Using Deep Learning-based Object Detection to extract structure information from scanned documents

Alice Nannini, Federico A. Galatolo[1][a], Mario G.C.A. Cimino[1][b] and Gigliola Vaglini[1][c]
[1]*Department of Information Engineering, University of Pisa, Largo L. Lazzarino 1, Pisa, Italy*
alice.nannini@gmail.com, {federico.galatolo, mario.cimino, gigliola.vaglini}@ing.unipi.it

Keywords: Deep Learning, Computer Vision, Object Detection, Region Proposal, Document Layout Analysis, Information Extraction, Transfer Learning.

Abstract: The computer vision and object detection techniques developed in recent years are dominating the state of the art and are increasingly applied to document layout analysis. In this research work, an automatic method to extract meaningful information from scanned documents is proposed. The method is based on the most recent object detection techniques. Specifically, the state-of-the-art deep learning techniques that are designed to work on images, are adapted to the domain of digital documents. This research focuses on play scripts, a document type that has not been considered in the literature. For this reason, a novel dataset has been annotated, selecting the most common and useful formats from hundreds of available scripts. The main contribution of this paper is to provide a general understanding and a performance study of different implementations of object detectors applied to this domain. A fine-tuning of deep neural networks, such as Faster R-CNN and YOLO, has been made to identify text sections of interest via bounding boxes, and to classify them into a specific pre-defined category. Several experiments have been carried out, applying different combinations of data augmentation techniques.

1 INTRODUCTION

This work aims to offer a cutting-edge approach to the processing of scanned documents that have a non-regular text structure. As a reference domain, the one of *play scripts* is considered. In this domain, a standard script includes the following structure: title, author, followed by a sequence of scenes and acts, a list of sentences (lines) with an associated character, and different types of descriptions and notes. Such sections may be in different positions of the document, with different formatting, including different font sizes and styles. The purpose of this research is to propose a deep learning approach to automatically identify position, content, and nature of each element, for script enrichment with metadata.

On one side, this problem can be considered as an *object detection* task, since it is based on identifying the text sections and their coordinates as objects in an image. On the other side, the problem can be formulated as a *document layout detection*, because it is based on associating each text box with its functionality within the script layout. A solution to this problem is composed by a set of labelled bounding boxes for the regions of interest (ROI) on each document page. The considered approach is language independent, because it does not explicitly consider the text content. Although the text contents of many regions can be useful for the purpose of classification, the approach based on vision has the advantage of generality with respect to language, without using text features.

This paper is organized as follows. Section 2 covers a survey of related works. Then, Section 3 discusses the architectural design. Experimental results are discussed in Section 4. Finally, Section 5 draws conclusions and future work.

[a] https://orcid.org/0000-0001-7193-3754
[b] https://orcid.org/0000-0002-1031-1959
[c] https://orcid.org/0000-0003-1949-6504

2 RELATED WORK

Several tools exist to manage texts extracted from scanned documents, and a variety of research works are based on processing document contents. A problem largely considered by the literature is related to the analysis of scientific papers. In (Soto & Yoo, 2019) an adaptation of the *Faster R-CNN* (Region Based Convolutional Neural Networks) object detection model is proposed to facilitate the automatic knowledge extraction from scientific articles. A similar approach is found in (Yang & Hsu, 2021), where the authors consider the analysis of layout of a scientific document as an object detection task on digital images. Here, the approach is based on fine-tuning the two stages of the Faster R-CNN, pre-trained on the Microsoft Common Objects in Context (MS COCO) dataset.

A different research domain is the analysis of historical documents, on which a variety of studies have been done, based on text recognition. With regard to this approach, in (Lombardi & Marinai, 2020) a survey of different deep learning techniques used on this type of documents is provided. The most used architectural models are Fully Convolutional Networks (FCNs), Faster R-CNN, Mask R-CNN and other models for object detection. In (Pondenkandath *et al.*, 2017) an architecture based on Long Short-Term Memory (LSTM) technology is proposed. Finally, in (Ziran *et al.*, 2019), an approach based on Faster R-CNN pre-trained on COCO data is carried out to solve a page layout analysis problem. On the other side, pixel-wise approaches are also widespread, performing page segmentation and classification by processing each pixel using a trained CNN (Barakat & Al-Sana, 2018).

