Digital Image Processing Final Project
Manga Panel Extraction

Eren ÇAKAR
1901042656

January 11, 2024

Contents

1 Introduction

Digital image processing plays a crucial role in various applications, and one such area is the extraction of panels from manga page scans. This project focuses on applying image processing techniques to automatically identify and separate individual panels within a manga page.

1.1 Background

Manga, a form of Japanese comic or graphic novel, typically consists of multiple panels arranged in a specific sequence to convey a storyline. Extracting these panels automatically can streamline the process of digitizing manga, making it easier to translate, analyze, and repurpose content.

1.2 Objective

The primary objective of this project is to develop a robust algorithm for manga panel extraction. This involves employing image processing methods to detect panel boundaries accurately, considering variations in panel sizes, shapes, and orientations.

1.3 Significance of the Study

The successful implementation of manga panel extraction can significantly benefit manga readers, publishers, and researchers. It can enhance the digital reading experience, facilitate content translation, and aid in automated analysis of manga content.

1.4 Scope of the Project

While the project aims to achieve accurate manga panel extraction, it is essential to acknowledge potential challenges and limitations. Factors such as complex page layouts, overlapping panels, and artistic variations may pose challenges in achieving 100% accuracy. Below is a list of challenges that should be considered:

1. Unaligned Panel Borders
2. Different Sized Panels
3. Panels On Edge
4. Tilted Panel Separators
5. Non-Straight Panel Separators
6. Joint Panels
7. Overlapping Panels
8. Colored Background

9. Image Background
10. Overflown Speech Bubble
11. Overflown Text
12. Overflown Panel Item
13. Borderless Panels

These challenges are ordered in terms of increasing complexity to handle.

2 Literature Review

The paper titled "A Robust Panel Extraction Method for Manga"[1], addresses the challenge of automatically extracting frames/panels from digital manga pages. The authors recognize the complexity of manga panel layouts, which often involve intricate designs and various visual symbols.

The proposed method in the paper takes a systematic approach to panel extraction. It starts by identifying the panel block through closing open panels and establishing a page background mask. Subsequently, a recursive binary splitting technique is employed to partition the panel block into sub-blocks, with an adaptive determination of optimal splitting lines at each recursive level. Finally, the method recovers accurate panel shapes from the computed sub-blocks.

The significance of this approach lies in its ability to robustly segment panels on manga pages with various styles. The paper contributes to the field by addressing the challenges specific to manga, providing a method that proves effective in handling the diverse and complex layouts encountered in Japanese comics.

However due to lack of including their implementation of their algorithm this project is done.

3 Panel Extraction Methods

In this section, we present the ideas and concepts behind the methodology employed for manga panel extraction. The algorithm consists of three distinct methods aimed at separating panels.

The detailed explanations and analysis are done in the nex section.

3.1 Base Method

The base method focuses on panel separation by identifying background segments and subtracting them from the original image. Then the result is split into segments which are extracted into individual files. This simple method is actually sufficient for more than half of the mangas. And if the method fails the output never lacks any information of the input image, which is considered user friendly since cutting an image is easier than combining two images for humans.

3.2 Second Method

Building upon the base method, the second method is designed to handle panels that are connected by speech bubbles or objects. After the background mask is obtained with the base method the segments that were cut off are continued and thus connected background mask is obtained. Then the rest of the process follows the same as the base method. However this can split panels that were joint, it introduces false-positive cases of cutting panels where it's not needed, resulting in elimination of said panels.

3.3 Third Method

The third method is tailored for high-variance, low-ranged intensity grayscale manga page images. A page with textured background that has a watercolor style is such an example. After obtaining the line-work both regular and adaptive threshold is applied to it and first is subtracted from the latter. Which results in basically deletion of only the panel edges, so the panels can be extracted via the segments obtained.

These three methods collectively provide a comprehensive approach to manga panel extraction, addressing various challenges encountered in different types of manga images.

4 Process and Result Analysis

In this section, we provide a detailed analysis of the manga panel extraction process, presenting visualizations for different scenarios handled by each method.

4.1 Base Method

We begin by examining the process of the base method. In this subsection the process and results of the base method will be analyzed through an example page.

