Automatic Detection of Formal Features of Comic Book Page

Siddhanth U Hegde, Abhishek Dhar and Nilesh Rathi[1]
[1]*Indiana University Bloomington*

I. INTRODUCTION

Scholarly attention towards comic books and graphic novels is on the rise, with fields such as literary studies, American studies, popular culture studies, and art history showing increasing interest in the medium. This study focuses on developing systems that can automatically detect various features of a comic book page. Some of the features of interests are the number of panels per page, the average size of panels, the size of the largest and smallest panels, the number of speech and thought balloons, per page and per panel. In the world of comic books, a panel is one of the squares on a page that depict a scene. The size of these panels can vary, often in relation to the importance of a scene. The speech and thought balloons exist inside each panel and help the reader associate the speech or thought with the source character.

The potential applications of this study are numerous. The identification of these features could aid in the classification of comic books and graphic novels, making it easier to organize and analyze large collections. Additionally, the analysis of these features could provide insight into the creative processes of comic book artists and writers, as well as the evolution of the medium over time. Ultimately, the study aims to contribute to the growing body of scholarship on comic books and graphic novels and to facilitate further research in this exciting and dynamic field.

Our eventual goal for this project is to build an accessible pipeline that can classify comic attributes and be used by researchers.

FIG. 1. Comic Page

II. BACKGROUND

Computer vision is a rapidly advancing field with a wide range of applications. One area of interest is the detection and recognition of comic book feature in images, such as speech bubbles, panels, and captions. While there have been several papers published on this topic, the majority of them are not easily reproducible for practical use. In many cases, the algorithms and techniques used in these papers require specialized knowledge and equipment, making it difficult for others to reproduce their results. Additionally, some of the data sets used in these studies may be difficult to obtain, limiting their usefulness to others who want to build upon the work.

Despite these challenges, the papers can still be useful to researchers and practitioners in the field. By reading these papers, one can gain insights into the latest techniques and methods being used to detect and recognize comic book features. There are several research papers published by faculty at la Rochelle Université that apply computer vision techniques to the field of computer vision. There is also a paper by faculty members at the University of Maryland that attempts to draw inferences between panels and comic narratives. These papers were incredibly educational for us, and guided us through the initial phase of the project

III. METHODS

A. Data set Description

The data set used in this study was provided by the University of Maryland and consisted of 500 images of comic book pages along with annotations for panels and text boxes. Each annotation was provided as a text file containing edge coordinates for each of the bounding boxes. Suppose we take a sample comic page as an example, where only the panel annotations are shown (as depicted in Figure 1). This comic page would have an accompanying annotation file (as displayed in Figure 2), which contains the edge coordinates outlining each bounding box of the panels and text boxes on the page.

In the case of the comic book data set, the annotations can be used to train object detection models to rec-

class	xmin	ymin	xmax	ymax
1	58	78	427	627
1	440	84	813	627
1	822	86	1257	633
1	70	636	593	1175
1	608	644	1251	1183
1	60	1194	703	1735
1	718	1190	1253	1741

FIG. 2. Annotations

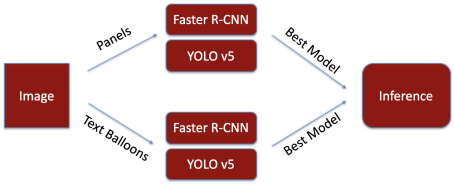

FIG. 3. Pipeline

ognize and localize panels and text boxes within comic book pages. This can help in automating the process of identify panels and text boxes, which can be a time-consuming task when performed manually. The trained models can then be used to process large amounts of comic book pages.

B. Pipeline Description

The pipeline proposed in this study 3 is designed to take a single comic book page as input and use trained models to predict the locations of panels and speech balloons within the page. There will be two models in the pipeline, one for text balloon detection and the other for panel detection. For each use case we will train a YOLO v5 and Faster R-CNN model and use the best performing ones in our final pipeline. Both of these models specialize in object detection.

Once the input image is processed by the two models, the resulting predictions for the panel and speech balloon locations are passed into an inference step. The inference step summarizes the results generated by the models and outputs informative features about the page, such as the number of panels, their respective sizes and locations, and the number and locations of speech balloons, the average panel/text balloon size as wells as maximum and minimum panel/text balloon size relative to the page size.

