263-3300-10L Data Science Lab: Lufthansa: Zero-shot damage detection on new aircraft engine parts

Lars Schuster*
Swiss Federal Institute of Technology Zurich (ETH Zurich)
Zurich, Switzerland
lschuster@ethz.ch

Ronan Tanios*
Swiss Federal Institute of Technology Zurich (ETH Zurich)
Zurich, Switzerland
taniosr@ethz.ch

Daniyar Zakarin*
Swiss Federal Institute of Technology Zurich (ETH Zurich)
Zurich, Switzerland
dzakarin@ethz.ch

Shyngys Aitkazinov*
Swiss Federal Institute of Technology Zurich (ETH Zurich)
Zurich, Switzerland
saitkazinov@ethz.ch

Sebastian Graulich
Lufthansa, Germany

Julia Kostin
ETH AI Center, ETH Zurich, Switzerland

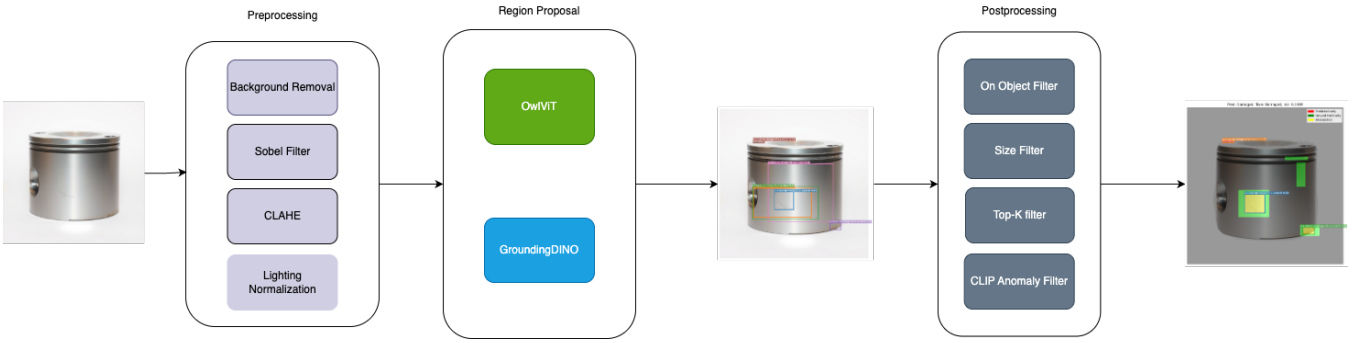

Figure 1: Damage detection pipeline

Abstract

Detecting subtle anomalies—scratches, dents, and nicks—on new aircraft engine parts is critical for safety and quality. However, current approaches typically rely on manual inspection or require extensive labeled datasets. In this project, we explore zero- and few-shot anomaly detection methods to automate this process without requiring labeled training data. Leveraging state-of-the-art vision-language models such as OWL-ViT, CLIP, and Segment Any Anomaly (SAA), we developed an efficient pipeline capable of detecting damage across a variety of aircraft components with minimal supervision. We evaluated our approach on a proprietary dataset of real part images and a synthetic dataset generated using Flux 1.0, demonstrating that OWL-ViT achieved the highest F1-score (0.83) and recall (1.0), outperforming other approaches (See Tab. 1). Our results show that zero-shot detection using foundation models is a promising direction for industrial visual inspection, although challenges remain in reducing overprediction and improving alignment between semantic prompts and visual anomalies.