Recent works are trying new approaches based on the *YOLO* (You Only Look Once) *v3* model, a state-of-the-art, real-time object detection system (Huang et al., 2019). YOLO achieves promising and robust performance, although it does not have a large application in the documents domain. Overall, the most used approach, on which more information and data are available, is the Faster R-CNN object detection model, pre-trained on a document-based dataset, and then fine-tuned on a custom dataset. As a consequence, in this paper Faster R-CNN and YOLOv5 will be compared.

3 DATASET STRUCTURE

In the literature, there is a lack of benchmark data in the domain of play scripts. As a consequence, it has been necessary to create a data set. Figure 1 shows an example of script. Specifically, an initial analysis of a large database of more than 10k publicly vailable play scripts (GTTempo, 2022), has been carried out. As a result, 6 classes of interest have been identified in the structure:

- *Title*: it refers to the title of the play, usually in the first page of a script;
- *Author*: it refers to the name of the script's author and related information;
- *Characters' List*: it refers to the list of all the characters in the play, usually placed on the first page after the title;
- *Subtitle*: it refers to the statement of an act or a scene, but also to other paragraph titles;
- *Description*: the text section contains a scene's description, a global note, or similar;
- *Dialogue*: it refers to each line in the text, and consists of the character's name, the text of the line, and any related notes.

The document annotation has been carried out using PAWLS (Neumann et al., 2021), a tool designed and optimized for scanned documents. The resulting machine learning set consists of 316 annotated pages, for overall 109 documents. The distribution of the classes is strongly unbalanced towards the *Dialogue* class.

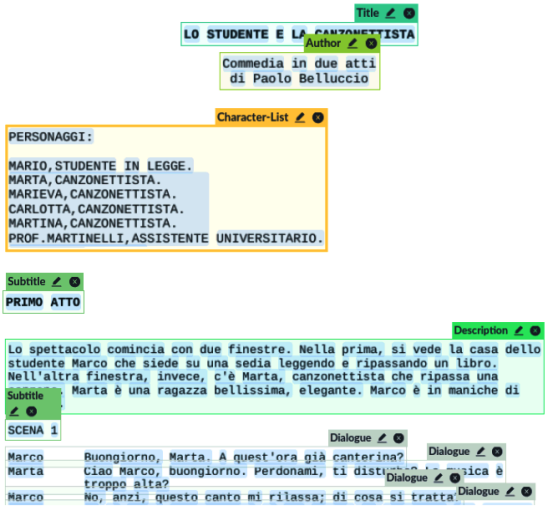

Figure 1: Example of annotated document page.

4 IMPLEMENTATION AND EXPERIMENTS

Experiments have been carried out with the *Google Colab* (Bisong, 2019) a free platform based on the open-source Jupyter project, that allows to write and run Python code through a browser with no

configuration required and with free access to GPUs. The device we got to use is a NVIDIA Tesla K80 GPU.

4.1 Faster R-CNN

Faster R-CNN (Ren et al., 2015) is a two-stage object detector that belongs to the region-based CNN family. We refer to the implementation provided by the framework *Detectron2*. It has a structure based on an initial stage where the Region Proposal is performed, implemented by a deep fully-connected network called RPN ("Region Proposal Network"). The first stage of the RPN starts with the backbone, based on a Residual Network, which extracts the features of the input image. Subsequently, it takes the feature maps coming out from different levels of the backbone, to work on multiple scales, and it searches for regions of interest via a sliding window and various anchors of different sizes. As a result, the bounding boxes of the Regions Of Interest (ROI) are generated. The second stage consists in the real detection, in which a network is trained to classify the ROIs. This stage is implemented by the convolutional head of the Fast R-CNN network (Girshick, 2015). It takes as an input both the ROIs and the feature maps extracted from the backbone.

The model is pre-trained on *PubLayNet* (Zhong, 2019), one of the largest datasets for document layout analysis, counting more than 360k annotated page images, mostly from scientific articles. The model has been fine-tuned for 10k epochs on 283 play script pages, and then tested on the remaining 33 pages, with a batch size of 256, and a starting learning rate of 5E-6 which then stabilizes at 250E-6 after 1000 warm-up iterations. Anchor scales of [32, 64, 128, 256, 512], anchor ratios of [0.5, 1.0, 2.0], and anchor angles of [-90, 0, 90] have been set. Regarding the data augmentation, the default setting has been included the scale jittering. With an Intersection-Over-Union (IOU) threshold of 0.5, the model achieved a mean average precision (mAP@0.5) of .566 over all six classes, with peak class performance on *Dialogue* sections (.623) and lowest performance on *Author* (.319).