By utilizing object detection models and an inference step, this pipeline can automate the process of panel and text balloon identification within comic book pages, which is a crucial task for various applications such as digitization, translation, and content analysis. This pipeline can also be adapted and extended to support additional features, such as character detection, to further enhance its utility in analyzing and processing comic book pages.

C. Data Preprocessing

Before training the models, we performed data augmentation and image resizing on the training data set. Data augmentation techniques such as rotation, flipping, and translation were applied to increase the size of the training data set and reduce over-fitting.

D. Model Training and Evaluation

We trained the YOLO v5 and Faster R-CNN models for 15 epochs across the data set. The Intersection over Union (IoU) score was used to evaluate the performance of the models. The IoU score measures the intersecting area between the predicted and ground truth box as a percentage of union of the two areas. We used the IOU score to compare the performance of the YOLO v5 and Faster R-CNN models and to select the best performing model.

E. Implementation Details

The pipeline was implemented using Python and PyTorch deep learning framework. The YOLO v5 model was trained using the PyTorch implementation of the YOLO v5 algorithm. The Faster R-CNN model was trained using the Detectron2 library. The models were trained on an NVIDIA GPU with a batch size of 32.

IV. RESULTS

The following figures show the training results of the YOLO v5 model. Figure 4 shows that the MAP, precision, and recall of the data set improved with each epoch for the YOLO v5 model while Figure 5 indicates that both the box loss and object loss decreased over the course of the training set. These metrics suggest the the YOLO v5's ability to accurately detect and identify objects improved with each successive epoch during training.

On the other hand, figure 6 illustrates the loss of the validation set, which initially decreased with each epoch, but then plateaued. This suggests that the YOLO v5 model was able to learn the data set up to a certain point, but additional training may not have resulted in significant improvements.

Figure 7 shows the loss values of both the validation and training sets for Faster R-CNN. Initially, the validation loss is lower than the training loss, and as the model trains, both the losses decrease with each epoch.

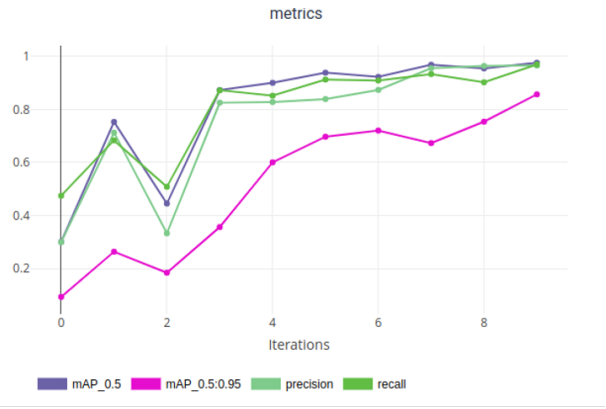

FIG. 4. Metrics per epoch (YOLOv5)

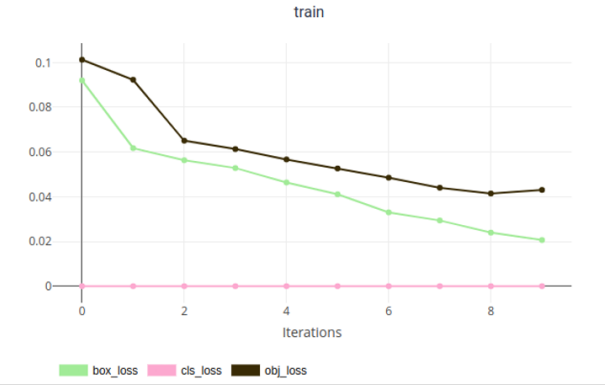

FIG. 5. Training Loss per epoch (YOLOv5)

However, at epoch 6, we can observe a change in the behavior of the losses. The training loss continues to decrease, but at a faster rate than the validation loss, which starts to level off. As a result, the training loss becomes lower than the validation loss and remains so in the subsequent epochs. This behavior suggests that the model has started to over fit the training data after epoch 6.

The test data set was used to detect panels using the two different models. An example of the results of panel detection is shown in Figures 8 and 9. Additionally, examples for the results of speech balloon detection in the test data set are shown in Figures 10 and 11.

