A Clustering Backed Deep Learning Approach for Document Layout Analysis

Rhys Agombar, Max Luebbering, Rafet Sifa

A Clustering Backed Deep Learning Approach for Document Layout Analysis

R. Agombar[0000−0001−5574−9754], M. Luebbering[0000−0001−6291−9459], and R. Sifa

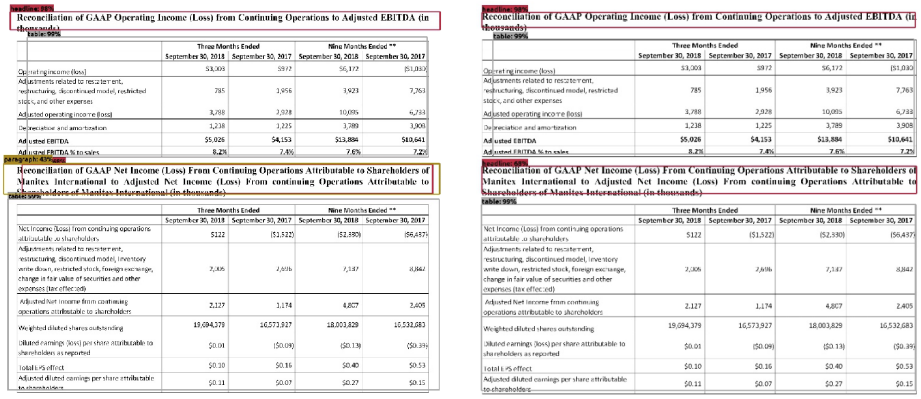
Fraunhofer Institute for Intelligent Analysis and Information Systems
Schloss Birlinghoven, 53757 Sankt Augustin, Germany

Abstract. Large organizations generate documents and records on a daily basis, often to such an extent that processing them manually becomes unduly time consuming. Because of this, automated processing systems for documents are desirable, as they would reduce the time spent handling them. Unfortunately, documents are often not designed to be machine-readable, so parsing them is a difficult problem. Image segmentation techniques and deep-learning architectures have been proposed as a solution to this, but have difficulty retaining accuracy when page layouts are especially dense. This leads to the possibilities of data being duplicated, lost, or inaccurate during retrieval. We propose a way of refining these segmentations, using a clustering based approach that can be easily combined with existing rules based refinements. We show that on a financial document corpus of 2675 pages, when using DBSCAN, this method is capable of significantly increasing the accuracy of existing deep-learning methods for image segmentation. This improves the reliability of the results in the context of automatic document analysis.

Keywords: Document Layout Analysis · Faster R-CNN · DBSCAN · Post-processing · Bounding Box Refinement

1 Introduction

Any significant organization generates documents as part of its day-to-day running. Examples of these include invoices, accounting records, and structured plans. Unfortunately, as an organization grows in size, the volume of these documents increase, often to the point where organizing and searching them becomes time prohibitive. Automated document processing systems have been proposed as a solution to this, but encounter a serious problem: the majority of documents are human, not machine, readable. For digitized documents, the PDF format is one of the most commonly used, but it is difficult to parse automatically. PDFs are designed to visualize elements without storing ordering or semantic information, which makes data retrieval difficult. Even when data can be extracted from them, PDF layouts are notoriously non-standard, with many organizations having their own unique ones. This makes effectively labeling the data difficult without human supervision.



(a) The unmodified output of an FRCNN. Note the text cut off in the first table, the duplicate prediction, and the sloppy fitting of the bounding boxes.

(b) The output, once refined using only a rules based approach.

Fig. 1: The Base and Rule-Refined bounding box predictions when classifying a document page. Here, a page excerpt containing two headers and two tables is segmented and labeled.

