Manga Speech Bubble Detection

Chun Meng (Amy) Yu

Abstract

This project compares three different approaches for manga speech bubble detection: algorithmic, generic model, specialized model trained on manga pages. We implemented an algorithmic approach using contour lines and found that the performance is similar to a generic model and almost as good as a specialized model. This means if there are patterns in the data, it's possible to implement an algorithm that approaches state-of-the-art performance.

1. Introduction

Manga are a type of comic book originally written in Japanese. Many of them have been manually translated to other languages, but the translation process is painstaking and tedious, where the translator would need to manually extract, translate, and replace the text in each speech bubble. Since there are still many high-quality works of manga that have not been translated, a tool that could automatically translate them would allow them to be available in different languages. Unlike text-based works such as novels, the manga translation process involves extracting text from an image before performing machine translation.

Speech bubble detection is an important step in this process. Without speech bubble detection, if we use OCR to extract text from the entire page directly, there will be many incorrect results where the OCR model would mistakenly identify the shapes and lines from the scenes in the manga as text (Figure 1). By applying speech bubble detection, we can limit the OCR model to extract text only from the regions inside the speech bubbles (Figure 2).

2. Related Works

There are existing implementations of manga translation tools that can detect speech bubble areas or extract the text from pages in a manga. There are also many existing algorithms and models that we can use to implement new speech bubble detection approaches.

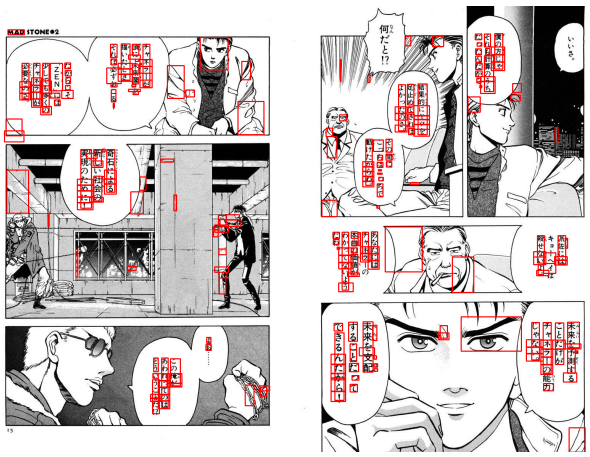

Figure 1. Extracting text without speech bubble detection

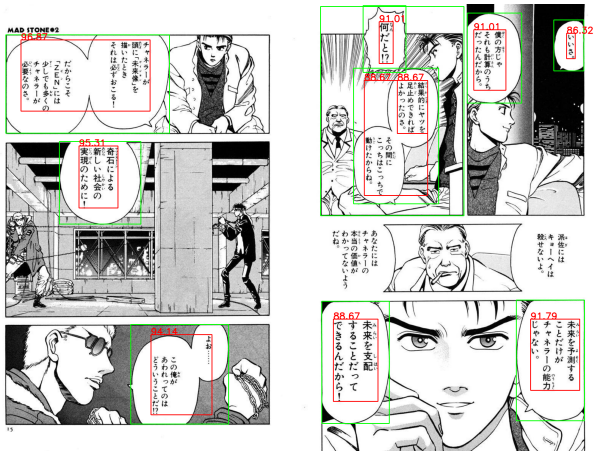

Figure 2. Extracting text with speech bubble detection

2.1. Canny Edge Detection

Canny Edge Detection [2] is a computer vision algorithm that could find the edges in an image. This could be used for speech bubble detection if there is a way to determine which edges belong to speech bubbles (Figure 3).

