Deep Learning for the Detection of Tabular Information from Electronic Component Datasheets

Mark Traquair[‡], Ertugrul Kara[‡], Burak Kantarci[‡] and Shahzad Khan[*]
[‡]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada
[*]Lytica Inc., 308 Legget Dr, Kanata, ON K2K 1Y6, Canada
E-mails: [‡] {mtraq059,ekara044,burak.kantarci}@uottawa.ca, [*]shahzad_khan@lytica.com

***Abstract*—The global electronic components supply chain consists of tens of thousands of e-component manufacturers who fabricate over a billion distinct components. These are described in datasheets that differ in style, layout and content, and frequently publish the salient product information in tables. Keeping up-to-date on this information consumes a great deal of human effort and corporate resources. Based on the motivation that AI-based techniques are strong candidates to minimize human intervention in many applications, in this paper, we aim at the first stage of this problem and conduct a comparison of deep learning methods in detecting tabular elements in these documents. Deep learning-based object detectors are shown to be state of the art in detection tasks in different domains therefore we chose two cutting-edge models to adapt to this field, namely Faster-RCNN and RetinaNet. We use backbone networks which are pre-trained on visually salient datasets then employ transfer learning techniques to adapt to our domain. We compare the two networks under two different datasets, namely a dataset that is widely used in academic studies and a private dataset that is used by the suppliers in real supply chains. Our numerical results show that the two networks adapt well to the domain with Faster-RCNN exhibiting marginally better precision with more than 1% difference. However, RetinaNet stands out with promising recall values indicating Feature Pyramid Network architecture can potentially detect supply chain component tables better.**

***Index Terms*—Deep Learning, Supply Chain Optimization, Document Processing, Object Detection, Page Object Detection**

I. Introduction

Modern electronic product supply chains span the globe and as reported in [1], supply chains defined as flow of services and information from a supplier to a customer with an entity in the middle. Having these entities operate together requires that they can share information. Much of this information is today shared manually using a combination of spreadsheets, PDF documents, emails, phone calls and faxes. These data representations are all unstructured, and thus human effort is required in almost every step of the supply chain to enable the firms to work together.

Automating content extraction from the documents is a well studied problem due to the increasing number of digitized documents. Most documents vary in style and shape as there are no standards governing their format. The increasing number of documents that are available introduces the complexity and diversity of table shapes. Since within tables important results, technical specifications, and prices are given, which are vital to supply chain function and optimization, accurate table detection is the first but the most important step to achieving a solution to this problem.

Object detection in images is a long standing open challenge in computing. There is no well defined rule to follow in understanding what objects are of interest within an image. Deep learning methods are researched in the state of the art applications as it was proven in [2] that deep convolutional neural networks (deep CNNs) are able to interpret and process the abstract environments in which object detection is needed with substantially greater results than all previously seen methods. As the way was paved showing the power of CNNs in this application, researchers continued to improve ([3]–[5]) and establish deep learning as the *de facto* solution to image segmentation.

Within this domain many methods exist which focus on the semantic deconstruction of documents such as those explored in [6]–[8] all focusing on identifying and labelling the unique elements of the documents. Each of these papers present desirable results within the general domain of semantic extraction of the elements, however we feel that emphasizing training on the specific task of table extraction may result in greater accuracy.

