A Two-Stage Multi-Modal LLM Fine-Tuning Framework for Analyzing Building Surface Defects

Gengyang Xu
Department of Computer Science
Hong Kong Baptist University
Hong Kong
21253277@life.hkbu.edu.hk

Feng Pan
Department of Computer Science
Hong Kong Baptist University
Hong Kong
fengpan@comp.hkbu.edu.hk

Pong C. Yuen
Department of Computer Science
Hong Kong Baptist University
Hong Kong
pcyuen@comp.hkbu.edu.hk

***Abstract*—Building surface defect detection plays a crucial role in structural health monitoring, ensuring the safety and aesthetics of buildings. Recently, Visual Question Answering (VQA) has been promising in architecture, especially for inspection automation and employee training. However, the insufficient pre-training on architectural knowledge and the limited defect detection accuracy of Large Multi-modal Models (LMMs) result in poor performance in multi-modal building surface defect analysis. Therefore, this paper proposes a two-stage fine-tuning framework for improving LMMs' performance in this task. Experiment results show that our framework significantly enhances the Visual Question Answering performance in the building surface defect analysis. Furthermore, our framework enhances the defect detection accuracy compared to conventional fine-tuning approaches, which leads to more accurate and reliable multi-modal analysis responses from the LMMs.**

***Index Terms*—Computer Vision; Large Multi-Modal Model; Fine-Tuning; Prompt Engineering; Defect Detection**

I. Introduction

As buildings age and experience wear and tear, Structural Health Monitoring (SHM) becomes crucial for ensuring safety and aesthetics. One of the most significant SHM approaches is building surface defect detection.

Building surface defect not only negatively impacts the building itself (e.g., the aesthetic appeal and lifespan) but also raise safety risks to the public (e.g., falling debris injuring pedestrians) [1]. Therefore, timely detection and maintenance of building surface defects are crucial for both preserving structural integrity and ensuring public safety.

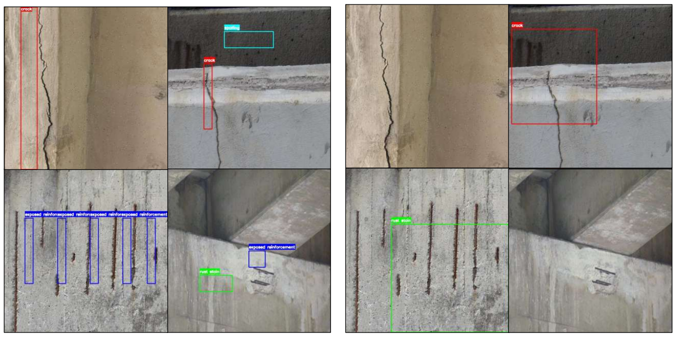
In the past few years, advancements in computer vision provided automated approaches for structural defect detection, particularly those driven by deep learning models [2], [3]. However, it is important to note that deep learning based defect detection methods can only focus on the visual modality [4], [5] and cannot conduct multi-modal analysis or provide knowledge-based insights (e.g., causes, urgency, and repair strategies). Human inspectors derive only limited information from them, which is insufficient for addressing the questions raised in defect assessment reports. Therefore, defect inspection models would be more practical if they supported open-ended visual question answering as a VQA system.

In traditional practices, conducting defect-related analysis often relies on professional engineers' knowledge, experience, and intuition. However, the introduction of the VQA system could transform the landscape. On the one hand, it could enhance the accuracy and efficiency of building surface defect inspection; on the other hand, VQA is user-friendly, allowing novices to conduct defect analysis in complex scenarios [6].

The implementation of VQA often relies on LMMs, which possess impressive zero-shot capabilities. Despite the success of LMMs in general domains, applying LMMs to building surface defect VQA tasks directly presents poor performance due to the existing challenges: 1) LMMs struggle with accurate defect object detection [7]. Figure 1 shows the bounding boxes generated by different LMMs, which either cover only part of the defect or fail to detect any defects. 2) The knowledge base of LMMs is limited by the pre-training datasets, particularly the absence of in-depth knowledge in specialized domains (e.g., Architecture). This leads to LMMs' insufficient knowledge to answer analytical questions about detected defects.