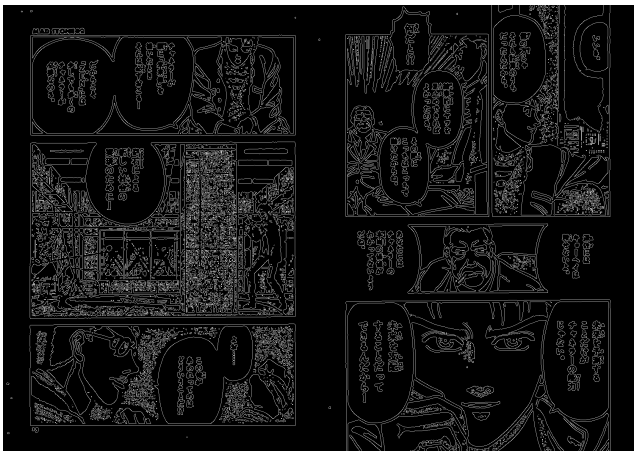

Figure 3. Canny Edge Detection

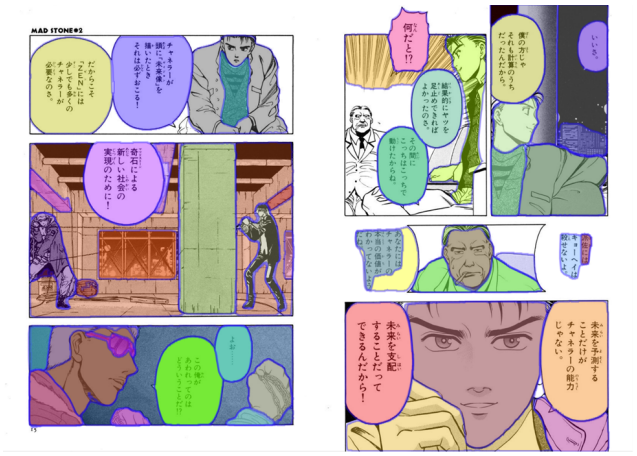

Figure 4. Segment Anything Model

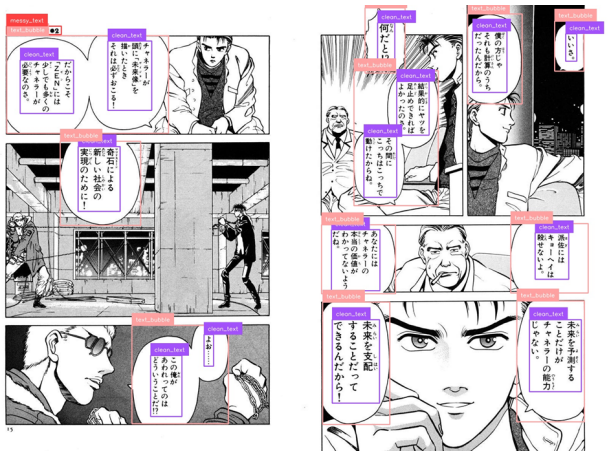

Figure 5. Manga Text Detection Computer Vision Project

2.2. Segment Anything Model

The Segment Anything Model [5] recently became available in 2023 that could detect segments in an image such as speech bubble regions (Figure 4). A model called Language Segment-Anything [7] builds on top of the Segment Anything Model with the added functionality of segmenting an image using text prompts. It uses the Grounding Dino model[9] that could detect parts of an image that match a text prompt. This makes it possible to extract the segments that are speech bubbles.

2.3. Existing work in manga speech bubble detection

There are many projects that have already been done in the topic of manga speech bubble detection, such as the Manga Text Detection Computer Vision Project [6]. This is a *yolov8n* model trained on 512 manga pages that could detect the bounding boxes of text and speech bubbles (Figure 5). It's available to use as an API in Roboflow.

2.4. Further research in automated manga translation

More advanced research has been done on translating manga. For example, the paper *Towards Fully Automated Manga Translation* [4] not only trained a model to extract and translate the text from speech bubbles. It describes a context-aware translation framework that includes information, such as text from other speech bubbles and metadata about the character speaking. This would result in more accurate translations that could further automate the translation process to remove the need for a human in the loop.

3. Approach

This project compares three approaches to determine how well each one performs and how much of an impact the recent innovations in computer vision have on the specific task of speech bubble detection.

1. Algorithmic approach: using Canny Edge Detection to obtain contour lines and determining which contours are speech bubbles based on the shape and level in the hierarchy.
2. Generic model: using a multipurpose model: Language Segment-Anything with text prompts: ("bubble", "text") to extract the speech bubbles.
3. Specialized model: using an existing model trained on 512 manga pages specifically for speech bubble detection.

For each approach, the input is the same subset of 1337 images from the Manga109 dataset[3] and the output should be the bounding boxes of the speech bubbles.

