InspectVLM: Unified in Theory, Unreliable in Practice

Conor Wallace* Zeitview
Isaac Corley Zeitview
Jonathan Lwowski Zeitview

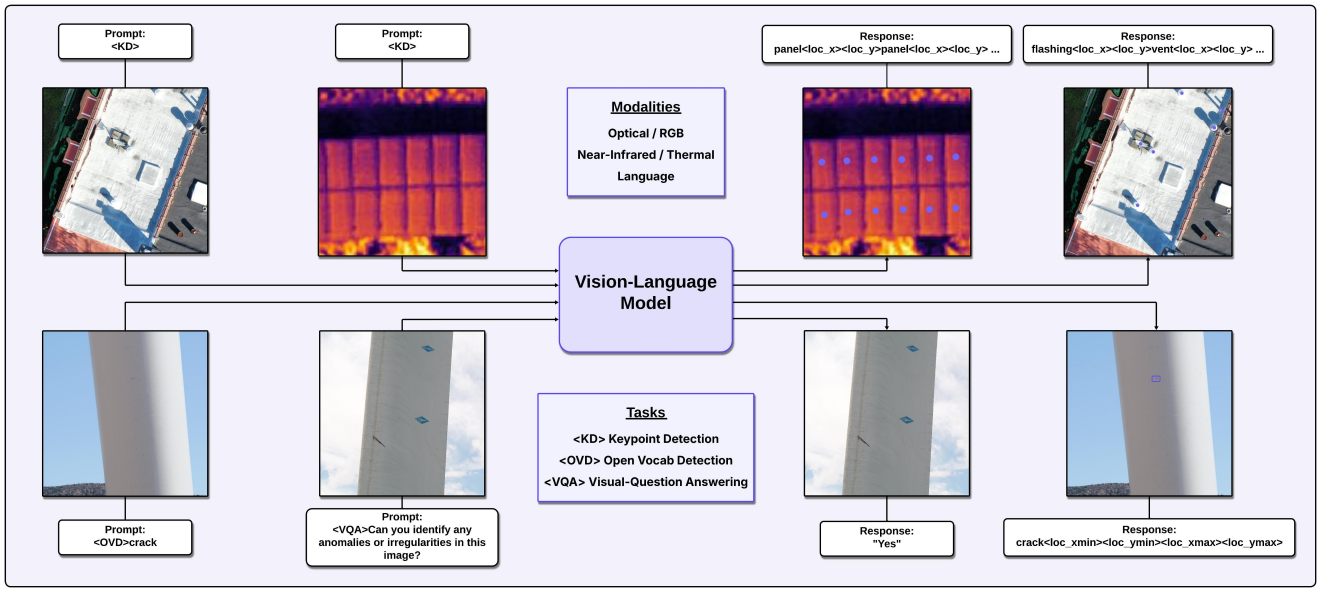


Figure 1. **Our InspectVLM multimodal multitask drone anomaly inspection architecture.** Industrial asset inspections involve multiple modalities (RGB & Thermal imagery) and multiple tasks such as image classification, object detection, and keypoint detection. By treating visual tasks as language, Vision-Language Models (VLMs) enable the unification of independent task-specific models into a single architecture.

Abstract

Unified vision-language models (VLMs) promise to streamline computer vision pipelines by reframing multiple visual tasks—such as classification, detection, and keypoint localization—within a single language-driven interface. This architecture is particularly appealing in industrial inspection, where managing disjoint task-specific models introduces complexity, inefficiency, and maintenance overhead. In this paper, we critically evaluate the viability of this unified paradigm using InspectVLM, a Florence-2–based VLM trained on InspectMM, our new large-scale multimodal, multitask inspection dataset. While InspectVLM performs competitively on image-level classification and structured keypoint tasks, we find that it fails to match traditional ResNet-based models in core inspection metrics. Notably, the model exhibits brittle behavior under low prompt variability, produces degenerate outputs for fine-grained object detection, and frequently defaults to memorized language responses regardless of visual input. Our findings suggest that while language-driven unification offers conceptual elegance, current VLMs lack the visual grounding and robustness necessary for deployment in precision-critical industrial inspections.

1. INTRODUCTION

Large-scale industrial asset inspection—across wind turbines, solar farms, and building rooftops—relies on a range of computer vision tasks, including anomaly detection, object localization, and inventory counting. Traditionally, these tasks are addressed by training and deploying separate models for classification, detection, and keypoint localization, each tailored to a specific data modality and use case. While effective, this approach introduces substantial opera-

*Corresponding authors: conor.wallace@zeitview.com

tional complexity and duplication: each task requires model tuning, deployment infrastructure, and long-term maintenance.

Recent advances in vision-language models (VLMs) offer an alternative. By casting vision tasks into a language interface—e.g., formulating detection as open vocabulary detection, or binary classification as visual-question-answering—VLMs promise to unify these disparate models into a single architecture. This unified approach could significantly simplify inspection systems: one model, one interface, multiple tasks.