Keywords

Zero-shot Learning, Anomaly Detection, Computer Vision, Vision-Language Models, Industrial Inspection

*Equal contribution

263-3300-10L Data Science Lab, August 8, 2025, Zurich, Switzerland

1 Introduction

Detecting surface-level anomalies on new aircraft engine parts is a critical task in the aerospace industry, ensuring both safety and quality standards. Even the smallest scratch, dent, or imperfection can render a part unusable due to the extreme speeds and stress conditions that modern jet engines operate under. In addition to

Author contributions using the CRediT framework [2]:
S1: Methodology, Software, Data Curation, Visualization, Investigation
S2: Methodology, Software, Data Curation, Visualization, Investigation
S3: Methodology, Software, Data Curation, Visualization, Investigation
S4: Methodology, Software, Data Curation, Visualization, Investigation
CG: Conceptualization, Supervision
AC: Supervision, Methodology, Project Administration

Table 1: (Image level) Evaluation metrics on the proprietary test set.

Method	F1-score	Accuracy	Precision	Recall	IoU
Finetuned Faster R-CNN	0.82	0.75	**0.77**	0.88	0.02
OWL-ViT	**0.83**	**0.75**	0.73	**1.00**	0.07
SAA (CLIPSeg)	0.75	0.675	0.76	0.74	**0.09**
CLIP	0.80	0.675	0.675	**1.00**	N/A

this, for some parts of aircraft, rigorous legislation prohibits the use of parts that have any detectable anomaly. This makes early detection essential to avoid downstream failure risks and costly operational delays.

Traditional quality assurance workflows are heavily based on manual inspection or models trained on large, labeled datasets, both of which are costly and difficult to scale. Moreover, variations in parts and lighting conditions make conventional computer vision methods brittle in real-world environments.

This project explores the potential of zero- and few-shot vision language models (VLMs) to perform anomaly detection on aircraft components without requiring retraining or access to large annotated datasets. Our aim is to design a deployable, lightweight, and interpretable damage detection pipeline suited for use by non-technical personnel on the shop floor. Ideally the images would be taken in a controlled environment of a photo box that ensures good lighting and a monochromatic background.

This work proposes a framework for anomaly detection built exclusively on open-source models. The pipeline is designed to run on a standard CPU or at most, a small GPU. At the same time, real-time inference is not required. We build on recent advances in foundation models, particularly models like OWL-ViT, CLIP, and SAM, and assess their capabilities in a zero-shot setting. To address the scarcity of publicly available aviation datasets, we use a small proprietary test set and generate synthetic images using the Flux 1.0 [10]generative engine, enabling systematic benchmarking.

Our findings suggest that prompt-based object detectors, such as OWL-ViT, outperform traditional supervised baselines when no real-world labels are available. The results demonstrate that general-purpose models pre-trained on Internet-scale data can transfer surprisingly well to specialized industrial domains like aerospace quality control, especially when integrated in a pipeline that uses various heuristic approaches to improve the quality of the detections.

2 Related Work

Anomaly Segmentation. Traditional deep learning approaches for anomaly segmentation have focused primarily on self-supervised learning, which typically requires large datasets and extensive training. Popular strategies include reconstruction-based methods [20, 21], student-teacher frameworks [6, 17], and distribution modeling techniques [8, 11].

Among training-free methods, memory bank-based approaches [5, 16] are particularly prominent. These methods store patch-level embeddings from normal example images and detect anomalies based on dissimilarity measures. While they typically utilize pre-trained models and require no further fine-tuning, they often depend on a large number of normal samples.

Recent advances in foundation vision-language models have enabled zero-shot anomaly detection via textual prompting [3, 7, 9]. Our work draws primary inspiration from Segment Any Anomaly (SAA) [3], which integrates foundation models such as DINO and SAM with hyper-prompt regularization. SAA achieves zero-shot anomaly segmentation without requiring additional training, demonstrating strong performance on previously unseen anomalies.

Foundation Vision-Language Models. OWL-ViT [13] employs vision transformers guided by natural language prompts to enable zero-shot object detection, facilitating the identification of novel object classes without retraining. Similarly, CLIP [15] offers a powerful image-text similarity framework, supporting open-vocabulary classification and localization through a shared vision-language embedding space.

These developments highlight the growing potential of foundation models for zero-shot defect detection in industrial settings. This serves as a central motivation for our investigation into the capabilities of such models on both proprietary and synthetic datasets.

