IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Automatic Translation of Mangas into Other Languages

[1] Saikumar S, [2] Raghavendra R

[1] PG Student, [2] Assistant Professor

[1] Department of Master of Computer Applications,

[1]School of CS & IT, JAIN(Deemed-to-be-University), Bangalore, India

Abstract: Mangas are getting more popular as time goes on, thanks to the convenience with which they can be accessed through online entertainment. Many mangas have grown in popularity to the point where they now outsell western comics, which have set the bar for sales in the last decade. As a result of the rising demand for localization of these works, publishers have pushed for mangas to be translated into multiple languages for easier consumption and therefore greater sales.

Cleaning manga panels and, if required, redrawing panels are the initial steps in the translation process. The Japanese text in the image is translated, and then the translated text is in-painted back into the cleaned image. To have the manga translated, numerous competent individuals with skills in picture editing and translation are necessary.

The goal of this article is to present an automated system for translating manga that will identify, translate, and in-paint the translated text. This will cut the number of man-hours and staff needed to complete the task while also integrating everything together into a single functional framework.

Index Terms **- Manga, Speech Balloon extraction, Text Detection, OCR, Translation**

I. INTRODUCTION

Mangas are Japanese comic books that are traditionally illustrated in black and white and are used for amusement sake. Manga's origins may be traced back to the end of the 18th century, although its most modern form of consumption stems from after WWII. That popularity has soared in the past two decades throughout the world, necessitating the translation of these works into languages native to the market.

English is the most widely translated language in Manga, and as a result, it has the largest readership. Manga has also been translated into French, German, and Indonesian, to mention a few. The trouble with translating manga is two-fold. First, there's the publisher's and distributor's interest in translating a certain work, which takes into consideration the aforementioned cost-benefit analysis of whether or not translating a work will be worthwhile. Second, the languages into which a publisher will translate it to. As previously said, English has the largest market share in translated mangas, and publishers seldom translate their works into languages other than English.

Even if a work is translated, there is no guarantee that the translated product will be published on time. This is due to a scarcity of skilled personnel and the time required to complete such tasks. Depending on how well the localised product's release does in sales, translation release timelines might be spaced out across several months or even years. As a result, many in-demand books are not translated due to competing, higher-performing titles released by the same publisher. This is where we propose our approach for digitally translating mangas and delivering them to customers. One of the things will not be the scope of this study is translating free-floating text outside balloons, this is because if we remove text covering a part of the image, we need to in-paint the part of the image back onto the image. This is currently a huge problem in manga in-painting because due to screen-tones that are used for shading in the image, disturb any model that can be created to in-paint it.

II. THE JAPANESE LANGUAGE AND THE ISSUES WITH TRANSLATION

When talking about translation in technical terms, we came up with the following steps: Detect Texts, Detect Speech Balloons, Translate Text, and finally in-paint the translated text back into the image. While we'll talk about the solutions we came up with to these challenges later, let's start with the obstacles we ran into when translating Japanese manga.

Because the Japanese language lacks whitespace between words to distinguish between them, it is unintelligible to use a conventional alphabet system. Unlike English, Japanese has three alphabetical systems: Hiragana (informal Japanese made up of syllables and hence a phonetic class), Katakana (Japanese for writing foreign words), and Kanji (An alphabet system where each letter represents a word) which are shown in Figure 1. As a result, Japanese is a mix of phonetic and non-phonetic languages. This necessitates for fine-tuning of text detection to the language. This challenge is worsened by the fact that Japanese is read from left

to right and written vertically from top to bottom in mangas, thus a conventional OCR would fail to recognise the language's peculiarity.

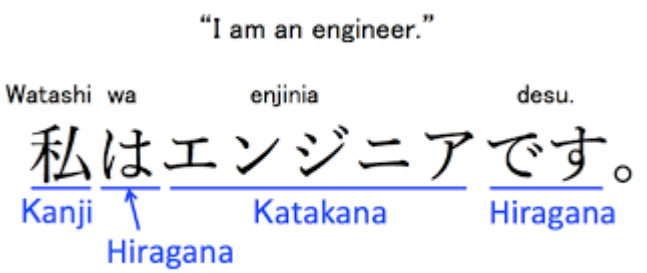

Fig. 1. Japanese Alphabets Systems

The goal of this article is to present an automated system for translating manga that will identify, translate, and in-paint the translated text. This will cut the number of man-hours and staff needed to complete the task while also integrating everything together into a single functional framework.