Additional data augmentation techniques have been integrated into the configuration. First, the default scale jittering parameters have been updated according to the size of the dataset images. Second, cropping and horizontal flipping have been enabled. Third, the augmentation techniques have been enabled also for the test images. As a result, the model has been retrained, achieving a mean mAP@0.5 of .622, with peak class-performance of .636 on *Dialogue* text sections, and a lowest performance of .346 on *Title* class. Table 1 shows a summary of the mAP@0.5 by class. Figure 2 shows the normalized confusion matrix on test set. Figure 3 shows the validation performance during the training phase. Table 2 shows various validation results against epochs. Finally, an example of inference on a test image is shown in Figure 4. Finally, an example of inference on a test image is shown in Figure 4.

Table 1: Faster R-CNN, summary of mAP@0.5 by class.

	mAP
Title	.346
Author	.395
Subtitle	.569
Description	.410
Character List	.420
Dialogue	.636

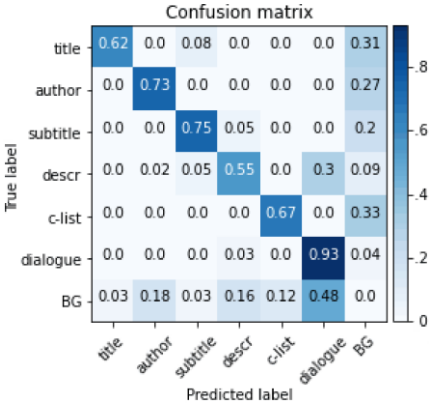

Figure 2: Faster R-CNN, normalised confusion matrix on test set.

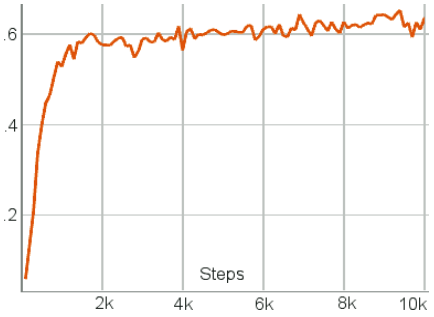

Figure 3: Validation mAP@0.5 trend of Faster R-CNN model during the training phase.

Table 2: Faster R-CNN, validation results again epochs.

Epochs	AP	AP_{50}	AP_{75}	Loss
1k	.351	.528	.382	1.073
2k	.389	.577	.427	.902
3k	.422	.589	.498	.831
4k	.388	.564	.445	.797
5k	.435	.598	.483	.796
6k	.439	.610	.528	.780
7k	.462	.626	.530	.775
8k	.465	.627	.531	.784
9k	.474	.643	.558	.782
10k	.472	.635	.543	.772

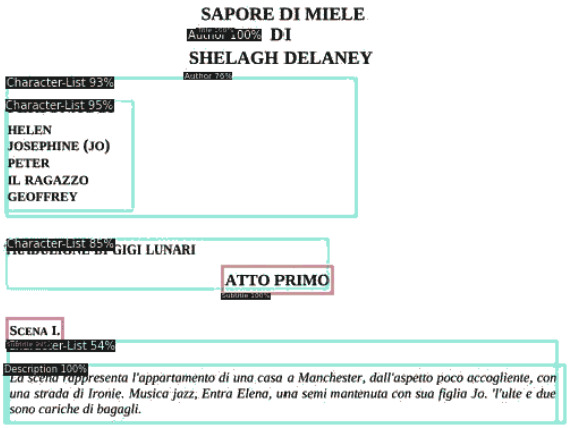

Figure 4: Faster R-CNN inference on test image.

4.2 YOLOv5

For a better assessment, the results achieved by the Faster-CNN model have been compared with a YOLO model (Redmon *et al.*, 2016). The purpose is to understand whether improvements can be achieved via a different approach.

Specifically, YOLO version 5 has been considered, because – although under development – it offers performance comparable to its previous versions with shorter training and inference time. Moreover, it is implemented on the PyTorch framework, which makes it easier to compare with the Faster R-CNN model. In contrast, the earlier versions of YOLO are based on the Darknet framework.

