AssemAI: Interpretable Image-Based Anomaly Detection for Manufacturing Pipelines

Renjith Prasad*
Artificial Intelligence Institute
College of Engineering and Computing
University of South Carolina
Columbia, SC 29208, USA
kaipplir@mailbox.sc.edu

Chathurangi Shyalika*
Artificial Intelligence Institute
College of Engineering and Computing
University of South Carolina
Columbia, SC 29208, USA
jayakodc@email.sc.edu

Fadi El Kalach
McNair Center
for Aerospace Innovation and Research
Department of Mechanical Engineering
College of Engineering and Computing
University of South Carolina
Columbia, SC 29208, USA
elkalach@email.sc.edu

Revathy Venkataramanan
Artificial Intelligence Institute
College of Engineering and Computing
University of South Carolina
Columbia, SC 29208, USA
revathy@email.sc.edu

Ramtin Zand
Intelligent Circuits, Architectures
and Systems Lab
College of Engineering and Computing
University of South Carolina
Columbia, SC 29208, USA
ramtin@cse.sc.edu

Ramy Harik
CU-ICAR
Clemson University
College of Engineering and Computing
Greenville, SC 29607, USA
harik@clemson.edu

Amit Sheth
Artificial Intelligence Institute
College of Engineering and Computing
University of South Carolina
Columbia, SC 29208, USA
amit@sc.edu

***Abstract*—Anomaly detection in manufacturing pipelines remains a critical challenge, intensified by the complexity and variability of industrial environments. This paper introduces AssemAI, an interpretable image-based anomaly detection system tailored for smart manufacturing pipelines. Utilizing a curated image dataset from an industry-focused rocket assembly pipeline, we address the challenge of imbalanced image data and demonstrate the importance of image-based methods in anomaly detection. Our primary contributions include deriving an image dataset, fine-tuning an object detection model YOLO-FF, and implementing a custom anomaly detection model for assembly pipelines. The proposed approach leverages domain knowledge in data preparation, model development and reasoning. We implement several anomaly detection models on the derived image dataset, including a Convolutional Neural Network, Vision Transformer (ViT), and pre-trained versions of these models. Additionally, we incorporate explainability techniques at both user and model levels, utilizing ontology for user-level explanations and SCORE-CAM for in-depth feature and model analysis. Finally, the best-performing anomaly detection model and YOLO-FF are deployed in a real-time setting. Our results include ablation studies on the baselines and a comprehensive evaluation of the proposed system. This work highlights the broader impact of advanced image-based anomaly detection in enhancing the reliability and efficiency of smart manufacturing processes. The image dataset, codes to reproduce the results and additional experiments are available at https://github.com/renjithk4/AssemAI.**

***Index Terms*—Smart Manufacturing, Object Detection, Image Processing, Anomaly Detection, Interpretability**

*These authors contributed equally to this work.

I. Introduction

The evolution of manufacturing has been driven by distinct technological milestones. Initially, mechanization marked the first industrial revolution, followed by the mass production techniques of the second revolution. The third revolution introduced automation and computerization, significantly enhancing operational efficiency. The modern smart manufacturing paradigm emphasizes the utilization of data and advanced analytics to inform decision-making, thereby optimizing productivity and efficiency [1]. The incorporation of the Internet of Things (IoT), Artificial Intelligence (AI) and other advanced technologies plays a pivotal role in this transformation, revolutionizing manufacturing processes and systems [2].

Anomaly detection is an essential methodology in manufacturing to identify deviations from normal production processes, which can indicate potential issues such as equipment failures, defects, or inefficiencies. Anomaly detection helps ensure product quality, reduce downtime, and optimize production processes [3] [4]. By monitoring various process parameters

and product characteristics, manufacturers can detect anomalies early and prevent costly production disruptions [5].

Image data are increasingly being leveraged in manufacturing systems due to advancements in computer vision and the availability of high-resolution cameras. In modern manufacturing facilities, image data are utilized for a wide range of applications, including quality inspection [6], [7], predictive maintenance [8], [9], process monitoring [10] and process optimization [11]. The integration of image data allows for real-time monitoring and anomaly detection, significantly improving the ability to identify defects and streamline production processes. Specifically, image-based anomaly detection has gained prominence, leveraging visual data to identify defects or irregularities in products. Automated visual inspection systems use image data to detect surface defects, assembly errors and deviations from design specifications [12], providing real-time feedback essential for maintaining high production standards and minimizing defects [4].

