Deep Learning for Table Detection in Scanned Document Images

Muhammad Ahmed Mohsin, Muhammad Umer, Tariq Umar
SEECS, NUST, Islamabad, Pakistan
Email: {mmohsin.bee20seecs, mumer.bee20seecs, tumar.bee20seecs}@seecs.edu.pk

***Abstract*—Table detection and information extraction from table images has always been an arduous task because of the wide range of table images in different images and sizes. The proposed technique for table detection and information extraction used a novel approach of object detection trained on publicly available ICDAR 2013 [1] dataset. We increased the efficiency of this proposed architecture by using image segmentation instead of object detection and by applying advanced image processing algorithms on both the dataset as well as the resulting masks from our model. The above-mentioned techniques were applied on the Marmot Extended for Table Structure Recognition, which was made publicly available by S. Paliwal et al. [2].**

***Index Terms*—Deep learning, convolutional neural networks, image segmentation, tablular structure recognition, image processing.**

I. Introduction

With the increasing number of devices with cameras the sharing of pictorial information is becoming increasingly popular. The need for extraction of information from those images is becoming pressing every day. In the present day, the information is being extracted from the documents manually and this takes up much time and is cost inefficient. The wide diversity on the nature of tables and the graphical information present along with them makes the task of information extraction from tables quite arduous.

Most of the techniques that are used now a days used tabular information extraction. The TabNet proposed in the approach utilized a base network of VGG-19 [3] and further used Tesseract [4] to highlight text for signifying semantic features.

The approach proposed by this paper was that the model predicts masks of the detected tables and then passed onto a pipeline of image processing. This pipeline spreads onto the nearest contours of the detected tables and outputs the coordinates of detected bounding boxes. Finally, the test image is cropped onto the dimensions of these bounding box coordinates and passed onto Tesseract for extracting text from the images using OCR.

We evaluated our model on ICDAR 2013 as well on the Extended Marmot dataset as made available by S. Paliwal et al. our model outperforms the efficiency of the base TabNet as well as models proposed by Tran et al. [5], S Schreiber et al. [6] and S. A. Siddiqui et al. [7]. Moreover, we further deduce that our model with minimal fine tuning can be used to generalize on other data sets and thus thereby enabling transfer learning.

In summary, the major contributions of this paper are as follows:

1) Increased the efficiency of already proposed TabNet by enhancing the architecture of the encoder and adapting a suitable series of dataset processing to the Extended Marmot dataset.
2) Deduced that an image processing pipeline, while reducing generalization, can transform unseen images into a form factor similar to the original dataset, thereby increasing the evaluative metrics.
3) Proposed advanced image filtering pipelines to increase OCR capabilities of Tesseract.
4) Implemented a third encoder specific to Rows, as proposed in the concluding remarks of the original paper.

II. Literature Review

Before the use of deep learning in the field of image recognition and data extraction most of the work on table detection was based on heuristics and metadata. TINTIN (Text Information-based Text Inquiry) [8] used structural information to identify tables and their respective fields.

Silva et al. [9] presented an approach based on Hidden Markov Models (HMMs); Probabilistic graphical models were used to detect tables which were modelled through the joint distribution over sequential observations of visual page elements and the hidden state of various lines to merge the potential table lines into a table.

Cesarini et al. [10] initialized the table detections task through the use of learning techniques. The proposition put forth by them, Tabfinder, predicts by turning the input documents into a Tree MXY representation and then looks for vertical and horizontal linings, deducing if a table is present or not.

U. Khan et al. [11] proposed TabAug [11], an approach in which the efficiency for the detecting tables was improved by increasing the data set through tabular augmentation. The rows and columns were detected and augmented which proved to be more efficient than the classical augmentation technique.

HybridTabNet [12] was also used for table detection and data extraction. It used two models ResNet-101 and Hybrid Task Cascade (HTC) to localize the tables in scanned document images. Moreover, the conventional neural networks were replaced by deformable neural networks which could detect tables of arbitrary layouts precisely.

