JSCE (Journal of System and Computer Engineering)
Vol.7, No.3, July 2026, pp. 266-276
ISSN (online): 2723-1240
DOI : https://doi.org/10.61628/jsce.v7i3.2712

Performance Comparison of YOLOv8 and YOLOv11 for Manga Text Area Detection Based on the Intersection over Union Metric

Kautsar Hasby Dastien Fredila [1,a]; Ericks Rachmat Swedia [2,a]; Margi Cahyanti [3,a]; M Ridwan Dwi Septian [4, a,*]

[1,2,4] *Departement of Informatics Engineering, Universitas Gunadarma, Indonesia*
[3] *Department of Computer Science and Information Systems, Universitas Gunadarma, Indonesia*
[a] *kautsar@lab-gunadarma.id;* [b] *ericks_rs@staff.gunadarma.ac.id;* [c] *margi@staff.gunadarma.ac.id;*
[d] *ridwandwiseptian@staff.gunadarma.ac.id*
** Corresponding author*

Abstract

Manga has complex visual characteristics, such as variations in speech bubble shapes, diverse text orientations, and dense background illustration, which complicate automatic text area detection. Detection errors in the form of false positives and false negatives can cause text areas to be localized inaccurately and affect the subsequent processing stages. This study compares the performance of YOLOv8m and YOLOv11m for text area detection in Japanese manga, evaluated with the Intersection over Union (IoU) metric together with standard detection metrics, namely precision, recall, F1-score, and mean Average Precision (mAP). The dataset consists of 551 annotated manga images covering three classes, namely clean_text, messy_text, and text_bubble, and the evaluation was performed on 50 test images containing 573 text objects. Both models were trained under identical configurations for 60 epochs. The two architectures achieved closely comparable accuracy. YOLOv8m and YOLOv11m obtained an mAP@50 of 0.8684 and 0.8734 and an F1-score of 0.8577 and 0.8638, respectively, while YOLOv8m attained a slightly higher mAP@50-95 (0.6061 versus 0.6006). A paired Wilcoxon signed-rank test on the per-image IoU showed no statistically significant difference between the two models ($p = 0.0997$). The main performance gap appeared across classes rather than across models, with text_bubble detected reliably (mAP@50 above 0.98) and messy_text remaining the hardest class for both models (recall around 0.52 to 0.55). YOLOv11m matched the accuracy of YOLOv8m while using fewer parameters (20.0M versus 25.8M) and lower computational cost (67.7 versus 78.7 GFLOPs), making it the more efficient option for the text detection stage of an automatic manga translation pipeline.

Keywords*—YOLOv11, YOLOv8, Object Detection, Japanese Manga, Intersection over Union*

1. Introduction

Manga is a popular visual media form that originates from Japan and has a very large readership across many countries (Auliawan & Ratna, 2024). The growing popularity of manga drives the need for automatic translation systems that can help international readers understand the story without waiting for manual translation. As artificial intelligence technology advances, various studies have developed automatic translation systems that integrate text detection, Optical Character Recognition (OCR), and machine translation to produce fast and efficient translations (Al-Ibrahim et al., 2024; Ghaffar Nia et al., 2023).

In an automatic manga translation system, the text area detection stage is a critical component because it determines the location of the text to be processed in the next stage (Muclis et al., 2023). Errors at the detection stage, such as false positives and false negatives, can cause the OCR to fail to recognize characters accurately, which affects the overall translation quality. Text detection accuracy is therefore the primary factor that determines the success of an automatic manga translation system (Puteri & Meirza, 2024).

Advances in deep learning technology have produced various object detection models capable of localizing objects quickly and accurately. One widely used model family is You Only Look Once (YOLO), which is known for its balance between speed and accuracy across various computer vision tasks (Mahadevkar et al., 2022; Sirisha et al., 2023). YOLOv8 is one of the widely used versions because of its stability and strong performance in object detection tasks (Shia & Ku, 2024). Meanwhile, YOLOv11, as a newer generation, offers architectural refinements designed to improve computational efficiency and object detection quality (Alif, 2024).