Despite the advancements in image-based anomaly detection, several challenges persist as follows: (i) In visual data, the distinctions between normal and anomalous conditions are often subtle, making it difficult for the model to differentiate these anomalies from typical image variations (ii) The need for large and diverse datasets for training can be resource-intensive, both in terms of data collection and annotation [5], [13]. (iii) Generalization issues that arise when a model trained in one manufacturing environment fails to adapt to others due to differences in lighting, camera angles, or product types [14]. (iv) Difficulty in interpreting detection results, hindering the ability to provide actionable insights for process improvement [15]. To address these challenges, this work presents the following contributions:

1) A new image dataset for anomaly detection in manufacturing, derived using a zero-shot object detection model OWL-ViT.
2) A novel anomaly detection model that leverages the pre-trained EfficientNet architecture [16], fine-tuned on the derived dataset for enhanced anomaly detection in assembly pipelines.
3) Integration of user-level explainability using process ontology and model-level explainability using SCORE-CAM for salient feature and model analysis,

II. Literature Review

A. Object Detection Models

Object detection has advanced significantly, with various models enhancing both accuracy and efficiency. Girshick et al. [17] introduced R-CNN, which applies CNNs to region proposals for object classification and bounding box refinement. Faster R-CNN [18] built on R-CNN by integrating a Region Proposal Network (RPN) for faster detection. The YOLO models, from YOLOv1 to YOLOv9, transformed object detection by formulating it as a single regression problem to predict bounding boxes and class probabilities directly [19]. ShuffleNet [20] and MobileNet [21] are lightweight models designed for efficiency on mobile devices. SqueezeNet [22] aimed to reduce model size while maintaining accuracy with its Fire module. Lastly, the Swin Transformer [23] introduced hierarchical transformers with shifted windows, setting new benchmarks for object detection and vision tasks. Despite these advancements, limitations such as the need for extensive labeled data and computational resources persist. Our work addresses these limitations by focusing on lightweight and efficient models tailored for anomaly detection in manufacturing settings, improving both detection speed and accuracy while maintaining interpretability.

B. Zero-Shot Object Detection

Zero-shot object detection (ZSD) recognizes objects without labeled training data but faces challenges like semantic noise and class imbalance. Foundational methods by Bansal et al. [24] and Rahman et al. [25] struggled with noise. Gupta et al. [26] and Zheng et al. [27] improved ZSD with symmetric mapping and cascade stages, but issues persisted. Hayat et al. [28] and Li et al. [29] used ResNet, KNN, and contextual information but faced noise and ambiguity. Zhu et al. [30] and Hayat et al. [31] advanced ZSD with Don't Even Look Once (DELO) model and Generative-ZSD. Liu et al. [32] and Li et al. [33] proposed contrastive learning and a semantics-aware framework. Our work focuses on specialized domains for precise anomaly detection.

C. Anomaly detection

Anomaly detection has seen significant advancements across diverse domains like networking [34], smart agriculture [35], healthcare [36], manufacturing [37]–[43]. Haselmann et al. [44] present an unsupervised one-class learning method using a deep CNN for surface inspection, outperforming other methods on decorated plastic parts. Xie et al. [37] propose a uniform benchmark for assessing image anomaly detection (IAD) algorithms in industrial settings. Maggipinto et al. [38] use convolutional autoencoders for monitoring semiconductor manufacturing, enhancing effectiveness and scalability. Jiang et al. [39] introduce YOLOv3 for balanced datasets and Fast-AnoGAN for unbalanced datasets in industrial production. Tan et al. [40] use an encoder-decoder for anomaly detection in sequential sensor data. Kim et al. [41] propose a self-supervised method using Gramian angular field and StyleGAN for time-series data. Bougaham et al. [42] demonstrate a three-step deep learning approach for Printed Circuit Board Assembly (PCBA) images, achieving high accuracy. Despite advancements, publicly available datasets for assembly processes are scarce, and existing methods lack interpretability and explainability for domain-specific insights.

III. Proposed Methodology

Figure 1 shows the overall architecture of the proposed method. The AssemAI pipeline begins with dataset preparation, where images are filtered and cropped to focus on relevant features. We then use a zero-shot object detection model to get the region of interest for the rocket in the image, deriving a dataset with masked regions of interest and

