Automatic Detection of Comic Characters: An Analysis of Model Robustness Across Domains

Javier Lucas, Antonio Javier Gallego(✉), Jorge Calvo-Zaragoza, and Juan Carlos Martinez-Sevilla

Department of Software and Computing Systems, University of Alicante, Alicante, Spain
jla45@alu.ua.es, {jgallego,jcalvo}@dlsi.ua.es, jcmartinez.sevilla@ua.es

Abstract. The popularity of comics has increased in the digital era, leading to the development of several applications and platforms. These advancements have opened up new opportunities for creating and distributing comics and experimenting with new forms of visual storytelling. One of the most promising research areas in this field is the use of deep learning techniques to process comic book images. However, one of the main challenges associated with the use of these models is adapting them to different domains because comics greatly vary in style, subject matter, and design. In this paper, we present a study on the problem of generalization across different domains for the automatic detection of characters in comics. We evaluate the performance of state-of-the-art models trained in different domains and analyze the difficulties and challenges associated with generalization. Our study provides insights into the development of more robust deep-learning models for processing comics' characters and improving their generalization to new domains.

Keywords: Comic Analysis · Character Detection · Domain Shift

1 Introduction

A *comic* is a means of artistic expression that combines text and images to tell a story. The history of comics goes back to the end of the 19th century, when the first publications of comic strips appeared in the press. Since then, comics have evolved and diversified, encompassing a wide variety of genres and styles, from science fiction and fantasy to horror and drama, among others.

In the digital era, comics have undergone a significant transformation owing to new technologies and the growing popularity of mobile phones and tablets. Currently, there are several applications and digital platforms that allow access to a wide variety of comics instantly. This has opened up new opportunities for reading, creating, and distributing comics, as well as experimenting with new forms of visual storytelling.

M. Coustaty and A. Fornés (Eds.): ICDAR 2023 Workshops, LNCS 14193, pp. 151–162, 2023.
https://doi.org/10.1007/978-3-031-41498-5_11

Within this context, the development of applications for comics has gained great importance in recent years, as for instance interactive reading and reading assistance for people with functional diversity [15]. Interactive reading allows the reader to explore the story in a more dynamic and immersive way, while assistive-reading applications help people with visual or learning disabilities enjoy the comic in a more accessible and personalized way.

The use of deep learning techniques to process comic book images is one of the most promising research areas nowadays to develop new applications. However, the associated challenges encompass several difficulties, such as the complexity of the illustrations, the variability in the drawing style, the composition of the strips, or the non-linear narrative [8]. To address these, several approaches based on computer vision have been proposed (cf. Sect. 2).

Furthermore, in addition to the challenges inherent in any task (as in comics here), one of the main drawbacks associated with the use of deep learning models is the difficulty of adapting the models trained for a particular comic to different domains. This is because comics can be highly variable in terms of style, subject matter, design, and other factors that influence the appearance and layout of the panels. This problem of learning neural networks that generalize across different domains is one of the main challenges faced by researchers in this area.

In this work, we present a study on the problem of generalization across different domains for the automatic detection of characters in comics. This is important because comics feature a wide range of artistic styles and character designs, making it challenging for a recognition system trained on one comic to perform well on another. Whether existing approaches can be well generalized to other domains in comics is a key piece for future strategies, not just to a task but general to the field of comic book processing itself.

In particular, we here evaluate the performance of state-of-the-art models trained in different domains of comics for character recognition, and we analyze the difficulties and challenges associated with adapting the models to other domains. Our work demonstrates improved cross-domain character detection through multi-dataset training.

2 Background

Compared to other graphics recognition fields, the automatic processing of comic sources is barely explored [11]. The new contributions usually focus on one specific task within the full spectrum of challenges.

In particular, Iyyer et al [8] explored whether computers can understand the implicit narrative conveyed by comic book panels, which often rely on readers to infer unseen actions. The authors built a dataset containing over 1.2 million panels with automatic textbox transcriptions. They asked deep models to predict narrative and character-centric aspects of a panel given context from preceding panels. In general, the automatic models underperformed human baselines, suggesting that comics present fundamental challenges for both vision and language understanding for current technology.

