-1 is transferred to both decoder branches meaning FCM-1 is trained by gradients from both tasks.

Each branch begins with its own fully convolutional module. In TableNet [24] and TransTab [25] the two individual modules are not the same. The column branch consists of two convolutions and a dropout, while the table consists of a single convolution and no dropout. Neither paper provide a reason for this asymmetry. Our approach is different in that we give both branches the single-convolution module design which we denote as FCM-3. Section 5.4 reports the ablation that motivates this choice.

3.1.3. ViT encoder

Each 512x8x8 output map from the branches is flattened to a sequence of 64 tokens, 512-dimensional vectors. Flattening discards spatial positing so a learned positional embedding is added to the sequence. The embedding is initialized from a truncated normal distribution with a standard deviation of 0.02. The sequence passes through a stack of three transformer encoder layers [11]. Each layer has eight attention heads, a feed-forward dimension of 2048, and a dropout of 0.1. In addition, we apply layer normalization

before each sublayer. We follow Dosovitskiy et al. [12] whereby the transformer operates on a convolutional feature grid as opposed to raw image patches. Self-attention used in each layer to model the global relationships between page elements. The output sequence of each ViT encoder is transposed and reshaped back into a 512x8x8 feature map.

3.1.4. Decoder

Each branch restores the 8x8 feature map to the input resolution of 256x256 through five upsampling stages. The first stage is a transposed convolution with kernel size 2 and stride 2 that takes 512 channels to 512 at 16x16. The output of the first stage is concatenated to the stride-16 skip connection resulting in a 1024 channel count. The second stage is a transposed convolution that takes the 1024 channel count to 512 at 32x32. The output is concatenated to the stride-8 skip connection producing a channel count of 768. The remaining three stages are upsampling blocks. Each stage consists of a transposed convolution that doubles the resolution, a ReLu, a (3x3) convolution, a second group normalization, and a second ReLu. The (3x3) convolution resolves checkerboard artifacts from the transposed convolutions by blending the boundaries between adjacent pixel patches. The three upsampling blocks map 768 channels to 256 at 64x64, 256 to 128 at 128x128, and 128 to 64 at 256x256. A final (1x1) convolution shrinks the 64 channels to one. A sigmoid function maps the resulting logits to per-pixel probabilities.

3.2 Loss function

For a branch with predicted probabilities M and ground truth G , the branch loss is a weighted sum of a dice term and a binary cross-entropy (BCE) term, where N is the number of pixels, ϵ is a smoothing constant set to 1, and α is set to 0.5. The total is the sum over both branches:

$$(1)$$

(2)

(3)

(4)

The BCE component of the loss remedies the Dice term’s inherent blind spots. The Dice term is computed over the entire mask, and since tables usually take up a small portion of pixels on a page, it is less sensitive to the imbalance between the foreground and background. BCE resolves this blind spot by providing a per-pixel gradient that stays informative in regions in which the Dice term goes flat. The Dice term in Eq. (2) does not operate on a thresholded mask. It operates on raw sigmoid probabilities so as to remain differentiable.

3.3 Implementation

We implemented ResForTable in PyTorch. We fine-tune an ImageNet-1k pre-trained group-normalized ResNet-50 on 454 training images. The fine tuning is end-to-end with no frozen stages. Input images are resized to 256x256 and the batch size is 2. We train for 120 epochs with AdamW at a peak learning rate of 0.0005 and a weight decay of 0.0001. We do this under a one-cycle learning rate schedule stepped once per batch with a warmup fraction of 0.1. We clip gradient norms at 0.5. The weights of the two output heads are initialized from a normal distribution with 0.001 standard deviation and a zero bias. We observed empirically that a larger initialisation saturates the sigmoid early in training and the branch never recovers. Checkpoint selection does not track the validation loss. Instead, it tracks the mean hard DICE of the two branches at a threshold of 0.5. The BCE continues to reward increased confidence after the decision at 0.5 has stopped changing so a loss-selected checkpoint would be worse on the metric we report.

We augment the training set with a rotation of up to degrees, scaling of up to , translation of up to of the image size, brightness and contrast perturbations of up to , and a horizontal flip with 0.5 probability. One set of geometric parameters is applied uniformly across the document image and masks in each example. We use bilinear resampling for the document images. We apply nearest neighbour resampling to the mask so that they stay binary. The photometric perturbations are applied to the document images only since it would be inconsequential to the masks. Finally, we apply ImageNet normalization so the pre-trained weights are exposed to a familiar input distribution. We implemented all the training runs on a single NVIDIA T4 GPU with two dataloader workers.