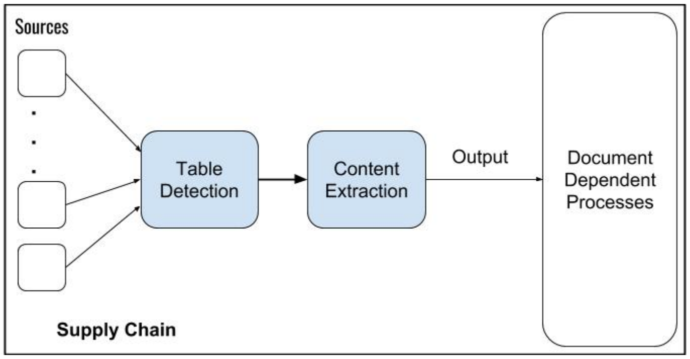

Fig. 1. Visualization of supply chain flow with table extraction

In this paper we compare two state of the art object detection techniques to table detection in documents in public datasets and a private dataset that was collected by the Lytica team from e-components manufacturers. We aim to establish the capabilities of the investigated methods within this specific domain. Determining the efficacy and accuracy of table extraction can aid us to establish a predictable standard for automated supply chain data interchange. This would make the case for supply chain and procurement automation and in turn eliminate redundancies and inefficiencies associated with human-mediated business transactions. Less human intervention in the supply chain would reduce man hours and expenses and enhance turnaround times for document delivery. Our solutions processes an image in 0.4 seconds on a GPU and gives promising results as discussed in the upcoming sections. It is clear that, deep learning models works well on this task and it is enough for humans to supervise the process instead of extracting tables themselves.

The rest of the paper is as follows: Section II provides a background on object recognition by using machine and deep learning techniques. Section III presents the two models compared on the sheets of supply chain data whereas Section IV presents and discusses the numerical results comparing the two methods. Finally, Section V concludes the paper.

II. Background and Motivation

Billions of documents flowing through supply chains contain vital information within the salient elements of these bodies of text. Documents containing vital information are largely varied in style and associated information, as such they present the challenge of accurately extracting meaningful text from these elements, and for our particular research interests we examine table extraction. Scanned documents contain no meta data or embedded text therefore methods for understanding the abstract and varied context surrounding the elements of these documents show that deep learning methods appear to be strong candidates for accurate extraction.

Object Detection: Deep learning methods in object detection continue to lead the industry. [9] In [10] a CNN is proposed which makes use of region proposal methods, and in turn improves the mean Average Precision (mAP) of competing methods by nearly two fold (from 35% to 53%) on Pascal VOC dataset [11]. With this advent the stage was set for many more advancements built on top of this, such as the fast and the faster Regions with CNN features (RCNN) [12], [13]. The advent of the region proposal network (RPN) was the key change enabling the Faster-RCNN which improved upon the accuracy of the Fast-RCNN by 8% (from 70% to 78%) on Pascal VOC dataset [11] and being able to process up to double the images when compared to its predecessor.

ResNet [4] is a very deep convolutional neural network with residual connections. ResNet [4] introduced residual connections between layers to eliminate vanishing gradient problem. Vanishing gradient problem is where we have large network and loss cannot backpropogate to the first layers due to how backpropagation works. Residual connections are shown to help solve this issue [4]. RetinaNet [14] uses ResNet as its backbone feature extractor to show that a simpler structure may be trained and improve upon mAP in object detection as compared to that of the Faster-RCNN. The focus of RetinaNet is primarily on the implementation of a new loss function which makes an RPN unnecessary, meaning simpler (no RPNs) structures may be used and introduce comparable, if not marginally better, results to those of Faster-RCNN and its other competing models.

Other Methods Considered: Statistical methods as seen in [15] process document images for margins and text blocks which then narrow down possible table locations and use them to create predictions. Though this method may work without prior training, the accuracy is considerably lower (roughly 85%) than what might be achieved through morphological or deep learning methods. Untrustworthy results may entail more work in corrections than simply manually labelling to begin with.

Applying mathematical morphology methods in [16] borrows the idea from the work in [17] which demonstrate considerably more capable results than the statistical methods. They apply morphological closing [17] hoping that the text in tables would be connected to each other since they would be close. Next, alignments of those connected components are checked to see whether they form a table. This shows that computationally expensive deep learning is not necessary to achieve desirable results. However, these methods do not attain equivalent results to which deep learning methods consistently improve.

Motivation: Due to the innate similarities between document images and natural images, deep learning methods are highly desirable as their strong results prove to perform greatly to object detection. To optimize supply chain processes, automation of tasks traditionally requiring human interaction (e.g. reading prices from component pricing sheets) are necessary. With the high volume of documents available in the supply chains, table detection with accuracy differences of even 1% mean thousands of incorrect detections, therefore sufficiently high accuracy of greater than 90% in table detection is required to lead into further research focusing on the semantic extraction and understanding of their contents.

