TOM: Text to Object Mapping Architecture for Library Bookshelves

Praneel Seth
Department of Computer Science
The University of Texas at Austin
Austin, United States
praneelseth@utexas.edu

Mukund Raman
Department of Computer Science
The University of Texas at Austin
Austin, United States
mkraman@utexas.edu

Sarthak Dayal
Department of Computer Science
The University of Texas at Austin
Austin, United States
sarthak@utexas.edu

***Abstract*—In this paper, we present a technique to detect books by title in a bookshelf image. The proposed technique includes four stages: full image segmentation, book segment identification, text detection, and text to object mapping. Firstly, segments (pixel level image masks) are generated across the entire image. Next, those segments are classified based on whether they are books or not. In a parallel process, text (a title) is detected across the whole image. Finally, the location of each segment relative to its title is used to map each segment to a title. Experimental results show that the model is better at identifying which books are not present than which books are present.**

***Index Terms*—Segment Anything Model, OpenAI CLIP, Google Cloud Vision**

I. Introduction

Libraries have been the basis for information store and exchange across the world, throughout history. Libraries were at first smaller storage spaces, with small quantities of books that enabled simple cataloging and management techniques. As the number of books in libraries scaled, management techniques evolved, and in an average American library today, there is likely to be some sort of digital cataloging or digital management in place. In fact, the national overdue rate, for example, was 0.7% pre-automation and 0.4% post-automation in the 1990s [1]. However, an important yet tedious task in library management that has yet to be automated is finding and cataloging books.

Robot that are capable of performing object manipulation tasks have become a natural solution to assist with daily time-consuming management activities, which can include finding, arranging, and storing books. Long-term research in this field would allow robots to create a connection between the digital and physical space in the library by understanding locations of books and the digital catalog that contains those books. We explore the task of getting the location of a particular book that the user requests within a camera frame of a bookshelf.

II. Background

Previous work in this area makes many assumptions about the placement and structure of the shelves in the library. The authors of [2] utilize Canny Edge Detection to segment each row of the bookshelf and each book. This method is advantageous because it can separate each row of the bookshelf, however, if the books are too slanted, the canny edge detection fails. In [3], a Hough line transform breaks the spine into three sections, the left, right and text-containing center. However, as explained in [4], this Hough transform approach is prone to be less accurate. Instead, Fatema, et al. fit the input image to a bookshelf model. This method tended to be more accurate, however all of these methods fail to produce high precision pixel-level representations of book edges that are helpful for robotics tasks. [4] also adds an additional requirement that adds management overhead, which requires more information than we argue is needed for detecting books in a frame. Other literature in the area including [5], [6], [7], and [8] show promising results but with similar qualms.

We propose a technique for book identification that works agnostic of the orientations, shapes, and locations of books in an image and can extract a pixel-level mask for the book based purely on a book title that utilizes newer innovations in deep learning.

The approach discussed in this paper can detect book segments and text in any orientation, both in the camera frame and in relativity to each other, which is advantageous over canny edge detection approaches. The model used to segment the image is highly generalized and so it detects objects precisely, which is an improvement relative to Hough line transforms. Although there are some complex models involved in this paper's approach, they still have greater precision than a rigid bookshelf model-fitting algorithm. The approach is modular, which simplifies the process of replacing a component of the computer vision system for future studies or iterations.

III. Proposed Technique

We propose the Text to Object Mapping architecture (TOM), a four-stage Computer Vision pipeline that can extract the correct segment for a particular book with some specified title. We assume that the provided title is perfect, obtained from common library databases. Using this title, we go through the following four steps. First, we segment the images using Meta AI's Segment Anything Model (SAM). Second, we feed the segments through a CLIP model to classify segments between book and not book. Third, we detect all the text in the image

and find bounding boxes for that text. Lastly, we match the coordinates of the detected title and the book to bring out the correct segment.

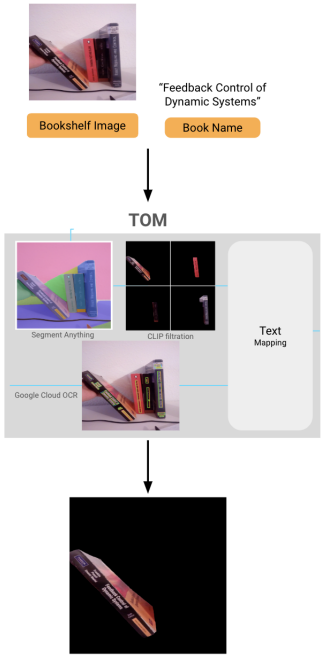

Fig. 1. The TOM Architecture.