III. Dataset Preparation

Lack of quality and clarity in dataset images for approaches centered around deep learning make or break the final results. Datasets for training models extensively for table detection are very scarce in their contents or are not publicly available and among them, even fewer fill the criterion for tabular structure recognition. Fortunately, S. Paliwal et al. [2] manually annotated the original Marmot [13] and open-sourced the dataset titled Marmot Extended for Table Structure Recognition. Although annotations specific to detecting rows as a separate entity is a shortcoming of this dataset, it serves the purpose for structural recognition of tables in images.

A. Upscaling and Sharpening

Each image in this extended dataset was upscaled to a resolution of 1024 * 1024 pixels in order to facilitate inferences for documents that were not of the standard size, Letter, 8.5 by 11 inches in dimensions. A sharpening filter also accommodated the upscaling process which makes the boundaries of tables and columns more prominent, essentially acting as an individual feature of the Marmot Extended Dataset.

B. Denoising

Although only a fraction of the total images had noise in them, denoising filters were manually implemented as the core of the architecture we proposed, instead of generalizing towards varying images, expects images processed through the same pipeline as the images in the dataset it was trained on.

C. Split

Since there was no default split for the dataset available, a standard 80/20 split was utilized, yielding 794 train and 199 test images. Among these images, those that did not contain any tabular data were later removed from the dataset.

D. Limitations

Upon analysis of the dataset, we find that there are close to no images having tables that are without a boundary. An exemplar figure of such a table is given in Fig. 1.

Tax Period	Check Date	Amount
2007 payable 2008 Fall Provisional	01-22-10	$ 45,439.03
Penalty 2007 payable 2008 Fall Provisional	01-22-10	9,087.80
2008 payable 2009 Spring Tax	01-22-10	48,989.80
2008 payable 2009 Fall Tax	01-22-10	48,989.80
Penalty 2008 payable 2009 Spring Tax	01-22-10	4,898.98
2009 payable 2010 Spring Tax	04-14-10	45,328.50
2009 payable 2010 Fall Tax	10-06-10	45,328.50
2010 payable 2011 Spring Tax	04-29-11	45,436.50
2010 payable 2011 Fall Tax	10-14-11	45,436.50
Total Property Taxes paid		$ 338,935.41

Fig. 1: Boundary-less Table

IV. TableNet: Proposed Architecture

A. Approach

Table detection and column detection are treated as two different problems in the previous proposed deep learning models as they can be solved independently. Columns are by definition vertically aligned characters, thus if all the columns in the document can be detected then we can construct the corresponding tables. But this method can produce many false results as there can be vertically aligned character without being in a table.

To counter this, we can use tabular detection along with the column detection, as both will appear at a common place in the document. By using these detection filters along we will see that the results of the model will increase significantly. The proposed model in this paper is based on this concept and the Long et al. [14], encoder-decoder model for semantic segmentation. The encoder for the column and table detection is common that enforces the encoding layers to use the ground truth while the decoders are separate for each. Thus, two computational graphs are to be trained.

B. Architecture

Residual Blocks [15] are a stack of layers that are set to take the output of one layer and add it to another layer deeper in the block. Non-linearity is applied after being added to the output of the appropriate layer in the main path. This bypass connection is called a shortcut connection or a skip connection. The traditional residual block has a structure with several channels (wide¿ narrow¿ wide). The input has many channels compressed with a 1x1 convolution. Then the number of channels increases again with a 1x1 convolution, allowing you to add inputs and outputs.

An Inverted Residual Block [16], sometimes called an MB-Conv Block, is a type of residual block used for image models that uses an inverted structure for efficiency reasons. It was originally proposed for the MobileNetV2 CNN architecture. It has since been reused for several mobile-optimized CNNs. An Inverted Residual Block follows a narrow $\Rightarrow$ wide $\Rightarrow$ narrow approach, hence the inversion. We first widen with a 1x1 convolution, then use a 3x3 depth-wise convolution (which greatly reduces the number of parameters), then we use a 1x1 convolution to reduce the number of channels so input and output can be added.