Nguyen et al. [13] described a method for indexing digital comic images. They considered deep learning to automatically split images into panels, and encode and index them through XML-based formats. The authors evaluated their method on a dataset consisting of online library content and proposed a new public dataset.

Concerning the automatic detection or recognition of characters—which is the scope of the present work—there also exists some previous work. Nguyen et al. [12] considered an object-detection model trained with a custom dataset called *Sequencity*, comprising a total of 612 pages. They tested the resulting models against different datasets, such as Sun60, Ho42, or Fahad18. Their method reported better results than those achieved until that date. Furthermore, Dutta and Biswas [2] not only trained a model to perform character recognition, but also panel detection. To accomplish these tasks, they applied Transfer Learning on a deep learning model using a new dataset they created: the Bengali Comic Book Image Dataset (BCBId). They tested the resulting model on different available datasets, such as eBDtheque, Manga109, or DCM, obtaining successful results.

Similarly to the approaches mentioned above, we also propose addressing character recognition in comics using deep learning. In our case, however, we particularly focus on studying the generalization capacity of these approaches to domains other than the one used for training.

3 Method

Object detection stands for the task of locating and delimiting within a bounding box the different elements present in an image for eventually classifying them [18]. Conceptually, these techniques model a continuous function that predicts the bounding boxes along with a discrete predictor that infers their associated labels.

A model that has stood out in object detection—which will be the architecture considered in this work—is the YOLO (You Only Look Once) algorithm. YOLO was proposed by Redmon et al. [16], and since then several versions have been developed that improve both the speed and accuracy of the method. The main idea behind YOLO is to look at the image only once (as its name indicates), as opposed to other proposals of the state of the art, such as DPM v5 [3], Fast-RCNN [5], Faster-RCNN [17], or Mask-RCNN [7], which perform this task in two steps: first, they detect the bounding boxes and then they perform a classification step. The reader is referred to the work by [19] that provides a comprehensive review of the existing formulations and architectures.

YOLO is composed of a backbone architecture that extracts image features and a neck that generates bounding boxes for objects and the associated class probabilities. It first divides the whole image into a $S \times S$ grid. Every cell in this grid will generate B bounding boxes, each represented by the spatial coordinates of the box, the confidence of containing an object, and the conditional probability of the object belonging to a particular class. The loss function used is a combination of these components: the sum of the mean squared error (MSE) between the predicted and true coordinates, the binary cross-entropy loss for the confidence score, and the cross-entropy loss for the class probabilities.

In our particular case, we will configure this architecture for the prediction of a single class: comic book characters. Since this type of data contains a wide variability of drawing styles, colors, types of characters, etc., and given the relatively limited amount of data available, we will resort to the use of transfer learning techniques for model initialization and data augmentation processes to improve the results and generalization capabilities of the models obtained (described in detail in Sect. 4.2).

Transfer learning consists of taking advantage of the knowledge learned for a domain with sufficient labeled data and applying it to another through a process of fine-tuning—i.e., using the weights of the trained model and using it as initialization for a new model. In our case, we will freeze the first part of the network (i.e. the backbone), so that these weights do not change during training. This solution helps to reduce the amount of labeled data needed for training, to make training converge faster, and to obtain models that generalize better.

In addition, since it is intended to analyze the performance of the model when training at the panel level as well as with the complete pages of the comics, it was necessary to perform a preprocessing of the datasets (see Fig. 1). First, the annotations available in the datasets were used to extract the panels. The characters were then associated with the panel in which they were contained, readjusting the coordinates of the panels in some cases. Finally, the character coordinates were recalculated to make them relative to the panel and to conform to YOLO's format (center x, center y, width and height, all rescaled in the range $[0, 1]$, so that the values are independent of the size of the image).

4 Experimental Setup

This section describes the configuration followed during the experimentation process, including the considered datasets, the details of the training process, and the evaluation metrics. All the experiments were performed using an Intel(R) Xeon(R) CPU @ 2.00 GHz with 12 GB RAM and an Nvidia Tesla T4 GPU with 12 GB of RAM.

4.1 Datasets

For the experimentation, we considered two datasets from the state of the art: eBDtheque [6] and Manga109 [4]. A description of them is provided below. Table 1 includes a summary of their characteristics and some random image examples can be consulted in Fig. 2, which shows the great variability of the characters and the appearance of the comics considered.