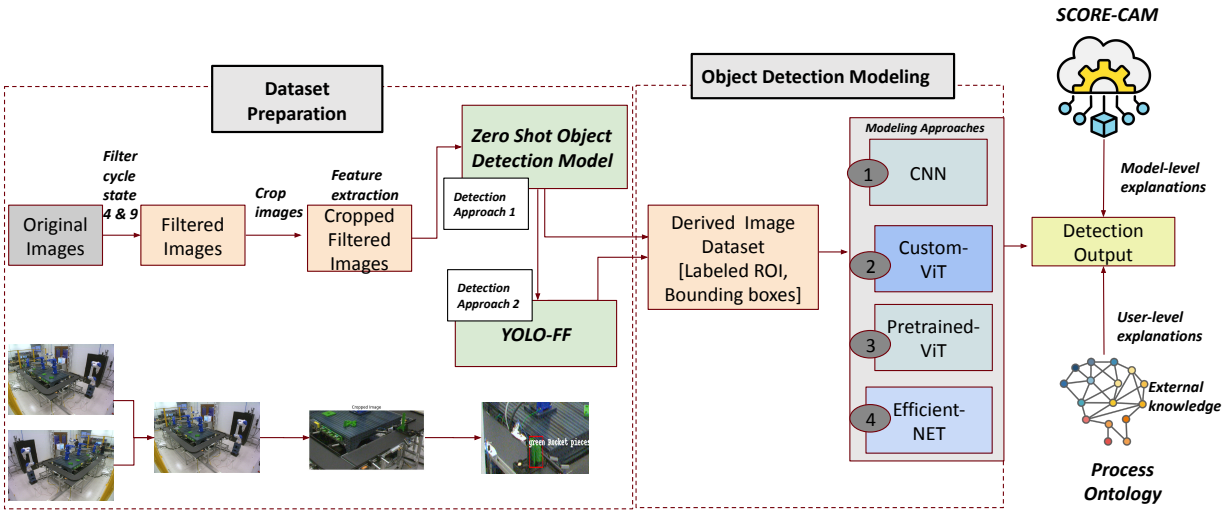

Fig. 1. Overall Architecture of AssemAI. The figure illustrates the AssemAI pipeline, beginning with dataset preparation, which includes filtering and cropping images. Next, object detection is performed using zero-shot detection and a fine-tuned model (YOLO-FF). This is followed by anomaly detection using CNN, Custom-ViT, Pre-trained-ViT, and EfficientNet. The detection output is explained using SCORE-CAM for model-level explanations and process ontology for user-level explanations.

bounding box details in a CSV file. A YOLO model is then fine-tuned on these images and bounding boxes. Subsequently, this derived dataset is used for anomaly detection experiments with several architectures, including CNN, Custom ViT, Pre-trained ViT, and EfficientNet.

TABLE I
ARTIFICATS OF THE ORIGINAL IMAGES IN FF MULTIMODAL DATASET

Dataset Artifact	Statistic
Rarity Percentage	35.73%
Frequency	0.367 Hz
Data collection period	30 hours
Total image count	332002
Original image size	720px*1080px
Types of anomalies	7
Types of classes	8

The detection output is then explained using SCORE-CAM for model-level explanations and integrated with process ontology for user-level explanations, providing technical and domain-specific insights.

A. Future Factories Multimodal Dataset

We use the Future Factories (FF) dataset [45] curated and publicly released by the Future Factories lab at the McNair Aerospace Research Center, University of South Carolina. The dataset consists of measurements from a simulation of a rocket assembly pipeline, which adheres to industrial standards in deploying actuators, control mechanisms and transducers[1]. The dataset comprises two versions; analog and multimodal dataset, for which we use the images included in the multimodal dataset in our study. Table I shows the statistics of the original images in the FF multimodal dataset.

[1]Refer to the images available in the GitHub repository for further details.

B. Image filtering

The rocket assembly process at the FF lab is divided into 21 distinct cycle states. Information about these cell cycle states is not directly available in the multimodal dataset and had to be extracted from the analog dataset using a mapping function provided by domain experts. By calculating the Structural Similarity Index (SSIM) between the normal and anomalous images as shown in Figure 2, we observed that the images are structurally very similar across most cycle states. This indicates that the differences between the images are subtle and localized, which suggests that further analysis is needed to filter and figure out the region of interest. Among these cell cycle states, the rocket and its parts are visible only in two specific cycle states due to the spatial location of robots and other machinery and the location of the cameras and camera angles. We focus on filtering cycle four and a section of cycle nine based on domain knowledge and observational insights for enhanced image understanding. Table II and III show the filtered dataset and its anomaly statistics, respectively.

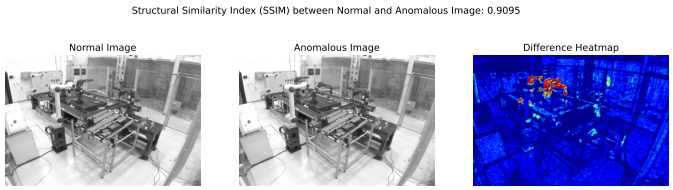

Fig. 2. Structural Similarity between Normal and Anomalous Images

C. Image Cropping

Our dataset comprises images captured at various stages of the manufacturing cycle. In each cycle, the rocket's position remains consistent within each respective state. Using this domain knowledge, we defined a cropping strategy by defining a bounding box for each cycle state to remove the

TABLE II
ARTIFICATS OF THE FILTERED IMAGES IN FF MULTIMODAL DATASET

Dataset Artifact	Statistic
Total image count	15594
Train image count	12475
Test image count	3119
Filtered image sizes	1:200px*70px (cycle state 4) 2:400px*205px (cycle state 9)

TABLE III
ANOMALY TYPES IN FILTERED MULTIMODAL FF DATASET

Anomaly type	Train image count	Test image count	Total count by anomaly	Percentage
No Anomaly	8006	2016	10022	64.26%
NoNose	872	238	1110	7.1%
NoNose, NoBody2	1222	308	1530	9.8%
NoNose, NoBody2, NoBody1	1310	310	1620	10.38%
NoBody1	1065	247	1312	8.4%
Total image count	12475	3119	15594	

background from all images. This consistent positioning of the rockets across cycles enables us to crop the images accurately, ensuring that only the relevant parts of the images are retained.