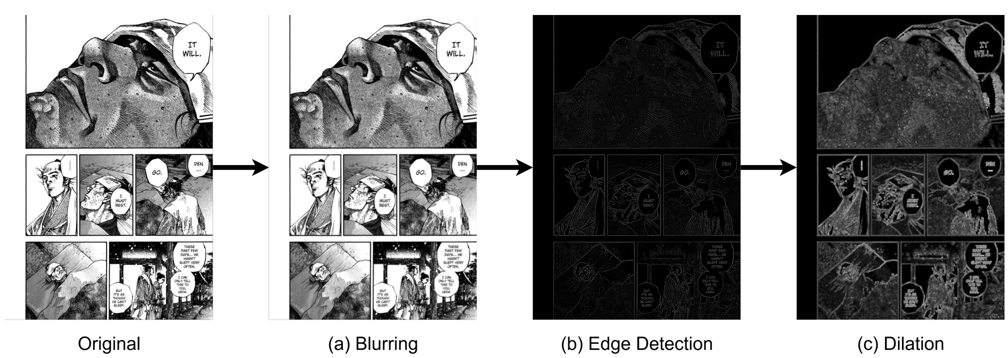
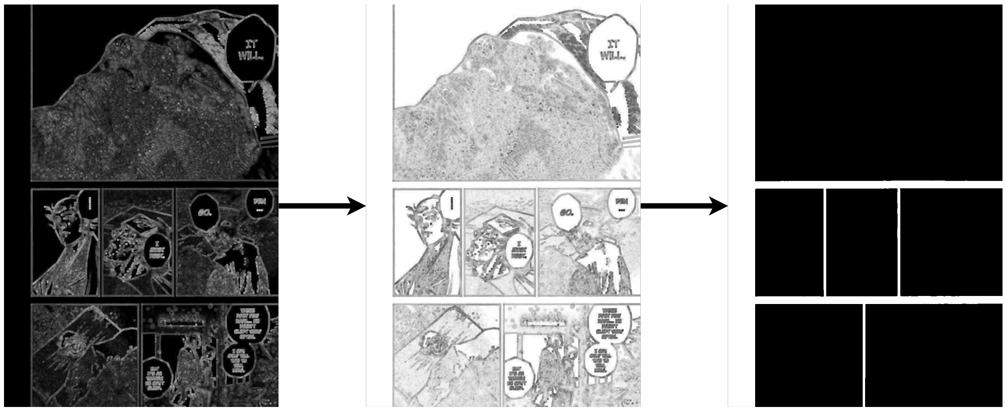


Figure 1: Line detection. (a) after Gaussian blur is applied. (b) after Laplacian is applied. (c) after dilation is applied.

Since manga page backgrounds vary in both color and sometimes imagery Laplacian is used for both turning the image into a grayscale image where only the line-work is white and the rest is black. Some manga styles include textured backgrounds, to account for that the input is blurred before edge detection is applied. This process is showcased in Figure 1.



(a) Edges (b) Inversion (c) Segment Filtering

Figure 2: Background mask generation. (a) edges obtained by previous steps. (b) after applying intensity inversion. (c) after filtering out every invalid white segment.

Then to extract the panels out first the background mask is formed via eliminating the invalid segments that were obtained by inverting the edges that were generated by previous steps. A segment is valid if their width or height close to input image's, respectively. Or their shape resembles a rectangle, which is checked by approximating their contour and counting the number of corners the resulting contour has. If the segment can't satisfy either of these conditions or its area is less than some percentage of the input image area then it's invalid and filtered out. An example of this segment elimination can be seen in Figure 2.

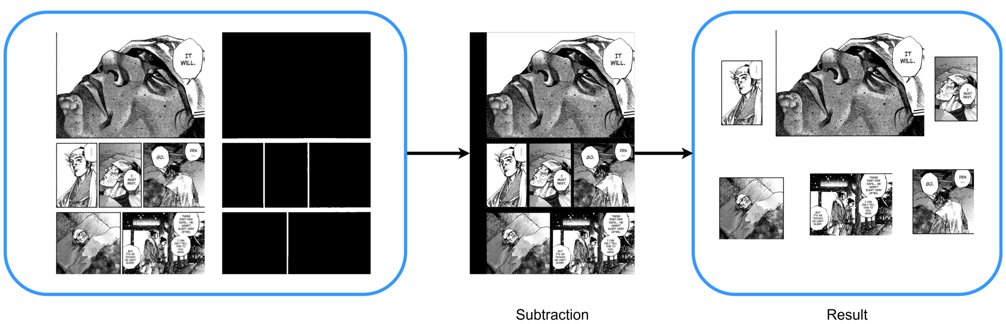


Figure 3: Panel extraction

As the final step the obtained background mask is subtracted from the original input then contour search is done on the resulting image. Filter conditions similar to that mentioned above are also applied at the result obtaining step. But the extracted images are cut-outs from the original input with the bounding rectangle of each valid segment, and not only the area that is under the valid segment mask. This decision was made for reducing false-negative cases for eliminating panels areas. An example of extracted panels can be seen in Figure 3

4.2 Split Method

Next, we evaluate the effectiveness of the split method by showcasing an example page where the method successfully handled panels connected by speech bubbles or objects.