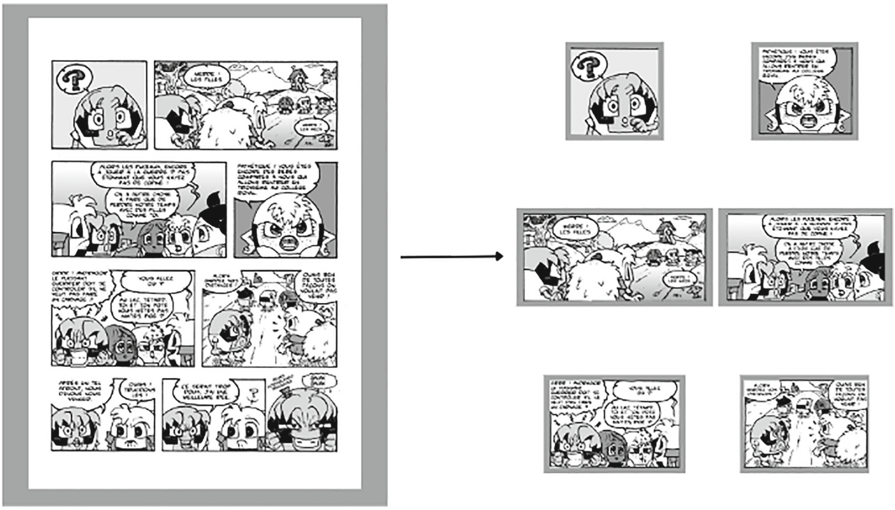


Fig. 1. Graphic representation of the panel extraction process from comic book pages.

eBDtheque. This dataset contains color images of hundreds of comics, most of them from French-Belgian authors. The ground truth includes labels for panels, text lines, and the main characters. We will use both the annotations for panels and characters.

Manga109. It contains 109 grayscale Japanese comics, with more than 21,100 pages in total. It includes labels for panels and characters, differentiating between the face and the whole body. In our experiments, we will use 25 % of all the data available in Manga109, both for pages and panels. We made this decision due to the resources available, as this helps us to reduce training time without notably reducing the results obtained.

Table 1. Summary of the characteristics of the datasets considered, including the number of samples and the minimum, maximum, and average resolution, both at the page level and at the panel level.

Dataset	Labeling level	# samples	Resolution (width×height px.)		
			Avg.	Min.	Max.
eBDtheque	Pages	96	1,841 × 2,230	800 × 301	5,320 × 3,632
	Panels	715	644 × 571	121 × 58	4,959 × 3,422
Manga109	Pages	2,517	1,666 × 1,179	1,654 × 1,170	2,960 × 2,164
	Panels	20,669	404 × 365	60 × 62	2,224 × 1,470