A. Segmenting the Image with SAM

Meta AI's Segment Anything Model (SAM) is an image segmentation model that can identify the precise location of every object in an image, but without labels. Segment Anything provides a foundational model capable of zero-shot segmentation for many tasks [9], including ours.

SAM is the ideal candidate for generating book segments from a bookshelf image because it is able to precisely detect book masks on a pixel level and is accurate regardless of the book's orientation, shape, and precise location within the frame. SAM provides the most information that can be obtained and represented in a pure 2D image for any given book.

B. Identifying Books with CLIP

Once we have the particular segments, we classify these segments into book and not book to avoid areas of the image that may have text in them and are not part of any book we may want to detect. CLIP provides an easy way to classify segments based on a set of labels, as it uses a similar embedding space for text and images and compares their cosine similarity [10].

We find that regular "book" and "not book" labels do not work well for a foundational model like CLIP and therefore use a custom set of labels that includes "wall" and "red book" among others. After this label alteration, however, we find that CLIP works with a high level of accuracy and propose that it be fine-tuned for book-related tasks in future work.

C. Text Detection with Google Cloud Vision OCR

For detecting the locations and content of text within our image, we use Google's Cloud Vision API, which uses vision transformers and classical OCR to provide us with all the text inside an image and the bounding boxes for all those pieces of text. Using the location for these bounding boxes, we can pinpoint the correct segment to extract.

Basically, this process runs in parallel to the first two models, since it takes in the original input image and returns a list of each piece of text in the image with its pixel coordinates. All that is left at this stage is to map the text to the correct detected block and by extension the correct pixel-level segment.

D. Extracting the Correct Segment

Finally, we resolve the text to its correct segment by merging our parts together. Google Cloud's Vision API provides text in "blocks", which contains "paragraphs" of text. We match our title text to a particular block by checking the combined text of the paragraphs and matching it to our title. After this, we transform the (x,y) location of the detected text to extract the correct segment in our set of SAM segments. We ensure that this detected text passes a similarity threshold with the title that the user has inputted. If we cross this threshold, we report that an image was found. Otherwise, we report a failed case.

Essentially, we map the location of the text on the image to the segment it falls inside of to create key-value pairs that outputs a pixel level image mask using the title of a book.

IV. Evaluation

We evaluate our method using a series of experiments on twelve bookshelf images that test the accuracy as well as the robustness of our methods. Our evaluation method tests true positives, false positives, true negatives and false negatives.

We define the true positive case to be the case where we find the correct segment for a book that is truly present in the image. The false positive case includes cases where a book segment is matched where no matches should exist for the provided title (i.e. the book is not inside our camera frame). True negative cases include when TOM determines that no books match the given title and there is no book in the image that matches the title. False negatives include both cases in which TOM detects that no segments match in the given image when a segment is present or when the wrong segment is detected by the TOM architecture.

Our evaluation involves running ten images through a series of five titles each. We make three of these five titles include books that are in the image (true positives) and two of the five titles be titles that are not included in the image (true negatives). The two negative tests include one where the text is very similar to the books in the image and one which is significantly different. These tests include a wide variety of images that test many different aspects of the model, including books in different orientations, books that are close together, books of similar color, and more. This allows us to see how

robust our model is to these different kinds of changes. For each of these tests, we run the image-text pairs through our TOM architecture and evaluate which of the four categories the particular combination fell into. We report these numbers below.

V. Results

After evaluation, we find that our architecture is better at finding failures than successes and determining when the incorrect title is provided to us. However, we find a high success rate for true positives as well as true negatives and report the specific numbers of the experiments below. Our results show improvements compared to other known methods and provide a clear setup for future work to improve on. Below we report our results and an analysis for why we believe the results have the particular numbers they do.

A. Overall Accuracy

TABLE I
Overall Results

	True Positives	True Negatives	False Positives	False Negatives
Image 1	3	1	1	0
Image 2	2	2	0	1
Image 3	2	1	1	1
Image 4	2	1	1	1
Image 5	3	2	0	0
Image 6	2	2	0	1
Image 7	2	2	0	1
Image 8	2	2	0	1
Image 9	2	2	0	1
Image 10	2	2	0	1

The number of true positives, true negatives, false positives, and false negatives per image.

We find that our algorithm reports a 73.33% true positive rate and a 85% true negative rate. This accuracy rate shows that our architecture performs better than comparable methods on many scenarios though it leaves room for improvement. We report some of the reasons for failures in the next section.