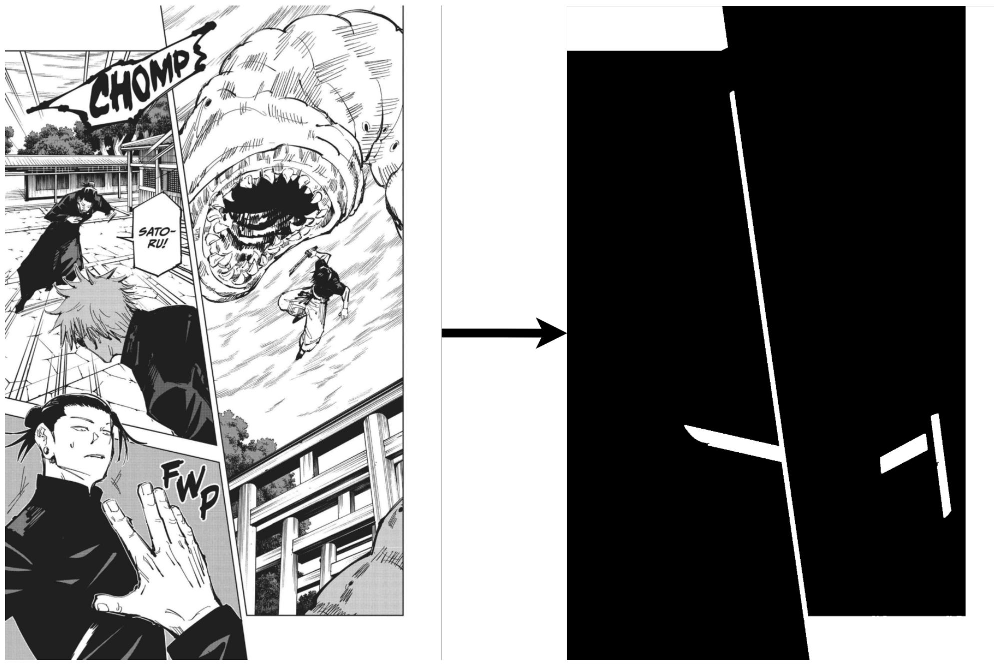


Figure 4: Background mask obtained with base method steps

First the background mask is obtained via the same steps described in the base method. Input for our example for this subsection can be seen in Figure 4. Then the obtained mask is thinned using Zhang-Suen[2] thinning algorithm. This way width of every segment is reduced to one pixel. The reason for doing so is to find the end-points of the segments easier. See Figure 5 for the thinning result of the mask above.

For finding the end-points of segments eight distinct direction kernels are convoluted over the thinned mask with the purpose of template matching. A pixel is put whenever there is a perfect match. Then these points are dilated with the same kernel that was used for pattern matching, this way the segment is only dilated in the direction it was cut. The size (or number of iterations) of the dilation was set linearly proportional to the width or the height of the input image depending on the direction of the identified segment. The example of this process is shown in Figure 5.