However, this promise is largely untested in real-world, high-stakes domains like industrial inspection. Existing VLM evaluations focus on curated web benchmarks or synthetic data, leaving open the question: Can unified VLMs actually replace task-specific vision models in practical inspection pipelines?

In this paper, we present a case study addressing this question. We introduce InspectMM, a large-scale multimodal multitask dataset spanning over 290,000 drone-acquired images with expert-labeled annotations for classification, object detection, and keypoint detection across wind, solar, and property domains. Using this dataset, we train InspectVLM, a Florence-2–based model fine-tuned across all three tasks simultaneously. We then compare its performance to traditional models: ResNet-50 classifiers, Faster R-CNN detectors, and Keypoint R-CNN localizers.

Our findings highlight the trade-offs of unified VLMs in practice:

- InspectVLM performs competitively on image classification and structured keypoint tasks, particularly in solar panel arrays.
- However, it fails significantly on fine-grained object detection, frequently producing degenerate bounding boxes.
- The model exhibits brittle language behavior, overfitting to fixed prompt templates and ignoring visual input under low variability.

These results suggest that while VLMs are architecturally elegant and appealing in theory, current models do not meet the accuracy, reliability, or robustness required for industrial-grade inspection. We conclude that VLMs offer a valuable unification strategy—but only when paired with careful prompt design, adequate visual grounding, and fallback mechanisms for safety-critical applications.

In this work, we explore the effectiveness of VLMs for large-scale asset inspection across a range of tasks and compare their performance to that of traditional task-specific computer vision models. We summarize our contributions below:

Multimodal Multitask Industrial Inspection Dataset We develop a novel multimodal multitask dataset, **InspectMM**, for large-scale asset inspection, encompassing diverse subtasks across multiple image domains and asset types.

A Unified VLM for Multimodal Multitask Inspection We investigate the applicability of VLMs for performing industrial inspection across different asset types and image modalities, resulting in the development of our **InspectVLM** architecture.

A case study in unified model performance We conduct an empirical evaluation of our InspectVLM unified model against traditional computer vision models tailored to individual sub-tasks.

A detailed analysis of VLM failure modes We identify and quantify key limitations of current VLMs in inspection settings, including **(1)** overfitting to low prompt variability, **(2)** defaulting to degenerate bounding boxes, and **(3)** reliance on spatial pattern priors over visual features. These findings highlight the challenges of deploying unified VLMs in high-precision industrial tasks.

1.1. Related Work

VLMs The combination of natural language processing and computer vision architectures into VLMs has progressed rapidly in recent years, showing strong multitasking and generalization abilities. Architectures such as LLaVA [12], MiniGPT-4 [20], InstructBLIP [5], GroundingDINO [13] have shown improved performance for multimodal tasks such as image captioning, visual question answering (VQA), and open-vocabulary detection. Furthermore, VLMs have also been used for visual-grounding tasks wherein the model is capable of understanding features from a referred image location. In a broader scope, general-purpose VLMs such as Florence-2 [19] and PaliGemma [2] are pretrained simultaneously on a combination of multimodal tasks. These advancements indicate that VLMs are a viable alternative to traditional task-specific vision models.

Asset Inspection Traditional deep learning methods have been used to great effect in multiple visual asset inspection tasks, including wind turbine inspection [1], building rooftop measurements [4], and solar farm inventory management [17]. VLMs have also been employed in inspection settings, although they have been limited to primarily single tasks or usage of simulated datasets. For example, AnomalyGPT [8] was trained on industry-specific data with simulated visual anomalies to generate descriptions of present anomalies and generate an approximate location via unsupervised learning using the feature maps generated by the vision encoder. Automotive-LLaVA [11] was proposed for answering questions about automotive part images. Similarly, Power-LLaVA [18] is a VQA model for power line inspections. Furthermore, while VLMs have been applied to industry-specific tasks, they have not been properly evaluated against traditional task-specific vision models.