III. Deep learning methods under study

In this section, we present the details of two state of the art object detectors and since the two architectures approach the object detection problem in different ways, we detail the superior elements of each.

TABLE I
Table of Notation

i	Index of an anchor in a mini-batch i
p_i	Objectness probability of anchor i
p_i^*	Ground-truth label for p_i
t_i	A vector representing coordinates of the predicted bounding box
t_i^*	A vector representing coordinates of the ground truth box associated with a positive anchor
L_{cls}	Classification loss
L_{reg}	Regression loss
x, y, w, h	Center coordinates (x, y) and width and height of the predicted box
x_a, y_a, w_a, h_a	Center coordinates (x, y) and width and height of the anchor box
x^*, y^*, w^*, h^*	Center coordinates (x, y) and width and height of the ground-truth box

A. Faster-RCNN

Faster-RCNN [13] became the state of the art in the object detection task when it was first released. It strengthened the defence of two-stage object detectors against single-stage detectors. Two-stage object detection methods include candidate bounding box proposals and are therefore supposed to be slower but more accurate. Single-stage detectors however, do not explicitly search for candidate boxes and this makes single-stage detectors faster, but since their prediction space is wider their accuracy is generally lower.

Faster-RCNN is an improved version of R-CNN [10] architecture. In Faster-RCNN, instead of using non-trainable search algorithms, authors switched to a faster method, to extract candidate boxes, the RPN [13]. RPNs share convolutional layers with the backbone feature extraction network and hence are end-to-end trainable. Authors of Faster-RCNN propose the following loss functions that were originally presented in [13] to end-to-end train the network:

$$\begin{aligned} L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \\ + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*). \end{aligned} \tag{1}$$

$$\begin{aligned} & t_x = (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \\ & t_w = \log(w/w_a), \quad t_h = \log(h/h_a), \\ & t_x^* = (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a, \\ & t_w^* = (\log(w^*/w_a), \quad t_h^* = \log(h^*/h_a) \end{aligned} \tag{2}$$

Classification loss is logarithmic loss over two classes: background and object. Whereas regression loss is adopted from Fast-RCNN [12] and called ’Smooth L1 Loss’. More details can be found in the paper [13].

Sharing layers in RPN makes Faster-RCNN faster than previous iteration of the architecture called Fast-RCNN [12] and the model reaches near real-time performance [13]. RPNs use features extracted from backbone feature extractor and generate class-agnostic candidate boxes with objectness score. Top performing boxes are then pooled by Region of Interest (RoI) Pooling and are sent to a classifier and a regressor. The classifier detects if the box contains an object or background, and the regressor refines the predicted bounding box for a better fit. The overall architecture can be seen in Fig. 2.

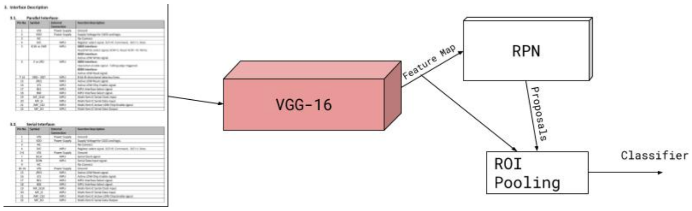

Fig. 2. General Faster-RCNN of architecture with VGG-16 backbone network

We chose Faster-RCNN because it is being widely adopted since the first publication of the architecture, and it has been used in many tasks and domains such as face detection [18], medical chart interpretation [19], [20], and many other abstract object detection challenges as shown in [21]. Thus, this architecture is shown to be suitable for transfer learning and domain adaptation. We use Faster-RCNN with the VGG-16 [3] backbone feature extractor network where we extract features from an intermediate convolutional layer (conv5). Using the VGG-16 backbone architecture provides a good trade-off between speed and detection performance compared to using ResNet backbone if enough computing power is not available.