The Kanji system, which was mentioned, is vast, with over 20,000 characters; n even native Japanese speakers find it difficult to remember them all, though only 2,000–5,000 are used on a daily basis; however, even this makes the use of kanji a double-edged sword for writers, as the meaning may not be conveyed. This is why a fourth informal alphabet called furigana was created, which is similar to hiragana but written next to the kanji so that it may be readily understood because hiragana is comprised of only syllables. This implies we need to get rid of the furigana in order to get a cleaner translation, as extracting text with furigana means we can extract the same word twice (the kanji and the furigana shown in Figure 2.), resulting in a less accurate and possibly incorrect translation.

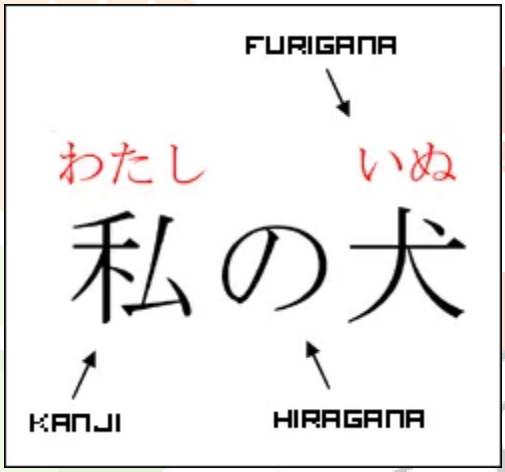

Fig. 2. Kanji, Furigana and Hiragana

III. Literature Survey

The use of a stroke-width ratio is proposed in [2] as a strategy for distinguishing text from the image. They calculate the width of each artefact in the image by looking at the stroke qualities of an artifact or object, which is how an image is drawn in the image. The artifacts are then filtered by calculating the stroke-width ratio, which distinguishes the textual and non-textual regions of the image and returns the text letters discovered in it. They next aggregate the discovered objects to determine the words and paragraphs, and finally extract the recognized text from the picture. The dataset for the training of this model is using [1], which is a collection of commercial mangas approved by their respective authors for academic usage, this is the only reliable dataset to conduct such experiements.

To distinguish between the text and the scenery in the image in [3] a Deep-Learning model is utilized. They suggest an approach in which the model that identifies character components proposes an area by placing all related components discovered by a Random Forest Classifier, thus minimising noise in character component extraction. The approach then offers an Area Classification algorithm that eliminates the portions that were incorrectly classified as text sections when the region was suggested. Finally, sections with appropriately detected text parts are provided.

A linked components technique is suggested in [4] for extracting text from a picture. The RGB picture is initially pre-processed to remove noise and smooth the image using a median filter. By applying a threshold range of 0 to 1, the RGB picture is transformed to a binary representation, which is black and white pixels. The picture is then subjected to a CCL (Linked Component Labelling) technique, which finds the image's connected components. Balloon detection is used to construct text and non-text blobs [5]. A text blob is defined as a blob that covers 8% or 10% of the original picture. The text is then retrieved using OCR from the text blobs or chunks that have been categorized.

An active contour model is suggested in [6] for detecting balloons in images. An active contour is a snake-like model made up of flexible points that covers a confined region in a picture. The snake will next traverse the image's space, returning the space points of the contour's edges for each recognized closed contour. It employs an energy-based approach, detecting energy in three ways for each component: an exterior energy of the picture relative to the closed contour, an internal energy of the contour, and the energy of the

text contained within the contour. The snake will be able to fit onto the boundaries of the identified contour while suppressing the coarse areas of the contour's limits due to the energy detected for the balloons.

[7] proposes using area extension to detect balloons in a picture. First, a connected component analysis is used to extract the image's background in relation to the balloons. This study employs area expansion, which is repeated as many times as the number of background points detected. Potential balloons are extracted based on their regions after the picture is converted to binary. To extract efficient balloons carrying text, heuristic techniques are utilised to select viable balloons, while visually cutting off the narrowest component of the balloon.

