Vision-Language In-Context Learning Driven Few-Shot Visual Inspection Model

Shiryu Ueno[1][a], Yoshikazu Hayashi[1] and Shunsuke Nakatsuka[1], Yusei Yamada[1],
Hiroaki Aizawa[2][b] and Kunihito Kato[1]

[1] *Faculty of Engineering, Gifu University, 1-1 Yanagido, Gifu, Japan*

[2] *Graduate School of Advanced Science and Engineering, Hiroshima University,
Higashihiroshima, Hiroshima 739-8527, Japan*

{*ueno, hayashi, nakatsuka, yyamada*}*@cv.info.gifu-u.ac.jp, hiroaki-aizawa@hiroshima-u.ac.jp*

Keywords: Visual Inspection, Vision-Language Model, In-Context Learning.

Abstract: We propose general visual inspection model using Vision-Language Model (VLM) with few-shot images of non-defective or defective products, along with explanatory texts that serve as inspection criteria. Although existing VLM exhibit high performance across various tasks, they are not trained on specific tasks such as visual inspection. Thus, we construct a dataset consisting of diverse images of non-defective and defective products collected from the web, along with unified formatted output text, and fine-tune VLM. For new products, our method employs In-Context Learning, which allows the model to perform inspections with an example of non-defective or defective image and the corresponding explanatory texts with visual prompts. This approach eliminates the need to collect a large number of training samples and re-train the model for each product. The experimental results show that our method achieves high performance, with MCC of 0.804 and F1-score of 0.950 on MVTec AD in a one-shot manner. Our code is available at https://github.com/ia-gu/Vision-Language-In-Context-Learning-Driven-Few-Shot-Visual-Inspection-Model.

1 INTRODUCTION

In this study, we propose a method that can detect defective locations in new product images by using Vision-Language Model (VLM) (Yin et al., 2024) (Liu et al., 2024b) and In-Context Learning (ICL) (Dong et al., 2023) (Zong et al., 2024).

With the advancements in deep learning technology, the automation of visual inspection has become increasingly common in recent years. However, current visual inspection models inspect specific products by collecting a large number of images of the target product and training the model. Thus, these models are only applicable to the target products on which they have been trained, and re-training is necessary for new products. Although some methods can inspect multiple products with a single model, they still require hyperparameter tuning or additional training for each product. In this study, we propose a general visual inspection model that leverages VLM and ICL allowing the inspection of new products without any hyperparameter tuning or model training.

Many of the current VLMs (Liu et al., 2024a) (Chen et al., 2023) leverage Large Language Model (LLM) to align visual and language features, demonstrating excellent performance in a wide range of tasks. These tasks range from basic image recognition tasks, such as classification, to advanced vision-language tasks, such as Visual Question Answering (VQA). However, these VLMs are not trained on specific tasks such as visual inspection.

In this study, we propose a general visual inspection model that can detect defective locations in new products without any hyperparameter tuning or model re-training, using VLM and ICL. The framework of our proposed method is shown in Fig. 1. First, we fine-tune the VLM for general visual inspection with a dataset constructed from a diverse set of non-defective and defective product images collected from the web. In this study, we use ViP-LLaVA (Cai et al., 2024), which has been trained on visual prompt recognition, as the foundation of our VLM, and fine-tune it with our dataset. In addition, in typical visual inspection processes by humans, inspectors use inspection standards for the target products. To em-

[a] https://orcid.org/0009-0006-0842-1362
[b] https://orcid.org/0000-0002-6241-3973

Ueno, S., Hayashi, Y., Nakatsuka, S., Yamada, Y., Aizawa, H. and Kato, K.
Vision-Language In-Context Learning Driven Few-Shot Visual Inspection Model.
DOI: 10.5220/0013088100003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 253-260
ISBN: 978-989-758-728-3; ISSN: 2184-4321