The suboptimal results obtained with uncropped images necessitated cropping of images. Specifically, our models performed poorly, with significant misclassification rates. We employed the explainability model SCORE-CAM [46] to understand the underlying issues. The insights provided by SCORE-CAM revealed that the models prioritized images' backgrounds rather than the rockets themselves, as shown in Figure 3. Therefore, images were cropped to remove the background to ensure the models focused on the important visual features. This image transformation improved the model performance, leading to better performance. As the rocket is the most important feature, the images are cropped accordingly, which resulted in a 200x70 pixel image for cycle state four and 400x205 pixels for cycle state nine.

D. Text Guided Zero-Shot Object Detection

We then incorporate a zero-shot object detection model to enhance the accuracy and robustness of our anomaly detection pipeline, which targets the detection of rockets and their parts. Specifically, we employ OWL-ViT [47], a model that leverages multimodal representations to perform open-vocabulary detection. This approach allows for the detection of objects based on free-text prompts, facilitating a flexible and powerful detection mechanism. OWL-ViT integrates CLIP (Contrastive Language-Image Pretraining) with lightweight object classification and localization heads. This integration enables the model to handle open-vocabulary detection by embedding free-text queries through CLIP's text encoder. These are then utilized as inputs for the object classification and localization

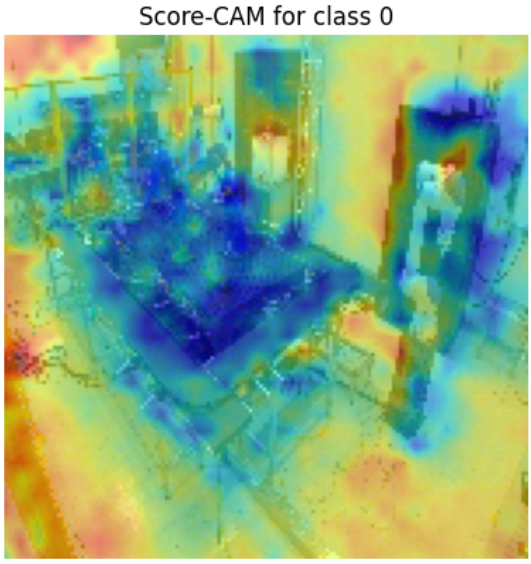

Fig. 3. Score-CAM visualization shows the model mistakenly focusing on background elements, highlighted in red as the most important regions, rather than the assembly pipeline marked in blue. This emphasizes the need for cropping and object detection to improve accuracy by isolating the relevant parts of the image.

heads. The ViT processes image patches as inputs, associating them with their corresponding textual descriptions.

E. Fine-tuning Object Detection Model with YOLOv9

To further enhance the object detection pipeline for rocket assembly, we fine-tune a YOLOv9 object detection model [19] using the dataset derived from OWL-ViT, which we refer to as YOLO-FF. While zero-shot models like OWL-ViT offer significant flexibility and adaptability, a fine-tuned model provides specific advantages as follows:

- Fine-tuning YOLOv9 using the derived dataset ensures the model parameters are optimized for the specific task, facilitating better accuracy and precision.
- YOLOv9 is optimized for high-speed inference, making it ideal for real-time applications, whereas models like OWL-ViT typically require more computation and have longer inference times.

IV. PROBLEM FORMULATION

Consider a dataset $\mathcal{D}$ comprising images captured at various stages of the rocket assembly process. The manufacturing process is divided into cycles, each consisting of 21 distinct states. Let $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ denote the set of manufacturing cycles, where each cycle C_j for $j = 1, 2, \ldots, n$ represents the complete assembly of one rocket. Each cycle C_j is divided into 21 states, represented by the set $\mathcal{S} = \{S_1, S_2, \ldots, S_{21}\}$. For our anomaly detection task, we focus on images corresponding to cycle states S_4 and S_9, where rocket parts are most likely to be visible. Let $\mathcal{I} = \{I_{j,s} \mid j = 1, 2, \ldots, n,\ s \in \{4, 9\}\}$ denote the set of images captured during these specific cycle states. Each image $I_{j,s}$ is associated with a label $y_{j,s} \in \mathcal{L}$, where $\mathcal{L} = \{\text{normal}, \text{anomaly}_1, \text{anomaly}_2, \ldots, \text{anomaly}_k\}$. Additionally, let $B_{j,s} = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$ denote the bounding

box for image $I_{j,s}$, obtained via object detection techniques to focus on the relevant part of the image. We define the dataset as in equation 1.