3.1. Approach 1 - Algorithmic

In this approach, we implemented speech bubble detection using Canny Edge Detection, which has been available

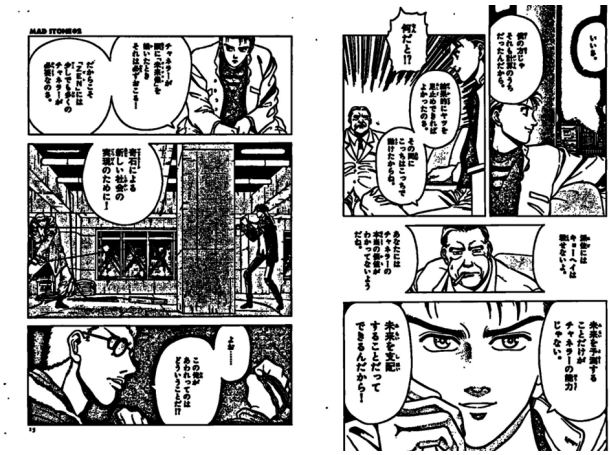

Figure 6. Approach 1 - Image Processing

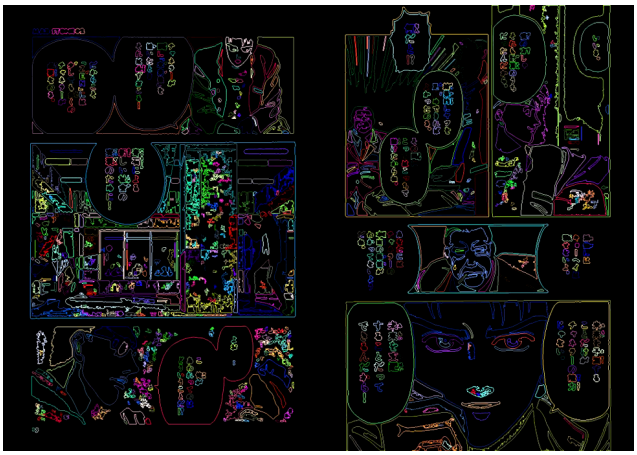

Figure 7. Approach 1 - Canny Edge Detection

since 1986, and contour lines. This would represent how speech bubble detection would have been implemented in the past before any of the developments in deep learning.

First, we would binarize and erode the input image to reduce complexity and simplify boundaries (Figure 6). We would also use the Canny Edge Detection algorithm from OpenCV to identify the edges in the image (Figure 7).

After processing the image, we would use the findContours function from OpenCV to get the contours represented as a list of shapes in the image along with a corresponding hierarchy tree containing the parent-child relations that indicate whether a shape is inside a shape.

We would use the hierarchy to extract the Level-2 contours (Figure 8), defined as the parents of the innermost contours. The reasoning behind this is that the text inside the speech bubbles are usually innermost contours, so the outlines of the speech bubbles would usually be Level-2. However, there are a significant number of speech bubbles that don't fit this criteria and we will describe these failure cases in another section.

Figure 8. Approach 1 - Level-2 Contours

From the Level-2 contours, we extract the contours that are shaped like speech bubbles (Figure 9). This means contours that are:

1. Not too big or too small. We filter the contours by their areas to remove those below a minimum threshold or above a maximum threshold. This would remove contours that belong to scenes from the manga (eg. character body parts, backgrounds).
2. Somewhat circular in shape. We determine how circular a shape is by determining how many times larger the perimeter is compared to a circle with the same area. We extract the contours where:

$$contour\_area > \frac{P^2}{4\pi N}$$

where P is the perimeter and N is the threshold of how many times larger the perimeter could be. We used N=2 such that it includes irregularly shaped speech bubbles that have spikes or sharp corners while excluding complex shapes that are Level-2 but are from scene in the manga rather than speech bubbles.

Finally, we use *TesseractOCR* [10] on the regions inside each of the detected speech bubble contours to extract the text along with a confidence score (Figure 10). We only exclude the speech bubbles with a confidence score that is too low since those are unlikely to contain text inside, and thus unlikely to be speech bubbles.

3.2. Approach 2 - Generic Model

The second approach uses the Language Segment-Anything model to extract segments based on a text prompt.