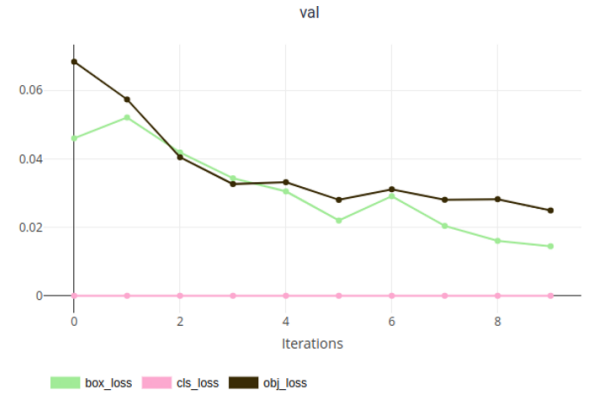

FIG. 6. Validation Loss per epoch (YOLOv5)

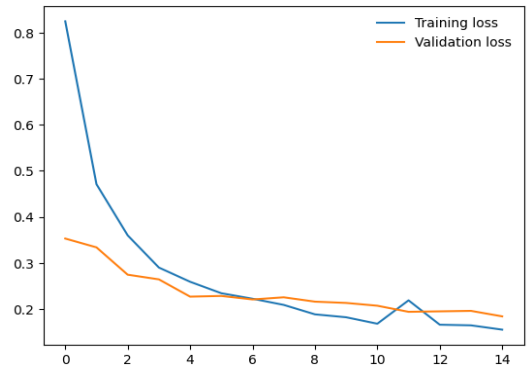

FIG. 7. Training/Validation Loss per epoch (Faster-RCNN)

FIG. 8. Panel Detection (Faster-RCNN)

Table 12 presents the Intersection over Union (IoU) accuracy values for panel and speech balloon detection for each of the two models: Faster R-CNN and YOLO v5. The results show that YOLO v5 outperforms Faster R-CNN in both panel and speech balloon detection, with higher IoU values of 97% and 95%, respectively. Using these results, we decided to proceed with the YOLO v5 models for our final pipeline.

The output of our detection pipeline is generated as a JSON that follows the format shown in figure 13. The output contains the coordinates of every detected panel and speech balloon along with some statistical inferences of the page. The panel coordinates are represented by the xmin, ymin, xmax, and ymax values, while the same is done for speech balloons.

Additionally, our JSON output includes statistics re-

FIG. 9. Panel Detection (YOLOv5)

FIG. 10. Speech Balloon Detection (Faster-RCNN)

FIG. 11. Speech Balloon Detection (YOLOv5)

lated to the detected panels and speech balloons, such as the number of panels and speech balloons, their average sizes, and the ratio of the biggest panel/speech balloon to the smallest panel/speech balloon. These statistics can help provide insights into the size distribution of the panels and speech balloons in the image.

V. CONCLUSIONS AND LIMITATIONS

Through this work, we were able to train two models to detect panels and speech bubbles and build a pipeline that allows a user to easily run a script to analyze comic book images. The inference script is available on GitHub and can be run using a single command. This tool will help make research in the field of comic books a lot easier by eliminating the need for manually annotating features.

There are many possible avenues for future development. One approach would be to add in OCR to provide

Model	Panel IoU (%)	Speech Balloon IoU (%)
Faster RCNN	92	88
YOLOv5	97	95

FIG. 12. Accuracy

```
{
    "image_name" : image_name,
    "panel" : {
        "xmin" : [coordinates of xmin of panels],
        "ymin" : [coordinates of ymin of panels],
        "xmax" : [coordinates of xmax of panels],
        "ymax" : [coordinates of ymax of panels],
        "stats" : {
            "count": number of panels,
            "avg_size": average area of all panels,
            "max_min_ratio": ratio of biggest panel to smallest panel
        }
    },
    "textbox" : {
        "xmin" : [coordinates of xmin of textboxes],
        "ymin" : [coordinates of ymin of textboxes],
        "xmax" : [coordinates of xmax of textboxes],
        "ymax" : [coordinates of ymax of textboxes],
        "stats" : {
            "count": number of textboxes,
            "avg_size": average area of all textboxes,
            "max_min_ratio": ratio of biggest textbox to smallest textbox
        }
    }

}
```

FIG. 13. Output Sample

statistical information about the words within speech bubbles, while another would be to add character recognition to provide information about the frequency and number of characters as well.