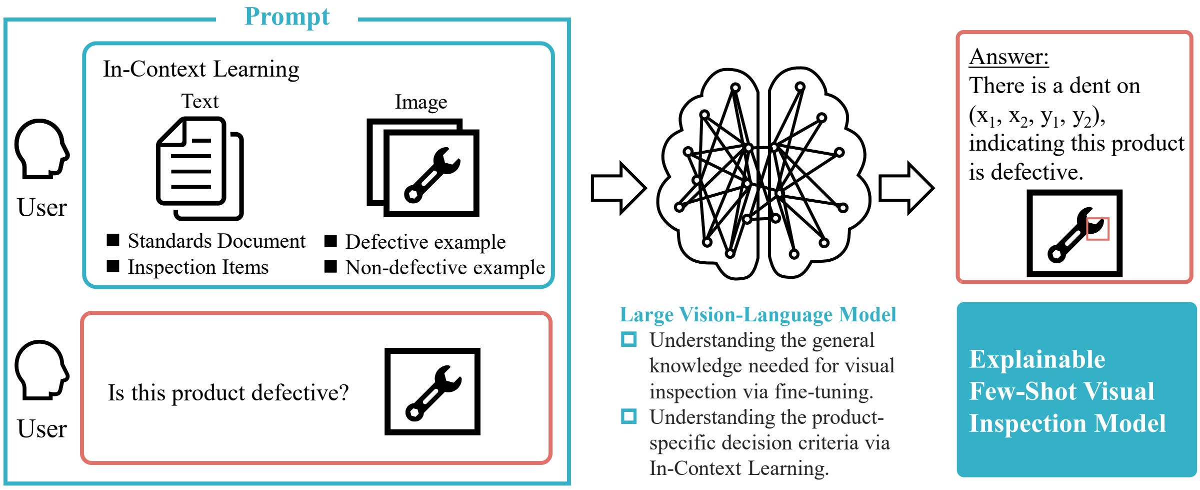

Prompt
User
In-Context Learning
Text
Image
■ Standards Document
■ Inspection Items
■ Defective example
■ Non-defective example
User
Is this product defective?
Large Vision-Language Model
□ Understanding the general knowledge needed for visual inspection via fine-tuning.
□ Understanding the product-specific decision criteria via In-Context Learning.
Answer:
There is a dent on (x_1, x_2, y_1, y_2), indicating this product is defective.
Explainable Few-Shot Visual Inspection Model

Figure 1: Framework of our proposed method. We utilize ICL for multiple image inputs to give VLM the inspection criteria of new products. Our framework gives the coordinates of the defective location, which helps the user understand the model's decision. In addition, it is easy to address by replacing the foundational model when a better VLM is proposed.

ulate this inspection process by human, we use ICL during the evaluation to provide an example of non-defective or defective product image along with explanatory texts that serve as inspection criteria. ICL is a method the model learns from few-shot input-output examples as prompts, without parameter update. Using ICL during the inference of new products, we provide VLM with inspection criteria, enabling specific inspection of target products. Since ICL performance varies significantly based on the provided examples, we propose an algorithm that can select high-quality example based on the distance in Euclidean space. Consequently, our proposed method does not need to collect a large number of images or to re-train the model for each target product.

In summary, our main contributions are:

- We propose a general visual inspection model capable of inspections and detecting defective locations for new products using VLM and ICL with only an example. In our proposed method, fine-tune VLM on visual inspection and utilize ICL enabling the inspection of specific products.
- We construct a new dataset consisting of diverse non-defective and defective products collected from the web, along with unified formatted output, for fine-tuning. Also, our dataset includes coordinates of defective locations for defective products, ensuring the explainability of the model.
- To empirically verify the proposed methodology, we evaluate on MVTec AD (Bergmann et al., 2019) and VisA (Zou et al., 2022). Our method achieves MCC (Chicco Davide and Jurman Giuseppe, 2020) of 0.804 and F1-score (Sokolova et al., 2006) of 0.950 on the MVTec AD dataset in a one-shot manner.

2 RELATED WORK

2.1 Visual Inspection

Many visual inspection methods based on deep learning are trained only on non-defective images (Yi and Yoon, 2020) (Defard et al., 2021). Thus, such methods require the collection of training samples and the re-training of the model for each target product. Consequently, it is challenging to apply the same model to different products without re-training.

Recently, visual inspection methods combining vision and language have been proposed. AnomalyGPT (Gu et al., 2024) can detect defective locations by learning an image decoder from non-defective and pseudo-defective images. However, AnomalyGPT utilizes PaDiM or PatchCore (Roth et al., 2022) for anomaly maps, and these methods need re-training for each products. WinCLIP (Jeong et al., 2023) calculates the similarity between images and texts of non-defective and defective images using CLIP (Radford et al., 2021) and can detect defective locations by using relative anomaly scores. However, WinCLIP only assigns anomaly scores to test samples during inference. To inspect correctly, it is necessary to experimentally determine the optimal threshold on test samples. Thus, these existing approaches cannot be considered general visual inspection models.