The success of object localization in YOLO models is generally evaluated using the Intersection over Union (IoU) metric, namely the ratio between the intersection area and the union area of the predicted and ground truth bounding boxes. IoU has become an evaluation standard in many object detection studies because it captures the positional and dimensional accuracy of the bounding boxes a model produces (Zou et al., 2023). In modern implementations, the IoU concept is used not only as an evaluation metric but is also integrated into the loss function to improve the model learning process. YOLOv8 and YOLOv11 employ Complete Intersection over Union (CIoU), which accounts for area overlap, the distance between bounding box centers, and aspect ratio consistency, producing a more effective optimization process (Li et al., 2024; Silmina et al., 2025).

Although YOLOv8 and YOLOv11 have been widely used in various object detection scenarios, research that specifically compares the two models for text area detection in Japanese manga remains relatively limited. Most prior studies focus on general objects or conventional text documents (Del Gobbo & Matuk Herrera, 2020), so the unique characteristics of manga, which has a complex visual structure, have not been extensively evaluated. A rigorous comparative analysis, supported by standard detection metrics and statistical testing, is therefore needed to assess how the two models perform for text area detection in Japanese manga.

This study aims to compare the performance of YOLOv8 and YOLOv11 in detecting text areas in Japanese manga using the Intersection over Union (IoU) metric. The dataset consists of manga page images annotated into three classes, namely clean_text, messy_text, and text_bubble. Both models were trained under equivalent training configurations so that any difference in the results can be attributed to the architecture itself. Beyond comparing the two models, this study also examines an evaluation issue specific to sparse-object pages and reports the statistical significance of the observed differences. The findings are expected to contribute to the selection of a more effective text detection model as an initial stage in developing an automatic manga translation system based on OCR and machine translation.

Prior studies have frequently compared YOLOv8 and YOLOv11 on general object detection, yet no study has been found that specifically evaluates the two models for Japanese manga text area detection with the IoU metric as the main evaluation focus. This study addresses that gap by conducting a comparative evaluation in the manga domain, which has visual characteristics that differ from conventional object detection datasets.

2. Method

This study compares the performance of the YOLOv8 and YOLOv11 models in detecting text areas in Japanese manga using the Intersection over Union (IoU) metric (Puteri & Meirza, 2024; Stodt et al., 2023). The methodology follows the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework, which provides a systematic structure for data-driven experiments and is widely applied in machine learning research. The research methodology

consists of data collection and annotation, preprocessing, model training, performance evaluation, and system implementation. Each stage was carried out sequentially, starting from understanding the manga text detection problem and ending with the integration of the better performing model into a working system. To keep the comparison fair, both architectures were placed under identical experimental conditions, namely the same dataset, hyperparameters, and hardware environment, so that any difference in performance could be attributed to the architecture itself rather than to external factors. The overall research flow is shown in Figure 1. This workflow outlines the sequence of data preparation, model training, and performance evaluation stages used in the study.

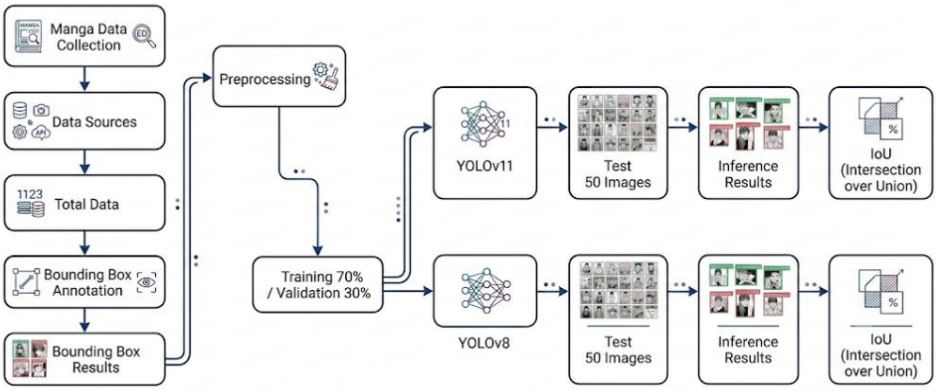

Figure 1. Model Development Workflow

2.1 Dataset Collection and Annotation

The dataset was obtained from the Roboflow platform and contains Japanese manga images. The initial dataset comprised 516 images already annotated for text areas. To increase the amount and variety of data, 35 additional manga images were added and manually annotated, bringing the total dataset to 551 images.