YOLOv5 is based on a one-stage detection, an approach resulting in a simpler structure and less time consumption, than two-stage networks such as Faster R-CNN. In essence, YOLOv5 processes the whole image at once through a CNN, without making predictions about many regions of an image. YOLOv5 divides an input image into an $S \times S$ grid. For each grid cell, it predicts B bounding boxes with related confidences (called "objectness score"), and related membership probabilities for all C classes.

The CNN is made by a backbone to extract the features, which are then received at 3 different scales as input by the detector. The detector predicts the bounding boxes and the classes. The model offers a very extensive data augmentation solution: by default, various colour adjustment techniques, translation, scaling, horizontal flipping, and mosaic augmentation are provided.

YOLOv5 offers multiple model implementations of different sizes, resulting in different trade-offs between inference time and accuracy. The adopted version is the "medium", which consists of about 21 million of trainable parameters (Faster R-CNN has 41 million of them), and is more robust with respect to previous versions.

The model has been configured with the weights pre-trained with the COCO dataset, and it has been trained for 500 epochs, with a batch size of 16. The anchors have been set through the execution of a k-means algorithm on the training data, while the learning rate was scheduled according to the One-Cycle policy (Smith, 2018). The Early Stopping technique with a patience of 100 have been set.

As a result, a mAP@05 of .703 has been achieved, which is higher than the performance of .622 achieved by Faster R-CNN. The peak performance of .899 has been achieved with the *Dialogue* class, while the worst class remains *Title* with a mAP of .481.

Table 3 shows a summary of validation results by class. Figure 5 shows the normalized confusion matrix on test set, whereas Figure 6 shows the precision-recall ROC curve. Figure 7 shows the validation performance during the training phase. It is clear that YOLO achieves a higher precision with less than an order of magnitude of steps with respect to Faster R-CNN. Finally, Figure 8 shows an example of inference on a test image.

Table 3: YOLOv5, validation results by class.

Class	Precision	Recall	AP_{50}	AP_{95}
All	.763	.671	.703	.560
Title	.726	.538	.481	.328
Author	.710	.727	.684	.591
Subtitle	.757	.737	.778	.581
Description	.724	.525	.611	.508
Character List	.789	.626	.767	.613
Dialogue	.872	.871	.899	.738

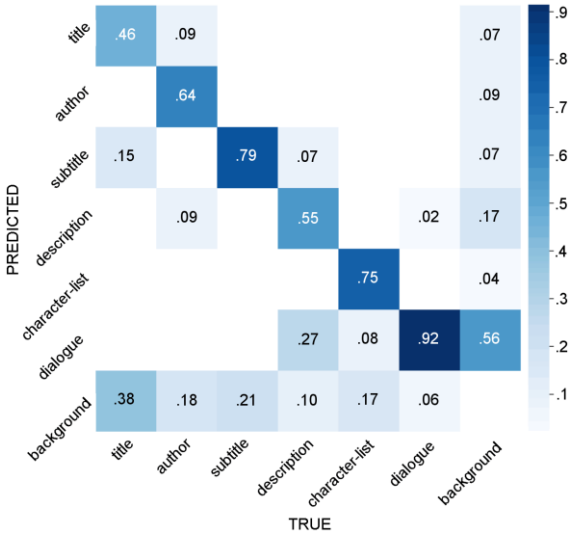

Figure 5: YOLOv5, normalised confusion matrix on test set.

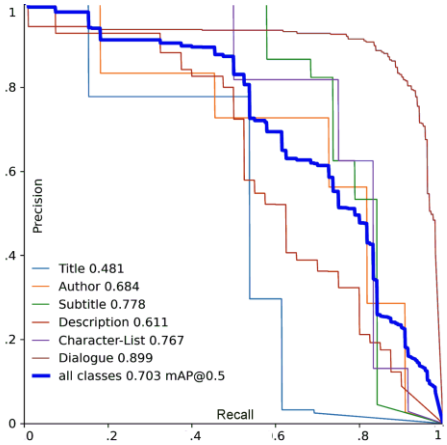

Figure 6: YOLOv5, precision-recall ROC curve

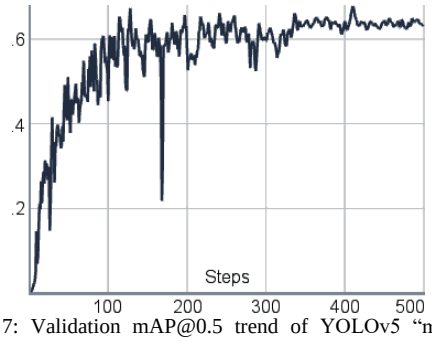

Figure 7: Validation mAP@0.5 trend of YOLOv5 "medium" model during the training phase.