Common benchmarks used in the field are the anomaly datasets MvTec [1] and VisA[22]. MvTec is a dataset for benchmarking anomalies in industrial applications. Both of the datasets test anomalies in a different domain from the aircraft parts. Therefore, the performance on these benchmarks does not transfer to this problem. Therefore, this work uses its own benchmarks.

3 Methodology

3.1 Synthetic data generation

To estimate the performance of different approaches on a diverse dataset of metal anomalies, a synthetic dataset was generated using diffusion models. The Flux-1.0-dev model [10] was used to generate a set of 150 images of different metal parts, such as rotors, turbines, oil pumps, etc. The prompts for the Flux model were optimized to create the parts in front of a white background while being photorealistic. The data was then manually filtered only to include realistic anomalies. The data was annotated with bounding boxes around the anomalies.

3.2 Damage detection pipeline

To do anomaly detection without training on real data, it turned out best to combine the base-models with a number of steps into a pipeline that extracts the most useful information out of the model output. This also makes the anomaly detection more modular,

allowing it to adapt to different datasets. The pipeline consists of some optional pre-processing steps that enhance the picture and reduce noise. Then the base models detect regions of interest for different anomalies. Finally these regions are post-processed to filter out unlikely predictions.

3.2.1 Pre-processing. As images that are used could come from different settings, the lighting, angles, and quality may change significantly. Therefore, it makes sense to preprocess the images. To enhance the local contrast of the images, we used the Contrast Limited Adaptive Histogram Equalization (CLAHE). Applying CLAHE has previously been shown to increase the capability of models to detect scratches [18]. The filter can be applied separately for the RGB channels or on a grey-scale version of the image. Another filter that can enhance anomaly detection is the Sobel-Filter [19]. It makes edges more visible. This can also lead to an abundance of false positives if the object contains many edges.

3.2.2 Post-processing. In the postprocessing, the outputs of the base models are assessed, and their scores are re-evaluated. For this the base-model or another model is used to segment the image and detect the background and the metal object. Then the bounding boxes, which mostly lie in the background, are removed because they usually find anomalies in the background (**OnObjectFilter**). In addition to this, bounding boxes that are too large are removed (**BBoxSizeFilter**). This reduces the number of false positives. In addition to these heuristic methods there is also the option to use a separate model to confirm the bounding boxes. For this the original image is cropped to the bounding box with some padding, and then the CLIP model is run to compute if the bounding box actually contains the anomaly detected by the base model. Some post-processing steps above are heavily inspired by the SAA [3], however, they were implemented separately for OWL-ViT and OWLv2.

While the pre-processing steps can be used for all models, the SAA already includes much of the post-processing. In the experiments, the pipeline was used for the OWL-ViT and OWLv2 models.

3.3 OWL-ViT

OWL-ViT [13] was evaluated in a zero-shot setting by leveraging natural language prompts to detect damage types such as "scratch," "dent," or "corrosion." The main focus was on prompt engineering and hyperparameter tuning to optimize performance without training data. We experimented with various thresholds for confidence scores, adjusted the number of top-k bounding boxes retained, and tested prompt variations to capture diverse anomaly types. Effective prompting significantly influenced detection accuracy, highlighting the importance of semantic alignment between text queries and visual anomalies.

Additionally, we used the recent OWLv2 [14] model, a self-learning based version of OWL-ViT, which scales the dataset using an existing detector to generate pseudo-box annotations on image-text pairs.

3.4 Finetune Faster R-CNN

To establish a supervised baseline, we manually labeled 70 synthetic images of engine parts using COCO-style bounding box annotations. These annotations included common damage types rendered in synthetic images. We then fine-tuned a pre-trained Faster R-CNN model on this dataset using standard augmentation and a learning rate scheduler. Despite the limited dataset, the model achieved competitive performance, confirming the utility of synthetically generated, labeled data for anomaly detection tasks.