B. Analysis

There were three clear causes for false negatives. The first was from SAM completely ignoring a book segment. In one example, a specific white book was between two other white books with similar lighting, so the error is almost understandable at a human vision level. As SAM ignores text lines and segments at the size of the text, it is understandable why it had difficulty differentiating adjacent white blocks as separate and not as a background element. The second cause for a false negative was an error in Google Cloud Vision text detection. For example, for one book, the text was likely detected incorrectly because of font color and contrast, as the text was white on a blue background in an irregular font. The third cause was due to similar titles between books.

There is only one cause for false positives: text similarity. The fake title was too similar to another title present in the image, and this happened because the similarity threshold was low enough to allow this difference.

VI. Discussion

The TOM architecture provides something unique compared to previous vision-based book detection systems. It provides pixel-level masks that can then directly be combined with a depth detector to help the robot know the exact location and orientation of a book in 3D space to pick it up or use it for its next task. Hough line transforms, for example, only provide a rough bounding box for each book, as compared to a precise pixel by pixel location for each book.

Our 73% true positive success rate proves to be better than comparable methods that use Hough Tranforms and Canny Edge detection (which largely failed to even replicate on our own tests). This provides a framework for future work to improve upon and creates a new way for robots to assist in library management tasks. Our work provides a baseline for other deep learning inspired approaches in this area that can improve the lives and work of many people. In addition, our work moves toward a future where robots can learn to connect language to physical objects, which would enable them to do much more beyond their current capacity.

VII. Conclusion

We propose the TOM vision architecture, which uses Meta's Segment Anything Model, OpenAI CLIP, and Google's Cloud Vision API to extract a segment that matches user-inputted text. Our results find that our model detects the correct segment with high accuracy and despite its false negative rate lays a strong foundation for future research that can utilize these models.

This vision system can enable robots to find the physical location of a book on a bookshelf, then use it as needed, whether that means picking it up or handing it off to someone. Ultimately, this automates the "busy work" that librarians have to do, saving time and energy while reducing the quantity of books that are lost or misplaced due to human error. In fact, the use of computer vision to manage books can be applied to other physical file systems, for example with filing cabinets in legal fields. Ultimately, a network of these autonomous agents could collaborate to run an entire physical file space, mapping physical files to digital spaces and managing files based on digital requests. It's the file system that files, logs, and audits itself.

References

[1] Hansel, Patsy J. "Managing overdues : a how-to-do-it manual for librarians". New York: Neal-Schuman, c1998.

[2] M. I. Jubair and P. Banik, "A technique to detect books from library bookshelf image," in 2013 IEEE 9th International Conference on Computational Cybernetics (ICCC), 2013, pp. 359-363. doi: 10.1109/ICCCyb.2013.6617619.

[3] K. Fatema, M. R. Ahmed, and M. S. Arefin, "Developing a System for Automatic Detection of Books," in Second International Conference on Image Processing and Capsule Networks. ICIPCN 2021, J. Chen, J. M. R. S. Tavares, A. M. Iliyasu, and K. L. Du, Eds. Lecture Notes in Networks and Systems, vol. 300. Cham: Springer, 2022.

[4] E. Taira, S. Uchida, and H. Sakoe (2004). "A MODEL-BASED BOOK BOUNDARY DETECTION TECHNIQUE FOR BOOKSHELF IMAGE ANALYSIS."

[5] Y. Aoki and M. Ishikawa, "Book Recognition from Color Images of Book Shelves," in MVA 1998 IAPR Workshop on Machine Vision Applications, Makuhari, Chiba, Japan, 1998, pp. 106-107.
[6] E. Takeda, S. Uchida, and H. Sakoe, "Block Boundary Detection and Title Extraction for Automatic Bookshelf Inspection," in 10th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV 2005), Fukuoka, Japan, 2004.
[7] P. E. Taira, S. Uchida, and H. Sakoe, "Book Boundary Detection from Bookshelf Image Based on Model Fitting," in International Symposium on Information Science and Electrical Engineering, 2003.
[8] M. Prats, P. J. Sanz, and A. P. del Pobil, "Model-based Tracking and Hybrid Force/Vision Control for the UJI Librarian Robot," in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada, 2005, pp. 1090–1095.
[9] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," arXiv, vol. 2304.02643, 2023.
[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," CoRR, vol. abs/2103.00020, 2021. [Online]. Available: https://arxiv.org/abs/2103.00020.