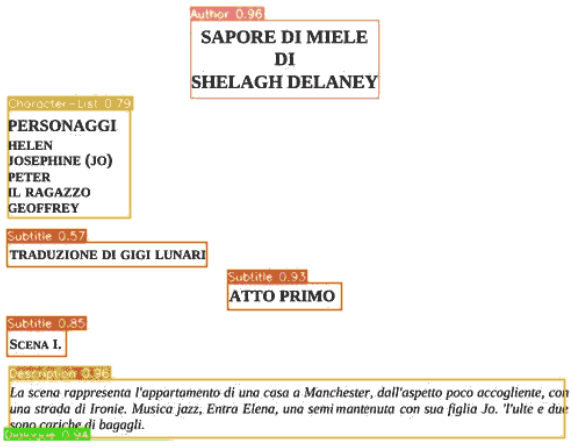

Figure 8: YOLOv5 inference on test image.

5 CONCLUSIONS

This paper compares state-of-the art deep learning solutions for the automatic extraction of structure information from scanned documents. The study considers play scripts as a type of document. For this purpose, the latest object detection techniques have been adopted. Given the unavailability of play script benchmark in the literature, a novel data set has been generated, via publicly available repositories.

In the architectural design, selected technologies for natural image processing have been adapted to the domain of digital documents. Specifically, Faster R-CNN and YOLOv5 have been considered.

Although a more in-depth exploration of the approaches, and an enrichment of the benchmark are needed, the experimental results are promising, and the object detection technology based on deep learning has proved to be easily adaptable and effective for the document domain. More recently, a novel approach of object detection has been proposed in the literature (Carion *et al.*, 2020). The approach, called DEtection TRansformer (DETR), streamlines the pipeline, by removing many hand-designed components. DETR demonstrates accuracy and run-time performance comparable with Faster-R-CNN. An extensive study in this direction can be a future work to bring a contribution in the field.

ACKNOWLEDGEMENTS

This work has been supported by: (i) the IT company *Wondersys* Srl, Leghorn, Italy; (ii) the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence); (iii) the Italian Ministry of University and Research (MUR), in the framework of the FISR 2019 Programme, under Grant No. 03602 of the project "SERICA". The authors thank teatropertutti.it, a blog founded by Rebecca Luparini. A demonstrator of the proposed system will be hosted on www.showteams.it

REFERENCES

Barakat, B. K., & El-Sana, J. (2018, March). Binarization free layout analysis for arabic historical documents using fully convolutional networks. In *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)* (pp. 151-155). IEEE.

Bisong E. (2019) Google Colaboratory. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8_7

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko S. (2020). End-to-End Object Detection with Transformers. *arXiv:2005.12872v3*

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

GTTempo, Copioni, https://www.gttempo.com/copioni, accessed 2022.

Huang, Y., Yan, Q., Li, Y., Chen, Y., Wang, X., Gao, L., & Tang, Z. (2019, September). A YOLO-based table detection method. In *2019 International Conference on Document Analysis and Recognition (ICDAR)* (pp. 813-818). IEEE.

Lombardi, F., & Marinai, S. (2020). Deep learning for historical document analysis and recognition—a survey. *Journal of Imaging*, *6* (10), 110.

Neumann, M., Shen, Z., & Skjonsberg, S. (2021). PAWLS: PDF Annotation With Labels and Structure. *arXiv preprint arXiv:2101.10281*.

Pondenkandath, V., Seuret, M., Ingold, R., Afzal, M. Z., & Liwicki, M. (2017, November). Exploiting state-of-the-art deep learning methods for document image analysis. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* . (Vol. 5, pp. 30-35),. IEEE.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*, 91-99.

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.

Soto, C., & Yoo, S. (2019, November). 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).

Yang, H., & Hsu, W. H. (2021, January). Vision-Based Layout Detection from Scientific Literature using Recurrent Convolutional Neural Networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, (pp. 6455-6462,). IEEE.

Zhong, X., Tang, J., & Yepes, A. J. (2019, September). Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)* (pp. 1015-1022). IEEE.

Ziran, Z., Marinai, S., & Schoen, F. (2019) Deep learning-based object detection models applied to document images. UniFi.