Inverted Residual Blocks [16] are a type of residual block used in image models that use inverse structures for efficiency reasons. Originally proposed for the MobileNet V2-CNN architecture. It has since been reused on several mobile-optimized CNNs. The inverted residual block follows the narrow $\Rightarrow$ wide $\Rightarrow$ narrow approach and is therefore inverted. First expand with a 1x1 convolution, then use a 3x3 depth (significantly reduce the number of parameters), then use a 1x1 convolution to reduce the number of channels and add inputs and outputs.

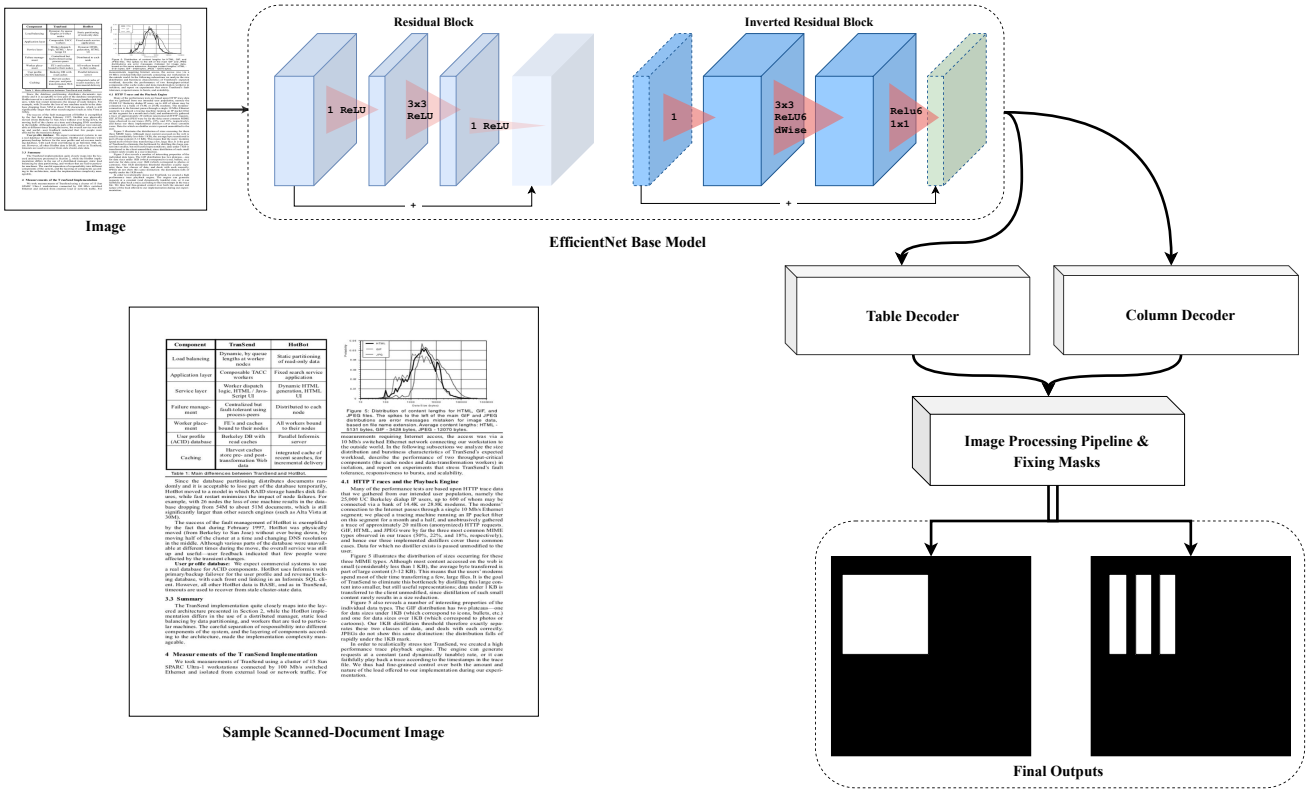


Fig. 2: TableNet - Proposed Model Pipeline

C. Methodology

As discussed in the previous sections, the input image is first converted into 1024 * 1024 resolution, then it is processed using Tesseract OCR [4]. Mask for the table and column will be produced by a single model. These masks will have the binary target pixel values, either for the table/column or the background. The table/column will be detected based on its visual features, which will have very small tolerance for noise. For this reason, instead of regressing the table or column boundaries, we used the method of predicting it pixel-wise. This method has been proved very effective in the recent semantic segmentation. FCN architecture, proposed by Long et al. [14], demonstrates the accuracy of encoder-decoder network architecture for semantic segmentation. This architecture has used the skip-pooling technique for combining low-resolution feature maps of decoder network and high-resolution features of encoder networks. The base layer for their model was VGG-16 and fractionally-stride convolution layers for upscaling the low-resolution semantic maps, which are then combined with the high-resolution encoding layers.