3.5 Segment Any Anomaly

SAA is a framework to detect various kinds of anomalies without needing more training. The general concept of the framework is first to use a region proposal model that generates bounding boxes for potential anomalies. In the next step a region refinement model then creates pixel-level masks based on the bounding box to pinpoint the anomaly. The authors of this method propose the backbones of Grounding-Dino and SAM for the region proposal and the region refinement models respectively. As SAM segments an image only based on a point or bounding box given to it, it is more fit to separate one object from another. It however struggles to separate an anomaly on top of another object, especially if they have very similar color and texture. To avoid this problem, this model uses CLIPSeg [12] instead, which is able to take a textual input and assign scores for each pixel on how much it matches the text. The next step of SAA is then to apply filters based on the size of the anomaly in comparison with the size of the object in the image. The final scores of the mask are then rescaled by a so-called saliency map, which is a measure of self-similarity of the object determined by a convolutional neural network. The final output can be transformed into a binary mask prediction by using a threshold for the anomalies.

4 Experimental Setup

For the experimental setup, different configurations were evaluated on a proprietary test dataset comprising 40 real images. Additionally, selected configurations were tested on a synthetic dataset containing 150 images. We compare the following baseline models on these datasets: OWL-ViT, OWLv2, SAA, and Faster R-CNN. All experiments were run on machines with only CPUs.

4.1 Metrics

To evaluate the proposed methods, we used a combination of object segmentation and localization metrics. At the bounding box level, we employed standard metrics: precision, recall, and Max-F1 score. A defect was considered correctly classified if a predicted bounding box had an Intersection over Union (IoU) of at least 0.1. This relatively low IoU threshold was chosen because the models often predict substantially larger bounding boxes than the actual anomalies.

For segmentation evaluation, we computed pixel-wise anomaly scores and reported the pixel-level Area Under the Receiver Operating Characteristic curve (pixel-AUROC). Additionally, we reported two-class image classification metrics: F1 score, precision, and recall. To avoid ambiguity, we clearly specify whether metrics are reported at the bounding box level or the image level.

Lastly, we calculated the average IoU between the predicted anomaly masks and the ground truth masks.

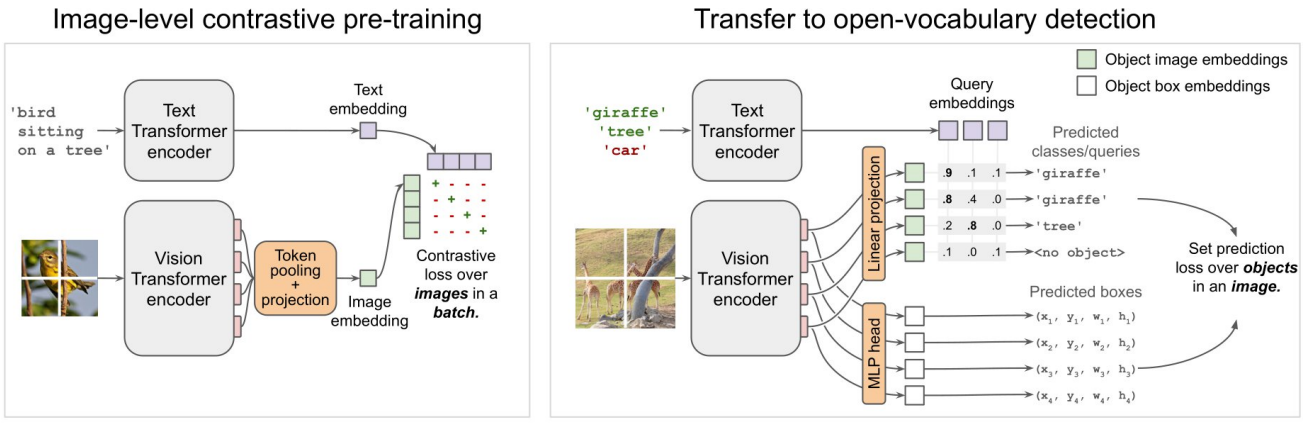

Figure 2: OWlViT architecture.