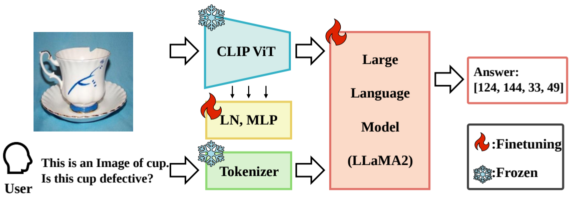


Figure 2: Architecture of ViP-LLaVA. After providing an image and the corresponding text, the image is tokenized by CLIP ViT, LayerNorm, and MLP layers, while the text is tokenized by tokenizer. Then the visual tokens and the text tokens are given to the LLM to generate the answer.

2.2 Vision-Language Model

VLMs leverage LLM to align visual and language features, demonstrating excellent performance across a wide range of tasks, from basic image recognition tasks such as classification, to advanced vision-language tasks, such as VQA. For example, LLaVA (Liu et al., 2023) inputs the vision embedding vectors and language embedding vectors into the LLM decoder to learn the alignment between vision and language. LLaVA has spawned many derivative methods, among which ViP-LLaVA focuses on visual prompt recognition by utilizing a dataset where arrows or visual cues are directly embedded in the input images, thereby strengthening the alignment between low-level image details and language. However, these VLMs have not been trained on visual inspection tasks and thus lack the general knowledge for visual inspection (Liu et al., 2024b).

2.3 In-Context Learning

ICL is a method that the model learns from few-shot input-output examples as prompts, without updating model parameters. For instance, given the input "Example input: (4, 2), Example output: 6, Question: (5, 6)," the model infers from the provided example that the task is addition and can answer "11." In multimodal ICL, the model makes inferences based on images, prompts, and their examples. Many VLMs are trained on diverse image-text pairs, enabling them to acquire ICL capabilities (Chen et al., 2024).

Some VLMs are explicitly built to enhance ICL capabilities. Otter (Li et al., 2023b) enhances ICL capabilities by fine-tuning Open Flamingo (Awadalla et al., 2023) on MIMIC-IT (Li et al., 2023a), which is in an ICL and Instruction Tuning format. At the same time, not to forget the knowledge of Open Flamingo, Otter only update parameters of Perceiver Resampler and Cross Attention Layer in language model. Similarly, LCL (Tai et al., 2023) proposes a new evaluation dataset, ISEKAI, which includes new concepts in the examples, making it challenging without seeing the examples. To address ISEKAI, LCL enhances its ICL capability by fully fine-tuning Shikra (Chen et al., 2023) on a custom dataset based on ImageNet (Deng et al., 2009). However, in practice, these VLM explicitly designed to enhance ICL capabilities do not necessarily outperform regular VLM (Chen et al., 2024) (Zong et al., 2024).

3 PROPOSED METHOD

3.1 Overview

In this study, we propose a general visual inspection model that combines VLM and ICL, enabling the specific inspection of new products without parameter optimization. In addition, by constructing unified output format dataset for fine-tuning, we enable quantitative evaluation of visual inspections using VLM. An overview of the proposed method is shown in Fig. 1.

3.2 Model

In this study, we use ViP-LLaVA (Cai et al., 2024) as the foundational VLM. ViP-LLaVA is a model that improves recognition capabilities for visual prompts by fine-tuning LLaVA 1.5 (Liu et al., 2024a) on a dataset where red circles or arrows are overlaid on the original images. In addition to this, ViP-LLaVA utilizes the multi-level visual features to address the tendency of CLIP's deeper features to overlook low-level details. These methodologies improves the recognition capability for low-level details, which is especially needed for visual inspection. ViP-LLaVA has not been trained on visual inspection tasks.

The model architecture of ViP-LLaVA is shown in Fig. 2. ViP-LLaVA consists of a vision encoder to extract visual features, LayerNorm (Ba et al., 2016) and an MLP to tokenize visual features, a tokenizer to tokenize the language, and an LLM to generate text from these tokens. The vision encoder is CLIP-ViT-L/14(Radford et al., 2021), and the LLM is LLaMA2 (Meta, 2023) During fine-tuning, we update the parameters of the LayerNorm, MLP, and LLM in accordance with the ViP-LLaVA procedure.