One proposed solution to this is to approach it as an image segmentation or object detection problem. Typically, this means we pass the PDF renderings through a neural network trained to output and label bounding boxes. In theory, the appearance of different elements in a PDF (images, headers, paragraphs, tables, etc.) should be distinct from each other, and allow a network to classify them. Previous work has shown that Faster R-CNN (FRCNN), [5, 7], or models inspired by it, [6], are suitable for this task, but require refinement in order to be feasible. As shown in Figure 1a, though a well-trained FRCNN model provides reasonably accurate bounding boxes, there are still errors that make it unusable as is. Overlaps, missing text, and sloppy box fitting all impact the accuracy of the system, especially when a document's layout is particularly dense. Indeed, [1] mentions a situation where FRCNN is known to struggle (dense tables) and claims a new approach is needed. Likewise, the problems described are present in the figures of [5], with their 'small table' example cutting off text, and 'page column alike' partially including the caption above it. To improve the usability of systems like this, accuracy must be increased, particularly when it comes to the precision of bounding box generation. Some of the mentioned problems can be fixed with rules based approaches, but alone, these can be unreliable. That is why, in this paper, we propose a clustering based approach that can improve bounding box generation in a more reliable and generalized manner, allowing automated systems to better handle diversities in layouts and improve precision when working with densely populated documents. This method is evaluated by computing a set of mAP scores for its predictions at multiple different IoU

Fig. 2: Intersecting bounding boxes pose a problem for rule based refinement, especially when they contain parts of the same text.

levels (0.75, 0.85, 0.95 and 0.99), and comparing them against the scores from predictions with only rule-based refinement, or without any refinement at all. For this, we used a custom dataset of assorted financial documents, consisting of 2675 individual pages that have been annotated with labeled bounding boxes for training and testing.

2 Layout Analysis with Deep Learning

As mentioned in the introduction, an FRCNN is an architecture for identifying and classifying regions in an image, producing a series of labeled bounding boxes as its output. The architecture consists of two parts: a region proposal network (RPN), and a classification layer. The RPN works by generating a set of 'anchor points' spread evenly across the input image. At each of these points, pyramids of boxes (called 'anchors') are computed at multiple different scales and aspect-ratios. During training, sets of ground-truth bounding boxes are provided and used to train the network to regress its anchor boxes into usable bounding boxes. Here, the anchors overlapping the ground-truths by the greatest amount (determined by calculating the intersection over union (IoU) score) are used, and the anchors that only overlap slightly or not at all are discarded. The bounding boxes generated by this proposal network are then passed to the classification layer for labeling, and the boxes, their labels, and confidence scores are output. To improve performance, the network shares certain layers, but the details of this are not necessary to understand the basic principle. If desired, more detailed explanation can be found in the paper introducing this architecture [4].

As shown in Figure 1a, this architecture is capable of reasonable bounding box generations, but merely reasonable accuracy is not sufficient for the use case of document analysis. In the first table prediction, the left-most column has been cut off, and any text extracted from this box would be missing letters. Additionally, the second header has two overlapping bounding boxes for it, which would lead to duplicated text being extracted. Though problematic, some of these issues can be fixed with rules based approaches. To deal with the overall sloppy fitting of the boxes, we take advantage of the fact that there is always a high contrast between the colour of the text and colour of the background, and apply a threshold to the image to retrieve the coordinates of the non-background pixels. The number of pixels yielded this way, however, can be substantial, especially if the page rendering contains a graphic or image. To reduce this to a more

manageable size, we pass the threshold image through a Canny edge detector and use the edge points for refinement instead. The bounding boxes can then be contracted until they fit exactly around the minimum and maximum x and y values of their contained edge points. Once the boxes have been shrunk, we compute the IoU scores between each of them. If a set of boxes overlap by more than a given threshold value (IoU of 0.75 in the first iteration, 0.3 in subsequent ones), we merge them and match the new box's label to the label from the set with the highest confidence. The two exceptions to this are: if an unacceptably large number of boxes are set to be merged, and if the set includes a table.

In the first case, we assume that an erroneous 'super-box' has been predicted, covering far too large of an area, and delete the box responsible for it instead. Though this can lead to good bounding boxes being discarded, the chance of a single box being wrong is much less likely than multiple boxes having errors. Experimentally, we found that this caused a significant increase in overall accuracy, making this trade-off acceptable.

In the special case of a table being merged, due to the typically large size of tables, and the tendency of elements within a table to be classified as something else (usually a paragraph), we assign a higher weight to the table's confidence, increasing the likelihood that it will be chosen. Again, this runs the risk of altering correct box predictions, but empirically we have observed these instances are few and far between.