B. RetinaNet

RetinaNet is a single-stage object detector which has state of the art performance in object detection that surpasses previous single-stage and many two-stage detectors. [14].

$$p_t = \begin{cases} p, & \text{if correct prediction} \\ 1-p, & \text{otherwise} \end{cases} \tag{3}$$

$$FL(p_t) = -(1-p_t)^{\gamma} log(p_t) \tag{4}$$

The novel contribution of RetinaNet and what makes it perform better than other single-stage object detectors such as YOLO [22] and Single Shot Detector [23] on natural object datasets such as MS COCO is the new loss function the authors introduced, namely the *Focal Loss (FL)* as formulated in eqs. 3 and 4 [14]. This loss function adds a tunable modulating factor, γ, to cross-entropy loss to give more weight to miss-classified examples.

In the single-stage object detection task, since we do not reduce the infinite number of possible box positions to a couple thousand, instead work is done on the possible hundreds of thousands of possible locations. Hence, boxes that are labeled as background class will overwhelm and dominate the gradient, thus creating class imbalance. RetinaNet solves this imbalance problem by reducing the effect of easily classified examples by it’s novel loss function [14].

Feature Pyramid Networks (FPN) [24] offer a network the capability of using different scales of an image without compromising the performance. Traditionally, in feature pyramids, this was done by scaling the image and using scaled images to extract features. FPNs, however, use a feature extractor (e.g. ResNet [4]) and extract features from different steps. Upper layers have semantically more value but spatial resolution decreases. Merging these features, by 1x1 convolutions and up-sampling, enables networks to be able to use lower and higher level features [24]. RetinaNet combines ResNet backbone feature extractor with FPN architecture and trained with their novel Focal Loss function.

We use the publicly available code base of RetinaNet [25] with backbone architecture of ResNeXt which performs slightly better than ResNet [26] in terms of precision on MS COCO [27]. ResNet [4] introduced residual connections between layers to eliminate vanishing gradient problem. Vanishing gradient problem is where we have large network and loss cannot backpropogate to the first layers due to how backpropagation works. Residual connections shown to help solve this issue [4]. In ResNext [26] authors use the same architecture but in a block consisting of different number of convolution layers, they have more than one path (e.g. 32 paths). Despite having many paths instead of one, they have the same amounts of parameters to train with. In [26] authors showed that increasing number of paths in a block from 1 to 32 decreases the error. Using ResNeXt as backbone instead of VGG-16, like we did in Faster-RCNN, requires more video ram which is only available through expensive hardware.

IV. Numerical results

Models	**ICDAR2013 Test Set**		
	Recall	**Precision**	**F1**
Faster-RCNN	0.9808	0.9738	0.9773
RetinaNet w/ ResNeXt-101	0.9865	0.9617	0.9742
RetinaNet w/ ResNet-50	0.9872	0.9440	0.9651
Kavaisidis et al. [8]	0.9810	0.9750	0.9780
DeepDeSRT [28]	0.9615	0.9740	0.9677
Tran et al. [16]	0.9636	0.9521	0.9578

TABLE II
Table detection performance comparison for models on ICDAR test set.

Models	**Lytica Test set**			**ICDAR2017 Test set**		
	Recall	**Precision**	**F1**	**Recall**	**Precision**	**F1**
Faster-RCNN	0.9528	0.9112	0.9315	0.9685	0.9385	0.9533
RetinaNet w/ ResNeXt-101	0.9099	0.7773	0.8384	0.9748	0.9241	0.9487
RetinaNet w/ ResNet-50	0.9442	0.7649	0.8452	0.9937	0.9025	0.9459

TABLE III
Table detection performance comparison for models on private Lytica test set.

A. Datasets

In this study, we use a combination of public and private datasets, namely the publicly available Marmot[1] dataset, DSSE-200[2] and ICDAR-2017 Page Object Detection (POD) dataset[3], ICDAR-2013 Table Competition Test Dataset[4] and private dataset provided to us by Lytica Inc[5], a supply chain company located in Kanata, Ontario, Canada. The private dataset includes millions of documents for electronic components. Hereafter, we will call this private dataset, the Lytica dataset. The Lytica dataset has interesting properties as it can be seen in Fig. 3. The sheets of data come from different suppliers and thus we have different table structures compared to other datasets that contains tables from scientific papers. This dataset contains documents from variety of suppliers and introduces extra layer of difficulty when compared to public datasets such as ICDAR table competition dataset. Page objects in public datasets are similar to each other and this is largely because public datasets are collected from same sources and these sources (e.g. conferences) have structural requirements to be met.