Therefore, this paper proposes a cost-effective two-stage fine-tuning framework for training the domain-specific VQA assistants. Specifically, we endow LMMs with the capability for multi-modal building surface defect analysis. In summary, we make the following contributions:

- We propose a two-stage multi-modal LLM fine-tuning strategy for building surface defect Visual Question Answering. Stage one focuses on defect detection augmentation, which enhances the LMMs' limited object detection capability. In stage two, we concentrate on enhancing the knowledge base and multi-modal analysis capabilities regarding detected building surface defects.
- We collect and annotate a dataset of building surface defects with bounding boxes. Plus, by utilizing knowledge distillation from the proprietary LMM, we construct the first multi-modal instruction-following dataset focused on defect analysis.
- We fine-tune several open-source LMMs under our two-stage fine-tuning framework. The experimental results show that our framework significantly improves the multi-modal defect analytic performance. Furthermore, compared to the conventional fine-tuning method, our framework particularly enhances defect detection accuracy.



(a) Detected by GPT-4o [8]

(b) Detected by LLaVA [9]

Fig. 1: LMMs give deviated defect detection results because of mentioned challenges

II. Related Work

A. Vision-based surface defect detection

Research on vision-based surface defect detection tasks primarily relies on deep learning-based approaches. The common tasks in deep learning-based computer vision include classification, object detection, and segmentation, which address visual data at the image level, object level, and pixel level, respectively. In the context of vision-based building surface defect detection, most research works consider the task as object detection [10]. Cha et al. [11] proposed a faster region-based CNN (Faster R-CNN) for structural visual defect detection of multiple types in quasi-real time. Wei et al. [12] proposed a building facade defect detection method based on the You Only Look Once (YOLO) model, which is renowned for its high speed and accuracy.

B. Knowledge distillation

Knowledge Distillation (KD) is a model compression technique that enhances the performance of a smaller, simpler "student mode" by transferring knowledge from a larger, more complex "teacher model [13]." With the rapid development of open-source LLMs, KD has become a popular technology for transferring in-depth knowledge from leading proprietary LLMs to other open-source LLMs.

The widespread application of knowledge distillation is primarily due to the limitations of proprietary LLMs: 1) **Limited accessibility and high costs.** Proprietary LLMs like GPT-4 [8] and Gemini [14] often require usage fees and have regional access restrictions, making them difficult for individuals to access. 2) **Data privacy and security concerns.** Accessing proprietary LLMs requires sending data to external servers, raising concerns about data privacy and security, especially for users handling sensitive data [15]. 3) **Adaptability limitations.** While proprietary LLMs exhibit strong general capabilities, they may not perform as well as specialized models for specific tasks. Consequently, the limitations in accessibility, cost, privacy, and adaptability are causing the development of model distillation to train more powerful open-source LLM [13].

III. Methodology

A. Two-Stage Fine-Tuning Framework for Multi-modal Defect Analysis

In this work, we propose a two-stage fine-tuning framework to specifically adapt LMMs for multi-modal building surface defect analysis. This framework emphasizes improving defect detection accuracy before achieving improved performance in domain-specific VQA tasks. The motivation is to prevent LMMs from analyzing incorrectly detected or entirely nonexistent defects due to their hallucinations.