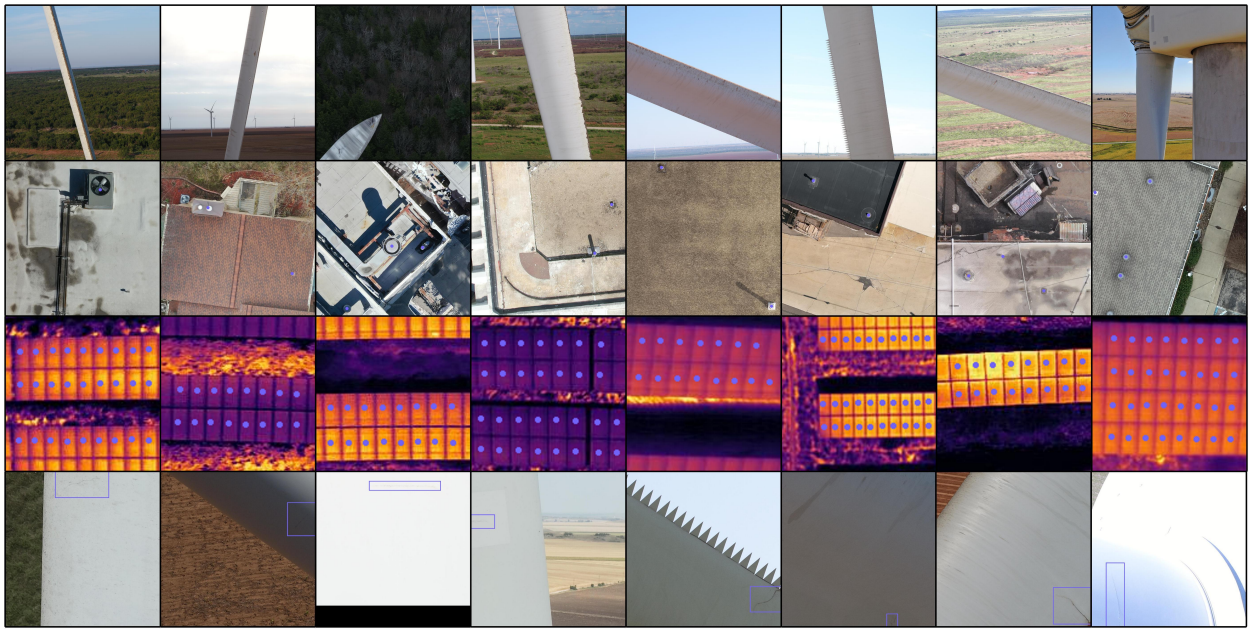


Figure 2. **Samples from our InspectMM dataset for multimodal multitask inspections.** Our dataset task types consist of keypoint detection, visual-question answering, and object detection. From top to bottom: wind turbine anomaly flagging, properties rooftop inventory counting, solar panel inventory counting, and wind turbine crack detection. *Best visualized while zoomed in.*

2. METHODS

Our model follows the Florence-2 architecture [19] consisting of a DaViT [7] image encoder and a unified multimodal transformer for processing both image and text tokens. Florence-2 is trained on multiple tasks including VQA, image captioning, and open vocabulary object detection. While other VLMs commonly train adapter modules to align frozen pretrained text and vision encoders' representations, Florence-2 trains both the vision multimodal encoders from scratch. This allows the model to transfer well to vision-specific tasks. Furthermore, Florence-2 employs an additional specialized tokenizer with 1,000 spatial task tokens allowing the model to encode and decode image locations. These tokens are used to represent image coordinates by normalizing pixel coordinates by image width and height and scaling by 1,000.

2.1. Region Representation

Similarly to the approaches described in Florence-2 [19] and Molmo [6], we represent image regions as language, employing the following spatial encoding formats:

Point Representation Points in the image are expressed as (x_c, y_c), where x_c and y_c represent the centroid coordinates of the object of interest.

Box Representation Bounding boxes are expressed as tuples of length 4: (x_1, y_1, x_2, y_2) corresponding to the top-left and bottom-right corners of the box.

2.2. Task Formulation

To create a unified multitask inspection dataset, we reformulate conventional vision tasks as language, employing and extending task-specific prompts as outlined in [19]. Examples of each task prompt and response can be found in Figure 1:

Binary Classification (VQA) Binary classification tasks are restructured as visual-question answering (VQA) problems. Using the prompt `<VQA>`, the model is provided with a task-specific question, and the response is a binary `yes` or `no`.

Keypoint Detection (Pointing) Inspired by the pointing approach used by Molmo [6], keypoint detection tasks are formulated with the prompt `<KD>`, followed by a task-specific question. The output consists of class names and their corresponding points.

Open-Vocabulary Object Detection Object detection tasks are reformulated as open-vocabulary object detection. The prompt `<OVD>` precedes the target object class of interest, and the response includes the class name and its associated bounding boxes.

3. DATASET

To train a unified model to perform industrial inspections across asset classes, image modalities, and visual tasks, we curate the InspectMM dataset. The dataset consists of