We repeat this shrink-merge process once more, to arrive at a much more accurate set of predictions. Occasionally some boxes will be predicted where there they contain either no elements, or insignificant ones (like specks of dust if the digital document was generated by scanning a physical one), so to finalize the refinement, we examine the threshold image again and delete any boxes containing only a sparse set of threshold points (less than 1% of the box area).

These rules lead to the output of Figure 1b, which improves upon the classification and fitting of Figure 1a, but still has the problem of boxes not fully capturing the desired text. At some point, these boxes need to be expanded to achieve this.

This problem is a difficult one because of the nature of the document layouts. Many layouts have sections that are densely populated with distinct elements. For example, a paragraph might have a header just above it, with very little white-space separating them, and relies on bold or underlined text to be easily distinguishable for humans. Additionally, sometimes bounding boxes intersect each other very slightly. In these cases, its usually not enough to justify a merge, but is enough for both of them to partially envelope the same line of text. An example of this is shown in Figure 2. These situations mean that simple, imprecise approaches like iteratively expanding bounding boxes or matching points to the best fitting ones are unlikely work. To maximize the odds of success, we propose an approach using clustering to intelligently handle these overlaps.

Fig. 3: By clustering edge points using DBSCAN, bounding boxes can be more accurately refined.

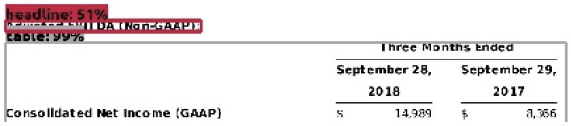


Fig. 4: Using DBSCAN, the problems of text being caught by multiple bounding boxes has been solved.

3 Improving Detection Performance by a Clustering based Refinement

In order to effectively classify items in a document that are covered by two or more bounding boxes, we need to understand the spatial extent of the intersected element. Using Figure 2 as an example, if a table's bounding box has been drawn slightly too large and it partially intersects a headline, we want to ensure that all of the text is assigned to the better fitting bounding box, rather than just parts of it. This will allow the other to contract to a more accurate shape. We solve this, and other inaccuracies, by using the clustering algorithm DBSCAN [2]. This process is our main contribution.

Since we do not know the number of 'clusters' that may be present in a given document page, DBSCAN's property of not requiring this number in advance is useful to us, and was the primary reason for the selection of this algorithm. The basic principle behind the algorithm is that it selects a data point, then counts to see if a certain number of points (including the selected one) are within a given radius ϵ of it. It then adds these points to a set of neighbours, labels the initial point as belonging to a specific cluster, and applies this process again to each of the neighbouring points, further expanding the set and classifying more and more points as belonging to the cluster at hand. Once it has exhausted its list of neighbours, it moves on to another unclassified point, increments the cluster label, and begins the process anew. In the end, the vast majority of points in the document should be labeled as belonging to different clusters. The method also classifies certain points as noise, but due to the clean images that most PDF renderings produced, these classifications were exceptionally rare. As such, we elected to ignore them, since they did not affect the system's accuracy. A more detailed description of the clustering algorithm can be found in [2].

Our use of DBSCAN clusters the given edge-points of the rendering such that each cluster will approximately cover an entire word in the text. In the cases of

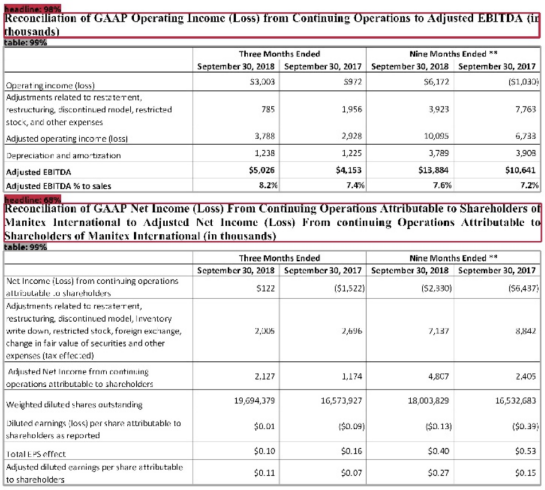


Fig. 5: The boxes from Figure 1a, now refined using both rules and clustering based techniques.