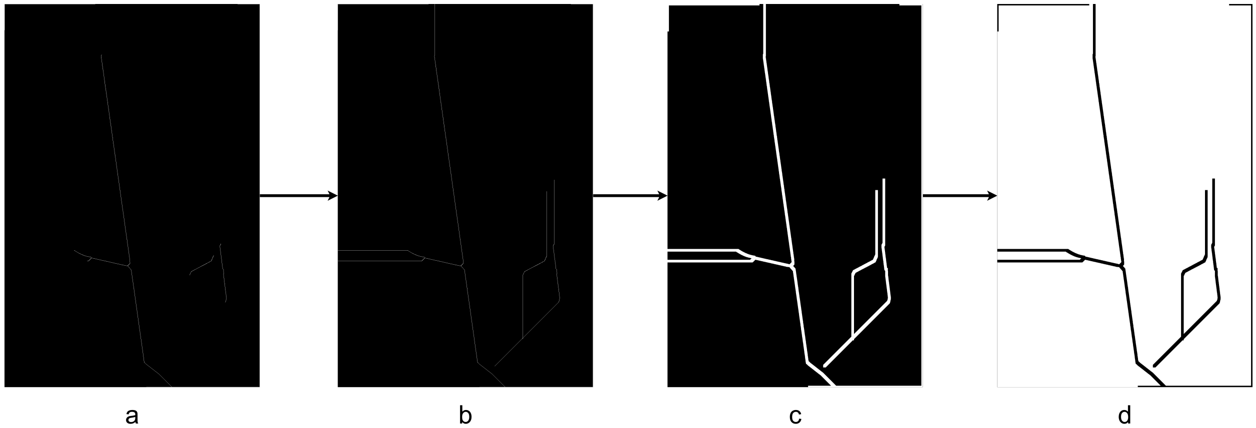


Figure 5: (a) skeleton obtained by thinning the background mask obtained in Figure 4. (b) after continuing segments with end-points. (c) after dilating. (d) after inverting.

After the end-points of background segments are connected a dilation of size proportional to initial background mask is applied to obtain the right size for the mask. And finally the obtained mask is subtracted from the original input image to extract the panels. The panels extracted for the example in Figure 4 are shown in Figure 6.

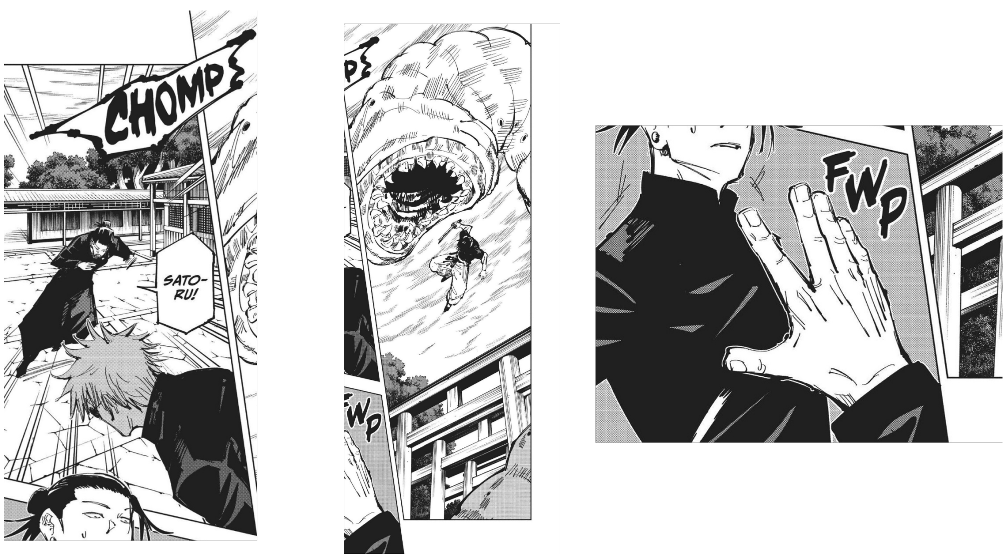


Figure 6: Extracted panel images

It is seen not all objects that belong to a panel are included in the related output, which is essentially unavoidable by definition.

Also due to thinning process the splitting method is computationally intensive, making it a slower option. Due to these reasons this method is not set to be the default method as the solution of the problem, but is left as an option for the user. Even then if the input image is classified as a strip comic style page then this method is prevented from executing regardless of user setting it on. The mentioned classification is a simple initial background mask area checker, relative to the total area of the page.

4.3 Fallback Method

For the third method, we present a page where the method somewhat successfully handles high-variance, low-ranged intensity grayscale manga pages.

Figure 7: Example page

The fallback method was developed closely related to the example that whose process will be explored in this section. This is due to the exceptional style of this manga, which was unique enough for us to not find any much more data to test this method on. The mentioned page can be seen in Figure 7.

The base method fails miserably against this example. The background mask generated by base method can be seen in Figure 8.