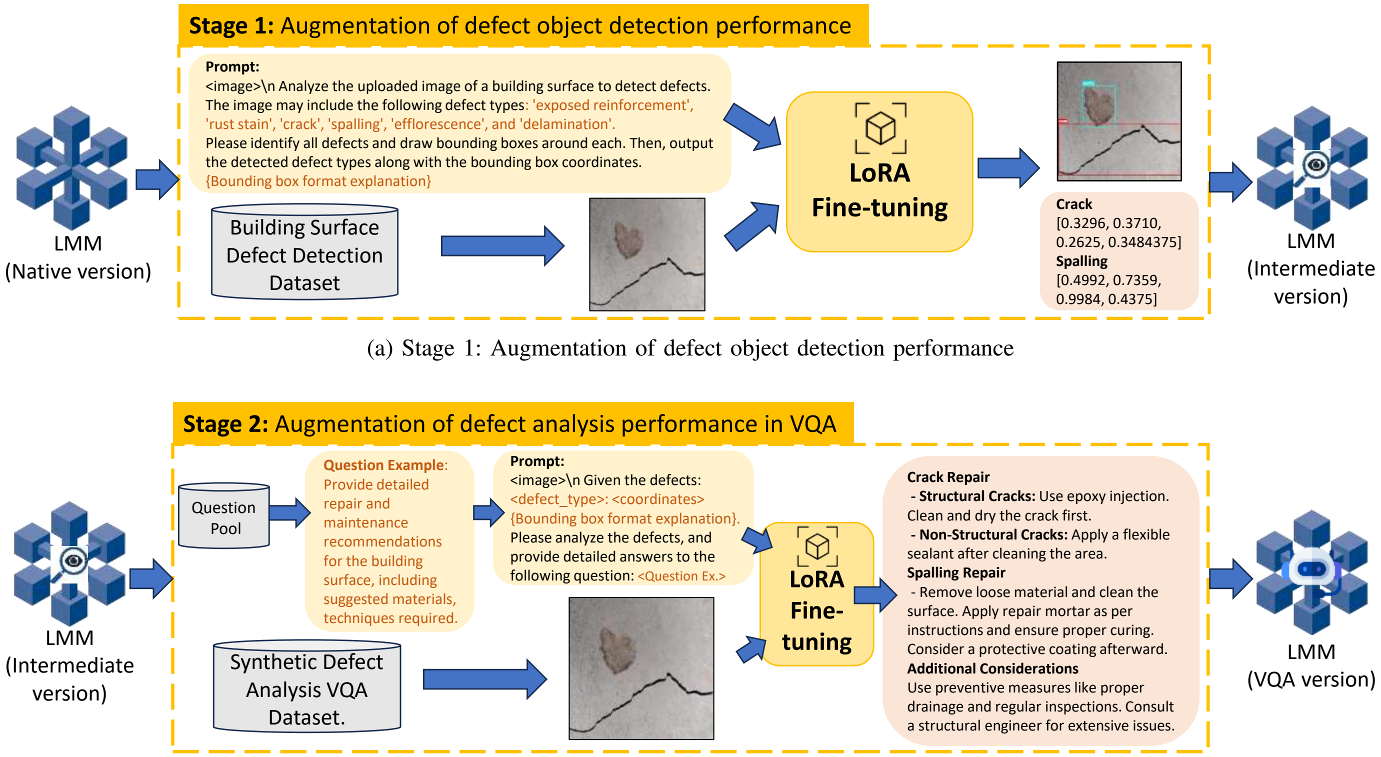
The limited object detection performance of LMMs on atypical objects (e.g., defects) significantly impacts the accuracy of VQA responses. Typically, VQA development involves fine-tuning LMMs using a visual question-answer pairs dataset and performing a one-stage tuning scheme [16]. However, defect detection accuracy is considerably compromised due to the poor object detection capability. Besides, general-purpose LMMs are not adequately equipped to understand specialized defects and their visual features. Therefore, the models trained by typical fine-tuning schemes cannot accurately identify defects and their locations, significantly constraining the credibility of the generated responses.

Our two-stage fine-tuning framework is designed to overcome such limitations. In the first stage, we focus on augmenting the defect object detection performance (as shown in Figure 2a). The prompt-completion pairs used in this stage consist of template questions and corresponding completions regarding the defect types and corresponding bounding box coordinates. In the second stage, we focus on enhancing the model's multi-modal defect analysis capabilities (as shown in Figure 2b). Based on the intermediate fine-tuned model after stage one, which already possesses improved defect detection capabilities, we further fine-tune it to learn expert knowledge related to defects. For the prompt-completion pairs, we design the prompt template containing VQA questions. Furthermore, to generate exemplary VQA answers, we propose a knowledge distillation pipeline to extract knowledge from the proprietary LMM cost-effectively.

This two-stage fine-tuning architecture not only adapts LMMs to support multi-modal defect analysis tasks within the VQA formats but also enhances the accuracy of defect detection for more precise responses.