The annotation was performed on the Roboflow platform using the bounding box method (Zhang et al., 2023). The detected objects were grouped into three classes, namely clean_text, messy_text, and text_bubble. Onomatopoeia text such as "ドキドキ" was not labeled because it is expressive and is not the focus of dialogue translation. Beyond the label categories, visual text variation was also considered during annotation. The dataset covers various font styles, such as the typical manga dialogue font, thin fonts, and bold fonts. This diversity affects the difficulty of detection and extraction by OCR, so it was retained so that the model can recognize variations in text appearance more comprehensively. An example of the dataset annotation process using Roboflow is shown in Figure 2.

Figure 2. Image Labeling

2.2 Preprocessing Data

The preprocessing stage converted the annotation results into a format compatible with the YOLO model. The dataset was then split into two parts, namely 70% training data (training set) and 30% validation data (validation set) (Mohammed & Alsunosi, 2022).

In addition, this study prepared 50 manga images as a testing set. This test data was not used during training and served only to evaluate model performance on previously unseen data.

2.3 YOLO Model Configuration and Training

This study compares two object detection models, namely YOLOv8m and YOLOv11m. Both models use pre-trained weights as initial weights to accelerate convergence during training (Jaiswal et al., 2023).

The medium (m) variant was selected for both architectures to keep the number of parameters and computational cost at a comparable scale, so that the comparison reflects the architectural differences between the two generations rather than differences in model capacity. Using pre-trained weights through transfer learning also allows the models to converge with a relatively small manga dataset.

To ensure an objective comparison, both models were trained using a fair parameter configuration. The training was carried out on Google Colab with an NVIDIA Tesla T4 GPU (Shah, 2024). The training parameters used are shown in Table 1.

Table 1. YOLOv8 and YOLOv11 Training Parameters

Parameter	Value
Image Size	640 × 640 pixels
Epoch	60
Batch Size	16
Optimizer	SGD
Learning Rate	0.01
Patience	0

Training was run for 60 epochs to obtain a stable model while minimizing the possibility of overfitting.

2.4 Evaluation Using Intersection over Union

Model performance was evaluated using the Intersection over Union (IoU) metric (Stodt et al., 2023). This metric measures the degree of agreement between the bounding box predicted by the model and the bounding box ground truth defined during annotation. The IoU value is computed using Equation (1).

$$IoU = \frac{Area\ Overlap}{Area\ Union} \tag{1}$$

The higher the IoU value obtained, the better the model capability in localizing objects (Tjahyanti & Pratama, 2025). In this study, the IoU value was computed for every detected object across the 50 test images and then averaged to obtain the overall performance of each model.

In addition to IoU, the models were evaluated using standard object detection metrics, namely precision, recall, F1-score, and mean Average Precision at IoU 0.50 (mAP@50) and averaged over IoU 0.50 to 0.95 (mAP@50-95), computed per class and overall (Zaidi et al., 2022). Two refinements were applied to the IoU analysis. First, images containing no ground truth object were reported separately as detection presence cases rather than being assigned a perfect IoU, because rewarding an empty prediction with a score of 1.0 conflates localization quality with detection presence. Second, to test whether the difference between the two models was statistically meaningful, a paired Wilcoxon signed-rank test was applied to the per-image IoU values, complemented by a paired t-test.

2.4.1 Implementation of the Intersection over Union (IoU) Computation

The system receives two coordinate inputs in the format (x1, x2, y1, y2) for each detected object, namely Box A (Ground Truth), which is the manual coordinate contained in the dataset label, and Box B (Prediction), which is the coordinate produced by the YOLOv8 or YOLOv11 model (Hidayatullah et al., 2025). To obtain the IoU value on the test data, the system performs the computation in the following order:

1. Determining the Intersection Area Coordinates
   The system locates the coordinates of the overlapping area using minimum and maximum comparison logic as defined in Equations 2 to 6 (Reza et al., 2025).