$$\mathcal{D} = \{(I_{j,s}, y_{j,s}, B_{j,s}) \mid j = 1, 2, \ldots, n,\ s \in \{4, 9\}\} \quad (1)$$

A. Task Description

The task is to develop a model $f : \mathcal{I} \rightarrow \mathcal{L}$ that can accurately classify each image $I_{j,s}$ into one of the predefined categories in $\mathcal{L}$. The classification involves the following steps:

1) Filtering the dataset to include only images from cycle states S_4 and S_9.
2) Cropping each image $I_{j,s}$ using its bounding box $B_{j,s}$, which is mapped to its specific state. This effectively isolates the regions of interest while removing background noise.
3) Detecting objects in the cropped images using an object detection model. This will generate the region of interest (i.e., rocket parts) and their bounding boxes.
4) Classifying anomalies from the detected objects using an anomaly detection model.

The main goal is to detect anomalies in the images by using bounding boxes to help the model focus on the important parts. To address class imbalance, we use a weighted cross-entropy loss, where higher weights are assigned to classes with fewer images. We aim to minimize the weighted classification error defined in equation 2:

$$\min_{\theta} \frac{1}{n} \sum_{j=1}^{n} \sum_{s \in \{4,9\}} w_{y_{j,s}} \mathcal{CE}(f(I_{j,s}; \theta), y_{j,s}) \quad (2)$$

where θ represents the model parameters, $\mathcal{CE}$ is the cross-entropy loss function, and $w_{y_{j,s}}$ is the weight assigned to the class $y_{j,s}$. By focusing on specific cycle states and using object detection to preprocess the images, we aim to improve the accuracy and reliability of our anomaly detection system.

V. Experiments

We outline the following experimental setup to evaluate the AssemAI's overall performance and the contribution of each sub-module.

A. Experiments with YOLO-FF Model

The YOLOv9 model is fine-tuned on the derived labeled image dataset resulting from the OWL-ViT model. It is trained for 50 epochs. During training, the model achieved an accuracy of approximately 99% on the validation dataset.

B. Common Hyperparameters and Training Setup

The derived image dataset is split into training (80%) and testing (20%) sets. We used Cross-Entropy Loss and the Adam optimizer [48] for baselines and the proposed models, tuning hyperparameters like epochs, batch size and learning rate. The best model is saved based on validation accuracy. The dataset preprocessing involved resizing images to the required input dimensions (224×224 pixels), normalizing ([0.485,0.456,0.406]) and standardizing ([0.229,0.224,0.225]) according to the specific requirements of each model.

C. Baselines

1) Custom CNN: The Simple CNN model architecture includes two convolutional layers (32 and 64 filters respectively), each followed by a max-pooling layer. After flattening, the output is passed through a fully connected layer with 512 neurons, then to the final layer corresponding to the number of classes (5).

2) Custom ViT: The Vision Transformer (ViT) model [49] is implemented from scratch, starting with a patch embedding layer of size 16 that splits the input image into non-overlapping patches, each embedded into a high-dimensional space of size 768. This is followed by a series of transformer blocks for learning spatial relationships within the image patches. Each transformer block consists of a multi-head self-attention mechanism and a feed-forward neural network, with layer normalization and residual connections applied at each step. The final classification head maps the output to the desired number of classes (5).

3) Pre-trained ViT: We utilize the pre-trained Vision Transformer (ViT) model, specifically google/vit-base-patch16-224. Images are preprocessed by dividing each image into a sequence of fixed-size non-overlapping patches, then linearly embedded. A [CLS] token is added to represent the entire image, facilitating classification. Absolute position embeddings are incorporated and the resulting sequence of vectors is fed to the standard Transformer encoder. The ViTImageProcessor resizes and normalizes images to the required 224x224 resolution. The model is loaded with half-precision (torch.float16). The training utilized PyTorch Lightning's Trainer, with a weighted Cross-Entropy loss function, and the AdamW optimizer with a learning rate scheduler. The best checkpoint is saved by monitoring the validation loss.

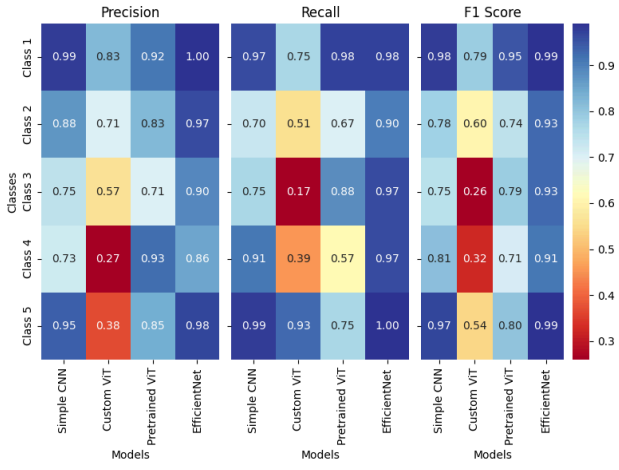

Fig. 4. Experimental results with OWL-ViT. Class 1-5 denotes anomaly types: [No Anomaly], [NoNose], [NoNose,NoBody2], [NoNose,NoBody2,NoBody1] and [NoBody1] respectively