3.3 Dataset for Fine-Tune

To enhance the general knowledge of existing VLM for visual inspection, we collect images of non-defective and defective products from the web. The image collection process consists of five main steps:

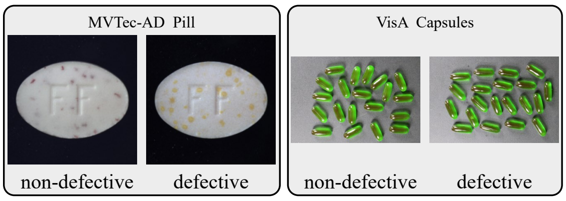


Figure 3: Examples of non-defective and defective images of “Pill” in MVTec AD and “Capsules” in VisA. For “Pill”, the non-defective image also contains red spots, making it difficult to inspect Similarly, for “Capsules”, the non-defective image also contains brown stains.

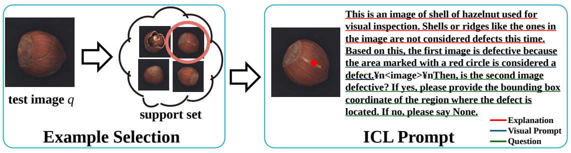
1. Generate product names and inspection-related keywords (e.g., “disk”, “broken disk”, “discolored disk”) by GPT-4 (OpenAI, 2023).
2. Expand the keywords into eight languages: English, Chinese, Spanish, French, Portuguese, German, Italian, and Japanese.
3. Perform image searches using the expanded keywords and collect images with selenium.
4. Remove duplicate or unclear images.
5. Annotate the defective location coordinates for the defective images in the remaining set.

Through this procedure, we collect images of various products. Each product category includes images of non-defective and defective products with up to five types of defects. Finally, we obtained a final set of 941 images of 84 categories.

After collecting the images, we construct a dataset for fine-tuning. The format of the dataset is based on VQA (i.e., a pair of question and answer for each images). Question is “This is an image of {product}. Does this {product} in the image have any defects? If yes, please provide the anomaly mode and the bounding box coordinate of the region where the defect is located. If no, please say None.”, answer is coordinates of the defective location for defective image, “None” for non-defective image.

3.4 In-Context Learning Driven Visual Inspection

It is challenging to inspect new products from a single image. An example is shown in Fig. 3. As shown in Fig. 3, some products need their specific inspection criteria for accurate visual inspection. Thus, in this study, we utilize ICL to provide an example of non-defective or defective image along with explanatory texts that serve as inspection criteria. Based on the example, model precisely inspects new products.



This is an image of shell of hazelnut used for visual inspection. Shells or ridges like the ones in the image are not considered defects this time. Based on this, the first image is defective because the area marked with a red circle is considered a defect.¥n<image>¥nThen, is the second image defective? If yes, please provide the bounding box coordinate of the region where the defect is located. If no, please say None.

Figure 4: Framework of evaluation. First, select the example based on Eq. (1), then infer the test image with ICL.

In addition, in multi-modal ICL, example images significantly influence the output of VLM (Baldassini et al., 2024). RICES (Yang et al., 2022) is an existing algorithm for selecting examples in ICL, it uses cosine similarity of features. However, cosine similarity can yield high values when the scales of features differ or when the feature dimensions are large, failing to accurately evaluate similarity (Steck et al., 2024). Thus, in this study, we propose a new selection algorithm. Our proposed method is shown in Eq. (1).

$$\operatorname{argmin}\left(\min\left\|f(x_i)-f(x_q)\right\|_2\right) \tag{1}$$

Where x denotes the image, f denotes the vision encoder (pre-trained ResNet50 (He et al., 2015)), and q denotes the index of the test image for inference, i denotes the index of the image except for q. Eq. (1) is an algorithm that selects neighboring image of the test image as example based on Euclidean distance. By this, Eq. (1) takes into account the scale and dimensions of the features, and expected to select better example compared to RICES.

4 EXPERIMENT

4.1 Settings

The dataset used for fine-tuning was collected and created as described in Sec. 3.3. We perform one-stage fine-tuning for 300 epochs using ZeRO2 with XTuner (XTuner Contributors, 2023) on 8 NVIDIA 6000Ada-48GB GPUs. The batch size is set to 4, thus the global batch size is set to 32. We utilized the AdamW (Loshchilov and Hutter, 2019) optimizer and 1e-4 learning rate, with the warm-up ratio set to 0.03. We also apply cosine decay to the learning rate.