D. Pipeline

The model proposed in this paper uses the same logic for encoder/decoder as in FCN architecture. As shown in figure 2, we used EfficientNet [17] as the base model for the network. The encoder part has two blocks: Residual block and Inverted Residual Block. In both of the blocks ReLU functions were used to find the efficiency of the models. Following the encoder, the architect is divided into two segments i.e., Table Decoder and Column Decoder. The output from the decoder is then passed through image processing which gives us the masks for the column and tables.

V. Table Detection and Extraction

A. Detection

Detection of the tables and columns producing binary masks as outputs is achieved by passing an input image into the end-to-end proposed model. The encoder, consisting of the EfficientNet base detects the Region of Interest in the input image and the decoders, one for columns and the other for tables, produce masks respective to their purpose. Fig. 3 displays an exemplar prediction of output. From these predicted outputs, we can move onto the extraction of the tabular data, either in the form of text by utilizing Tesseract OCR or in the form of an image using common libraries.

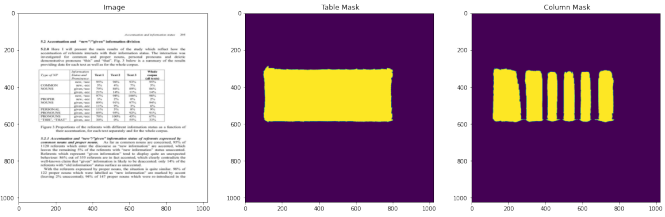
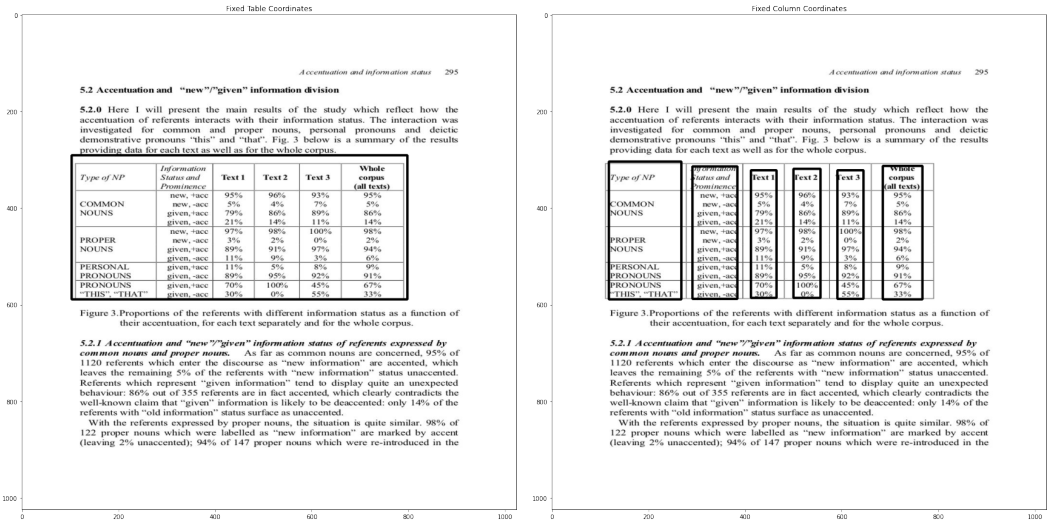


Fig. 3: Predicted Masks by the Proposed Model



(a) Table Bounding Box Coordinates (b) Column Bounding Box Coordinates

Fig. 4: Conversion of Fixed Masks to Bounding Boxes

B. Extraction

After the generation of masks by the model for each table and for columns within that table, these masks are then fed into the proposed image processing pipeline. Efficient erosion-dilation proposed by J. Y. Gil et al. [18], algorithms renders any column mask overlapping with adjacent column masks out of the output. The aforementioned process also helps with filtering out masks smaller than the preset area, which cannot possibly be either a table or a column.