4. EXPERIMENTAL RESULTS AND DISCUSSION

We assess ResForTable’s performance against TableNet [26] and TransTab*. We do not use TableNet from the original authors [24] because they did not publish their Marmot dataset performance. TransTab* is our own implementation of the original TransTab [25] architecture. The authors of the original TransTab paper compare their Marmot dataset performance against CascadeTabNet [21], DeepDeSRT [19], Table Transformer [23], and TableNet [24], all of whom did not publish Marmot dataset performance results. Additionally, they report IoU as 0.9332 and F1 Score as 0.9277. Since IoU cannot exceed F1-Score for the same pair of masks, these two figures cannot both be correct. Considering these concerns, we implemented the architecture for ourselves and report it as TransTab* throughout.

4.1 Model performance

Figure 2 shows a column and table segmentation result by ResForTable. Each

prediction is side-by-side with its corresponding ground truth. Table 1 displays performance across models based on table DICE, column DICE, table XOR, column XOR, table ASSD, and column ASSD. The TableNet rows report the best Marmot benchmark results at each input resolution in [26], which is 9.18% at 256x256 after 500 epochs and 13.83% at 1024x1024 after 150 epochs. There are clear differences in performance, with ResForTable being top in all metrics. In Table DICE, ResForTable (0.9352) marginally exceeds TransTab* which recorded 0.9290. TransTab* (0.8915) also trails ResForTable (0.9045) in the column DICE by 0.0130. In terms of Table

XOR, ResForTable’s error rate (1.86%) is roughly five times lower than that of TableNet (256x256) and roughly seven times lower than TableNet (1024x1024) who scored 9.18% and 13.8% respectively. TransTab* (2.08%) has a lower error rate than the TableNet variants but has a slightly higher error rate than ResForTable. TransTab* (2.61%) has a higher column XOR than ResForTable (2.31%). ResForTable scores table and column ASSDs of 2.654 and 1.818. It performed better than TransTab*, which scored 2.795 and 1.958 on the table ASSD and column ASSD respectively.

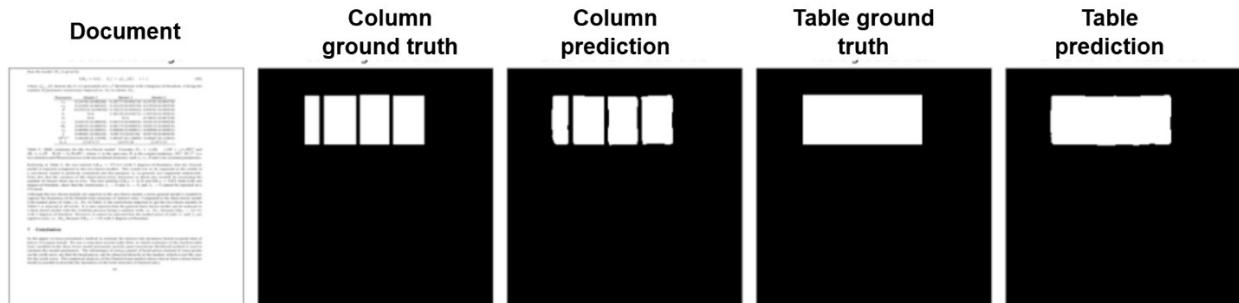


Figure 2: ResForTable predictions vs ground truth masks

Table 1: Performance comparison

Model	Table DICE	Column DICE	Table XOR	Column XOR	Table ASSD	Column ASSD
TransTab*	0.9290	0.8915	2.08%	2.61%	2.795	1.958
TableNet (256x256) [26]	-	-	9.18%	-	-	-
TableNet (1024x1024) [26]	-	-	13.83%	-	-	-
ResForTable	0.9352	0.9045	1.86%	2.31%	2.654	1.818

4.2 Backbone comparison Fully convolutional module ablation

Table 2 shows the results of when we switched out our ResNet-50-GN backbone for a VGG-19. ResForTable outperforms VGG19-backbone architecture across all the metrics. ResForTable has approximately 4.6M more parameters than VGG19-backbone; however, it shows a 42.7% reduction in FLOPs.

The canonical set up of a column branch’s fully convolutional module contains two convolutional layers and one dropout layer [24], [25], [26]; on the other hand, the table branch’s fully convolutional module contains one convolutional layer and no dropout layer. We added an extra convolution and a dropout layer to the column branch of ResForTable, restoring the canonical asymmetry, and called the result ResForTable-extra. Table 3 shows the results. ResForTable slightly

outdid ResForTable-extra across all the table and column overlap metrics. ResForTable-

extra marginally outperformed ResForTable on ASSD.