To evaluate the performance, we used MVTec AD and VisA. These datasets were not used at all during training. During the evaluation, one example image is given by using ICL (i.e., one-shot manner). The method for selecting the example image follows the procedure described in Eq. (1). Also, to evaluate the effectiveness of the proposed example image selection algorithm, we compare its performance without example image and with that of RICES. Finally, framework of the evaluation is shown in Fig. 4.

We use F1-score (Sokolova et al., 2006) and Matthews Correlation Coefficient (MCC) (Chicco

Table 1: Result of MVTec-AD. 'N/A' means that zero division occurred. Bold means the highest performance.

Settings	Vanilla		w/o ICL		ICL (RICES)		ICL (Ours)	
Product Name	F1-score	MCC	F1-score	MCC	F1-score	MCC	F1-score	MCC
Bottle	N/A	N/A	0.863	N/A	0.892	0.510	0.917	**0.610**
Cable	N/A	N/A	0.400	0.338	0.795	0.564	0.899	**0.754**
Capsule	N/A	N/A	0.750	0.426	0.912	0.384	0.946	**0.658**
Carpet	0.044	0.074	1.000	**1.000**	0.983	0.929	1.000	**1.000**
Grid	N/A	N/A	0.973	0.910	0.884	0.476	0.982	**0.935**
Hazelnut	0.228	0.226	0.780	N/A	0.795	0.257	0.800	**0.289**
Leather	0.043	0.076	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**
Metal Nut	N/A	N/A	0.832	0.540	0.912	0.468	0.989	**0.947**
Pill	N/A	N/A	0.838	0.402	0.922	0.368	0.968	**0.814**
Screw	0.209	0.140	0.851	0.244	0.903	0.506	0.925	**0.673**
Tile	N/A	N/A	0.957	0.870	0.977	**0.916**	0.977	**0.916**
Toothbrush	N/A	N/A	0.906	0.633	0.866	0.418	0.921	**0.697**
Transistor	N/A	N/A	0.762	0.592	0.871	0.780	0.894	**0.821**
Wood	N/A	N/A	0.976	0.752	0.992	**0.965**	0.992	**0.965**
Zipper	N/A	N/A	0.741	0.541	0.975	0.879	0.987	**0.941**
All category	0.042	0.068	0.860	0.519	0.917	0.665	0.950	**0.804**

Davide and Jurman Giuseppe, 2020) for the evaluation. F1-score, as shown in Eq. (2) , is a common metric for binary classification.

$$\text{F1-score} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{2}$$

As shown in Eq. (2), F1-score does not use prediction of true negative. Thus, when there is a large number of positive samples during inference, the performance can be significantly inflated by predicting all samples as positive. For instance, in MVTec AD, with 1,258 positive samples and 467 negative samples in the test data, F1-score shows a high value of 0.844. This shows F1-score is suspicious when there is a bias in the test data. Thus, we use not only F1-score but also MCC as evaluation metrics. MCC is reported to be adequate for binary classification, particularly for better consistency and less variance (Grandini et al., 2020) (Gösgens et al., 2022). MCC is shown in Eq. (3).

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \tag{3}$$

MCC ranges from -1 to 1, where 1 indicates perfect prediction of all samples, -1 indicates incorrect prediction of all samples, and 0 indicates random prediction. In the previously mentioned example, MCC cannot be calculated because the denominator becomes zero. Thus, in this study, we use both F1-score and MCC for evaluation. The evaluation methods include assessing the performance for each product within each dataset, as well as the overall performance across the entire dataset.

4.2 Evaluation of Results

4.2.1 Result of MVTec AD

The results for MVTec AD are shown in Tab. 1. The settings are as follows: "Vanilla" for ViP-LLaVA before fine-tune, "w/o ICL" for ViP-LLaVA after fine-tune without using an example during inference, "ICL (RICES)" for using a selected example image with the RICES algorithm during inference, and "ICL (Ours)" for using a selected example image with Eq. (1) during inference. In each settings, results of F1-score and MCC are in a row. From the table, we confirm that providing an example significantly improves performance. This demonstrates the effectiveness of our framework. Additionally, compared to RICES, our selection algorithm achieves improvement in performance with an increase in MCC, demonstrating the effectiveness of our algorithm.