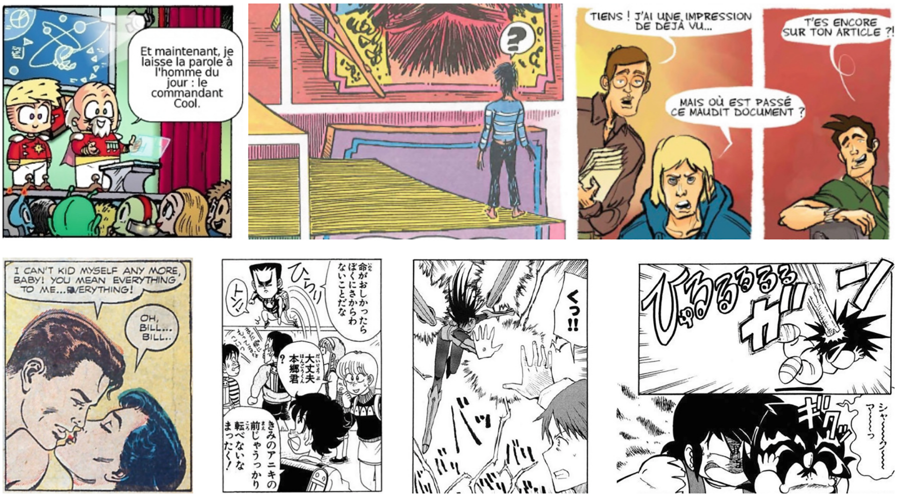

Et maintenant, je laisse la parole à l'homme du jour : le commandant Cool.
?
TIENS ! J'AI UNE IMPRESSION DE DÉJÀ VU...
MAIS OÙ EST PASSÉ CE MAUDIT DOCUMENT ?
T'ES ENCORE SUR TON ARTICLE ?!
I CAN'T KID MYSELF ANY MORE, BABY! YOU MEAN EVERYTHING TO ME...EVERYTHING!
OH, BILL... BILL..
命がおしかったらぼくにさからわないことだな
ひらり
トンッ
大丈夫本郷君
?
きみのアニキの前じゃうっかり転べないなまったく!
くっ!!
バッ
ひゅるるるるる
ガ
ギクッ
シャ〜〜ク〜〜マ〜〜っ

Fig. 2. Random examples from the datasets considered for the experimentation.

For the experimentation, we used 80% of the available data to train the models, 10% for validation, and the remaining 10% for testing. Since there are many more samples from Manga109 than from eBDtheque, we applied *oversampling* to generate the models that combine both datasets during training. This technique consists of duplicating the images of the dataset with fewer samples until the number of samples is balanced, so that the model is trained with the same amount of images of each of them. This approach eliminates the possible bias towards a type of data and also ensures that the result obtained cannot be attributed to the use of more or less data for a specific dataset.

4.2 Network Configuration

In our experiments, we used the YOLO v5 version of the algorithm [9], which provides several network sizes (from small to large) with different numbers of parameters. We selected the medium size (with a total of 21.2 million parameters) since after several preliminary tests we determined that it presented a good balance between efficiency (in the resources available) and efficacy.

The original model was initialized using the pre-trained weights obtained for the Common Objects in COntext (COCO) corpus [10] for object detection. From this initialization, we carry out the fine-tuning process for the datasets considered in this work. In this process, the first part of the network (i.e. the backbone) was frozen and only the final layers were trained during 100 epochs, which, as will be seen, was sufficient for the convergence of the model due to the good initialization. The learning rate was fixed to 10^{-3}, with a weight decay regularization of 10^{-4}, and considering a warm-up process of 3 epochs with a learning rate of 10^{-1}. Stochastic Gradient Descent [1] was selected as the optimization function with a mini-batch size of 32 samples.

Regarding the data augmentation, we considered a collection of transformations that were randomly applied to each training image: horizontal flips (with a 50% of probability), mosaics, translations of the images by -10% to 10% on both axes, axis-independent scale changes within -50% and 50% of their original size, and color alterations in the Hue-Saturation-Value (HSV) color space by a fraction of 0.015, 0.7, and 0.4, respectively to each of these channels.

4.3 Metrics

In terms of evaluation, we considered different figures of merit commonly used in the assessment of object detection methods [14]: Average Precision (AP), F-measure (F_1), Precision (P), and Recall (R).

Considering a predicted character $\hat{m}$ and its associated m ground-truth annotation, we first calculate the Intersection over Union (IoU) as the ratio between their overlapping area and the total surface covered by their union. Based on this indicator, a character is considered correctly detected *iff* $\mathrm{IoU} \geq \delta$, where $\delta \in [0, 1]$ represents an evaluation threshold that relates to the severity of the assessment.

The figures of merit considered build upon this definition using the IoU value obtained for one or more thresholds. F_1, P, and R are defined as:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

$$\mathrm{F}_1 = \frac{2 \cdot \mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{3}$$