Model	OnObjectFilter	BBoxSizeFilter	max-F1	Precision	Recall	Pixel-AUROC
OWL-ViT	with	with	**0.58**	0.40	**0.8**	0.62
		w/o	0.46	0.25	0.65	0.61
	w/o	with	**0.58**	0.34	**0.8**	0.59
		w/o	0.41	0.22	0.63	0.61
OWLv2	with	with	0.49	0.27	**0.8**	0.67
		w/o	0.53	0.26	0.78	**0.71**
	w/o	with	0.48	0.26	0.78	0.66
		w/o	0.50	0.26	0.78	0.70
FasterRCNN	w/o	w/o	0.21	0.11	0.4	0.51
SAA	w/o	w/o	0.29	0.875	0.175	0.53

Table 2: (Bounding box- and pixel-level) Performance comparison of OWL-ViT and OWLv2 under various filter configurations evaluated on the real-world dataset.

Model	OnObjectFilter	BBoxSizeFilter	max-F1	Precision	Recall	Pixel-AUROC
Owl-ViT	with	with	0.63	**0.61**	0.64	0.58
		w/o	0.54	0.42	0.67	0.68
	w/o	with	**0.64**	0.57	0.69	0.58
		w/o	0.55	0.35	0.66	0.66
OWLv2	with	with	0.60	0.32	0.90	0.62
		w/o	0.61	0.32	0.90	0.67
	w/o	with	0.60	0.32	**0.91**	0.62
		w/o	0.63	0.31	**0.91**	**0.71**

Table 3: (Bounding box- and pixel-level) Performance comparison of OWL-ViT and OWLv2 under various filter configurations evaluated on the synthetic dataset.

5 Results

We evaluated our zero-shot damage detection methods on a proprietary dataset of 40 real aircraft engine part images (See Tab. 1). The models were assessed using standard metrics: F1-score, accuracy, precision, recall, and Intersection over Union (IoU), where applicable.

Among the methods, OWL-ViT achieved the best overall performance with an F1-score of 0.83, perfect recall (1.0), and an accuracy of 0.75, demonstrating its strong potential for zero-shot industrial

inspection. The fine-tuned Faster R-CNN model closely followed with an F1-score of 0.82 and slightly higher precision (0.77), albeit at the cost of requiring labeled training data.

CLIP-based classification yielded perfect recall but lower precision (0.675), indicating a tendency to overpredict. The Segment Any Anomaly (SAA) method, adapted with a CLIP-based segmentation model, achieved the highest IoU (0.09) but lagged in overall classification performance compared to OWL-ViT.

Visual inspection confirmed that OWL-ViT provided the clearest bounding box detections with high recall, while SAA produced more detailed masks but often lacked semantic accuracy in localizing true damage regions.

In addition to the quantitative results presented above, we provide a qualitative example in Figure 3 to visualize the segmentation output of the selected model. The image shows a synthetic aircraft part alongside the corresponding segmentation mask predicted by the model. This example highlights the model's ability to localize surface anomalies based solely on prompt-guided inference, without requiring task-specific training. As seen in the figure, the predicted mask aligns well with the damaged regions, capturing fine-grained surface variations. However, some noise remains in areas with complex textures, suggesting opportunities for further refinement through prompt tuning or post-processing. Common areas that were difficult to classify for most models were screw-holes, bolts or edges of turbine blades.

In terms of inference time, OWL-ViT is faster than SAA. Depending on postprocessing, OWL-ViT runs for about 20-30 seconds, OWLv2 for approximately 40 second, and SAA for approximately a minute per image.