The balloon extraction in [8] is based on the RGB image being converted to an HSV colour space. We can detect the likely speech balloons using these characteristics and a region growth method similar to [7], because all white regions have a high Value (V) and Saturation (S) (S). They are then filtered depending on the size and shape of the discovered balloons. A linked component analysis is then used to find the text pieces within the potential balloons. The text is then dilated in order for the algorithm to recognize it. Finally, the retrieved text is extracted, as well as the position of the speech balloons.

In addition to these in [9] and [10], CNN models are used to extract panels from mangas and detect objects such as character's face, panels, in-image objects are done. These are really complex CNN models that are computationally exhaustive and need a huge dataset such as the dataset [1] to train these models.

IV. IMPLEMENTATION

The discussion of the implementation will be split into four sections each covering the problems we set forth in the introduction.

4.1 Detect Text in the Image

As previously mentioned, mangas have the language written vertically rather than horizontally, as seen in Figure 3., which makes extraction harder. It's a benefit in terms of text detection. The goal of the detection is to locate related components that are densely packed together. This is accomplished by creating a binary structure in a picture and then looking for labels inside it. We may then detect bounding boxes for these components, which will comprise the most tightly clustered components, by finding related components that are immediately near to each other.

This may be done directly on the image without any pre-processing, improving efficiency. Connected components must be of a size that is neither too huge nor too tiny in relation to the image in order to be classified as text. The bounding boxes are created by grouping the linked components that are the closest to one other, either by actual intersection or by the minimal mean distance between all of the components. It is critical to have a mean distance to the centre of the detected text or related components because it allows the components to be judged as a group rather than as individual components, increasing the efficiency of grouping them into a bounding box.

On the linked components, a bounding box is drawn as a rectangle that covers the detected components. While this strategy works in the majority of circumstances, it does occasionally leave text behind. As a countermeasure, we run it through the algorithm once more, but this time we process the image first. The image's recognized text bounding boxes are first deleted, resulting in all pixels within the bounding box becoming white.

The picture is then converted to a binary image, with a threshold over which all pixels become white and the remainder of the pixels become black. The lettering will still be dark at this stage, so we'll invert the binary picture, swapping all the black pixels for white and vice versa, then dilate the image. Dilation increases the number of white pixels while decreasing the number of dark pixels. Dilation is performed using a kernel of any size or shape that is overlapped with the picture to add extra white pixels where the algorithm considers appropriate.

The dilated picture will now have more linked components, allowing the text detection algorithm to be run again, ensuring that all of the text in the image is detected. The bounding boxes are then filtered to see if the rectangles resemble the shape of histograms in a graph; the rectangles that do not match the related components well are removed. The reason we filter the rectangles in this way is due to the aforementioned peculiarity of mangas where Japanese is written vertically, which guarantees that the image's bounding box looks vertically upright or horizontal.

The disadvantage of this approach in general is that it will invariably designate some non-text components as text, as shown in Figure 4., regardless of whether it is run twice with an inverted binary image. While this would be an issue if we were to translate the picture from this point alone, there are still text balloons to be recognized, which can assist in recognizing the correct text bounding boxes.

Fig. 3. Detected Vertical Text

Fig. 4. Falsely Detected Text Components

4.2 Detect Balloons in the Image.

We transform all the pixels within the text bounding boxes to white after we've found them. This will ensure that all identified text is removed from the image. The picture is now free of any text that might hinder subsequent operations. The image is now transformed to a binary representation of itself, which is then inverted as discussed earlier. The picture is then dilated to fill in all the missing pixels, using the same process as before. Now we'll subject the image to a search for contours.

A contour is a set of points that detects a closed component in an image and allows all of the points to be deformed and adjusted to the picture's closed bounds. To do this, we use contour approximation methods on the picture, which begin by locating the image's related components in the same manner that text does, but we look at a different approach of extracting the components here. We locate all the components that are nested within a much bigger component. To do so, we begin by locating the largest component based on its size, subsequently dig down inside their border to locate lesser components and components underneath them, and so on. We'll be able to locate contours instead of text blocks because they are unfiltered.

This will provide us with all of the contour points on all of the contours in the image. After that, we discover the bounding box for each contour we've identified by detecting the lowest x and y coordinates, which will decide the top-left corner of the bounding box, and the highest x and y values, which will determine the bottom-right corner. In picture coordinates, unlike a traditional x-y axis, x and y coordinates get larger as you move towards the right and bottom of the image. We can get all the bounding boxes for all the contours in the image using these coordinates as shown in Figure 5.