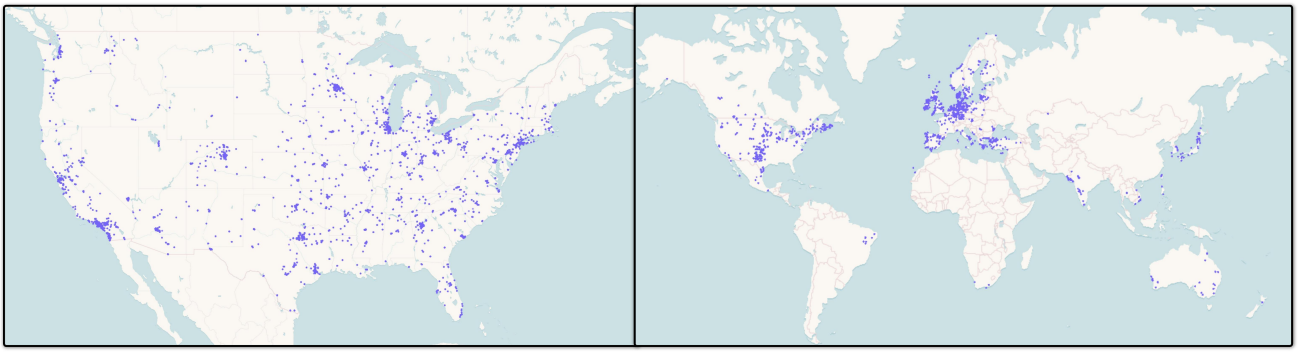


Figure 3. **Diverse geographic locations of the industrial assets in our InspectMM dataset.** InspectMM consists of imagery from inspections of properties & solar assets (left) and wind (right). For properties and solar we sample 97k images from inspections across the continental U.S.; for wind we draw 145k images from from turbine inspections across the globe.

292,341 images taken during aerial drone inspections and 694,905 region-level annotations labeled by industrial inspection experts. The dataset, detailed in Table 1, consists of 3 visual tasks: image classification, object detection, and keypoint detection for 3 asset types: solar, wind, and properties.

Anomaly Flagging This subset of the dataset consists of imagery from wind turbine surfaces and building rooftops. This task requires global-level image understanding of whether an anomaly, which can vary from a large to small area, is present within a high-resolution image. The wind turbine imagery captures a variety of structural surfaces, including blades and hubs, while the buildings imagery spans both commercial and residential structures.

Anomaly Detection For anomaly detection, we use the ZVCD wind turbine crack detection dataset proposed in [1], and improve it by adding bounding box annotations, resulting in the ZVCD+ dataset. In this case, due to the small size of the cracks, the dataset is composed of small 1024×1024 patches of the original image. This adds complexity for a multitask model to be performant on both high-resolution imagery and patches.

Inventory Counting This subset consists of imagery extracted from a mixture of RGB and NIR orthomosaics of building rooftops and solar farms and is annotated with point-level coordinates for identifying and counting components such as HVAC units, vents, skylights, and solar panels. This method allows for efficient identification of objects of varying sizes and orientations without the complexity of detailed bounding box or polygon representations. Keypoint localization simplifies and lessens the annotation cost by focusing on identifying and marking the centroid of each component.

Inspection Task	Asset Type	Annotation Type	# Images	# Annotations
Anomaly Flagging	Properties Wind	CLS	145k	145k
Anomaly Detection	Wind	OD	50k	20k
Inventory Counting	Properties Solar	KD	97k	674k

Table 1. **Overview of the InspectMM dataset for multitask multimodal drone inspections.** Industrial assets include wind turbines, solar farms, and residential and commercial buildings. Inspection tasks include anomaly flagging, anomaly detection, and inventory counting. The tasks can be mapped to the following machine learning problem types, respectively: Image Classification (CLS), Object Detection (OD), and Keypoint Detection (KD).

4. EXPERIMENTS

4.1. Zero-Shot

We initially select the Florence-2 [19], GroundingDINO [13], and PaliGemma [2] models as candidate VLMs. We use the ZVCD+ wind turbine crack detection dataset [1] as our benchmark to evaluate the zero-shot performance of each VLM. As detailed in Table 3, we find that Florence-2 outperforms other VLMs with respect to the number of parameters and IoU. While GroundingDINO provides decent performance, its non-language-based decoder makes it complicated to adapt to multiple tasks other than open-vocabulary object detection. Therefore, we select Florence-2 as our architecture for the following experiments throughout.

4.2. Single & Multitask Evaluation

Experimental Details We train the Florence-2 architecture and initialize the model weights from the original authors' checkpoints [19]. We use the AdamW optimizer [14] with

Task	Type	Model	# Params (M)	Accuracy	Precision	Recall
Anomaly Flagging	CLS	ResNet-50 [9]	24	66.7	62.1	59.3
		InspectVLM (Ours)	232	**75.9**	**73.6**	**89.5**
Anomaly Detection	OD	Faster R-CNN [15]	42	-	**46.1**	**43.7**
		InspectVLM (Ours)	232	-	16.5	19.8
Inventory Counting	KD	Keypoint R-CNN [10]	59	-	**36.6**	**68.9**
		InspectVLM (Ours)	232	-	34.1	60.7

Table 2. **Results across the sub-tasks within our industrial asset inspection dataset, InspectMM.** For Anomaly Flagging classification we report overall accuracy, precision, and recall. For the Anomaly Detection and Inventory Counting we report precision and recall at a 50% IoU threshold. Faster R-CNN and Keypoint R-CNN both use a ResNet-50 FPN backbone. The tasks can be mapped to the following machine learning problem types: Image Classification (CLS), Object Detection (OD), and Keypoint Detection (KD). *Note that InspectVLM is trained for all tasks simultaneously while each model comparison (e.g. ResNet-50) can only be trained on individual tasks.*

a learning rate of $\alpha = 1e-6$, a cosine annealing schedule, mixed-precision training, a batch size of 8, and resize images to 768×768. We train each experiment for 10 epochs.