$$x_{min\_inter} = \max(x_{min\_A}, x_{min\_B}) \quad (2)$$

$$y_{min\_inter} = \max(y_{min\_A}, y_{min\_B}) \quad (3)$$

$$x_{max\_inter} = \min(x_{max\_A}, x_{max\_B}) \quad (4)$$

$$y_{max\_inter} = \min(y_{max\_A}, y_{max\_B}) \quad (5)$$

$$If \ x_{max\_inter} > x_{min\_inter} \ and \ y_{max\_inter} > y_{min\_inter} \ \text{Then} \quad (6)$$
$$Area_{inter} = (x_{max\_inter} - x_{min\_inter}) \times (y_{max\_inter} - y_{min\_inter})$$

2. Combining the Total Area (Union)
   The IoU equation is obtained by dividing the Area of Intersection by the Area of Union, as in Equation (1). The computed IoU value is then classified using a defined threshold, namely 0.5 (50%). If the IoU value is > 0.5, the detection is declared a True Positive (TP). Conversely, if the IoU is < 0.5, it is declared a False Positive (FP) (Wu et al., 2022).

3. Results And Discussion

The comparison between YOLOv8 and YOLOv11 is examined from two complementary angles, namely the quantitative measurement of IoU and inference time across the 50 test images, and the qualitative behavior observed on individual manga panels. The following subsections first present the tabulated results for all 573 text objects, then analyze the detection performance and several representative cases, and close with the implications for the manga translation pipeline.

3.1 Model Testing Results

The testing was conducted by implementing the YOLOv8 and YOLOv11 models in a Python programming environment using a notebook (.ipynb) format. Unlike the training process, in the testing stage the system was configured to automatically extract bounding box coordinates from the ground truth using the Intersection over Union (IoU) metric. In addition to the IoU value, inference time was measured to evaluate the computational efficiency of each model. The test results are shown in Table 2. For each test image, Table 2 lists the number of ground truth objects, the IoU obtained by both models, and the corresponding inference time. The per-image IoU values were then averaged across the test set to summarize the overall localization performance of each model.

Table 2. Comparison Results of YOLOv8 and YOLOv11

No.	Image ID	Number of Objects	IoU		Inference Time (ms)	
			YOLOv8	YOLOv11	YOLOv8	YOLOv11
1	manga_001	3	0.2424	0.2805	1304.75	1048.72
2	manga_002	19	0.8968	0.8502	1151.03	1073.93
3	manga_003	10	0.9217	0.9158	1083.76	986.4
4	manga_004	16	0.7771	0.7428	1174.47	1037.18
5	manga_005	13	0.7893	0.9078	1178.82	1382.51
6	manga_006	0	1	1	1073.86	1507.73
7	manga_007	7	0.9225	0.9041	1683.47	1603.7
8	manga_008	9	0.8626	0.9561	1658.8	1529.86
9	manga_009	4	0.7932	0.7191	1678.7	973.96
10	manga_010	16	0.9347	0.9333	1711.76	980.06
11	manga_011	12	0.8694	0.9389	1319.56	1039.25
12	manga_012	0	0	1	1158.21	1046.7
13	manga_013	8	0.302	0.2943	1082.85	998.68
14	manga_014	6	0.2748	0.419	1093.05	992.53
15	manga_015	17	0.7521	0.8675	1079.09	985.07
16	manga_016	6	0.7415	0.8619	1076.77	986.74
17	manga_017	10	0.5539	0.8118	1076.37	984.32
18	manga_018	15	0.6327	0.6516	1081.19	1013.51
19	manga_019	20	0.5761	0.5249	1090.57	1519.6
20	manga_020	19	0.6905	0.7248	1606.38	1522.2
21	manga_021	16	0.8019	0.7648	1691.74	1500.69
22	manga_022	14	0.5703	0.5186	1699.72	1308.09
23	manga_023	18	0.7517	0.6551	1353.8	971.66
24	manga_024	19	0.7841	0.6913	1089.11	984.76
25	manga_025	12	0.7805	0.8106	1079.94	981.16
26	manga_026	11	0.7454	0.8261	1082.06	983.62
27	manga_027	7	0.8484	0.8609	1083.71	982.46
28	manga_028	4	0.7882	0.7996	1084.42	981.7
29	manga_029	10	0.6478	0.7402	1107.07	987.8
30	manga_030	18	0.7189	0.7424	1091.75	977.82
31	manga_031	20	0.6533	0.6758	1085.09	986.54
32	manga_032	2	0.8195	0.8818	1384.11	1238.89
33	manga_033	8	0.7797	0.8328	1684.34	1542.99
34	manga_034	7	0.5499	0.7187	1728.25	1520.81
35	manga_035	11	0.644	0.7	1597.62	1531.22
36	manga_036	22	0.7449	0.6809	1098.15	1109.14
37	manga_037	14	0.6972	0.823	1080.27	984.55
38	manga_038	18	0.6161	0.6635	1096.6	969.05
39	manga_039	2	0.3953	0.4367	1098.8	991.89
40	manga_040	31	0.7409	0.8016	1099.64	986.9
41	manga_041	13	0.7436	0.835	1099.01	985.48
42	manga_042	3	0.4725	0.5005	1098.37	979.88