Fig. 1. Detected Balloons

This procedure will also produce undesired outlines in the image, which will need to be filtered out. We would identify all the bounding boxes within whose limits text bounding boxes also exist for this filtration. Because most free-floating text in mangas is not surrounded by closed limits by definition, this ensures that we locate all text bounding boxes that are solely contained within a balloon. When two or more bounding boxes cover the same text box, we chose the one with the smallest area because it will be more efficient when we in-paint text into the image later on.

If two bounding boxes overlap, we test to see if one of them occupies the text boxes within it; if so, the parent bounding box is the one that contains both text boxes. We won't try to extract and translate free-floating text in the image because, as previously said, we can't in-paint free floating text.

4.3 Extracting Text from the Image

We must first remove the aforementioned furigana from the text before performing any OCR on the bounding boxes. This means we'll have to run the same text component algorithm on the bounding boxes again, but this time we'll be looking for components that are much smaller than the text components we found before. This can be done by tuning the ratio of text components to furigana components, which happens dynamically with each text we find, because a larger text will have a larger furigana than normal text. We remove the furigana's bounding boxes from the text by setting all pixels in that bounding box to white after we've detected them. We now have all the text that we need to extract as shown in Figure 6.

Fig. 2. All detected text that will get extracted

After that, we can extract text from the bounding boxes using an OCR, in our case the tesseract OCR. The tesseract OCR can extract Japanese text, but we'll need some more parameters to extract it vertically, as that's how they're written, so we can obtain an exact translation. We can then translate and in-paint the text once we receive it.

4.4 Translating Test and In-painting text.

For translation, we employed the Helsinki NLP model, which is capable of multilingual translation and can be deployed with few resources. We will use the API and provide the text that we extracted and then have it translated into Japanese. One of the reasons we looked for balloon bounding boxes containing many text boxes is because they are intended to be read in one stretch in writing, therefore translating the combined text of the text boxes included within the balloon bounding box would result in a more accurate translation.

We may begin in-painting the text onto the image after we receive the translated text. We divide the sentences into lines anytime a word is too big for the bounding box to fit in since we need it to fit within the balloon bounding box. This is accomplished by adding a y offset to each new line. This signifies that the text will not extend beyond the bounding box of the balloon and guarantee that the in-paint is cleaner.

IV. RESULTS

Fig. 3. Manga image to be translated

We applied our method on the manga image shown in Figure 7. We will now walk through all the stages of detection in the process.

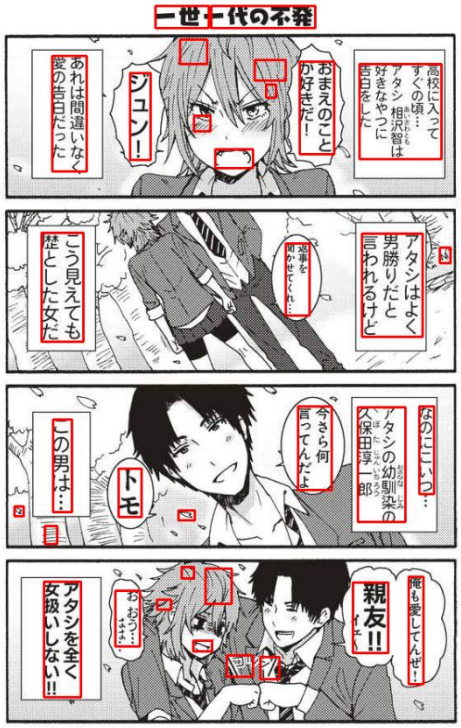

Fig. 4. Detected Text

As evident from Figure 8. we can see that some areas of the image where detected as text even though they are not.