We have hand-labeled around 2400 documents from Lytica dataset. As deep learning methods are data-driven methods, this number is not enough for fully training a model or applying transfer learning, therefore we use the public datasets as well. 700 images for validation and for testing purposes, we use a test set of ICDAR-2017 POD that contain 817 images, in total we have 5600 images for training.

For the sake of simplicity, we do not employ any pre-processing or post-processing techniques to further increase the performance of the models except adding horizontally flipped images to the training data to increase the dataset size. However, applying different techniques to further improve the results are in our future agenda.

B. Training Details and Experiments

Since the amount of available annotated data is not enough to avoid overfitting, our backbone feature extractor networks are pre-trained on MS COCO [27] dataset which is one of the widely used datasets in object detection benchmarks.

We adopt the Faster-RCNN method in [13], and utilize the pre-trained VGG-16 network while RPN and fully connected layers are employed for classification and regression in lieu of the classification layer. We scale the image to make its shortest side 600 pixels long and trained with mini-batch size of 1 and 128 RoI for 8 epochs while monitoring the validation loss to avoid possible overfitting. We start with learning rate of 0.001 and then reduce it after 30000 steps.

For RetinaNet [14], we compare ResNeXt-101 [26] and ResNet-50 [4] feature extractors. ResNet-50 architecture consists of 50 layers and residual connections where ResNeXt-101 architecture is 101 layers and have residual connections as well as different pathways inside convolutional blocks. To construct the FPN, the last activations output are employed for each stage block as proposed in [24]. Images are scaled to make their shortest side 600 pixels long and then trained with mini-batch size of 2 and 64 RoI for 5 epochs for ResNeXt-101 backend and 9 epochs with every mini-batch containing 6 images for ResNet-50 backend. We monitored and stopped the training when the models started to overfit.

To compare the performance, we use Intersection over Union (IoU) to calculate precision, recall and F1 score. For predicted bounding box (P) and ground truth bounding box (Gt), IoU can be described as;

$$IoU(P, Gt) = \frac{P \cap Gt}{P \cup Gt} \tag{5}$$

[1] http://www.icst.pku.edu.cn/cpdp/data/marmot_data.htm
[2] http://personal.psu.edu/xuy111/projects/cvpr2017_doc.html
[3] http://www.icst.pku.edu.cn/cpdp/ICDAR2017_PODCompetition/dataset.html
[4] https://roundtrippdf.com/en/data-extraction/icdar-2013-table-competition/
[5] https://www.lytica.com/