No.	Image ID	Number of Objects	IoU		Inference Time (ms)	
			YOLOv8	YOLOv11	YOLOv8	YOLOv11
43	manga_043	7	0.9173	0.9159	1097.64	982.93
44	manga_044	2	0.9562	0.9506	1259.68	984.02
45	manga_045	12	0.9399	0.8759	1821.04	1044.45
46	manga_046	12	0.9307	0.9279	1702.29	1463.97
47	manga_047	13	0.8846	0.743	1679.96	1514.5
48	manga_048	12	0.8823	0.8258	1094.21	1884.27
49	manga_049	10	0.7595	0.6081	1094.36	1312.65
50	manga_050	15	0.8815	0.8767	1185.43	1049.56
Total / Average		573	0.7196	0.7598	1276.23	1158.72

Based on Table 2, YOLOv11 obtained an average IoU of 0.7598 across the 50 test images, compared with 0.7196 for YOLOv8. This raw average, however, includes two images that contain no ground truth object, where a model that correctly predicts nothing is assigned an IoU of 1.0. Because this conflates localization quality with detection presence, the IoU was recomputed using only the 48 images that contain objects. Under this corrected measure, the gap narrows considerably, as shown in Table 3. The corrected IoU of YOLOv11 (0.7498) and YOLOv8 (0.7287) differ by only 0.0211, and the per-image difference is not statistically significant (Wilcoxon signed-rank, $p = 0.0997$, paired t-test $p = 0.0752$). The two models should therefore be regarded as statistically comparable in localization quality rather than one outperforming the other.

Table 3. Raw and Corrected Mean IoU

Model	Mean IoU (50 images)	Corrected Mean IoU (48 images, GT > 0)	Median IoU
YOLOv8m	0.7196	0.7287	0.7558
YOLOv11m	0.7598	0.7498	0.8006

The two images without ground truth objects are reported separately. YOLOv11 correctly produced no detection on both, whereas YOLOv8 generated one false positive, so the apparent IoU advantage of YOLOv11 on these pages reflects false positive avoidance rather than tighter localization.

3.2 Standard Detection Metrics

Table 4 reports the standard detection metrics for both models. The two architectures are closely matched. YOLOv11 holds a marginal edge in precision, recall-weighted F1-score, and mAP@50, while YOLOv8 is slightly higher in mAP@50-95, indicating marginally tighter localization at strict IoU thresholds. All overall differences are within 0.02, consistent with the statistical test.

Table 4. Overall Detection Metrics

Model	Precision	Recall	F1-score	mAP@50	mAP@50-95
YOLOv8m	0.9027	0.817	0.8577	0.8684	0.6061
YOLOv11m	0.923	0.8119	0.8638	0.8734	0.6006

A per-class breakdown (Table 5) shows that the dominant performance gap is between classes, not between models. The text_bubble class is detected reliably by both models (mAP@50 above 0.98), and clean_text is also handled well (mAP@50 above 0.95). The messy_text class, which

covers expressive text fused with dense background art, is the clear bottleneck, with recall around 0.52 to 0.55 and mAP@50 around 0.67 for both models. On this class YOLOv11 improves precision (0.79 to 0.87) but lowers recall, so the change is a trade-off rather than an overall gain.