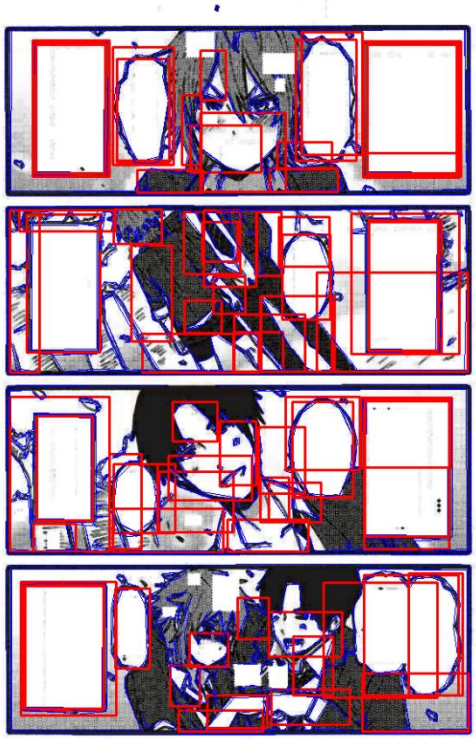

Fig. 5. Detected Balloons

The bounding box in Figure 9. shows how the balloon detection algorithm recognizes the contours, and you can see how it incorrectly detects different areas of the image as being made up of balloons.

Fig. 6. Filtered Text and Balloon

After filtration, you can see in Figure 10. how we were able to correctly locate the location of the text and balloons in the image.

Fig. 7. Translated Image

The translated text is in-painted onto the image in Figure 11. The image at the top of the page is not translated as part of the study's objectives, nor is any free-floating text on the image translated.

V. CONCLUSION

This work shows what may be accomplished when translating mangas using a pure mathematical and image morphological method. It produces a good result by keeping the translated image as clean as feasible and as close as possible to a perfect machine translation. Because speech balloons in mangas are exceedingly variable and do not follow a standard within the same piece of work, the project's future focuses on producing a more appropriate balloon bounding box. Machine learning will almost certainly be used in the strategy. Another area that can be improved is translation. Translation models tailored to a certain manga can be developed, resulting in a more accurate translation. In-painting of free floating text is one area where there will be little improvement for the foreseeable future. This is owing to the usage of screen tones in mangas for shading and toning in the image, which even the most complex CNN models cannot address. However, a great deal of research is being done in these areas, not just specific manga, but on a variety of other computer vision issues.

REFERENCES

[1] K. Aizawa et al., "Building a Manga Dataset 'Manga109' With Annotations for Multimedia Applications," IEEE multimed., vol. 27, no. 2, pp. 8–18, 2020, doi: 10.1109/mmul.2020.2987895.

[2] B. Piriyothinkul, K. Pasupa, and M. Sugimoto, "Detecting text in Manga using stroke width transform," in 2019 11th International Conference on Knowledge and Smart Technology (KST), 2019, pp. 142–147.

[3] Y. Aramaki, Y. Matsui, T. Yamasaki, and K. Aizawa, "Text detection in manga by combining connected-component-based and region-based classifications," in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 2901–2905.

[4] M. Sundaresan and S. Ranjini, "Text extraction from digital English comic image using two blobs extraction method," in International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012), 2012, pp. 449–45

[5] W.-T. Chu and C.-C. Yu, "Text detection in Manga by deep region proposal, classification, and regression," in 2018 IEEE Visual Communications and Image Processing (VCIP), 2018, pp. 1–4.

[6] C. Rigaud, J.-C. Burie, J.-M. Ogier, D. Karatzas, and J. Van De Weijer, "An active contour model for speech balloon detection in comics," in 2013 12th International Conference on Document Analysis and Recognition, 2013, pp. 1240–1244.

[7] X. Liu, Y. Wang, and Z. Tang, "A clump splitting based method to localize speech balloons in comics," in 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 901–905.

[8] K. N. Ho, J.-C. Burie, and J.-M. Ogier, "Panel and speech balloon extraction from comic books," in 2012 10th IAPR International Workshop on Document Analysis Systems, 2012, pp. 424–428.

[9] V. Nguyen Nhu, C. Rigaud, and J.-C. Burie, "What do we expect from comic panel extraction?," in 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), 2019, vol. 1, pp. 44–49.

[10] H. Yanagisawa, T. Yamashita, and H. Watanabe, "A study on object detection method from manga images using CNN," in 2018 International Workshop on Advanced Image Technology (IWAIT), 2018, pp. 1–4.

[11] J. Tiedemann, "The tatoeba translation challenge -realistic data sets for low resource and multilingual MT," Arxiv.org [Online]. Available: https://arxiv.org/pdf/2010.06354.pdf