D. Proposed Modeling Approach

We implement an EfficientNet-B0 model [16] for image classification tasks. The EfficientNet-B0 architecture is se-

TABLE IV
EXPERIMENTAL RESULTS OF ASSEMAI

Model	WP	WR	WF1	Accuracy	Support*
Simple CNN*	82.00± 1.00%	81.00± 1.00%	83.00± 1.00%	82.05± 1.00%	3119
Custom ViT*	85.00± 1.00%	86.00± 0.05%	83.00± 1.00%	84.10± 1.00%	3119
Pre-trained ViT*	88.50± 0.50%	87.50± 0.50%	87.50± 0.50%	88.50± 0.50%	3119
EfficientNet with original images	62.50± 0.50%	60.50± 0.50%	61.50± 0.50%	61.50± 0.50%	332002
EfficientNet with filtered images	70.50± 0.50%	72.50± 0.50%	73.50± 0.50%	72.50± 0.50%	3119
EfficientNet*	**95.00± 1.00%**	**96.00± 1.00%**	**95.00± 1.00%**	**96.00± 1.00%**	**3119**

Models marked in * are experimented with the derived image dataset. EfficientNet model is experimented for original images (includes all cycle states) and for filtered images (includes cycle states four and nine). Bold indicates the best performance.

lected for its state-of-the-art performance and efficiency in image classification. It leverages a compound scaling method to optimize model depth, width and resolution, achieving high accuracy with fewer parameters compared to traditional networks. The model is pre-trained on ImageNet and adapted for our task by modifying the final classification layer to match the number of classes in our dataset (5 classes).

VI. RESULTS

Table IV summarizes the results of our experiments on the test set and the ablation studies across different baselines. We evaluate the performance using four metrics: weighted averages of precision, recall, F1-score and accuracy. The weighted averages are calculated based on the four types of anomaly classes and the normal class. The EfficientNet model achieves 96% overall accuracy and 95% of precision, 96% of recall, 95% of F1 score, respectively. Figure 4 depicts the performance of detecting various anomaly types and the normal class across various modeling approaches with the YOLO-FF object detection approach. It can be observed that among all the models, EfficientNet gives the best results in detecting all the types of classes.

A. Additional experiments on Explainability

1) User level explainability: In this work, we employ process ontology designed and developed for the Future Factories Rocket assembly line. In contrast to conventional ontologies, process ontology not only captures the definition of sensors but also captures the procedural nature of the assembly process. This aids in understanding the involvement of sensors and types of equipment at a given point in the assembly process. The Future Factories assembly is divided into 21 cycle states, which form the basis of the ontology construction process. The specific features of process ontology are as follows: (i) consists of definition and item specification of sensors and types of equipment, (ii) relationship between the sensors and types of equipment, (iii) function and involvement of each sensor and robot with respect to the cycle states (iv) expected (or anomalous) values of sensor variables in with respect to each cycle state (v) type of anomaly that could be associated with each cycle state (iv) sensor values and other knowledge can be dynamically updated as per the change in experiment set up. Capturing the procedural nature of the assembly process aids in understanding the contribution of sensors in anomalies. For example, if the initial stage of the assembly line is being analyzed for anomaly, it can be understood from the ontology that Robot-4 and its corresponding sensors do not contribute to this anomaly.

The goal of process ontology is to explain and assess the output from the models. Given an image, if the model detects an anomaly, the expected values of the sensors present in that image can be obtained and provided to the user. On the other hand, process ontology can also verify if the model predictions are correct in certain cases (Figure 5). Since each image is associated with a timestamp which in turn can be mapped to the cycle state, the predicted output from the model can be verified using anomaly types defined in the process ontology. Figure 5 illustrates that the input image is associated with cycle state 4. The model predicted the image to be anomalous with type *NoNose*. This is an incorrect prediction as per the ontology as *NoNose* anomaly can happen only from cycle stage 8 onwards. Using this knowledge and verifying the outputs of model predictions, it is found that the model incorrectly predicted *NoBody2* anomaly 51 out of 801 times and *NoNose* anomaly 106 out of 1145 times. The ontology also acts as an additional layer that catches the misclassification of the model, enhancing the robustness of the proposed anomaly detection pipeline.