special formats like underlined or italicized text, they may span an entire line. This behavior is intended, as said formatted text likely all belongs to the same element in the page. Once the clusters are computed, the following set of rules based on their points are applied:

1. If only a single bounding box contains points from the cluster, the box will be expanded to encompass all of them.
2. If multiple bounding boxes contain points from the cluster, the box that will expand the least is chosen.
3. If multiple bounding boxes contain points from the cluster, but require no expansion, a bounding box is computed from the cluster. The IoU scores between it and the other boxes are then calculated and the box with the highest score is the one that will be chosen.

In the end, this allows the box predictions from Figure 2 to be refined to produce the image shown in Figure 4.

4 Experiments

Though the system shows good results qualitatively (Figure 5), a quantitative evaluation is needed to truly demonstrate the effectiveness of this extension. To do this, we ran a set of experiments and recorded the resulting mAP scores at multiple IoU levels.

The dataset we used for this consists of 2675 pages from assorted financial documents, with 2169 pages used for training the FRCNN and 506 used for the

	Base	Rule	Cluster
mAP@0.75	0.7839	0.7537	**0.7948**
mAP@0.85	0.5036	0.7521	**0.7948**
mAP@0.95	0.2654	0.7115	**0.7268**
mAP@0.99	0.0000	0.0000	**0.3183**

Table 1: The mAP scores at different IoU levels for the base network predictions, predictions refined using rules, and predictions refined using a combination of rules and clustering techniques.

evaluation. To compensate for this relatively small amount of data, the network was pretrained using the MS COCO dataset [3], before being refined by our training set. Regions of the dataset pages are classified using labeled bounding boxes corresponding to five different categories ('header/footer', 'headline', 'image', 'paragraph' and 'table'). Due to the level of precision needed when extracting text from documents with dense layouts, we restricted our evaluations to mAP scores using between 0.75 and 0.99 IoU. IoUs any lower than this would be wholly unusable in the context of text extraction.

The 'Base' mAP values are computed from the unmodified output of the trained FRCNN, and as shown in Table 1, do not perform well when high levels of precision are required. Our rule-refined method results in higher levels of accuracy, but due to the problems mentioned before, the lack of expansion rules limit its ability to perform optimally. For our clustering experiments, we keep the existing rule-refinements in place and add our clustering rules to allow for the expansion of too-small bounding boxes. This results in a higher level of accuracy overall, and even allows the system to function (albeit with a lower mAP score) when the required IoU ratio is raised to 0.99. The ability to work at such high levels of precision mean that the idea of backing rules based refinements with a clustering algorithm shows merit when attempting to analyse documents with dense layouts.

5 Conclusion and Future Work

In conclusion, automatic document analysis has the potential to be very useful for large organizations, but problems with non-standard layouts and densely populated documents mean approaches based on image segmentation need improvement in order to be viable. To solve this, we propose a clustering based approach using DBSCAN that can be combined with simple, rules based, refinements. As our experiments show, this successfully increases the performance of a deep-learning system when high levels of precision are required. For future work, we would like to improve the run-time of this method. Our clustering approach solves the problem it set out to, but does so at a significant computational cost. While it is usable as is, we would like to investigate other, less computationally expensive methods to refine our bounding boxes. This would improve run-time

performance and increase our contribution's viability when dealing with mass amounts of documents.

References

1. Déjean, H., Meunier, J.L., et al.: Versatile layout understanding via conjugate graph. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 287–294. IEEE (2019)
2. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. p. 226–231. KDD'96, AAAI Press (1996)
3. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014)
4. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
5. Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: 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, pp. 1162–1167. IEEE (2017)
6. Singh, P., Varadarajan, S., Singh, A.N., Srivastava, M.M.: Multidomain document layout understanding using few shot object detection. arXiv preprint arXiv:1808.07330 (2018)
7. Soto, C., Yoo, S.: Visual detection with context for document layout analysis. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3464–3470. Association for Computational Linguistics, Hong Kong, China (Nov 2019)