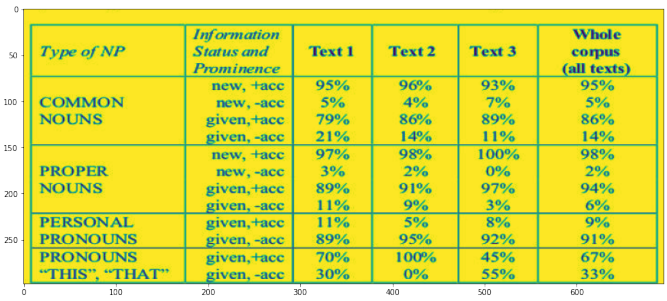
Although the prediction of row through indirect approaches was possible, it did not improve the performance of the model in any way, and this sole fact served as a motif for the manual annotation of the Marmot dataset; that without a decoder specifically applicable to rows, row mask generation was not feasible.

C. Fixing Masks

After the erosion-dilation process, the masks were spread onto the page until each individual prediction lined up with a horizontal or vertical boundary of a table. This was achieved by implementing an edge and line oriented contour detection algorithm [19]. We coined the term for this two-part process in our proposed architecture as fixing masks.

D. Saving

To then extract an image from the predicted and processed masks, a simple function was used to extract the bounding box coordinates from the respective masks, as no image processing library currently supports cropping an image to a limiting mask directly. Lastly, the cropped image is directly fed onto a sharpening filter and the same processes that each image of the training data underwent are applied onto it. Continuing the demonstration of inference from the proposed model, Fig. 5 displays the final cropped image output.



Type of NP	*Information Status and Prominence*	Text 1	Text 2	Text 3	Whole corpus (all texts)
COMMON NOUNS	new, +acc	95%	96%	93%	95%
	new, -acc	5%	4%	7%	5%
	given,+acc	79%	86%	89%	86%
	given, -acc	21%	14%	11%	14%
PROPER NOUNS	new, +acc	97%	98%	100%	98%
	new, -acc	3%	2%	0%	2%
	given,+acc	89%	91%	97%	94%
	given, -acc	11%	9%	3%	6%
PERSONAL PRONOUNS	given,+acc	11%	5%	8%	9%
	given, -acc	89%	95%	92%	91%
PRONOUNS "THIS", "THAT"	given,+acc	70%	100%	45%	67%
	given, -acc	30%	0%	55%	33%

Fig. 5: Cropped Image Output

VI. Experiments

Our proposed model, like the base TableNet, requires both table and structure annotated data for training. Training was achieved through the same Extended Marmot data, however, manual annotations for rows was achieved to enable adding the third branch of the decoders, the row decoder. The architecture of our model has been implemented in PyTorch and trained on a cloud system with 16GB RAM and Tesla P100, provided by Google Colab. Multiple experiments were conducted with the proposed architecture, specifically oriented towards changing the hyperparameters.

Proposed model is trained over a range of epochs, keeping in mind the time constraints, and with a batch size of 2. Adam proved to be the most suitable optimizer for our case, as it converges relatively quicker, with characteristic parameters of beta β_1 = 0.9, beta β_2 = 0.999 and epsilon ϵ = 1e-08. A small set of the Extended Marmot dataset was used to monitor the overfitting and convergence behavior through dynamic graphs. Severe limitations were encountered on both the number of experiments that were feasible to conduct and how long each experiment took to complete, as the authors of this paper are undergraduate students and short on computational resources.

VII. Evaluation and Results

Generalization capabilities of deep learning models improve as the amount of data increases [20], and as proposed by S. A. Siddiqui et al. [7] combining datasets help improve model performance. We adopted a leave-one-out approach to evaluate our model's performance in order to remove any unnecessary deviations of our scores, and also to have a fair comparison to other counterpart propositions for tabular structure recognition. The two of some of the popular datasets for testing tabular recognition models are the ICDAR 2013 [1] and the Marmot dataset [13].