First, we extract speech bubbles from the input image using the text prompt "bubble" (Figure 11). This would extract most of the speech bubbles, but also segments that fit the description "bubble" but are not speech bubbles. We

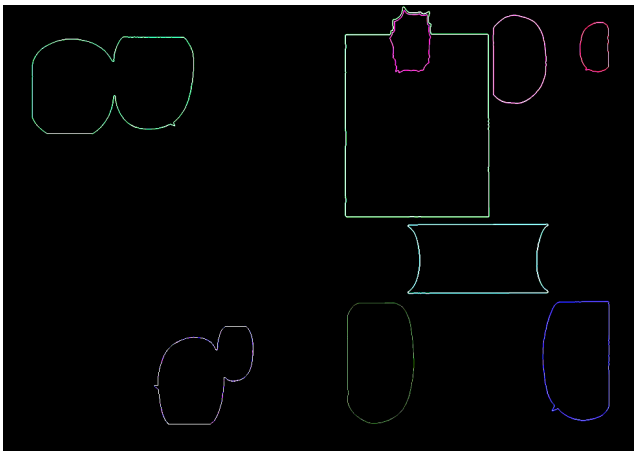

Figure 9. Approach 1 - Bubble-like Contours

Figure 10. Approach 1 - OCR

tried different text prompts such as ”speech bubble” and ”text balloon”, but found that those prompts often failed to extract most of the bubbles in the page. It’s a trade-off between having many false positives or many false negatives, so we would rather extract as many potential speech bubbles as possible and then filter them out.

We also extract text areas from the input image using the text prompt ”text” (Figure 12). We filter out the speech bubble segments that don’t overlap with a text segment to get only the speech bubbles that have text inside (Figure 13).

If a speech bubble is shaped like two speech bubbles stuck together, the Segment Anything Model would consider the speech bubble as two separate bubbles instead of one (Figure 14). We developed this algorithm to merge detected bounding boxes of the bubbles in this situation:

```
for b1 in bubbles
  for b2 in bubbles
    if b1 == b2
                                    continue
```

Figure 11. Approach 2 - ”bubble” segments

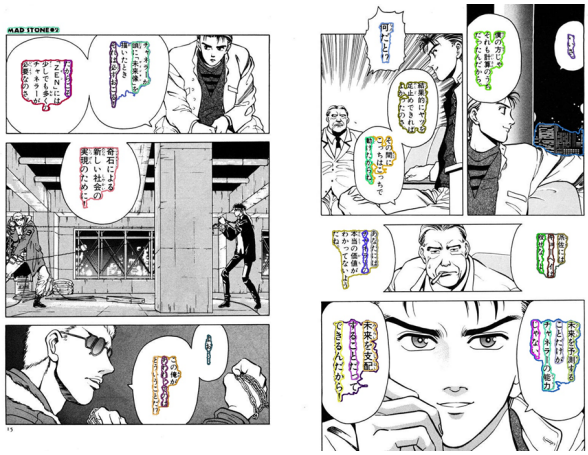

Figure 12. Approach 2 - ”text” segments

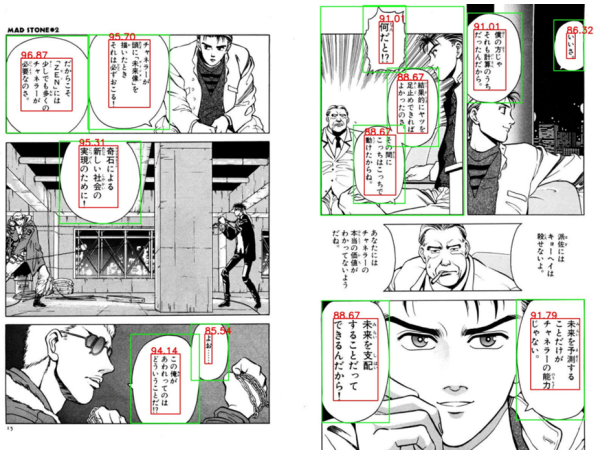

Figure 13. Approach 2 - Overlap ”bubble” and ”text” to get speech bubbles

Figure 14. Approach 2 - Merge bubbles

```
if not has_overlap(b1, b2)
  continue
b_new.x0 = min(b1.x0, b2.x0)
b_new.y0 = min(b1.y0, b2.y0)
b_new.x1 = max(b1.x1, b2.x1)
b_new.y1 = max(b1.y1, b2.y1)
b_new.conf = max(b1.conf, b2.conf)
for c in contours
  if get_bounding_box(c) == b_new
    bubbles b1, b2 = b_new
    break
```

This would try to merge two overlapping bounding boxes of bubbles *b1* and *b2* that might belong to two speech bubbles stuck together into one. If there is a bounding box of a contour *c* that is equal to the merged bounding box, we should keep the merged bounding box because it means there is a speech bubble with contour *c* that looks like the two speech bubbles stuck together.