One limitation with this approach is that we cannot provide accurate results regarding the area of speech bubble. This is due to the fact that we are running object detection models for speech bubbles. As a result, the bounded area predicted by the model is often more than the actual area of the bubble as we're predicting boxes instead of the actual bubble shape. This could be improved upon by using segmentation techniques to precisely determine the bubbles. This would allow for more statistical information about the page and provide deeper insights.

Finally, we had the privilege of presenting our project to Professor John Walsh, who provided us with the initial idea. Professor Walsh was very positive about our work, and we are grateful for his feedback and guidance throughout the project. His insights and resources were valuable in developing our project, and we appreciate his support.

We are pleased that our work was well-received and deemed interesting and professional by Professor Walsh. Additionally, we are thrilled that our code can be reused for future research. We hope to continue working with Professor Walsh to refine our project and potentially publish a conference or journal article related to our work.

In conclusion, we are proud of our achievements and grateful for Professor Walsh's support and mentorship. We are excited about future collaborations and opportunities to enhance our skills and knowledge in this field.

"I am writing to discuss my experience working with

Abhishek Dhar, Nilesh Rathi, and Siddhanth U Hedge on their computer vision project. I had a very positive experience working with this team. They initially reached out to me over email requesting more information about the proposal I submitted to the class. In response to this email I provided the team with some relevant articles and sample comic book page images. Some time later we met over Zoom to discuss the project in more detail. At the end of the semester the team got back in touch with me to share their work. Over zoom we reviewed their code and results.

I was very impressed by both the students' work and by their professionalism in presenting it to me. They produced reusable code, which I can use in my research, and interesting results. I very much look forward to exploring their work more on my own. I hope to continue working with this team to refine their initial work and potentially work on a conference or journal article related to this project"—John A. Walsh, *Associate Professor, Luddy School of Informatics, Computing Engineering*

VI. REFERENCES

1. Nguyen N-V, Rigaud C, Burie J-C. Digital Comics Image Indexing Based on Deep Learning. Journal of Imaging. 2018; 4(7):89. https://doi.org/10.3390/jimaging4070089

2. Nguyen, NV., Rigaud, C. Burie, JC. Comic MTL: optimized multi-task learning for comic book image analysis. IJDAR 22, 265–284 (2019). https://doi.org/10.1007/s10032-019-00330-3

3. Rigaud, Christophe Burie, Jean-Christophe. (2018). Computer Vision Applied to Comic Book Images: Digital, Multimodal, and Cognitive Methods. 10.4324/9781315185354-6.

4. Nina S. T. Hirata, Igor S. Montagner, and Roberto Hirata. 2016. Comics image processing: learning to segment text. In Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding (MANPU '16). Association for Computing Machinery, New York, NY, USA, Article 11, 1–6. https://doi.org/10.1145/3011549.3011560

5. Nina S. T. Hirata, Igor S. Montagner, and Roberto Hirata. 2016. Comics image processing: learning to segment text. In Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding (MANPU '16). Association for Computing Machinery, New York, NY, USA, Article 11, 1–6. https://doi.org/10.1145/3011549.3011560

6. Rigaud, Christophe Nguyen, Nhu-Van Burie, Jean-Christophe. (2021). Text block segmentation in comic speech bubbles.

9. Clément Guérin, Jean-Christophe Burie, and Jean-Marc Ogier. 2016. Detection of comic books twin pages with a non-overlapping stitching method. In Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding (MANPU '16). Association for Computing Machinery, New York, NY, USA, Article 1, 1–6. https://doi.org/10.1145/3011549.3011550

10. Rigaud, Christophe Nguyen, Nhu-Van Burie, Jean-Christophe. (2021). Text block segmentation in comic speech bubbles.

11. Iyyer, M., Manjunatha, V., Guha, A., Vyas, Y., Boyd-Graber, J., Daume, H., Davis, L. S. (2017). 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).

A. Code

The code for the project and the dataset used can be found here: https://github.iu.edu/cs-b657-sp2023/bubble-vision

Data set: https://obj.umiacs.umd.edu/comics/index.html

B. Links

1. https://debuggercafe.com/custom-object-detection-using-pytorch-faster-rcnn/

2. https://github.com/ultralytics/yolov5