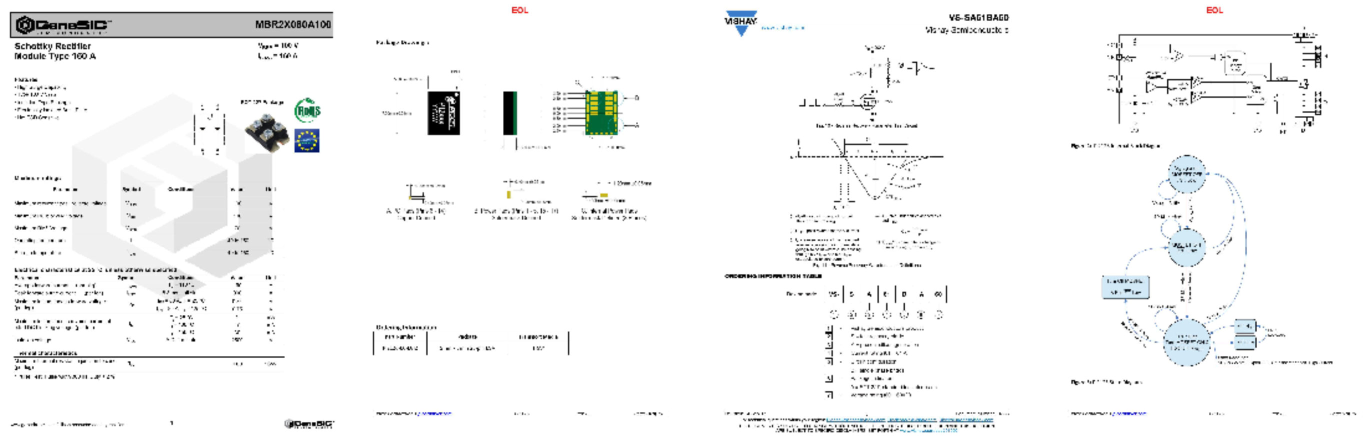

Fig. 3. Examples from private Lytica dataset. Dataset includes colored figures, state diagrams and electronical component drawings which are similar to tables. Tables have various formats and cell sizes which makes models unstable. (Content has been blurred intentionally.)

Higher IoU score denotes significantly more overlap and if IoU score is more than a predetermined threshold, a prediction is considered to be true, else false.

C. Performance Comparison

In Table II, we present the comparison of the recall, precision, and F1 scores of Faster-RCNN, RetinaNet with ResNeXt-101 backbone, and RetinaNet with ResNet-50 backbone as well as the state of the art in the task under the public ICDAR 2013 dataset, then in Table III under our private dataset provided by Lytica.

As seen in the Table II and III, the precision of RetinaNet with ResNeXt-101 backbone network is higher than the precision of with a backbone of ResNet-50 network. However, we can observe that the ResNet-50 backbone achieves a phenomenal recall value. Both models are eclipsed in overall F1 score on both datasets by Faster-RCNN, however as a large set of datasheets need to be automatically read to support supply chain automation, we believe the higher recall of RetinaNet would provide greater value to the later supply chain processes as not detecting tables may omit vital information. RetinaNet with ResNet-50 backbone achieves 2.52% higher recall than Faster-RCNN on ICDAR 2017 test set. Thus, even though ResNet-50 backbone fails to fit bounding boxes precisely, we can rely on the model to detect tables.

Faster-RCNN takes 0.4 seconds on average per image whereas, RetinaNet with ResNeXt-101 network requires 0.45 seconds on average and RetinaNet with ResNet-50 network infers an image in 0.2 seconds. When dealing with supply chains documents, it is crucial to consider inference times of the methods used in addition to the recall and precision.

The difference between our models indicates RetinaNet is more capable of detecting tables but the model cannot find a well-fitted bounding box for tables. This phenomenon can be observed on the right side of Fig. 4. Considering the ICDAR test set is composed of similar documents containing simpler table structures, a drop in precision in the Lytica dataset means that it is harder for the models to identify the structures of tables from different sources. Tabular data is inherently linearly aligned, and many of the misclassifications in the test data used (both ICDAR and Lytica) consist of linearly structured elements, such as formulas or charts. In Fig. 5, we can observe this problem clearly. A substantial amount of data points from the Lytica dataset in our training set decreased the overall detection performance under ICDAR test set as well.

It is worth mentioning that PDF-based techniques have access to the meta-data of the documents and cannot be compared with other models directly. Using PDF meta-data is helpful but scanned documents or tables that reside on web pages have different or no meta-data available to use. We are aiming to detect tables in images which is a harder problem to solve.

V. Conclusion

In this paper, we have compared solutions to automatically extract data from millions of documents that represent products in the global e-components supply chains and save valuable human time and effort. Using the deep learning models of Faster-RCNN and RetinaNet, even without any pre or post processing, high precision and recall can be used to detect and extract tables under minimal human supervision.

Checking performance on the public ICDAR dataset as well as private document sets from Lytica, both models performed well on ICDAR but a sharp 10% F1 score drop off has been observed by RetinaNet under Lytica dataset. We believe improvements could be made on RetinaNet to increase its precision as it shows promise with high recall value (99.37%), therefore our work shall continue to focus on improving the RetinaNet with ResNet-50 backbone network.