B. Parameter-Efficient Fine-tuning (PEFT)

Due to the large number of trainable parameters in LMM (for instance, an LMM of 7B contains approximately 7 billion trainable parameters), performing full parameter fine-tuning brings significant computation and time costs. Therefore, to achieve an efficient tuning process within a shorter timeframe and lower computational costs, we employ low-rank adaptation (LoRA) for fine-tuning. The key idea of LoRA is to freeze the pre-trained model weights and inject trainable rank decomposition matrices into each layer of the Transformer architecture [17]. Therefore, Lora significantly reduces the number of trainable parameters, making the training process


Stage 1: Augmentation of defect object detection performance
Prompt:
<image>\n Analyze the uploaded image of a building surface to detect defects.
The image may include the following defect types: 'exposed reinforcement',
'rust stain', 'crack', 'spalling', 'efflorescence', and 'delamination'.
Please identify all defects and draw bounding boxes around each. Then, output
the detected defect types along with the bounding box coordinates.
{Bounding box format explanation}
LMM
(Native version)
Building Surface
Defect Detection
Dataset
LoRA
Fine-tuning
Crack
[0.3296, 0.3710,
0.2625, 0.3484375]
Spalling
[0.4992, 0.7359,
0.9984, 0.4375]
LMM
(Intermediate
version)
Stage 2: Augmentation of defect analysis performance in VQA
Question
Pool
Question Example:
Provide detailed
repair and
maintenance
recommendations
for the building
surface, including
suggested materials,
techniques required.
Prompt:
<image>\n Given the defects:
<defect_type>: <coordinates>
{Bounding box format explanation}.
Please analyze the defects, and
provide detailed answers to the
following question: <Question Ex.>
LMM
(Intermediate
version)
Synthetic Defect
Analysis VQA
Dataset.
LoRA
Fine-
tuning
Crack Repair
- Structural Cracks: Use epoxy injection.
Clean and dry the crack first.
- Non-Structural Cracks: Apply a flexible
sealant after cleaning the area.
Spalling Repair
- Remove loose material and clean the
surface. Apply repair mortar as per
instructions and ensure proper curing.
Consider a protective coating afterward.
Additional Considerations
Use preventive measures like proper
drainage and regular inspections. Consult
a structural engineer for extensive issues.
LMM
(VQA version)

(a) Stage 1: Augmentation of defect object detection performance

(b) Stage 2: Augmentation of defect analysis performance in VQA

Fig. 2: Two-Stage Fine-Tuning framework

more efficient and cost-efficient. In our experiment, we fine-tune LMM using LoRA in a generative manner, as shown in Equation 1:

$$L(\theta) = -\sum_{t=1}^{T} \log P_\theta(y_t|x, y_{<t}) \tag{1}$$

where θ represents the LoRA trainable parameters, T is the output sequence length, and $P_\theta(y_t|x, y_{<t})$ is the probability of the model with parameters θ generating a token y_t given the context x and all previous tokens $y_{<t}$. In this generative approach, the output at time step t is conditioned only on the previous time steps ($< t$).

C. Knowledge Distillation from Proprietary LLM

The lack of visual question-answering datasets regarding building defects has limited the development of the VQA system. To bridge the gap, we propose a knowledge distillation-based pipeline for generating a multi-modal instruction-following dataset from proprietary LMMs and use it for visual instruction tuning.

Before the knowledge distillation process, we need to select a qualified "teacher model." Firstly, by referring to various building surface defect inspection report templates, we design a set of questions that most inspectors are concerned with, forming a question pool. Next, we randomly select questions from the question pool and input them into several proprietary LMMs. Subsequently, we evaluate the generated answers based on industry standards. Ultimately, we selected GPT-4o [8] as the "teacher model" because of the outstanding performance of its responses.

In the knowledge distillation process, we extract the building surface defect image and the corresponding labels first. Subsequently, we select questions from the question pool and add them to the default visual instruction prompts. Then we input both images and prompts into GPT-4o, followed by collecting the generated responses as exemplary answers. An example of this process is illustrated in Figure 3.