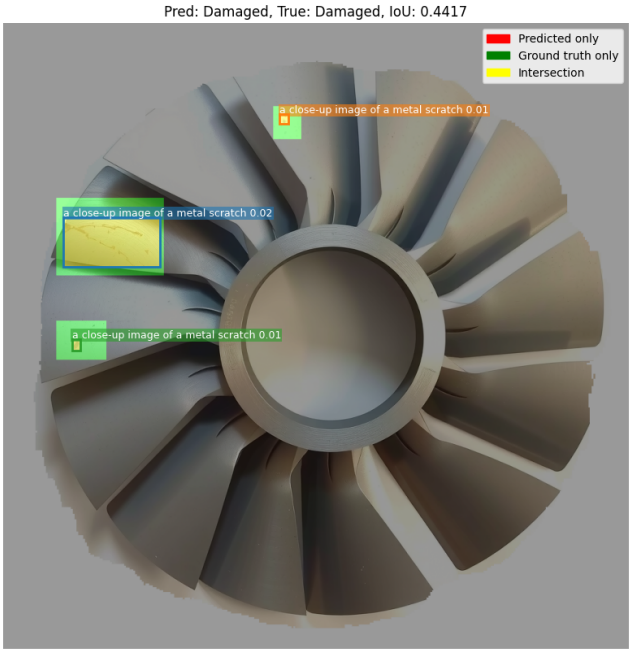

Figure 3: Example segmentation output on a synthetic aircraft part. Using OWLViT model with post-processing

5.1 Postprocessing Ablation

We evaluated OWL-ViT and OWLv2 pipelines using various post-processing configurations on both real-world and synthetic datasets.

Real-world dataset. (See Tab. 2) The best performance in terms of the precision-recall trade-off was achieved by OWL-ViT with both **OnObjectFilter** and **BBoxSizeFilter**, yielding a maximum F1 score of 0.58, with precision of 0.4 and recall of 0.8. Both **OnObjectFilter** and **BBoxSizeFilter** consistently improved performance across all metrics. However, for OWLv2, the impact of these filters was less consistent. Although OWL-ViT outperformed OWLv2 on classification metrics, OWLv2 achieved higher pixel-level AUROC across various post-processing settings. Notably, both OWL-ViT and OWLv2 outperformed Faster R-CNN and SAA across all evaluated metrics.

Synthetic dataset. (See Tab. 3) On the synthetic dataset, OWLv2 significantly outperformed other configurations in terms of Pixel-AUROC and Recall, although it achieved lower Precision.

Overall, the results demonstrate that both OWL-ViT and OWLv2 are effective for anomaly detection at both the bounding box and pixel levels. While OWL-ViT offers a better trade-off between Precision and Recall, OWLv2 delivers superior pixel-level performance and higher Recall. This suggests that with appropriately tuned post-processing, OWLv2 has the potential to surpass OWL-ViT.

5.2 Preprocessing

For image pre-processing, we found that applying different filters generally did not enhance detection performance. In particular, some filters—such as CLAHE—tended to introduce numerous false positives, as they made regular surfaces appear more scratched or textured than they actually were. However, there were cases where the CLAHE filter improved performance, especially in poorly lit environments (See 4) where contrast enhancement is beneficial.

In certain OWLv2 configurations, applying the CLAHE filter led to a notable increase in Recall, which also resulted in a higher F1 score. Nonetheless, this improvement was not consistent across all settings. For the Sobel filter, no configuration seemed to yield better results.

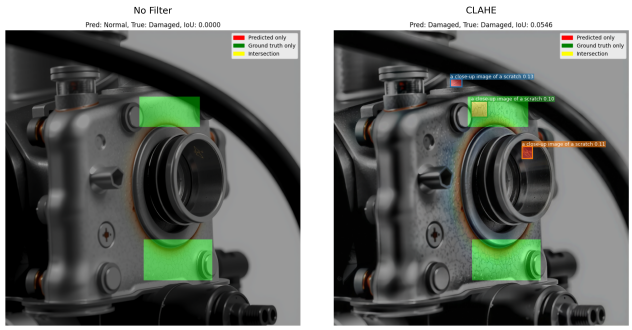

Figure 4: Example of CLAHE filter detecting more anomalies along with additional false positives

6 Discussion

The results underscore the viability of zero-shot learning techniques for anomaly detection in high-stakes industrial applications. OWL-ViT emerged as the most reliable method, balancing accuracy and generalization without requiring retraining, which is critical for deployment in settings with limited or no annotated data. Its reliance on well-crafted prompts and pre-trained vision-language alignment enabled robust generalization to unseen damage types across varied aircraft components.