Table 2: Comparative analysis of ResForTable and VGG19-backbone

Model	Table DICE	Column DICE	Table XOR	Column XOR	Table ASSD	Column ASSD	Parameters (M)	Complexity (GFLOPs)
VGG-19 backbone	0.9266	0.8939	2.12%	2.55%	2.769	1.883	50.40	46.53
ResForTable	0.9352	0.9045	1.86%	2.31%	2.654	1.818	54.99	26.64

Table 3: Comparative analysis of ResForTable-extra and ResForTable

Model	Table DICE	Column DICE	Table XOR	Column XOR	Table ASSD	Column ASSD
ResForTable-extra	0.9320	0.9017	1.99%	2.43%	2.628	1.651
ResForTable	0.9352	0.9045	1.86%	2.31%	2.654	1.818

4.3 DICE and geometric bias

Across all the model configurations we trained, the table DICE is consistently higher than the column DICE as shown in Figure 3. In contrast, all the same model configurations have column ASSD lower than table ASSD.

The metrics consistently ranked the two branches in opposite orders. We conducted correlation tests across all model configurations. In each configuration, the collinearity between ASSD and P/A was less than 0.35 in magnitude.

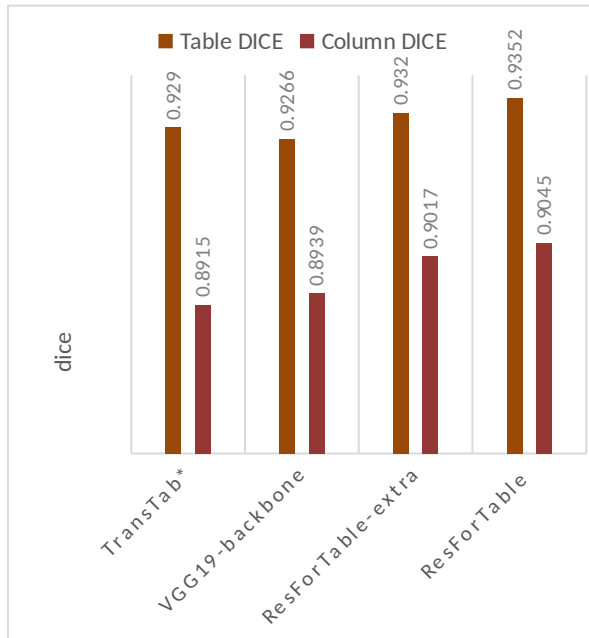
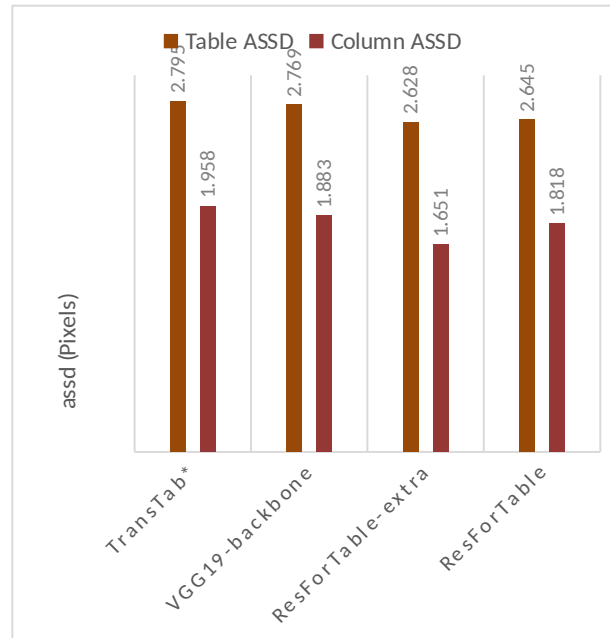


Figure 3: DICE and ASSD performance across model configurations

4.4 Discussions

ResForTable exceeds TransTab* on every metric we report. For the XOR error rate, ResForTable beats both TableNet variants from [26] by considerable margins. This positions ResForTable as the current state of the art in pixel-level table and column

segmentation tasks. The ResNet-50-GN encoder results in 42.7% less arithmetic cost compared to a VGG-19 encoder. On top of that, the ResNet-50-GN architecture beats the VGG-19 architecture on all reported metrics. This reveals a correctness and computational efficiency gain. Document processing

usually happens with huge loads and modest hardware, which makes the abovementioned an important discovery.