Table 5. Per-Class Detection Metrics

Class	Model	Precision	Recall	F1-score	mAP@50	mAP@50-95
clean_text	YOLOv8m	0.9349	0.9318	0.9334	0.956	0.6370
clean_text	YOLOv11m	0.922	0.9399	0.9308	0.9524	0.6252
messy_text	YOLOv8m	0.788	0.5522	0.6494	0.6674	0.4194
messy_text	YOLOv11m	0.8746	0.5205	0.6526	0.6859	0.4281
text_bubble	YOLOv8m	0.9852	0.9669	0.9760	0.9817	0.7618
text_bubble	YOLOv11m	0.9723	0.9752	0.9738	0.9820	0.7485

3.3 Detection Performance Analysis

Based on Figure 4, most images show a higher IoU for YOLOv11 than for YOLOv8. As reported in Table 3, however, this per-image difference is not statistically significant, so the chart reflects a tendency rather than a decisive advantage.

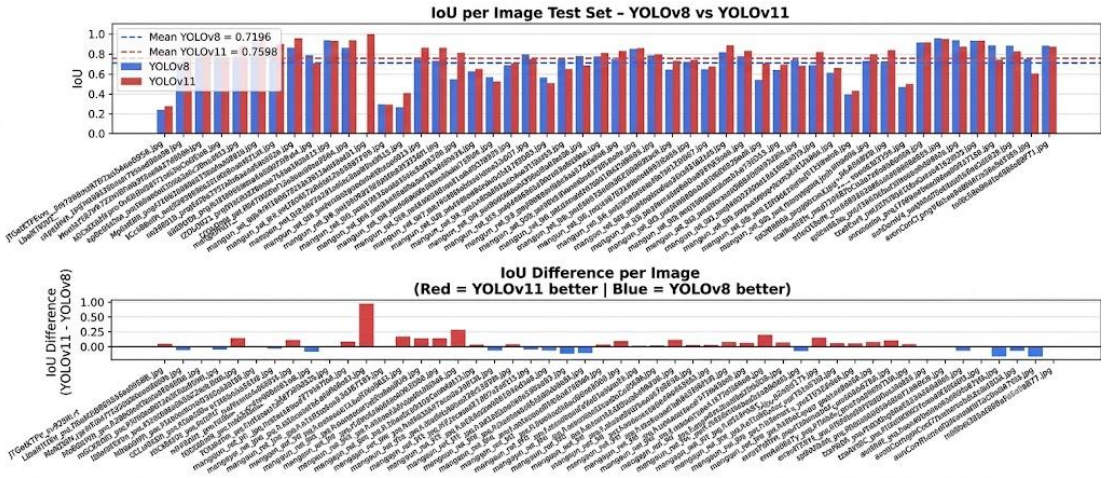

Figure 4. IoU Comparison Diagram of YOLOv8 and YOLOv11

Although the overall metrics are comparable, the C2PSA module (Cross Stage Partial with Spatial Attention) introduced in YOLOv11 appears to influence its behavior on cluttered pages, where spatial attention helps separate text regions from surrounding artwork. This is consistent with the higher precision of YOLOv11 on the messy_text class. The benefit, however, does not translate into a statistically significant overall gain, which suggests that for this dataset the architectural change mainly redistributes the precision and recall balance rather than raising absolute accuracy.

In addition, YOLOv11 showed more stable performance on manga pages with a relatively large number of objects. On several images with more than 10 text objects, the model maintained a high IoU value without a significant performance drop.

3.4 Detection Case Analysis

The visual object detection results of both models are shown in Figure 5, illustrating the detected manga text areas and the corresponding bounding boxes generated by each model. In the figure, the three classes are distinguished by color, namely clean_text, messy_text, and text_bubble, so that the localization behavior of each model can be compared directly against the ground truth. This qualitative inspection complements the quantitative metrics by revealing how each model behaves on individual panels rather than on aggregate scores.