Method	# Params (B)	IoU	mAP
PaliGemma [3]	3.00	0.00	0.00
GroundingDino [16]	0.31	0.47	**4.48**
Florence-2 [19]	**0.23**	**0.54**	3.11

Table 3. **Zero-shot performance of multimodal language models on the ZVCD test set [1].** We report box IoU and mAP as metrics to evaluate zero-shot performance in addition to model size in parameters.

Baselines For comparison to traditional single-task computer vision models, we use ResNet-50 [9], Faster R-CNN [15] with a ResNet-50 Feature Pyramid Network (FPN) backbone, and Keypoint R-CNN [10] with a ResNet-50 Feature Pyramid Network (FPN) backbone for the Anomaly Flagging, Anomaly Detection, and Inventory Counting experiments, respectively, each initialized with ImageNet pretraining weights.

5. DISCUSSION

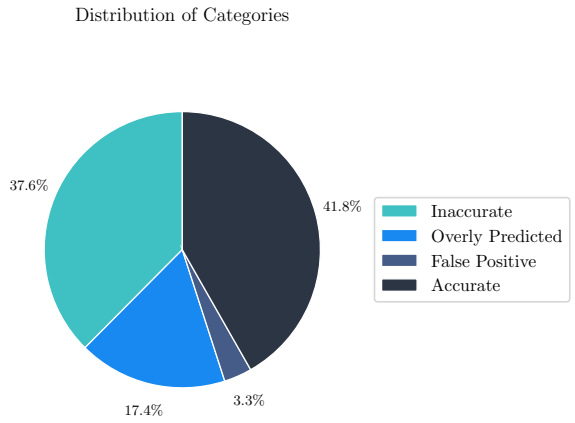
Domain Specific Datasets Due to VLMs being trained on natural images to be general-purpose models, zero-shot performance tends to be inadequate when transferring to domains like industrial inspection, which contain out-of-distribution imagery from the original pretraining set. As a result, their ability to generalize off-the-shelf to tasks such as anomaly detection, component identification, and inventory counting is limited. To address this gap, we find it necessary to construct a large-scale domain-specific dataset with high-quality annotations from human inspectors, like InspectMM, for fine-tuning VLMs through to performing accurate inspections across diverse asset types. Furthermore, we anecdotally find many openly available industrial inspection datasets are inadequate due to poor labeling quality or easily identifiable defects.

Experimental Results The results for the 3 tasks in our InspectMM dataset are presented in Table 2. For Anomaly Flagging, our Florence-2 based InspectVLM significantly outperforms the traditional ResNet-50 classifier by 10%+ precision and recall. For Inventory Counting, Florence-2 and Keypoint R-CNN achieve comparable performance. However, for object detection, we find that the VLM performs significantly worse than the Faster R-CNN model. We intuit that this is due to the nature of the ZVCD+ dataset being difficult with barely visible cracks requiring the extraction of fine-grained visual features which the VLMs haven't become capable of.

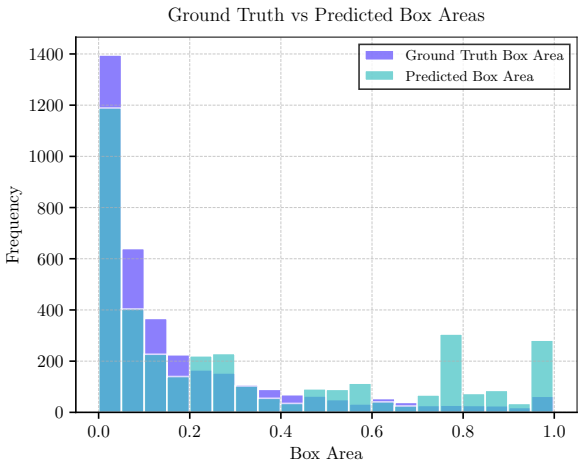
VLMs for Inspections Vision-language models (VLMs) offer a strong platform for multi-task learning, demonstrating versatility across a wide range of image domains and applications. Their ability to quickly adapt to new tasks, such as keypoint detection, makes them an appealing choice for dynamic environments where flexibility is crucial. However, traditional models like ResNet consistently achieve peak performance within their specific tasks, particularly when fine-tuned for specialized applications. This highlights a tradeoff between the benefits of streamlined model architecture and deployment, which VLMs provide, and the superior, task-specific performance that traditional models deliver. While VLMs excel at handling a variety of tasks, traditional models remain a viable choice for achieving high performance in narrowly defined problems. This tradeoff underscores the need for careful consideration when selecting models, depending on whether the goal is broader adaptability or maximum performance in a specialized context.