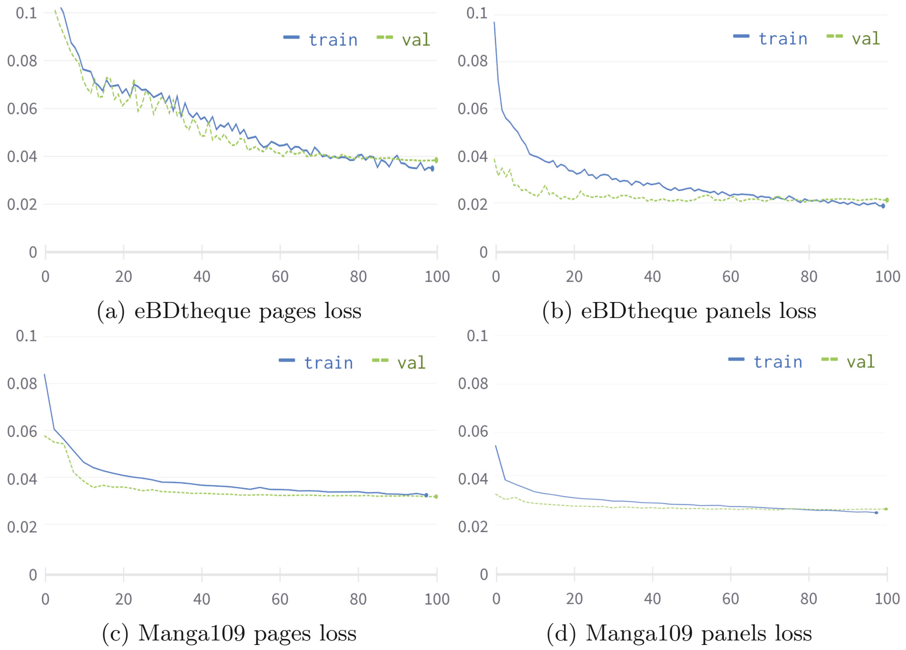
where TP, FP, and FN denote the True Positives (number of correctly detected characters), False Positives (number of incorrectly detected characters), and False Negatives (number of non-detected or missed characters) for a given δ threshold, for which we will use $\delta = 0.5$ since it is the value commonly used for this type of tasks.

Regarding the AP metric, it measures the area under the precision-recall curve, representing the average of the individual ratios between the number of correctly estimated characters and the number of elements to retrieve. Since this metric also depends on the δ threshold, we resort to the values usually reported in related works: $\delta = 0.5$ (denoted as $\mathrm{AP}^{0.50}$) as well as the average of the AP scores (referred to as AP^{m}) when considering 11 equispaced threshold values in the range $\delta \in [0.5, 1]$.

5 Results

In this section, the proposed method is evaluated using the experimental setup previously described.

In the first place, we analyze the training process of the proposed architecture in order to assess the model's learning capacity on the training data both at the page and panel level, and the generalization made with the validation set. Figures 3a to 3d show the evolution of the loss values across the epochs for the training and validation sets of the two datasets considered. As can be seen, the models manage to converge in all cases and do not perform overfitting. In addition, learning at the panel level seems to be easier, since the error is reduced faster and a lower error rate is reached.



(a) eBDtheque pages loss

(b) eBDtheque panels loss

(c) Manga109 pages loss

(d) Manga109 panels loss

Fig. 3. Evolution of the bounding box loss (vertical axis) across the epochs (horizontal axis) for the training and validation sets of eBDtheque and Manga109.

We now analyze the results obtained with the models trained for the different scenarios contemplated: train at the page and panel level and carry out the evaluation both in the same data domain (i.e. intra-domain case) and in the other domain considered (i.e. inter-domain case). Besides, the case of combining both domains for training is also studied. The results of these experiments can be seen in Table 2, which details the dataset used for training (*source* column) and for the evaluation (*target* column), and whether training was conducted at the page or panel level.

Analyzing these results in general, it is observed that training at the page level and the panel level achieves a similar average performance for all the metrics, which slightly improves or worsens depending on the dataset used. Looking in detail at the case of training with a single domain and performing the intra-domain evaluation, eBDtheque obtains a better result at the page level and Manga109 at the panel level, although with fairly close figures for all metrics.

In the case of the inter-domain evaluation, it also depends, since when training with Manga109 and evaluating with eBDtheque (Manga109→eBDtheque) a better result is reported at the panel level, while for eBDtheque→Manga109 a certain improvement is obtained at the page level. In both cases, the results are quite good considering that the models were not trained with data from the same domain. It is important to remember that the styles are very different, one in color and the other in grayscale.

Table 2. Results of the experimentation carried out considering the different possibilities of domain changes for training (source column) and evaluation (target column), as well as the combination of data domains for training. The intra-domain evaluation case for each scenario is underlined. The best results for each metric, labeling level, and target test set are highlighted in bold.