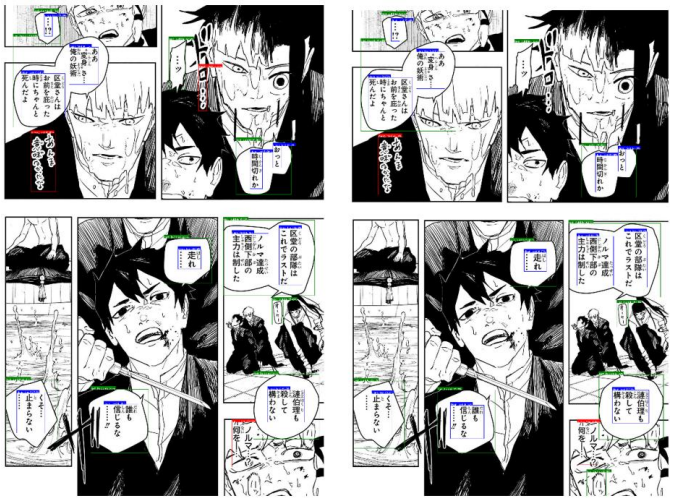

Figure 5. Bounding Box Results of YOLOv8 (Left) and YOLOv11 (Right)

Visually, YOLOv11 produces bounding boxes that are tighter around the text areas than YOLOv8 on many panels, although this visual impression is not backed by a statistically significant difference in IoU.

On images with clear, high-contrast text characteristics, both models produced nearly similar detection results. However, on images with small text, closely spaced text, or complex backgrounds, YOLOv11 tended to produce more precise localization.

Nevertheless, both models still experienced a performance decline on several images of low visual quality. Conditions such as weak contrast, unclear text edges, and illustration elements covering part of the text area were factors that affected detection accuracy.

3.5 Impact on the Manga Translation System

Text area detection quality has a direct effect on the OCR and automatic translation process. The more accurate the bounding box coordinates produced, the higher the likelihood that the OCR system obtains the complete Japanese characters.

Because the two models perform comparably, either is suitable for this stage. The more efficient YOLOv11 is preferred, and its tight localization helps the OCR module receive cleaner text regions. An example of the implementation result is shown in Figure 6.

Figure 6. Translation Results of YOLOv8 (Left) and YOLOv11 (Right)

Overall, the two models deliver comparable detection accuracy, and the principal limitation for the translation pipeline lies in the messy_text class rather than in the choice between YOLOv8

and YOLOv11. Given this parity in accuracy, YOLOv11 is the more practical choice for the detection stage because it reaches the same performance with fewer parameters (20.0M versus 25.8M) and lower computational cost (67.7 versus 78.7 GFLOPs).

4. Conclusions

This study compared YOLOv8m and YOLOv11m for text area detection in Japanese manga using the Intersection over Union (IoU) metric together with standard detection metrics. Testing on 50 manga images containing 573 objects showed that the two models achieve closely comparable accuracy, with an mAP@50 of 0.8684 for YOLOv8m and 0.8734 for YOLOv11m, and F1-scores of 0.8577 and 0.8638, respectively. A paired Wilcoxon signed-rank test on the per-image IoU found no statistically significant difference ($p = 0.0997$), so the two architectures are best reported as statistically comparable rather than one being superior. The analysis also showed that the common practice of assigning a perfect IoU to images without objects inflates the apparent gap. After correcting for this, the IoU difference narrows from 0.0402 to 0.0211. The dominant performance gap is between classes, where text_bubble is detected reliably (mAP@50 above 0.98) while messy_text remains difficult for both models (recall around 0.52 to 0.55).

Two contributions follow from these findings. First, the evaluation highlights a methodological pitfall in IoU averaging on sparse-object pages and reports significance testing that is often omitted in YOLO comparison studies. Second, although accuracy is comparable, YOLOv11m reaches it with fewer parameters (20.0M versus 25.8M) and lower computational cost (67.7 versus 78.7 GFLOPs), which makes it the more efficient option for the detection stage of an OCR-based manga translation system. Future work can focus on enlarging and rebalancing the messy_text class, applying targeted data augmentation, and extending the evaluation to a larger annotated test set so that smaller performance differences can be detected with greater statistical power.

References

Al-Ibrahim, N., Al-Ibrahim, M., & Al-Awadhi, W. (2024). Text Extraction and Recognition in Manga Comics Using Image Processing Techniques. *Science and Information Conference*, 408–418.

Alif, M. A. R. (2024). Yolov11 for vehicle detection: Advancements, performance, and applications in intelligent transportation systems. *ArXiv Preprint ArXiv:2410.22898*.