We would get the final results: Figure 2

3.3. Approach 3 - Specialized Model

The third approach involves calling the Roboflow API to run the model from the Manga Text Detection Computer Vision Project [6] with each page of the manga as the input image. This would obtain the bounding boxes for all of the speech bubbles labeled "text_bubble" (Figure 5).

4. Results

From the precision and recall for speech bubbles with confidence greater than 50% (Table 1), the baseline Approach 3 has the highest performance as expected, while Approach 1 and Approach 2 have similar performance. This is true for any reasonable confidence threshold.

	Precision	Recall
Approach 1	0.8715	0.7056
Approach 2	0.8736	0.6721
Approach 3	0.9602	0.7710

Table 1. Confidence threshold = 50%

4.1. Evaluation Method

We evaluated the results from each approach against ground truth annotations for the same subset of 1337 manga pages

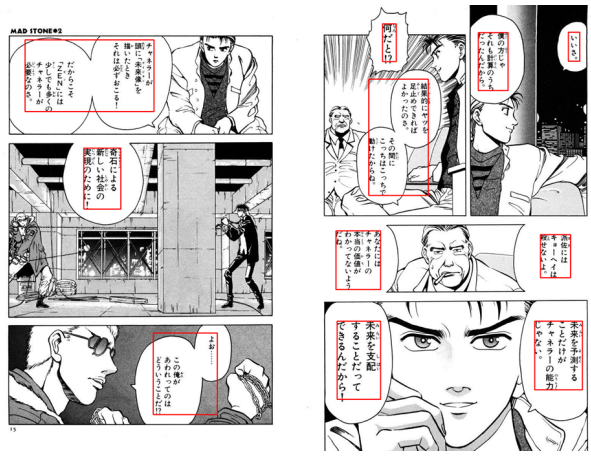

Figure 15. Manga109 Ground Truth Annotations

from the Manga109 dataset[3] (Figure 15) to obtain precision and recall metrics.

$$precision = \frac{relevant\_retrieved}{all\_retrieved}$$

$$recall = \frac{relevant\_retrieved}{all\_relevant}$$

where *all_retrieved* is the number of speech bubbles retrieved with a specific approach, *all_relevant* is the number of speech bubbles annotated in the ground truth Manga109 dataset, and *relevant_retrieved* is the number of speech bubbles retrieved with the approach that are also found in the Manga109 dataset.

We determine whether a retrieved speech bubble is found in the Manga109 dataset based on whether its bounding box overlaps completely with a bounding box of a text area from the Manga109 annotations.

4.2. Approach 1 - Algorithmic

We found that Approach 1 (algorithmic) (Figure 16) is almost as good as Approach 3 (specialized model). This is because there are characteristics we can use (Level-2, circular shape) that apply to most speech bubbles. This means that we can detect speech bubbles fairly accurately even without training a model.

Some failure cases (Figure 17) for this approach include:

A. The shape is too complex. This would cause the contour to get filtered out due to the circular shape criteria.
B. The bubble isn't a closed shape. This would cause the contour to not be Level-2 since it doesn't enclose a text contour.
C. Detected a Level-2 circular shape with text inside that isn't actually a speech bubble. This is when a shape from a scene in the manga or the outline of the panel box happens to enclose an innermost contour and the shape isn't too complex.