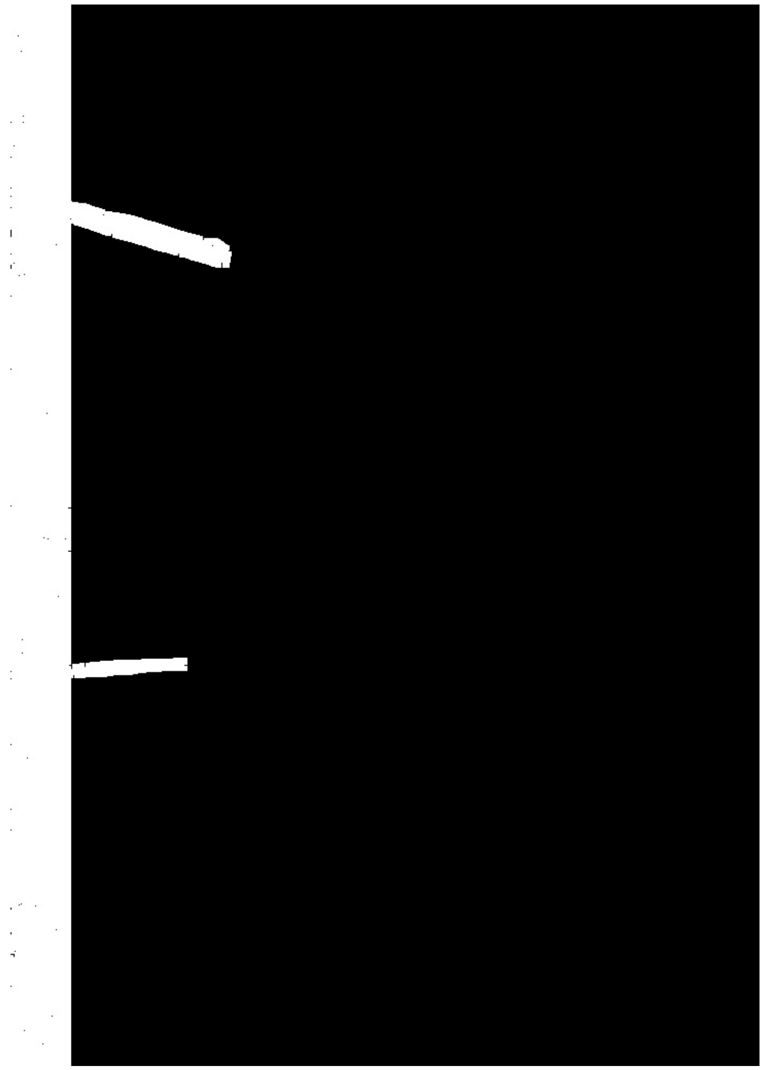


Figure 8: Background mask of example in 7 generated by the base method

The failure of generation of a proper background mask is due to the reasons; thin and separated background segments, tight intensity range, textured background, and joint pages.

With trial and error it was noticed due the highly varying background and low range of intensity, the adaptive threshold results of the Laplacian result of image is practically regular threshold result with dense noise all over the page. Deeming such results would be obtained always for such inputs (high-variance, low-range) this was made use of by subtracting the regular threshold result from the adaptive threshold result to obtain basically the panel segments. This process is shown in Figure 9.

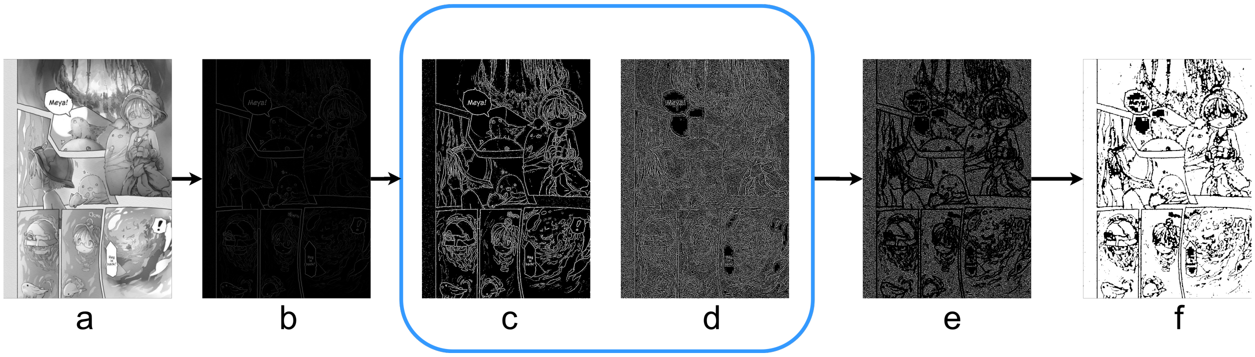


Figure 9: (a) original input. (b) edges obtained by Laplacian. (c) threshold result of (b). (d) adaptive threshold result of (b). (e) result of subtraction of (c) from (d). (f) after dilation

Then dilation is applied for a better mask, since the noise is dense, applying dilation connected the noise inside the panels but not the individual panels to each other.