Auliawan, A. G., & Ratna, M. P. (2024). Strategi Cool Japan yang baru untuk menyelamatkan pertumbuhan ekonomi dan citra populer Jepang di era digital. *KIRYOKU*, *8*(2), 590–604.

Del Gobbo, J., & Matuk Herrera, R. (2020). Unconstrained text detection in manga: A new dataset and baseline. *ECCV 2020 Workshops*.

Ghaffar Nia, N., Kaplanoglu, E., & Nasab, A. (2023). Evaluation of artificial intelligence techniques in disease diagnosis and prediction. *Discover Artificial Intelligence*, *3*(1), 5.

Hidayatullah, P., Syakrani, N., Sholahuddin, M. R., Gelar, T., & Tubagus, R. (2025). Yolov8 to yolo11: A comprehensive architecture in-depth comparative review. *ArXiv Preprint ArXiv:2501.13400*.

Jaiswal, A., Liu, S., Chen, T., Wang, Z., & others. (2023). The emergence of essential sparsity in large pre-trained models: The weights that matter. *Advances in Neural Information Processing Systems*, *36*, 38887–38901.

Li, Y., Yan, H., Li, D., & Wang, H. (2024). Robust miner detection in challenging underground environments: An improved yolov11 approach. *Applied Sciences*, *14*(24), 11700.

Mahadevkar, S. V, Khemani, B., Patil, S., Kotecha, K., Vora, D. R., Abraham, A., & Gabralla, L. A. (2022). A review on machine learning styles in computer vision—techniques and future directions. *Ieee Access*, *10*, 107293–107329.

Mohammed, M., & Alsunosi, R. (2022). Effect of selecting validation dataset on building random forest and decision tree models. *AlQalam Journal of Medical and Applied Sciences*, 470–478.

Muclis, P. A., Kharisma, I. L., & others. (2023). Penerapan Algoritma Backpropagation Untuk Text Recognition Yang Ditranslate Ke Bahasa Daerah. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, *5*(1), 21–32.

Puteri, N. R., & Meirza, A. (2024). Implementasi Metode YOLOV5 dan Tesseract OCR untuk Deteksi Plat Nomor Kendaraan. *Journal of Computer Science and Visual Communication Design*, *9*(1), 424–435.

Reza, R. A., Bintoro, A., Multazam, T., Hasibuan, A., & Badriana, B. (2025). Analysis comparison effect of image capture speed on rice pest detection using yolov 5 and yolov 7. *Journal Geuthee of Engineering and Energy*, *4*(2), 81–91.

Shah, R. (2024). *Global GPU Network (GGN): Performance Benchmarking and Comparative Analysis Against Traditional GPU Computing Models*.

Shia, W.-C., & Ku, T.-H. (2024). Enhancing microcalcification detection in mammography with YOLO-v8 performance and clinical implications. *Diagnostics*, *14*(24), 2875.

Silmina, E. P., Sunardi, S., Yudhana, A., & others. (2025). Comparative Analysis of YOLO Deep Learning Model for Image-Based Beef Freshness Detection. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *11*(1), 250–265.

Sirisha, U., Praveen, S. P., Srinivasu, P. N., Barsocchi, P., & Bhoi, A. K. (2023). Statistical analysis of design aspects of various YOLO-based deep learning models for object detection. *International Journal of Computational Intelligence Systems*, *16*(1), 126.

Stodt, J., Reich, C., & Clarke, N. (2023). Unified intersection over union for explainable artificial intelligence. *Proceedings of SAI Intelligent Systems Conference*, 758–770.

Tjahyanti, L. P. A. S., & Pratama, P. A. (2025). Deteksi Objek Menggunakan Convolutional Neural Network (CNN) pada Pengolahan Citra Digital. *KOMTEKS*, *4*(2), 35–40.

Wu, S., Yang, J., Wang, X., & Li, X. (2022). Iou-balanced loss functions for single-stage object detection. *Pattern Recognition Letters*, *156*, 96–103.

Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing, 126*, 103514.

Zhang, H., Xu, C., & Zhang, S. (2023). Inner-IoU: more effective intersection over union loss with auxiliary bounding box. *ArXiv Preprint ArXiv:2311.02877*.

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object Detection in 20 Years: A Survey. *Proceedings of the IEEE, 111*(3), 257–276.