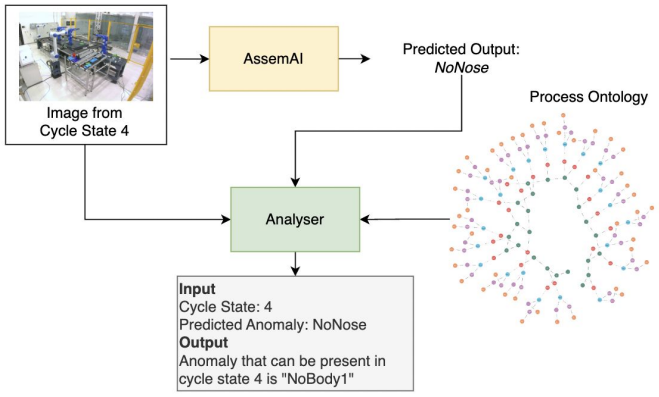

Fig. 5. Verification of the explanations through process ontology

VII. DEPLOYMENT OF ASSEMAI

After training and testing the model, we deploy AssemAI on the Future Factories lab. This deployment setup can be seen in Figure 6. AssemAI requires two separate types of inputs to operate in real time. The script that utilizes the model should be able to obtain the current cycle state of the assembly process. This data tag allows the script to ensure that the images captured and input into the model are from cycle states four and nine. This tag is obtained by connecting to the OPC UA server running on the Programmable Logic Controller (PLC) and constantly reading the tag as it is

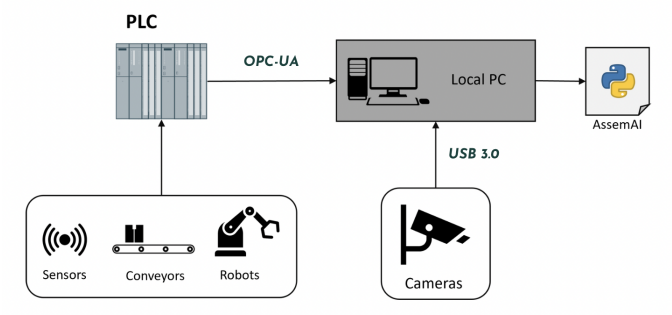

Fig. 6. Deployment Setup of AssemAI

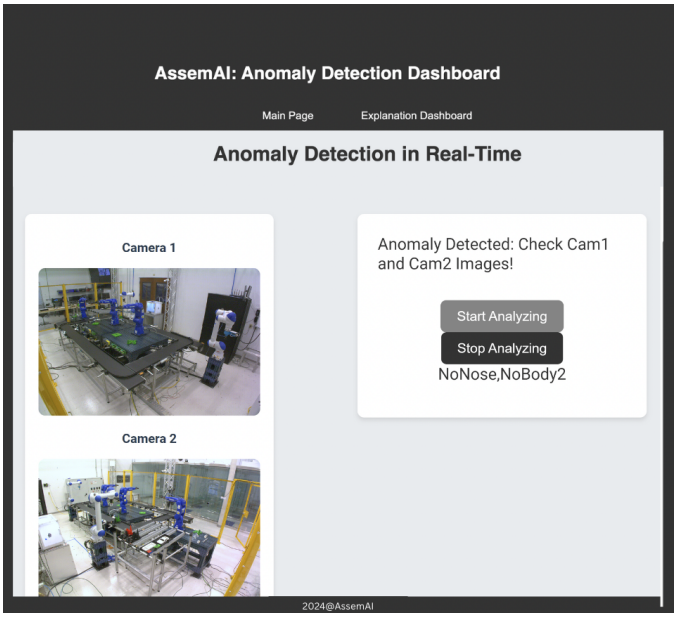

Fig. 7. Web User interface of AssemAI

updated. The cameras used in this deployment are separate from the PLC's network. Images taken by the cameras can be acquired by connecting directly to them through a USB 3.0 wired connection. These cameras are industry-grade Basler cameras with a Python library, pyplon, which simplifies the image-capturing process. Since this is a wired connection, we can ensure that there is minimal communication lag time between the image request and the capturing process. As such, once the cycle state tag is read as either four or nine, the images are captured from the cameras and sent into the model for detection. Figure 7 illustrates the web user interface of the deployed AssemAI.

VIII. Conclusion and Further Work

In this study, we derived an industry-standard dataset tailored for assembly processes using the novel YOLO-FF model and introduced AssemAI, a standardized image-based anomaly detection pipeline. We began with a simple CNN Model, which provided a baseline for anomaly detection through a straightforward architecture and standard training techniques. Building on this, we explored ViT and EfficientNet models to enhance classification accuracy and efficiency. A significant aspect of our approach is the focus on interpretability, where we aimed to understand both model and user lever explainability to high-level phenomena such as structural integrity. This not only improved model understanding but also provided actionable insights for anomaly detection. Our findings indicate that EfficientNet offers significant improvements over traditional methods. Future work should explore hybrid architectures, which take other modalities like time series and textual, further interpretability techniques and deployment on edge devices for real-time anomaly detection applications to extend these findings to other domains and production environments. Also, to improve how domain experts understand our approach, we suggest creating abstract representations of causal factors associated with anomalies. This approach focuses on linking sensor data to broader concepts like structural issues or gripper malfunctions, providing a more comprehensive view of anomalous events.