Immediate extensions of this study include applying pre-processing to the datasets as well as possible post-processing methods to highlight table structures and regress bounding boxes, thereby potentially increasing detection accuracy. Greater efforts may be placed into labelling larger quantities of data as the documents from Lytica accounted for less than half of training data. This may account for the difficulty in detecting tables on the Lytica test dataset as the e-component

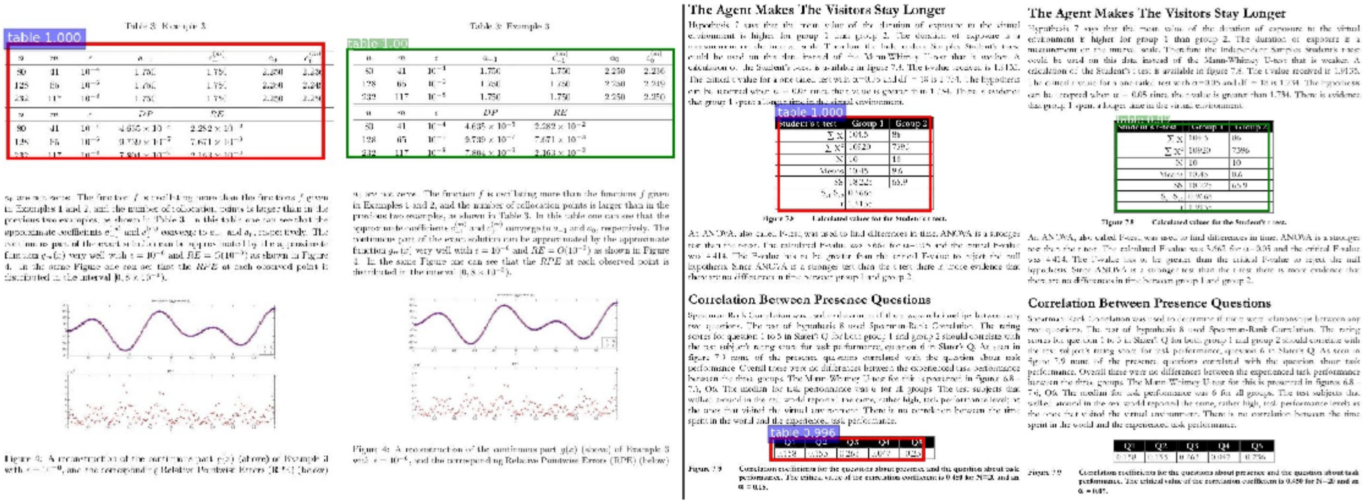

Fig. 4. Visual comparison of the Faster-RCNN and RetinaNet with ResNeXt-101 backbone models. Red detections are from Faster-RCNN and green detections are from RetinaNet. For the image on the left, detection performance is the same. But on the right, both networks have no difficulty to detect the table at the top. However, for the table on the bottom, RetinaNet cannot even detect the table whereas Faster-RCNN detects the table with high confidence.