Next, for qualitative evaluation, the visualization of the model prediction is shown in Fig. 5. As shown in the figure, our approach can roughly detect defective locations, which means the model recognizes the defects in the image. However, the model cannot detect multiple defects or logical defects, such as those in "Cable". This is due to the lack of variety in the training dataset. Thus, further image collection and an enlarged training dataset are required for performance improvement.

Also, for some products like "Hazelnut", while our approach improved the performance, it is still insufficient for real-world conditions. For "Hazelnut", the model detected thin parts as defective, indicat-

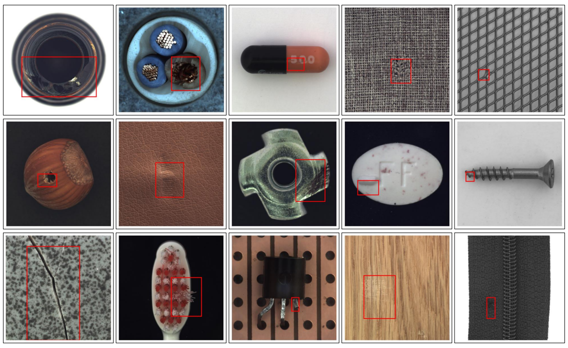


Figure 5: Visualize the model prediction for MVTec AD.

ing that the model does not fully leverage ICL. Thus, providing detailed inspection criteria is necessary. It has been reported that increasing the number of examples improves ICL performance (Agarwal et al., 2024) (Bertsch et al., 2024). Alternatively, further performance improvement is expected by proposing an optimal selection algorithm that selects multiple example images (based on the query strategies, including those from Deep Active Learning (Pengzhen Ren et al., 2021) (Ueno et al., 2023)).

Additionally, for all products, although coordinates are output, their positions deviate from the actual defective locations. Indeed, pixel-level AUROC was 0.730, which is very low compared to the existing methods. This is because the CrossEntropyLoss used for training uniformly calculates the loss for differences in token values. For example, when the ground truth of the starting x-coordinate is 100, the loss is the same when the model outputs 101 and 900 (assuming the prediction probabilities are equal). Thus, CrossEntropyLoss is not optimal for tasks requiring specific numerical outputs like coordinates. However, existing VLMs are trained with CrossEntropyLoss, meaning their outputs are text-based cannot be safely converted to floats (with gradient flow intact), thus performance improvement is expected by constructing a multi-head VLM for defect detection and modifying the loss function to alternatives like Mean Squared Error or GIoU Loss.

While “Bottle” has the same product in the training dataset, their performance is lower compared to “Wood”, which also has the same product in the training dataset. This is likely due to the significant differences in appearance between the images in the training data and those in MVTec AD, as shown in Fig. 6. However, despite the differences in appearance, “Tile” shows high performance, confirming the generalization capability for some products. Also, to prevent the forgetting of knowledge acquired during pre-training when fine-tuning, it is necessary to use Parameter Efficient Fine-Tuning methods, such as Low-Rank Adaptation (Hu et al., 2021), which forget less than fine-tuning (Biderman et al., 2024).

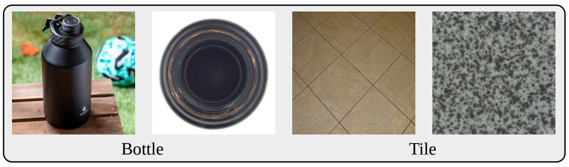


Figure 6: Examples of the images of “Bottle”, and “Tile” from the collected images and MVTec AD.

4.2.2 Result of VisA

The results for VisA are shown in Tab. 2. The table follows the same format as Tab. 1. From the table, it can be confirmed that the performance improves by using ICL in VisA as well, demonstrating the effectiveness of the proposed framework. However, compared to RICES, our selection algorithm does not show significant improvement. This is because both RICES and our selection algorithm are based on similarity, which depends on the data distribution. Most of the products in VisA are too widely distributed (e.g., “Macaroni”, “PCB”). Thus, proposing a more distribution-robust selection algorithm could potentially improve performance. Also, it can be seen that the performance does not improve regardless of the presence of ICL when there are two or more products in the image, especially if those products are not aligned. In fact, “Macaroni1”, which is neatly aligned, shows higher qualitative and quantitative performance compared to “Macaroni2”, which is randomly arranged. This is likely due to the lack of training dataset that considers differences in product positions and orientations. Thus, performance improvement is expected by collecting fine-tuning data and performing data augmentation, such as rotation and flipping. Simultaneously, it should be noted that for some products, positional shifts or orientation differences may be defined as defects.