5.1. Object Detection Failure Modes

While InspectVLM shows modest zero-shot capabilities and performs competitively on classification tasks, its object detection performance is significantly worse than traditional detectors. In this section, we analyze the model's failure modes on the ZVCD+ subset of InspectMM, which requires detecting fine-grained cracks on wind turbine sur-



(a) Distribution of detection quality categories, showing the proportion of accurate detections versus various error types.



(b) Distribution of ground truth vs. predicted normalized bounding box areas.

Figure 4. Object detection performance metrics. The histogram in (a) displays the proportion of different detection quality categories, while (b) shows the distribution of ground truth and predicted bounding box areas normalized by image size.

faces—arguably one of the most visually difficult and safety-critical inspection tasks in the dataset.

Rather than producing a spectrum of plausible detections, InspectVLM exhibits a trichotomy of failure modes: it either predicts inaccurate bounding boxes, overly large degenerate boxes, or hallucinates false defects that do not exist.

5.1.1. Bounding Box Type Categorization

We categorize each prediction from InspectVLM on the ZVCD+ test set into four groups:

- **Accurate**: Predicted IoU ≥ 0.5 with a ground truth crack box.
- **Overly Predicted**: IoU < 0.2 and the predicted box covers more than 30% of the image area.
- **Inaccurate**: IoU < 0.5 and the predicted box covers less than 30% of the image area.
- **False Positive**: Falsely identified anomalies that do not correspond to any ground truth boxes.

Figure 4a shows the distribution of these results across the entire evaluation set. This distribution reflects a failure to achieve reliable localization - InspectVLM either guesses too broadly or fails to respond.

5.1.2. Bounding Box Area Distributions

To better understand this issue, we plot the relative size of predicted boxes normalized by the image size. Figure 4b shows the normalized box area distributions for both InspectVLM predictions and ground truth annotations.

This mismatch confirms that the model often produces bounding boxes that are spatially incoherent, likely as a result of failing to learn proper attention mappings from prompt to image.

Figure 5 shows representative examples of the three failure modes: an accurate detection with a tight bounding box, an overprediction where the model outputs a box spanning the majority of the image, and a missing prediction despite a clearly annotated crack. These examples illustrate that even when visual cues are present, the model either lacks the resolution to localize small cracks or defaults to a fallback decoding strategy that is only weakly grounded in the visual input.

5.1.3. Limits of Visual Grounding in Current VLMs

These object detection failure modes likely stem from several limitations inherent to current VLM architectures like Florence-2: Coarse spatial resolution from tokenized spatial embeddings may limit the model's ability to attend to small visual structures. Decoder fallback behavior may default to predicting a large bounding box when confidence is low or grounding is ambiguous. Lack of multiscale visual processing, as used in traditional detectors (e.g., Feature Pyramid Networks), further weakens performance on small objects.

Moreover, because the ZVCD+ dataset contains real-world turbine surface conditions—variable lighting, textures, and scale—the model's underperformance here highlights the gap between benchmark-style VLM pretraining and real industrial data.

5.2. Structured vs. Unstructured Keypoint Detection

The InspectMM dataset includes keypoint detection tasks for component counting across two domains: solar arrays and building rooftops. These tasks vary not only in object class (e.g., solar panels, HVAC units, vents), but also