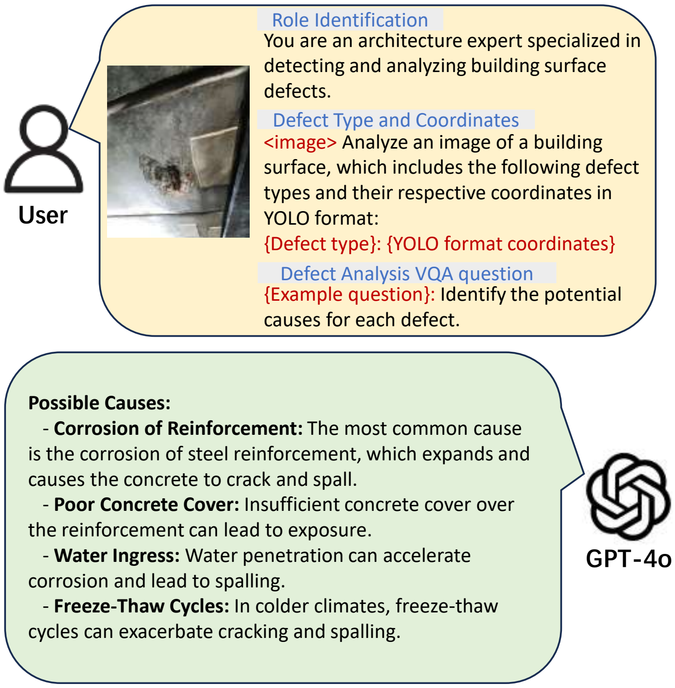
IV. Experiment

A. Experimental Setting

1) Dataset Statistics: Based on the different focuses in the two stages (as discussed in §III-A), we organize two datasets: 1) the Building Surface Defect Detection Dataset and 2) the Multi-modal Defect Analysis VQA Dataset.

Building Surface Defect Detection Dataset. We manually collect and annotate a dataset comprising 7,353 images of building surface defects, each with a size of 640 pixels by 640 pixels. It includes six distinct visible defect types. All images are annotated by qualified building defect inspectors, with labels of the defect types and the corresponding bounding box coordinates in YOLO format. The distribution of the defect types in the dataset is shown in Table I.

In our experiments, we randomly select 80% (5,882 images) of the dataset as the training set and the remaining 20% (1,471 images) as the test set.



Role Identification

You are an architecture expert specialized in detecting and analyzing building surface defects.

Defect Type and Coordinates

<image> Analyze an image of a building surface, which includes the following defect types and their respective coordinates in YOLO format:
{Defect type}: {YOLO format coordinates}

Defect Analysis VQA question

{Example question}: Identify the potential causes for each defect.

Possible Causes:
- **Corrosion of Reinforcement:** The most common cause is the corrosion of steel reinforcement, which expands and causes the concrete to crack and spall.
- **Poor Concrete Cover:** Insufficient concrete cover over the reinforcement can lead to exposure.
- **Water Ingress:** Water penetration can accelerate corrosion and lead to spalling.
- **Freeze-Thaw Cycles:** In colder climates, freeze-thaw cycles can exacerbate cracking and spalling.

Fig. 3: An example of knowledge distillation from GPT-4o [8]

TABLE I: The distribution of six visible defect types in dataset

Defect Types	Number of Images [1]	Proportion
Exposed reinforcement	3309	45.00%
Rust stain	2362	32.12%
Crack	3933	53.49%
Spalling	2568	34.92%
Efflorescence	1848	25.13%
Delamination	746	10.15%
Total	7353	100%

[1] An image can contain multiple defect types.

Multi-modal Defect Analysis VQA Dataset. This Dataset is constructed based on the 5,882 images (i.e., training set) from the Building Surface Defect Detection Dataset. Through knowledge distillation, the "teacher model" GPT-4o generates exemplary answers in five distinct perspective questions for each image. Figure 4 shows the token count distribution for GPT-4o's responses to five perspective questions. Then, we combine the images with GPT-4o's exemplary responses to construct the dataset.

2) Implementation Details: The fine-tuning process utilizes two NVIDIA® GeForce RTX™ 4090 GPUs, each with a memory capacity of 24 GB. For each selected open-source LMM, we employ our designed two-stage fine-tuning framework for one epoch.