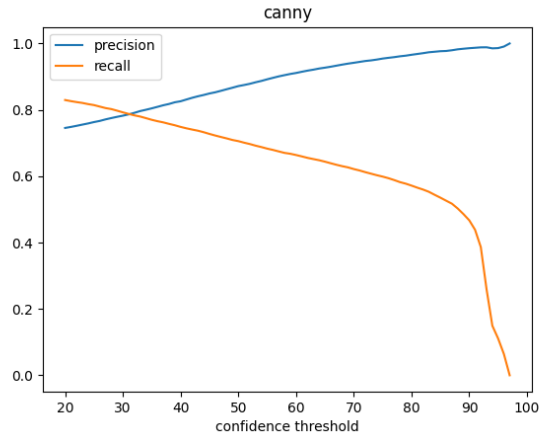

Figure 16. Precision/Recall for Approach 1 (Algorithmic)

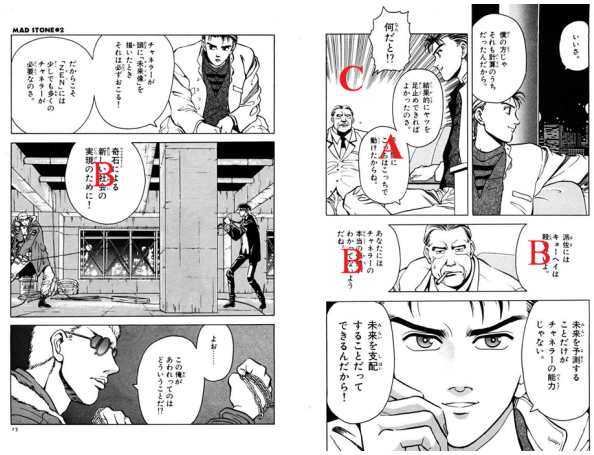

Figure 17. Approach 1 Failure Cases

Failure case A can be mitigated by adjusting the threshold N in the constraint for determining if the shape is circular enough.

Failure case B may be difficult to mitigate without significant changes to the algorithm. It could work if we detect text first from the entire image and then determine if the regions with text are inside a speech bubble by determining if the area surrounding the text is empty space.

Failure case C can be mitigated using additional filtering criteria, such as detecting whether there are non-text objects inside the contour, since speech bubbles should only contain text. This could be done by using OCR for each contour enclosed inside the potential speech bubble to determine if the inner contour contains text.

4.3. Approach 2 - Generic Model

We found that Approach 2 (generic model) (Figure 18) performed worse than Approach 3 (specialized model) and is only as good as Approach 1 (algorithmic).

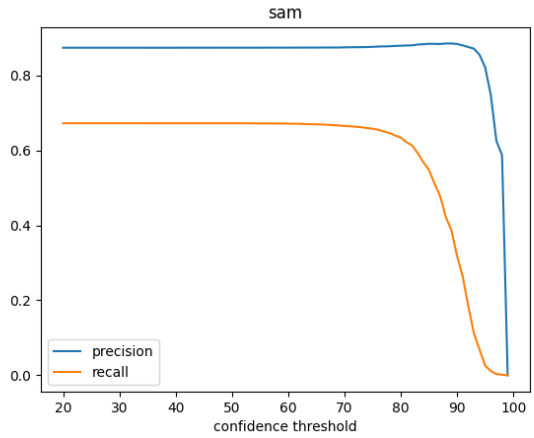

Figure 18. Precision/Recall for Approach 2 (Generic Model)

Figure 19. Approach 2 Failure Cases

This is because Approach 3 uses a model specifically trained to detect speech bubbles while the Language Segment-Anything model is generalized to detect objects based on a text prompt. Thus, the performance depends on how well Language Segment-Anything could understand the prompt. The model does not understand the prompt "bubble" correctly enough to fetch all of the speech bubbles and nothing else.

The failure cases (Figure 19) include:

A. Language Segment-Anything failed to detect a speech bubble.
B. Language Segment-Anything failed to detect a text area.
C. Language Segment-Anything detected a speech bubble with text inside that isn't actually a speech bubble.

These failure cases could be mitigated through improvements in natural language understanding so the Language Segment-Anything model could understand the prompts more accurately. One solution would be to use the original Segment Anything model together with a multimodal LLM such as GPT4o or Qwen-VL[1] on each segment with a prompt asking if the segment is a speech bubble or if the segment contains only text and nothing else. A multimodal LLM could understand and describe images[8] and might perform better than the model (Grounding DINO[9]) used