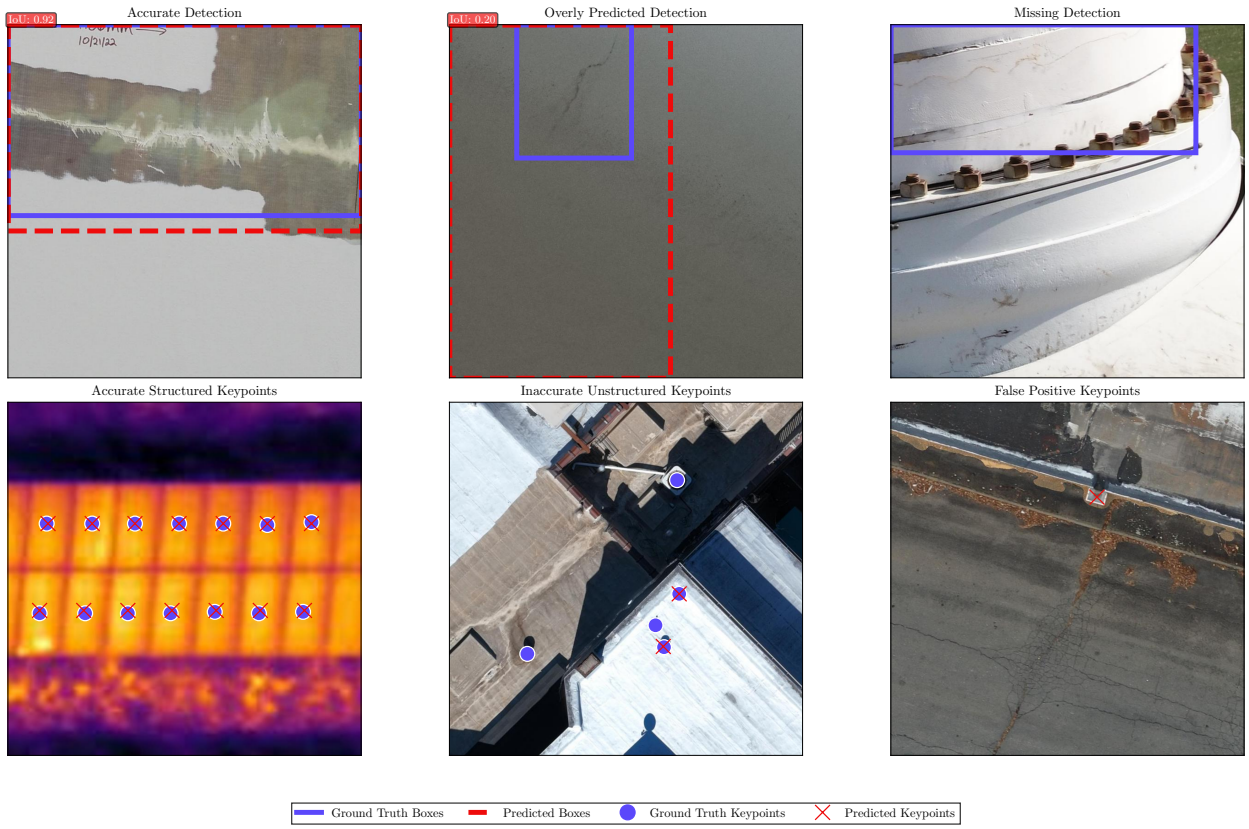


Figure 5. **Sample predictions from our InspectVLM on object and keypoint detection tasks.** Example behavior from the unified inspection model. Top: object detection behavior can be highly accurate (left), over-localized spanning the entire image (center), or misses important defects (right). Bottom: keypoint detection accuracy is largely based on structured data (left), or unstructured data (center and right). *Best visualized while zoomed in.*

in spatial structure. Solar panels are typically arranged in regular, grid-like rows, while rooftop components are irregularly placed and visually diverse.

While the overall performance of InspectVLM trails traditional models, we find that it performs notably better on structured layouts. This suggests that VLMs may leverage spatial priors from pretraining or internal representations to guide keypoint localization in predictable scenes.

5.2.1. Performance by Layout Type

To quantify this, we divide the keypoint validation set into: Structured layouts: Scenes containing solar panels in rows or grids. Unstructured layouts: Scenes with scattered rooftop fixtures.

We evaluate precision and recall at a 20-pixel distance threshold, comparing InspectVLM with a task-specific Keypoint R-CNN baseline. Results are shown in Table 4.

InspectVLM's performance in structured layouts is within 2–3% of Keypoint R-CNN, but in unstructured layouts, it lags by over 20% in both precision and recall.

Figure 5 presents visualizations of InspectVLM predictions for both layout types: a solar array scene with well-aligned predicted keypoints, a rooftop with scattered HVAC units and partial or missed detections, and a rooftop with false positives placed in empty regions.

These examples demonstrate that the model is able to "fill in" keypoints along spatially regular patterns, even in cases of partial occlusion or shadowing. However, when objects do not follow predictable spatial arrangements, the model lacks sufficient visual sensitivity to local textures or edges to correctly localize them.

Layout Type	Model	Precision (%)	Recall (%)
Structured	InspectVLM	94.5	92.2
	Keypoint R-CNN	97.7	91.6
Unstructured	InspectVLM	58.4	48.5
	Keypoint R-CNN	69.1	73.9

Table 4. Keypoint detection performance comparison for structured and unstructured layouts. We report precision and recall at a 10-pixel threshold. InspectVLM performs competitively in structured scenes like solar arrays but significantly underperforms in unstructured rooftop environments.

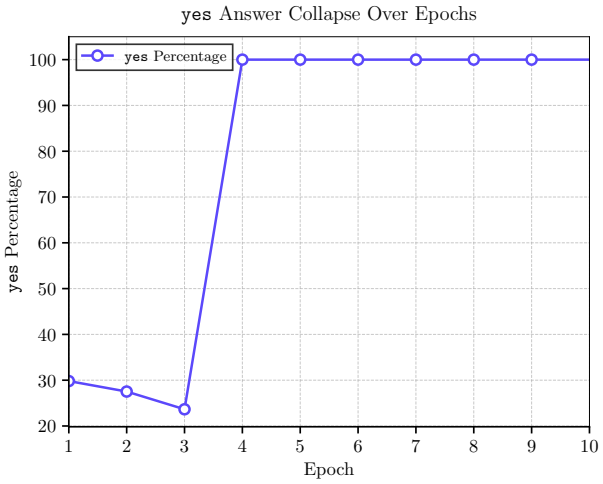


Figure 6. Evolution of `yes` responses across training epochs. The plot shows a clear pattern where the model initially maintains relatively low levels of `yes` responses (around 20-30%) for the first few epochs, but then exhibits a collapse to 100% `yes` responses starting from epoch 4 onwards.