3) Baselines: To evaluate the performance of our two-stage fine-tuning framework, we compare the *LMMs tuned by our framework* against *direct prompting of LMMs* and *conventional fine-tuned LMMs*. In our experiment, the open-source LMMs include LLaVA-v1.5-7B [9] and Qwen-VL (Qwen-7B) [18], while the proprietary LMMs include Gemini 1.5 Pro [14], and GPT-4o–2024-08-06 [8].

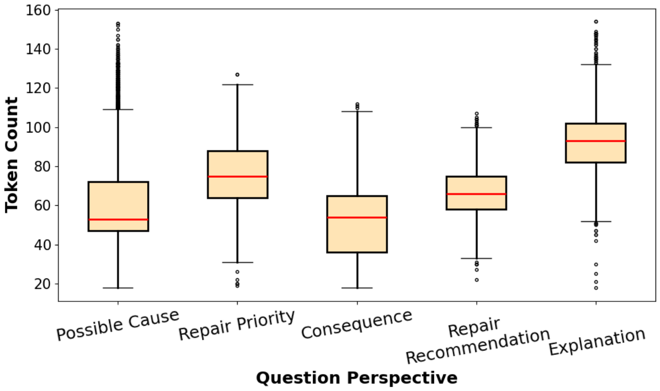


Fig. 4: Distribution of token counts for generating answers to five distinct perspective questions

[Instruction]
Please act as an impartial judge, reference the exemplary answers, and evaluate the quality of the response provided by the AI assistant to the user question displayed below. Your evaluation should consider factors include the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, please rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]"
{Question}: {Exemplary answer}
[The Start of Assistant's Answer]{answer}[The End of Assistant's Answer]

Fig. 5: The default prompt for "observer model" as a judge

4) Evaluation Metrics: Based on the different enhancement targets of the two stages, we use different metrics to evaluate separately:

Defect Object Detection. First, we evaluate the target emphasized in the first stage: the capabilities of defect object detection. It is important because the premise of meaningful visual question-answering is accurately detecting the defect types and locations within an image. To evaluate it, we utilize *Precision*, *Recall*, and *F1-Score* metrics, which are standard in object detection tasks. The evaluation process involves calculating the *Intersection over Union (IoU)*, which is the ratio of the intersection area of two bounding boxes to their union area. This metric measures how closely the predicted results align with the ground truth. In our experiment, we set a predefined IoU threshold of 0.5.

Defect Analysis VQA. To compare the performance in the defect analysis VQA task, we employ an "observer model (Llama 3.1 [19])" as a judge to evaluate the responses by different LMMs. The "observer model" should reference the exemplary answers and assign a score ranging from 1 to 10 to assess the quality of the generated responses, with a higher score indicating superior performance. The default prompt for the judge is shown in Figure 5 [20].

B. Result and Discussion

To evaluate the performance of our framework across different LMMs, we utilize a test set comprising 1,471 images

TABLE II: Performance evaluation in defect object detection, measured by Precision (P), Recall (R), and F1-score (F1).

Models		Direct Prompting			Conventional Fine-tuning			Two-stage Fine-tuning		
		P	R	F1	P	R	F1	P	R	F1
Proprietary LMMs	Gemini 1.5 Pro [14]	4.62%	3.67%	3.96%	N/A			N/A		
	GPT-4o–2024-08-06 [8]	4.86%	3.83%	3.99%						
Open-source LMMs	LLaVA-v1.5-7B [9]	2.33%	1.84%	1.76%	40.67%	34.76%	37.22%	43.42%	**39.80%**	**39.24%**
	Qwen-VL (Qwen-7B) [18]	3.68%	2.66%	2.92%	41.21%	36.11%	37.67%	**44.14%**	36.21%	37.70%

TABLE III: Performance evaluation in defect analysis VQA, measured by score graded by the "observer model"

Models	LLaVA [9]	Qwen [18]	Gemini [14]
Direct Prompting	5.2	6.0	7.8
Conventional Fine-tuning	6.7	7.2	N/A
Two-stage Fine-tuning	7.1	7.5	N/A

that were not seen by any participating LMMs during the fine-tuning process. We follow the evaluation metrics defined in §IV-A4. Furthermore, the metrics calculation involves averaging the metrics across all test samples to compare our selected open-source LMMs with proprietary LMMs under three conditions: 1) direct prompting without fine-tuning, 2) under the conventional fine-tuning approach, and 3) our proposed two-stage fine-tuning framework. The comparison is conducted across two tasks: 1) defect object detection and 2) defect analysis in VQA.