Source	Target	AP	$AP^{0.5}$	P	R	F_1
Pages						
eBDtheque	eBDtheque	**50.3**	**85.3**	**88.8**	**78.9**	**83.6**
	Manga	38.0	70.2	76.4	63.7	69.5
Manga	eBDtheque	17.7	42.0	61.1	39.2	47.8
	Manga	**58.2**	**86.4**	86.3	**79.1**	**82.5**
eBDtheque + Manga	eBDtheque	48.7	81.9	80.8	76.0	78.3
	Manga	56.9	86.1	**86.6**	78.1	82.1
Panels						
eBDtheque	eBDtheque	41.6	77.4	**81.9**	**77.2**	**79.5**
	Manga	36.4	68.2	70.8	68.7	69.7
Manga	eBDtheque	26.3	61.5	63.7	59.3	61.4
	Manga	**61.2**	**86.2**	**83.0**	**84.7**	**83.8**
eBDtheque + Manga	eBDtheque	**42.4**	**79.2**	78.4	75.9	77.1
	Manga	61.0	**86.2**	82.7	84.4	83.5

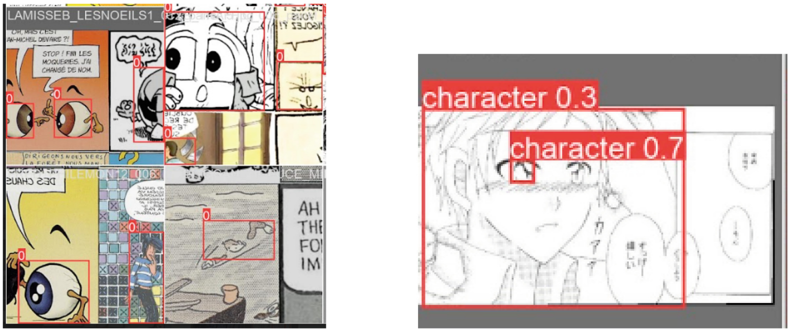
Finally, if we analyze the case of combining domains for training (eBDtheque + Manga109), quite similar results are also obtained at the page and panel levels, being slightly better for eBDtheque at the page level and for Manga109 at the panel level. Remarkably, the result obtained equals or even improves in some cases the individual training (see P for Manga109 at the page level or $AP^{0.5}$ at the panel level, and, in the case of eBDtheque, the AP and $AP^{0.5}$ metrics at the panel level). If we compare these results with the case of the inter-domain evaluation when training with a single domain, a remarkable improvement is observed (an average improvement of 22.7% of AP and 18.2% of F_1). Therefore, the proposed strategy for the combination of domains is much more appropriate when it is intended to use a model in several domains.

5.1 Discussion

As seen in the experimentation carried out, the training at the page level and the panel level have reported quite similar results, not the case for the resources and time used for training. Specifically, training the model with eBDtheque + Manga109 using panels required 51 h and 23 min, whereas training at the page level took only 13 h and 18 min. Therefore, it is more convenient to train at the page level, since having more individual images does not bring any improvement. It seems that having a larger variability of characters within a single sample outweighs the benefits of having more samples but with fewer characters.

Another relevant finding is the competitive results obtained when combining domains for training. This approach allows obtaining models that are usable in multiple domains, increasing their generalizability and range of application, which is very convenient for their practical use.

The task of detecting characters in comics represents a great challenge due to the wide variability of elements that we can find, which is only limited by the imagination. This variability not only makes labeling and model building difficult, but can lead to other unexpected problems. For example, we have detected that in some comics there are characters that are parts of the body, like the one that can be seen in Fig. 4a, in which the characters are eyes. This has caused the model to learn to detect the eyes of some characters as characters (see Fig. 4b).



(a) Ground truth

(b) Prediction

Fig. 4. Examples of panels from eBDtheque with characters that are eyes. The model predicts the eye of a character as an actual character.

Another problem related to the aforementioned variability is that we can find many secondary characters, which are often not labeled in these datasets, either because they are not important, they do not participate in the main action, or simply because the labeling is not complete. However, the generated models are capable of detecting these missing characters (see Fig. 5). This can help improve the quality of available datasets, as well as make it easier to label new ones. Therefore, it would be of great help in the tedious and laborious task of labeling.