5.3. Overfitting to Low Textual Variability

Despite the promise of VLMs as flexible unified interfaces for vision tasks, we observe a critical failure mode in InspectVLM. The model easily overfits to fixed prompt structures and low-diversity answer spaces. This issue is particularly evident in the binary visual question answering (VQA) formulation of anomaly flagging, where prompts and answers are highly templated. Rather than grounding responses in image content, the model memorizes linguistic patterns and ignores visual evidence—resulting in performance collapse after a few epochs.

5.3.1. Answer Frequency and Collapse

We analyze the distribution of predicted answers across all tasks. Although the ground truth annotations contain a variety of (`yes`, `no`), bounding boxes, and keypoints, we observe that the model response distribution suddenly collapses. As shown in Figure 6, by epoch 4, all responses break down to `yes`, regardless of the ground truth or task grounding. This finding indicates that the model learns to optimize loss by memorizing the dominant label-response mapping, effectively ignoring the image modality.

5.3.2. When Language Overpowers Vision

These findings raise concerns about the core assumption behind treating vision tasks as language: while VLMs offer architectural unification, they can behave as language-only systems when the task setup encourages shortcut learning. In our anomaly flagging task, a combination of: Fixed prompts, Limited answer space, Class imbalance, and Repetitive training examples led the model to disregard its visual input and converge on a degenerate "always-yes" response.

This behavior highlights a serious limitation for applying VLMs in critical industrial contexts. When standardizing prompt formats for deployment (as one would in a production inspection pipeline), we may inadvertently create brittle models that appear accurate on validation data but fail to generalize to even slight variations in prompt phrasing or domain conditions.

We argue that prompt standardization, while desirable for consistency, should be accompanied by: prompt variation during training (e.g., paraphrased prompts), answer diversification (e.g., explanations or references), visual grounding checks (e.g., requiring spatial justifications), and language entropy regularization to penalize degenerate outputs.

Limitations & Future Work Due to the explosion in the number of VLMs being developed each week, it is impossible to compare to state-of-the-art architectures efficiently. Furthermore, we note that VLMs also excel at visual grounding tasks such as referring segmentation or object detection. However, in this paper, we do not evaluate VLMs for these industrial inspection tasks as creating these datasets is costly. We leave both of these for future work.

6. CONCLUSION

In this work, we introduced InspectVLM, a vision-language model designed for multimodal, multitask industrial asset inspections. Leveraging the InspectMM dataset, we demonstrated that VLMs can unify traditionally independent inspection tasks, including anomaly detection, keypoint localization, and inventory management, with a single architecture. Our empirical results highlight the strengths of VLMs in multitask adaptability and language-based task unification, offering a streamlined alternative to maintaining separate task-specific models.

However, our findings also underscore key limitations. While VLMs exhibit competitive performance in anomaly flagging and keypoint detection, they struggle with fine-grained object detection, where specialized models like Faster R-CNN still outperform them. This suggests that while VLMs provide a flexible and scalable framework for industrial inspections, task-specific architectures remain essential for high-precision, domain-specific tasks.

References

[1] Sourav Agrawal, Isaac Corley, Conor Wallace, Clovis Vaughn, and Jonathan Lwowski. Barely-visible surface crack detection for wind turbine sustainability. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5933–5939. IEEE, 2024. 2, 4, 5

[2] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 2, 4

[3] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023. 5

[4] Isaac Corley, Jonathan Lwowski, and Peyman Najafirad. Zrg: A dataset for multimodal 3d residential rooftop understanding. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4635–4643, 2024. 2

[5] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[6] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models, 2024. 3

[7] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *European conference on computer vision*, pages 74–92. Springer, 2022. 3

[8] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(3):1932–1940, 2024. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 5

[11] Aman Kumar, Mahbubul Alam, Ahmed Farahat, Maheshjabu Somineni, and Chetan Gupta. Diagnostics-llava: A visual language model for domain-specific diagnostics of equipment. In *Annual Conference of the PHM Society*, 2024. 2

[12] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, pages 34892–34916. Curran Associates, Inc., 2023. 2

[13] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2, 4

[14] Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5, 2017. 4

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 5

[16] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024. 5

[17] Conor Wallace, Isaac Corley, and Jonathan Lwowski. Solar panel mapping via oriented object detection. In *ICLR 2023 Workshop on Tackling Climate Change with Machine Learning*, 2023. 2

[18] Jiahao Wang, Mingxuan Li, Haichen Luo, Jinguo Zhu, Aijun Yang, Mingzhe Rong, and Xiaohua Wang. Power-llava: Large language and vision assistant for power transmission line inspection. *2024 IEEE International Conference on Image Processing (ICIP)*, page 963–969, 2024. 2

[19] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024. 2, 3, 4, 5

[20] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*, 2024. 2