Finally the obtained mask is segmented and used for extracting the panels. The results can be seen in 10.

Figure 10: Extracted panel images

It is clear the results are not pretty and even wrong for some. Which is why this is the **fallback** method which is only used when the user turns on the option for it and the base method fails, where failing means producing only one output.

4.4 Failure Cases

Finally, we analyze three pages where the manga panel extraction process failed to produce satisfactory results. Refer to Figures 11, 12, and 13 for the mentioned examples.

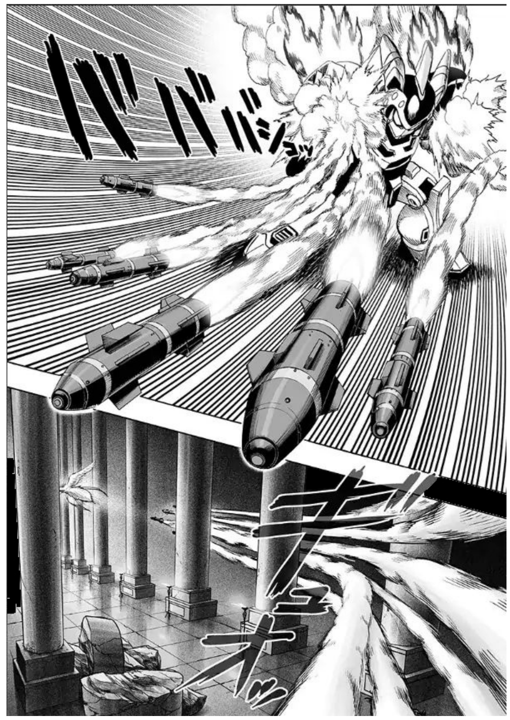


Figure 11: Failed input

Figure 12: Failed input

Figure 13: Failed input

Though it should be mentioned for the example in Figure 13 the split method manages to export lower panels the rest are not only identified but also not outputted without splitting, which means loss of input data which was deemed failure.

These visualizations provide a comprehensive overview of the manga panel extraction process, showcasing successful outcomes and identifying areas for improvement in handling challenging scenarios.

5 Conclusion

In conclusion, this project endeavors to contribute to the field of digital image processing by addressing both the classical challenges of image processing and the unique challenges posed by manga panel extraction.

It is clear that the artistic styles are not only hard to handle in technical aspects but also in semantic aspects. It is inherently unclear to when and where to split joint panels for some cases, where the user should interfere and specify their preference. Even then it's totally possible that the author purposefully created entangled visuals with the purpose of storytelling.

Apart from such cases the technically complex aspects are seemed to be more fit for artificial intelligence systems to handle, such as convolutional neural networks.

6 Implementation

The algorithms were implemented using Python 3 and OpenCV. The source code can be found at this GitHub repository.

References

[1] Xufang Pang, Ying Cao, Rynson W. H. Lau, and Antoni B. Chan. *A Robust Panel Extraction Method for Manga*. Video, Image, and Sound Analysis Lab (VISAL) at the City University of Hong Kong, 2014.

[2] Zhang, T. Y., & Suen, C. Y. (1984). *A fast parallel algorithm for thinning digital patterns.* Communications of the ACM, 27(3), 236-239.

[3] Python Software Foundation. (Year Accessed). Python 3. Retrieved from `https://www.python.org/`

[4] OpenCV Contributors. (Year Accessed). OpenCV (for Python). Retrieved from `https://opencv.org/`

[5] Akihito Tsukushi, *Made in Abyss*, Manga series.

[6] Gege Akutami, *Jujutsu Kaisen*, Manga series.

[7] Takehiko Inoue, *Vagabond*, Manga series.

[8] Hiroaki Yoshikawa, *Bocchi the Rock*, Manga series.

[9] ONE, Yusuke Murata, *One Punch Man*, Manga series.

[10] Kentaro Miura, *Berserk*, Manga series.