TableNet [24] introduced an asymmetric fully convolutional module in which the column branch receives more capacity than the table branch. TransTab [25] sustained it, and the re-evaluation in [26] retains it, yet none of the papers justify the design choice. Our symmetric design beats the asymmetric one on all four-overlap metrics while using fewer parameters. However, the asymmetric design beat ours on the ASSD metrics. Which suggests that added column capacity improves boundary placement but has little to no effect on region overlap.

The central observation of this paper is that the overlap metric and boundary metric rank the two branches in opposite orders, and they do this in every configuration we trained. Eq. (13) is used to try and explain this divergence. ASSD reports the average boundary displacement and nothing else. The dice deficit reports that same displacement multiplied by the perimeter-to-area ratio of the target. So, we see that the two measures look at a similar underlying error, only that one is scaled multiplicatively by the perimeter-to-area ratio of the target. Our validation masks have a mean P/A of 0.1455 for columns against 0.0755 for tables. A column target transforms a given boundary error to twice the overlap penalty that the same error would receive on a table target.

The correlation analysis was used to empirically observe the relationship between DICE deficit and P/A. The statistically significant partial correlation shows that shape is partially responsible for the variation in the dice deficit among images that share a boundary error. The practical consequence of this in segmentation tasks is that DICE must not be used as the sole evaluation metric between structures with different geometries. As recommended in [28], it must be reported alongside a boundary metric. In the context

of column and table segmentation, practitioners must consider that a DICE performance gap between tables and columns could be an artefact of the target’s perimeter-to-area ratio.

The study relies on a single validation split and single dataset. Because of the single validation split, ground truth P/A values are identical across all model configurations, and only the predicted masks vary. Future studies should vary the validation split in different runs so that the P/A distribution can vary. This would help strengthen the study since it mirrors independent replication. Extending the evaluation to a second dataset would test whether the architecture generalises beyond domains from which the Marmot documents were sourced.

5. CONCLUSION AND FUTURE WORKS

This study introduced ResForTable, an end-to-end dual-branch model for joint table and column segmentation using a pretrained group-normalized ResNet-50 encoder and transformer-enhanced decoder branches. Evaluation on the Marmot dataset achieved DICE scores of 0.9352 for table segmentation and 0.9045 for column segmentation, with XOR error rates of 1.86% and 2.31%, respectively. The model also obtained ASSD values of 2.654 for tables and 1.818 for columns, while outperforming the reimplemented TransTab across the evaluated metrics. Furthermore, replacing the VGG-19 backbone with group-normalized ResNet-50 reduced computational requirements by 42.7% without compromising segmentation performance. The analysis also showed that the asymmetric fully convolutional branch design introduces additional parameters without providing an overlap-based performance advantage. Finally, the geometric analysis demonstrated that DICE is inherently more sensitive to thinner structures with higher perimeter-to-

area ratios, supporting the use of overlap and boundary metrics together when evaluating table and column segmentation.

Future research will investigate more efficient architectural configurations and alternative segmentation metrics that can better account for the geometric characteristics of tables and columns. Further studies will also evaluate ResForTable on larger and more diverse document datasets to assess its generalization across different document layouts, table structures, and imaging conditions.