For qualitative evaluation, the visualization of the model prediction is shown in Fig. 7. As shown in Fig. 7, for products that have multiple objects like “Candle” or “Capsules”, the model prediction gets worse. As mentioned, our dataset is still insufficient for generalization because there are limited products and they are mostly single object. In addition, images with multiple objects are highly distributed compared to the images with single object, which influences the performance of ICL because the selection algorithms depend on the distribution.

Table 2: Result of VisA.

Settings	**Vanilla**		**w/o ICL**		**ICL (RICES)**		**ICL (Ours)**	
Product Name	F1-score	MCC	F1-score	MCC	F1-score	MCC	F1-score	MCC
Candle	N/A	N/A	0.635	**0.539**	0.692	0.241	0.694	0.253
Capsules	N/A	N/A	0.599	0.415	0.841	**0.513**	0.809	0.389
Cashew	N/A	N/A	0.814	0.623	0.890	0.670	0.889	**0.674**
Chewinggum	N/A	N/A	0.921	0.758	0.921	0.758	0.935	**0.804**
Fryum	N/A	N/A	0.867	0.699	0.917	**0.741**	0.888	0.648
Macaroni1	N/A	N/A	0.760	**0.502**	0.685	0.204	0.683	0.190
Macaroni2	N/A	N/A	0.669	**0.071**	0.667	N/A	0.667	N/A
PCB1	N/A	N/A	0.131	0.190	0.891	**0.792**	0.875	0.762
PCB2	N/A	N/A	0.347	0.343	0.772	**0.493**	0.763	0.471
PCB3	N/A	N/A	0.243	0.248	0.747	0.503	0.751	**0.513**
PCB4	N/A	N/A	0.622	0.516	0.801	0.594	0.817	**0.610**
Pipe Fryum	N/A	N/A	0.870	0.726	0.920	0.744	0.929	**0.774**
All category	N/A	N/A	0.671	0.429	0.800	**0.492**	0.795	0.479

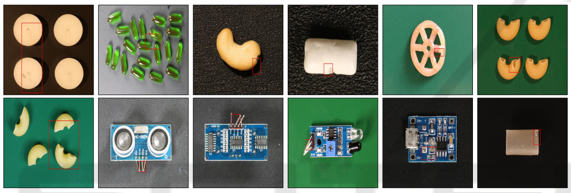


Figure 7: Visualize the model prediction for VisA.

5 CONCLUSION

In this study, we propose a general visual inspection model based on a few images of non-defective or defective products along with explanatory texts serving as inspection criteria. For future work, further performance improvement is expected by collecting more images for fine-tuning. In this study, we enabled visual inspection using VLM by training on a dataset consisting of only 941 images, which is very small compared to the pre-training dataset of VLM. Another consideration is to construct the multi-head VLM and change of the loss function. Furthermore, introducing the example image selection algorithm is another way for improvement. Specifically, existing algorithms are for selecting one example image for the inspection, so proposing an optimal selection algorithm for many example images improves model performance. Finally, the proposed method is based on VLM, so by adding the rationale statements for the decision in the response, model explainability is expected to improve, and performance could be enhanced through multitasking.

REFERENCES

Agarwal, R., Singh, A., Zhang, L. M., Bohnet, B., Rosias, L., Chan, S., Zhang, B., Anand, A., Abbas, Z., Nova, A., Co-Reyes, J. D., Chu, E., Behbahani, F., Faust, A., and Larochelle, H. (2024). Many-Shot In-Context Learning.

Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., Jitsev, J., Kornblith, S., Koh, P. W., Ilharco, G., Wortsman, M., and Schmidt, L. (2023). OpenFlamingo: An Open-Source Framework for Training Large Autoregressive Vision-Language Models.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer Normalization.

Baldassini, F. B., Shukor, M., Cord, M., Soulier, L., and Piwowarski, B. (2024). What Makes Multimodal In-Context Learning Work?

Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. (2019). MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592. IEEE.

Bertsch, A., Ivgi, M., Alon, U., Berant, J., Gormley, M. R., and Neubig, G. (2024). In-Context Learning with Long-Context Models: An In-Depth Exploration.

Biderman, D., Ortiz, J. G., Portes, J., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., Chiley, V., Frankle, J., Blakeney, C., and Cunningham, J. P. (2024). LoRA Learns Less and Forgets Less.