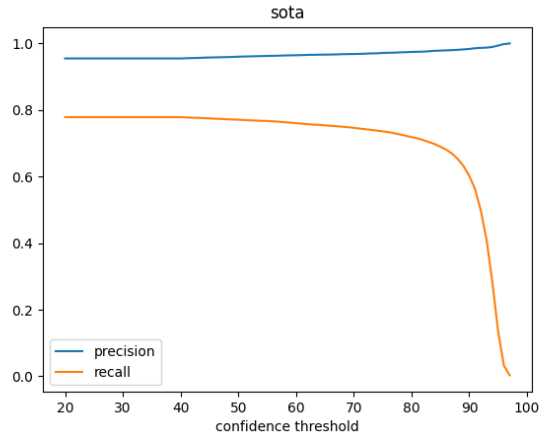

Figure 20. Precision/Recall for Approach 3 (Specialized Model)

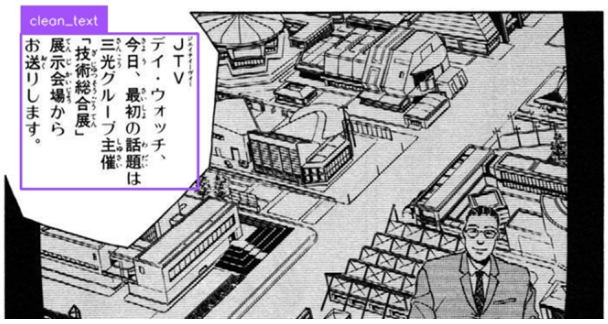

Figure 21. Approach 3 Failure Case

by Language Segment-Anything.

4.4. Approach 3 - Specialized Model

The performance for Approach 3 (Figure 20) is the highest compared to Approach 1 and 2. This is expected since Approach 3 uses a state-of-the-art model trained specifically for the task of speech bubble detection.

There are still some failure cases where it fails to detect some speech bubbles that don't look like parts of a circle (Figure 21). This could be mitigated by increasing the size and diversity of the training data so that the model would be trained on a larger number of irregularly shaped speech bubbles.

5. Conclusion

In conclusion, if we can find patterns in the data, an algorithmic approach could perform almost as well as training a model. Although the impact of deep learning has been profound in many areas of computer vision, for this specific task, we only saw a small increase in performance if we train a model. If we apply the mitigations described in the previous sections for the failure cases, we may further improve the performance of Approach 1 to be even closer to the state-of-the-art model.

The benefit of using an algorithmic approach is that it requires fewer computational resources. However, a model training algorithm is easier to implement, and the model can be retrained for many different tasks with little change to the algorithm. This is because instead of requiring a human to put in the effort to identify the patterns in the data, the model would be trained to fit those patterns automatically.

We also see that a specialized model trained for a specific task performs better compared to using a generic model such as Segment Anything, which is only as good as the algorithmic approach. This is expected since a generic model was trained to identify all kinds of objects from images instead of only speech bubbles. It's possible that a more powerful generic model such as state-of-the-art multimodal LLMs (GPT4o, Qwen-VL[1]) could perform better and approach or even surpass the performance of a specialized model, so this is worth studying further.

This project in speech bubble detection could be used in a tool for automatically translating manga. For future work, we can create a manga translation application using a specialized model that we determined performs best for speech bubble detection.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 6, 7

[2] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 1

[3] Azuma Fujimoto, Toru Ogawa, Kazuyoshi Yamamoto, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Manga109 dataset and creation of metadata. In *Proceedings of the 1st international workshop on comics analysis, processing and understanding*, pages 1–5, 2016. 2, 5

[4] Ryota Hinami, Shonosuke Ishiwatari, Kazuhiko Yasuda, and Yusuke Matsui. Towards fully automated manga translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12998–13008, 2021. 2

[5] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[6] mangaseer. Manga text detection computer vision project. https://universe.roboflow.com/mangaseer/manga-text-detection-xyvbw, 2024. 2, 5

[7] Luca Medeiros. Language segment-anything. https://github.com/luca-medeiros/lang-segment-anything, 2023. 2

[8] Fanqing Meng, Jin Wang, Chuanhao Li, Quanfeng Lu, Hao Tian, Jiaqi Liao, Xizhou Zhu, Jifeng Dai, Yu Qiao, Ping Luo, et al. Mmiu: Multimodal multi-image understanding for evaluating large vision-language models. *arXiv preprint arXiv:2408.02718*, 2024. 6

[9] IDEA Research. Grounding dino. https://github.com/IDEA-Research/GroundingDINO, 2023. 2, 6

[10] Ray Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, pages 629–633. IEEE, 2007. 3