REFERENCES

- [1] W.-L. Huang, “A case study of lean digital transformation through robotic process automation in healthcare,” *Sci. Rep.*, vol. 14, no. 1, p. 14626, 2024, doi: 10.1038/s41598-024-65715-9.
- [2] J. Krzywy, K. Dorofiejczuk, F. Nowak, and M. Jasiulewicz-Kaczmarek, “Study of the use of robotic process automation in supporting customer order process,” *IFAC-PapersOnLine*, vol. 58, no. 19, pp. 1018–1023, 2024, doi: 10.1016/j.ifacol.2024.09.142.
- [3] N. M. Soto Leiva, Y. R. Cubas Pecho, J. A. Taquíá Gutiérrez, and J. C. Quiroz Flores, “Document Process Automation in an International Logistics Company Using OCR, RPA, and Text Analytics,” in *Actas del Congreso Internacional de Ingeniería Industrial (CIII)*, Lima, Peru: Universidad de Lima, 2026, pp. 161–168. doi: 10.26439/ciii2025.8661.
- [4] R. Zanibbi, D. Blostein, and J. R. Cordy, “A survey of table recognition: Models, observations, transformations, and inferences,” *International Journal on Document Analysis and Recognition*, vol. 7, no. 1, pp. 1–16, 2004, doi: 10.1007/s10032-004-0120-9.
- [5] L. R. Dice, “Measures of the Amount of Ecologic Association Between Species,” *Ecology*, vol. 26, no. 3, pp. 297–302, 1945, doi: 10.2307/1932409.
- [6] V. Yeghiazaryan and I. D. Voiculescu, “Family of boundary overlap metrics for the evaluation of medical image segmentation,” *Journal of Medical Imaging*, vol. 5, no. 1, p. 015006, 2018, doi: 10.1117/1.JMI.5.1.015006.
- [7] A. Reinke *et al.*, “Understanding metric-related pitfalls in image analysis validation,” *Nat. Methods*, vol. 21, no. 2, pp. 182–194, 2024, doi: 10.1038/s41592-023-02150-0.
- [8] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
- [10] Y. Wu and K. He, “Group Normalization,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.
- [11] A. Vaswani *et al.*, “Attention Is All You Need,” in *Advances in Neural Information Processing Systems*, 2017.
- [12] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.
- [13] A. Gray, *Tubes*, 2nd ed., vol. 221. in Progress in Mathematics, vol. 221. Basel: Birkhäuser, 2004. doi: 10.1007/978-3-0348-7966-8.
- [14] P. Pyreddy and W. B. Croft, “TINTIN: A System for Retrieval in Text Tables,” in *Proceedings of the Second ACM International Conference on Digital Libraries (DL '97)*, Philadelphia, PA, USA: Association for Computing Machinery, 1997, pp. 193–200. doi: 10.1145/263690.263816.
- [15] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, “Trainable table location in document images,” in *Proceedings of the 16th International Conference on Pattern Recognition (ICPR 2002)*, Quebec City, QC, Canada: IEEE, 2002, pp. 236–240. doi: 10.1109/ICPR.2002.1047838.
- [16] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet, “Learning to Detect Tables in Scanned Document Images Using Line Information,” in *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, Washington, DC, USA: IEEE, 2013, pp. 1185–1189. doi: 10.1109/ICDAR.2013.240.

- [17] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT '92)*, Pittsburgh, PA, USA: Association for Computing Machinery, 1992, pp. 144–152. doi: 10.1145/130385.130401.
- [18] A. Costa e Silva, "Learning Rich Hidden Markov Models in Document Analysis: Table Location," in *2009 10th International Conference on Document Analysis and Recognition (ICDAR)*, Barcelona, Spain: IEEE, 2009, pp. 843–847. doi: 10.1109/ICDAR.2009.185.
- [19] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan: IEEE, 2017, pp. 1162–1167. doi: 10.1109/ICDAR.2017.192.
- [20] S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, "DeCNT: Deep Deformable CNN for Table Detection," *IEEE Access*, vol. 6, pp. 74151–74161, 2018, doi: 10.1109/ACCESS.2018.2880211.
- [21] D. Prasad, A. Gadpal, K. Kapadni, M. Visave, and K. Sultanpure, "CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, 2020, pp. 2439–2447. doi: 10.1109/CVPRW50498.2020.00294.
- [22] M. Agarwal, A. Mondal, and C. V Jawahar, "CDeC-Net: Composite Deformable Cascade Network for Table Detection in Document Images," in *2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy: IEEE, 2021, pp. 9491–9498. doi: 10.1109/ICPR48806.2021.9411922.
- [23] B. Smock, R. Pesala, and R. Abraham, "PubTables-1M: Towards Comprehensive Table Extraction From Unstructured Documents," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, 2022, pp. 4624–4632. doi: 10.1109/CVPR52688.2022.00459.
- [24] S. S. Paliwal, D. Vishwanath, R. Rahul, M. Sharma, and L. Vig, "TableNet: Deep Learning Model for End-to-end Table Detection and Tabular Data Extraction from Scanned Document Images," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, NSW, Australia: IEEE, 2019, pp. 128–133. doi: 10.1109/ICDAR.2019.00029.
- [25] K. Sahukara, Z. Bettouche, and A. Fischer, "Synthetic Data Augmentation for Table Detection: Re-evaluating TableNet's Performance with Automatically Generated Document Images," in *2025 15th International Conference on Advanced Computer Information Technologies (ACIT)*, Šibenik, Croatia: IEEE, 2025, pp. 841–846. doi: 10.1109/ACIT65614.2025.11185625.
- [26] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool," *BMC Med. Imaging*, vol. 15, no. 1, p. 29, 2015, doi: 10.1186/s12880-015-0068-x.
- [27] L. Maier-Hein, "Metrics reloaded: recommendations for image analysis validation," *Nat. Methods*, vol. 21, no. 2, pp. 195–212, 2024, doi: 10.1038/s41592-023-02151-z.
- [28] Y. Wang, "TransTab: A transformer-based approach for table detection and tabular data extraction from scanned document images," *Machine Learning with Applications*, vol. 20, p. 100665, 2025, doi: 10.1016/j.mlwa.2025.100665.