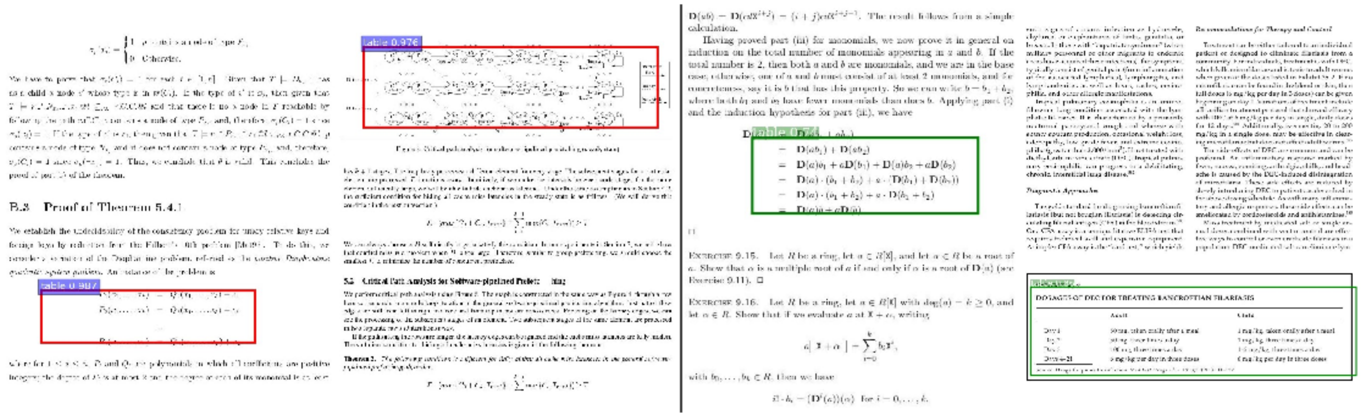

Fig. 5. Missclassification under Faster-RCNN and RetinaNet. On the left, two examples of miss classifications of Faster-RCNN model are presented. The model detected formulas as tables because they have structure as well. On the right, one miss classification and one boundary issue for RetinaNet network with ResNeXt-101 backbone are presented. RetinaNet results in less miss classifications. For example, for the examples on the left, RetinaNet does not detect a table. But it failed on a different formula for the same reason. On the far right, RetinaNet detects a table but have a difficulty to have a fair detection performance.

datasheets are largely varied and challenging, therefore greater numbers of examples are required to solidify network adaption to diverse data. Training with higher mini-batch sizes with better hardware and comparing with lower mini-batch sized models is an interesting and a controversial topic. After the detection of tables, we will work on semantic extraction from the tables.

Acknowledgement

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) ENGAGE Program and Ontario Centres of Excellence Voucher for Innovation and Productivity(VIP) I Program under Project Number 29931, and Lytica Inc.

References

[1] J. T. Mentzer, W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith, and Z. G. Zacharia, "Defining supply chain management," *Journal of Business Logistics*, vol. 22, no. 2, pp. 1–25.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[5] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, p. 640–651, Apr 2017.

[6] H. . S. S. . B. M. Shetty, Shravya Srinivasan, "Segmentation and labeling of documents using conditional random fields," 2007.

[7] S. A. Oliveira, B. Seguin, and F. Kaplan, "dhsegment: A generic deep-learning approach for document segmentation," 2018.

[8] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, and P. Messina, "A saliency-based convolutional neural network for table and chart detection in digitized documents," 2018.

[9] W. G. Hatcher and W. Yu, "A survey of deep learning: Platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies

for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results." http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[12] R. Girshick, "Fast r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, p. 1137–1149, Jun 2017.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[15] F. Shafait and R. Smith, "Table detection in heterogeneous documents," *ACM International Conference Proceeding Series*, pp. 65–72, 01 2010.

[16] 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," *International Journal of Contents*, vol. 11, no. 4, pp. 77–85, 2015.

[17] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc., 1983.

[18] X. Sun, P. Wu, and S. C. Hoi, "Face detection using deep learning: An improved faster rcnn approach," *Neurocomputing*, vol. 299, pp. 42 – 50, 2018.

[19] X. Mo, K. Tao, Q. Wang, and G. Wang, "An efficient approach for polyps detection in endoscopic videos based on faster r-cnn," *arXiv preprint arXiv:1809.01263*, 2018.

[20] A. Akselrod-Ballin, L. Karlinsky, S. Alpert, S. Hashoul, R. Ben-Ari, and E. Barkan, "A cnn based method for automatic mass detection and classification in mammograms," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp. 1–8, 2017.

[21] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *IEEE CVPR*, vol. 4, 2017.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[24] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection.," in *CVPR*, vol. 1, p. 4, 2017.

[25] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron." https://github.com/facebookresearch/detectron, 2018.

[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 5987–5995, IEEE, 2017.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *Lecture Notes in Computer Science*, p. 740–755, 2014.

[28] 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. 01, pp. 1162–1167, Nov 2017.