Interestingly, the CLIP-based approach, despite its strong recall, showed limited precision, suggesting that raw image-text similarity is insufficient for precise localization of subtle anomalies. Likewise, SAA with CLIPSeg demonstrated promise in segmentation detail but occasionally misinterpreted surface textures or benign structures as defects, highlighting the need for better semantic alignment.

The supervised baseline (Faster R-CNN) performed competitively but required synthetic labeled data, which may not always represent real-world conditions. This underscores a common trade-off in industrial computer vision: data-hungry models can outperform zero-shot counterparts under ideal conditions but lack flexibility and scalability.

Overall, the study demonstrates that foundation models like OWL-ViT can significantly reduce development time and data requirements while maintaining high performance. However, challenges remain in balancing precision with recall, improving prompt robustness, and bridging the semantic gap between general-purpose vision models and domain-specific anomalies.

7 Future Work and Conclusion

7.1 Future Work

The output of the models that were used is usually very dependent on the prompt. Therefore, it could be useful to develop an agent framework where the first step would be to pass the model to a large Vision Language Model (VLM), which can then describe the image and trigger different preprocessing steps such as contrast enhancement or light normalization. It could further be used to describe the piece to allow for better segmentation of the object and the background. Finally, the VLM could propose possible anomalies that it sees or that would be likely, which could then be added to the standard prompts for the basemodels. Another approach would be to use an ensemble of base models to propose regions of interest instead of just one. As the base models are not trained on the same datasets, their scores usually do not align. To solve this, normalization techniques could be used or a stacked approach, where a meta-model is trained on the outputs of the ensemble members.

7.1.1 Other Attempted Approaches: WinCLIP and AnomalyDINO. The core of our investigation focused on zero-shot approaches that utilize vision-language models (VLMs) for industrial anomaly detection. While these methods offer the convenience of requiring no training, they rely heavily on precisely describing each possible defect within the model prompt. Moreover, the diverse set of potential visual anomalies may not be sufficiently represented in the model's pretraining data.

To address these limitations, k-shot patch-based methods have been proposed in the literature, such as WinCLIP [9] and AnomalyDINO [4]. These approaches allow for the use of example images of normal instances, from which patch-level embeddings are extracted. Anomalies in test images are then detected based on dissimilarities to these normal patch embeddings. Such methods perform best when the dataset contains multiple examples of the same part.

This assumption presented a challenge in our case, as the synthetic dataset consistently generated slightly different parts, and the real-world dataset included various distinct components with varying backgrounds. Nevertheless, we believe that the flexibility offered by these approaches makes them promising candidates for industrial anomaly detection tasks.

Additionally, we note that patch-based methods such as PatchCore [16] and PaDiM [5] are widely used in the literature. However, they require maintaining a large memory bank of normal images, which may be impractical in some industrial settings.

7.1.2 Unexplored: Multimodal Reasoning Models. Modern multimodal reasoning models demonstrate strong capabilities in image understanding and can be readily applied to anomaly classification tasks. Furthermore, their ability to process large input contexts allows for straightforward incorporation of few-shot examples.