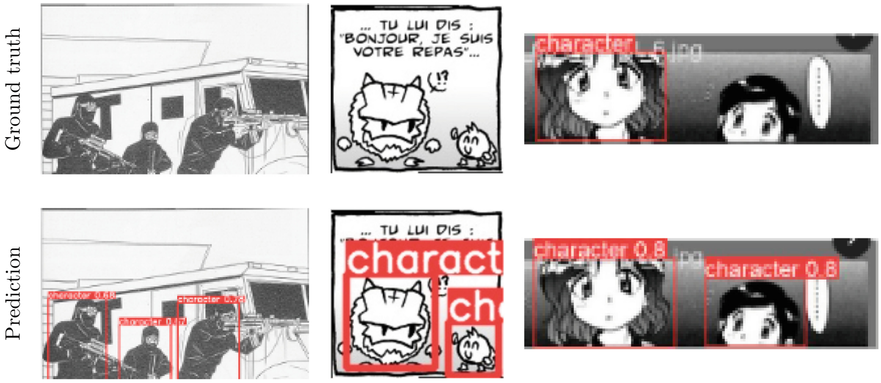


Fig. 5. Examples of incomplete labeling in the eBDtheque and Manga109 ground truth (first row) and how the generated model exhaustively predicts all characters even if they were not labeled (second row).

6 Conclusions

This work has presented a study on the problem of generalization in different domains for the automatic detection of characters in comics. The results of the experimentation show that when the model is trained with characters from different datasets, the model has a higher precision for the detection of characters in different domains. Moreover, training the model using panels instead of pages is not worth it at all, because the results obtained are similar, but the time it takes to train the model is significantly longer.

To keep on the development of automatic character recognition in comics, there are several ways to explore in future work. One is to try other architectures that take into account the context of the panels, for which the Transformers could be very effective. The datasets used for the training and evaluation of the models can also be expanded, in order to obtain more robust and representative results, as well as the combination of data coming from different domains, such as comics of different styles and genres, to further assess the generalization capacity of the trained models.

References

1. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Lechevallier, Y., Saporta, G. (eds.) Proceedings of COMPSTAT'2010, pp. 177–

186. Springer-Verlag, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-7908-2604-3_16

2. Dutta, A., Biswas, S.: CNN based extraction of panels/characters from Bengali comic book page images. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), vol. 1, pp. 38–43. IEEE (2019)
3. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
4. Fujimoto, A., Ogawa, T., Yamamoto, K., Matsui, Y., Yamasaki, T., Aizawa, K.: Manga109 dataset and creation of metadata. In: Proceedings of the 1st International Workshop on Comics Analysis, Processing and Understanding, pp. 1–5 (2016)
5. Girshick, R.: Fast r-cnn. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015). https://doi.org/10.1109/ICCV.2015.169
6. Guérin, C., et al.: eBDtheque: a representative database of comics. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1145–1149. IEEE (2013)
7. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
8. Iyyer, M., et al.: The amazing mysteries of the gutter: drawing inferences between panels in comic book narratives. In: Proceedings of the IEEE Conference on Computer Vision and Pattern recognition, pp. 7186–7195 (2017)
9. Jocher, G.: YOLOv5 by Ultralytics, May 2020. https://doi.org/10.5281/zenodo.3908559, https://github.com/ultralytics/yolov5
10. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
11. Lladós, J.: Two decades of GREC workshop series. Conclusions of GREC2017. In: Fornés, A., Lamiroy, B. (eds.) GREC 2017. LNCS, vol. 11009, pp. 163–168. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02284-6_14
12. Nguyen, N.V., Rigaud, C., Burie, J.C.: Comic characters detection using deep learning. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 3, pp. 41–46. IEEE (2017)
13. Nguyen, N.V., Rigaud, C., Burie, J.C.: Digital comics image indexing based on deep learning. J. Imaging **4**(7), 89 (2018)
14. Padilla, R., Passos, W.L., Dias, T.L., Netto, S.L., Da Silva, E.A.: A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics **10**(3), 279 (2021)
15. Rayar, F.: Accessible comics for visually impaired people: challenges and opportunities. In: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 3, pp. 9–14. IEEE (2017)
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: 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)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
18. Xiao, Y., et al.: A review of object detection based on deep learning. Multimed. Tools Appl. 23729–23791 (2020). https://doi.org/10.1007/s11042-020-08976-6
19. Zhao, Z.Q., Zheng, P., Xu, S.t., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3212–3232 (2019)