Cai, M., Liu, H., Park, D., Mustikovela, S. K., Meyer, G. P., Chai, Y., and Lee, Y. J. (2024). ViP-LLaVA: Making Large Multimodal Models Understand Arbitrary Visual Prompts.

Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., and Zhao, R. (2023). Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic.

Chen, S., Han, Z., He, B., Buckley, M., Torr, P., Tresp, V., and Gu, J. (2024). Understanding and Improving In-Context Learning on Vision-language Models. arXiv.

Chicco Davide and Jurman Giuseppe (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13.

Defard, T., Setkov, A., Loesch, A., and Audigier, R. (2021). Padim: A patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., Li, L., and Sui, Z. (2023). A Survey on In-context Learning.

Gösgens, M., Zhiyanov, A., Tikhonov, A., and Prokhorenkova, L. (2022). Good Classification Measures and How to Find Them. *neural information processing systems*, 34(17136-17147).

Grandini, M., Bagli, E., and Visani, G. (2020). Metrics for Multi-Class Classification: An Overview.

Gu, Z., Zhu, B., Zhu, G., Chen, Y., Tang, M., and Wang, J. (2024). AnomalyGPT: Detecting Industrial Anomalies Using Large Vision-Language Models. In *AAAI Conference on Artificial Intelligence*, volume 38, pages 1932–1940. arXiv.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models.

Jeong, J., Zou, Y., Kim, T., Zhang, D., Ravichandran, A., and Dabeer, O. (2023). WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19606–19616. arXiv.

Li, B., Zhang, Y., Chen, L., Wang, J., Pu, F., Yang, J., Li, C., and Liu, Z. (2023a). MIMIC-IT: Multi-Modal In-Context Instruction Tuning.

Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., and Liu, Z. (2023b). Otter: A Multi-Modal Model with In-Context Instruction Tuning.

Liu, H., Li, C., Li, Y., and Lee, Y. J. (2024a). Improved Baselines with Visual Instruction Tuning.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. (2023). Visual Instruction Tuning. In *Advances in Neural Information Processing Systems*, volume 36. arXiv.

Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., Chen, K., and Lin, D. (2024b). MMBench: Is Your Multi-modal Model an All-around Player?

Loshchilov, I. and Hutter, F. (2019). Decoupled Weight Decay Regularization.

Meta (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models.

OpenAI (2023). GPT-4 Technical Report.

Pengzhen Ren, Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., and Wang, X. (2021). A Survey of Deep Active Learning. *ACM computing surveys (CSUR)*, 54(9):1–40.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, pages 8748–8763.

Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., and Gehler, P. (2022). Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328.

Sokolova, M., Japkowicz, N., and Szpakowicz, S. (2006). Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In *AI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science*, volume 4304, pages 1015–1021.

Steck, H., Ekanadham, C., and Kallus, N. (2024). Is Cosine-Similarity of Embeddings Really About Similarity? In *Companion Proceedings of the ACM on Web Conference 2024*, pages 887–890.

Tai, Y., Fan, W., Zhang, Z., Zhu, F., Zhao, R., and Liu, Z. (2023). Link-Context Learning for Multimodal LLMs.

Ueno, S., Yamada, Y., Nakatsuka, S., and Kato, K. (2023). Benchmarking of Query Strategies: Towards Future Deep Active Learning.

XTuner Contributors (2023). XTuner: A Toolkit for Efficiently Fine-tuning LLM.

Yang, Z., Gan, Z., Wang, J., Hu, X., Lu, Y., Liu, Z., and Wang, L. (2022). An Empirical Study of GPT-3 for Few-Shot Knowledge-Based VQA. In *AAAI Conference on Artificial Intelligence*, volume 36 of *3*, pages 3081–3089. arXiv.

Yi, J. and Yoon, S. (2020). Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation. In *Asian Conference on Computer Vision (ACCV)*. arXiv.

Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., and Chen, E. (2024). A Survey on Multimodal Large Language Models.

Zong, Y., Bohdal, O., and Hospedales, T. (2024). VL-ICL Bench: The Devil in the Details of Benchmarking Multimodal In-Context Learning.

Zou, Y., Jeong, J., Pemula, L., Zhang, D., and Dabeer, O. (2022). SPot-the-Difference Self-Supervised Pre-training for Anomaly Detection and Segmentation. In *European Conference on Computer Vision*, pages 392–408. arXiv.