Common performance metrics such as the F-Score, Precision and Recall were compared across DeepDeSRT [6], DeCNT [7], and the original TableNet [2]. Table 1 shows these previously mentioned metrics on the ICDAR 2013 competition dataset and similarly, Table 2 does so on the Marmot dataset.

Model	Recall	Precision	F1 Score
TableNet + Advanced Filters	**0.938**	**0.959**	**0.948**
TableNet	0.910	0.898	0.884
DeepDeSRT	-	-	-
Tran et al. [5]	-	-	-
DeCNT	0.946	0.849	0.895

TABLE I: Results on Marmot Dataset

Model	Recall	Precision	F1 Score
TableNet + Advanced Filters	**0.969**	**0.974**	**0.971**
TableNet	0.950	0.954	0.943
DeepDeSRT	0.961	0.974	0.967
Tran et al. [5]	0.963	0.952	0.957
DeCNT	0.996	0.945	0.972

TABLE II: Results on ICDAR 2013

VIII. Conclusions

This prepare takes an already well-established architecture, TableNet, a novel deep learning model, and enhances its prediction capabilities by heavily incorporating image processing techniques to assist in text recognition at the final phase of the model. Table detection and tabular structure recognition are treated as two distinct problems in other approaches and are solved independently. TableNet bridges this gap and deals with both tasks at the same time through exploitation of inherent interdependence between table detection and tabular structure recognition. Transfer learning techniques were also utilized, and the original model acted as the basis of the proposed architecture in this paper, enabling enhanced performance capabilities even when the training data is sparse or of subpar quality. We also show that image processing, while reducing generalization, can help transform unseen images into what the model may detect to be an image from the original dataset. Future work from the original paper was also accommodated in this paper as manual annotation of the Extended Marmot dataset was incorporated and a third branch to identify rows was implemented.

References

[1] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "Icdar 2013 robust reading competition," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1484–1493.

[2] 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)*. IEEE, 2019, pp. 128–133.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[4] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2. IEEE, 2007, pp. 629–633.

[5] D. N. Tran, T. A. Tran, A. Oh, S.-H. Kim, and I. S. Na, "Table detection from document image using vertical arrangement of text blocks," *Informs Journal on Computing*, vol. 11, pp. 77–85, 2015.

[6] 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)*, vol. 1. IEEE, 2017, pp. 1162–1167.

[7] 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. 74 151–74 161, 2018.

[8] 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*, ser. DL '97. New York, NY, USA: Association for Computing Machinery, 1997, p. 193–200. [Online]. Available: https://doi.org/10.1145/263690.263816

[9] A. C. e. Silva, "Learning rich hidden markov models in document analysis: Table location," in *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ser. ICDAR '09. USA: IEEE Computer Society, 2009, p. 843–847. [Online]. Available: https://doi.org/10.1109/ICDAR.2009.185

[10] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable table location in document images," in *Object recognition supported by user interaction for service robots*, vol. 3. IEEE, 2002, pp. 236–240.

[11] U. Khan, S. Zahid, M. A. Ali, F. Shafait *et al.*, "Tabaug: Data driven augmentation for enhanced table structure recognition," *arXiv preprint arXiv:2104.14237*, 2021.

[12] D. Nazir, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Hybridtabnet: Towards better table detection in scanned document images," *Applied Sciences*, vol. 11, no. 18, p. 8396, 2021.

[13] J. Fang, X. Tao, Z. Tang, R. Qiu, and Y. Liu, "Dataset, ground-truth and performance metrics for table detection evaluation," in *2012 10th IAPR International Workshop on Document Analysis Systems*, 2012, pp. 445–449.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019.

[17] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[18] J. Y. Gil and R. Kimmel, "Efficient dilation, erosion, opening, and closing algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1606–1617, 2002.

[19] G. Papari and N. Petkov, "Edge and line oriented contour detection: State of the art," *Image and Vision Computing*, vol. 29, 02 2011.

[20] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution," 2018.