Below is a simple example of querying ChatGPT with an image for reasoning:

Figure 5: Example synthetic image of a scratched turbine

```
User: Describe this image.
[Image: example-image.png]

ChatGPT (Reasoning): The image shows a metallic turbine blade
with visible surface scratches. These scratches may indicate
wear or potential manufacturing defects, suggesting that
the part should be further inspected for anomalies.
```

7.2 Conclusion

In this work, we evaluated several approaches for industrial anomaly detection of airplane engine parts. While the evaluation data is too limited to draw strong conclusions, we conclude that our implemented pipeline with OWL-ViT and OWLv2 background provides a promising solution. We reported an ablation study to confirm the efficacy of post-processing and showed examples of pre-processing benefits. Additionally, we evaluated the SAA [3] and finetuned

FasterRCNN approaches. Lastly, we discussed the main shortcomings and potential improvements to our solution.

References

[1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019. doi: 10.1109/CVPR.2019.00982.

[2] Amy Brand, Liz Allen, Micah Altman, Marjorie Hlava, and Jo Scott. Beyond authorship: Attribution, contribution, collaboration, and credit. *Learned Publishing*, 28(2):2, 2015.

[3] Yunkang Cao and et al. Personalizing vision-language models with hybrid prompts for zero-shot anomaly detection. *IEEE Transactions on Cybernetics*, 55(4), April 2025. ISSN 2168-2275. doi: 10.1109/tcyb.2025.3536165. URL http://dx.doi.org/10.1109/TCYB.2025.3536165.

[4] Simon Damm, Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. Anomaly-dino: Boosting patch-based few-shot anomaly detection with dinov2, 2025. URL https://arxiv.org/abs/2405.14529.

[5] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization, 2020. URL https://arxiv.org/abs/2011.08785.

[6] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9737–9746, June 2022.

[7] Hanqiu Deng, Zhaoxiang Zhang, Jinan Bao, and Xingyu Li. Bootstrap fine-grained vision-language alignment for unified zero-shot anomaly localization, 2024. URL https://arxiv.org/abs/2308.15939.

[8] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 98–107, January 2022.

[9] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation, 2023. URL https://arxiv.org/abs/2303.14814.

[10] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[11] Jiarui Lei, Xiaobo Hu, Yue Wang, and Dong Liu. Pyramidflow: High-resolution defect contrastive localization using pyramid normalizing flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14143–14152, June 2023.

[12] Timo Lüddecke and Alexander S. Ecker. Image segmentation using text and image prompts, 2022. URL https://arxiv.org/abs/2112.10003.

[13] Matthias Minderer and et al. Simple open-vocabulary object detection with vision transformers. *arXiv preprint arXiv:2205.06230*, 2022. URL https://arxiv.org/abs/2205.06230.

[14] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection, 2024. URL https://arxiv.org/abs/2306.09683.

[15] Alec Radford and et al. Learning transferable visual models from natural language supervision. 2021. URL https://arxiv.org/abs/2103.00020.

[16] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection, 2022. URL https://arxiv.org/abs/2106.08265.

[17] Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan T.M. Duong, Chanh D. Tr. Nguyen, and Steven Q. H. Truong. Revisiting reverse distillation for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24511–24520, June 2023.

[18] Jorge Vasquez, Tomotake Furuhata, and Kenji Shimada. Image-enhanced u-net: Optimizing defect detection in window frames for construction quality inspection. *Buildings*, 14(1), 2024. ISSN 2075-5309. doi: 10.3390/buildings14010003. URL https://www.mdpi.com/2075-5309/14/1/3.

[19] Yujia Wang, Tao Yin, Xiaojun Chen, et al. A steel defect detection method based on edge feature extraction via the sobel operator. *Scientific Reports*, 14:27694, 2024. doi: 10.1038/s41598-024-79205-5. URL https://doi.org/10.1038/s41598-024-79205-5.

[20] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem - a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8330–8339, October 2021.

[21] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Dsr – a dual subspace re-projection network for surface anomaly detection. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 539–554, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19821-2.

[22] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. SPot-the-Difference Self-Supervised Pre-training for Anomaly Detection and Segmentation, 2022. Visual Anomaly (VisA) dataset accessed on DATE from https://registry.opendata.aws/visa.