As shown in Table II, the open-source LMMs fine-tuned by our two-stage framework achieve optimal performance in the defect object detection task, surpassing proprietary models such as GPT-4o and Gemini 1.5 Pro, as well as those fine-tuned using the conventional approach. Furthermore, as illustrated in Table III, open-source LMMs fine-tuned by our two-stage framework achieve higher scores compared to direct prompting and conventional fine-tuning approach in the defect analysis VQA task, exhibiting exceptional capabilities in multi-modal defect analysis. Moreover, the two-stage fine-tuned Qwen's average score is comparable to proprietary LMMs Gemini 1.5 pro, which demonstrates their robust multi-modal analysis abilities and in-depth knowledge base.

V. Conclusion

This paper proposes a two-stage fine-tuning framework designed for tuning open-source LMMs specifically for multi-modal building surface defect analysis. Through knowledge distillation and Low-Rank Adaptation fine-tuning, we effectively fine-tuned selected LMMs at a low cost and within a short timeframe. Furthermore, our two-stage design significantly enhances the performance of LMMs in defect detection, which in turn improves the accuracy and credibility of VQA responses. We believe that our work could provide new insights for the development of domain-specific, multi-modal analytic question-answering systems.

References

[1] H. Perez, J. H. Tah, and A. Mosavi, "Deep learning for detecting building defects using convolutional neural networks," *Sensors*, vol. 19, no. 16, p. 3556, 2019.

[2] B. Marc, P. Foucher, F. Forbes, and P. Charbonnier, "Normalizing flows with task-specific pre-training for unsupervised anomaly detection on engineering structures," in *2024 32nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2024, pp. 1937–1941.

[3] K. Lee, G. Hong, L. Sael, S. Lee, and H. Y. Kim, "Multidefectnet: Multi-class defect detection of building façade based on deep convolutional neural network," *Sustainability*, vol. 12, no. 22, p. 9785, 2020.

[4] X. Ye, T. Jin, and C. Yun, "A review on deep learning-based structural health monitoring of civil infrastructures," *Smart Struct. Syst*, vol. 24, no. 5, pp. 567–585, 2019.

[5] S. Sony, K. Dunphy, A. Sadhu, and M. Capretz, "A systematic review of convolutional neural network-based structural condition assessment techniques," *Engineering Structures*, vol. 226, p. 111347, 2021.

[6] T. Yamane, P.-j. Chun, J. Dang, and T. Okatani, "Bridge damage cause estimation using multiple images based on visual question answering," *arXiv preprint arXiv:2302.09208*, 2023.

[7] J. Wang, M. Zhu, Y. Li, H. Li, L. Yang, and W. L. Woo, "Detect2interact: Localizing object key field in visual question answering (vqa) with llms," *IEEE Intelligent Systems*, 2024.

[8] OpenAI, "Gpt-4o release page," 2024. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[9] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024.

[10] J. Guo, Q. Wang, and Y. Li, "Evaluation-oriented façade defects detection using rule-based deep learning method," *Automation in Construction*, vol. 131, p. 103910, 2021.

[11] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.

[12] G. Wei, F. Wan, W. Zhou, C. Xu, Z. Ye, W. Liu, G. Lei, and L. Xu, "Bfd-yolo: A yolov7-based detection method for building façade defects," *Electronics*, vol. 12, no. 17, p. 3612, 2023.

[13] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, "A survey on knowledge distillation of large language models," *arXiv preprint arXiv:2402.13116*, 2024.

[14] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[15] X. Wu, R. Duan, and J. Ni, "Unveiling security, privacy, and ethical concerns of chatgpt," *Journal of Information and Intelligence*, vol. 2, no. 2, pp. 102–115, 2024.

[16] Y. Ding, M. Liu, and X. Luo, "Safety compliance checking of construction behaviors using visual question answering," *Automation in Construction*, vol. 144, p. 104580, 2022.

[17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[18] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond," 2023. [Online]. Available: https://arxiv.org/abs/2308.12966

[19] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[20] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 595–46 623, 2023.