Acknowledgment

This work is supported in part by NSF grants #2119654, "RII Track 2 FEC: Enabling Factory to Factory (F2F) Networking for Future Manufacturing" and SCRA grant "Enabling Factory to Factory (F2F) Networking for Future Manufacturing across South Carolina".

References

[1] N. Anumbe, C. Saidy, and R. Harik, "A primer on the factories of the future," *Sensors*, vol. 22, no. 15, p. 5834, 2022.

[2] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 157–169, 2018.

[3] L. Wu, L. Xu, and H. Xu, "Machine learning for anomaly detection in industrial systems: A review," *Journal of Manufacturing Processes*, vol. 31, pp. 476–487, 2018.

[4] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, no. 2, p. 1–38, Mar. 2021. [Online]. Available: http://dx.doi.org/10.1145/3439950

[5] V. Chandola and V. Banerjee, A.and Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[6] M. Babic, M. A. Farahani, and T. Wuest, "Image based quality inspection in smart manufacturing systems: A literature review," *Procedia CIRP*, vol. 103, pp. 262–267, 2021.

[7] S. Sundaram and A. Zeid, "Artificial intelligence-based smart quality inspection for manufacturing," *Micromachines*, vol. 14, no. 3, p. 570, 2023.

[8] M. Drakaki, Y. L. Karnavas, I. A. Tziafettas, V. Linardos, and P. Tzionas, "Machine learning and deep learning based methods toward industry 4.0 predictive maintenance in induction motors: State of the art survey," *Journal of Industrial Engineering and Management (JIEM)*, vol. 15, no. 1, pp. 31–57, 2022.

[9] K. S. Kiangala and Z. Wang, "An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment," *Ieee Access*, vol. 8, pp. 121 033–121 049, 2020.

[10] Y. Du, Y. Chen, G. Meng, J. Ding, and Y. Xiao, "Fault severity monitoring of rolling bearings based on texture feature extraction of sparse time–frequency images," *Applied sciences*, vol. 8, no. 9, p. 1538, 2018.

[11] Y. Yang, B. Yang, S. Zhu, and X. Chen, "Online quality optimization of the injection molding process via digital image processing and model-free optimization," *Journal of Materials Processing Technology*, vol. 226, pp. 85–98, 2015.

[12] Y. Jing, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, "Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges," *Materials*, vol. 13, p. 5755, 12 2020.

[13] "CVPR2023 Tutorial on AD — sites.google.com," https://sites.google.com/view/cvpr2023-tutorial-on-ad/, 2023, [Accessed 18-07-2024].

[14] H. Yao and W. Yu, "Generalizable industrial visual anomaly detection with self-induction vision transformer," *arXiv preprint arXiv:2211.12311*, 2022.

[15] e. a. Tian, F., "A review of interpretability methods for deep learning models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3225–3241, 2021.

[16] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 97, pp. 6105–6114, 2019.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[19] C.-Y. Wang, I.-H. Yeh, and H. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," *ArXiv*, vol. abs/2402.13616, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267770251

[20] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," in *arXiv preprint arXiv:1704.04861*, 2017.

[22] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[23] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 012–10 022.

[24] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran, "Zero-shot object detection," in *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2018, p. 397–414. [Online]. Available: https://doi.org/10.1007/978-3-030-01246-5_24

[25] S. Rahman, S. Khan, and F. Porikli, "Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 547–563.

[26] D. Gupta, A. Anantharaman, N. Mamgain, S. K. S, V. N. Balasubramanian, and C. Jawahar, "A multi-space approach to zero-shot object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[27] Y. Zheng, R. Huang, C. Han, X. Huang, and L. Cui, *Background Learnable Cascade for Zero-Shot Object Detection*. Springer International Publishing, 2021, p. 107–123. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-69535-4_7

[28] M. Hayat, S. Khan, S. Ali, S. W. Zamir, F. S. Khan, and L. Shao, "Synthesizing the unseen for zero-shot object detection," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2020, pp. 449–464.

[29] Z. Li, Z. Zhang, Z. He, and H. Yang, "Context-guided super-class inference for zero-shot detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 960–967, 2020.

[30] X. Zhu, Z. Wang, J. Dai, L. Yuan, and Y. Wei, "Don't even look once: Synthesizing features for zero-shot detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1201–1210.

[31] M. Hayat, S. Khan, S. W. Zamir, F. Shahbaz Khan, and L. Shao, "Generative zero-shot detection: Synthesizing the unseen for zero-shot object detection," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2020, pp. 297–312.

[32] Y. Liu, H. Zhang, Y. Huang, and W. Liu, "Contrastive learning for zero-shot object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[33] Y. Li, A. Kumar, A. K. Roy-Chowdhury, and H. Pirsiavash, "Semantics-aware detection transformer for zero-shot object detection," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2023.

[34] S. S. Kim and A. N. Reddy, "Image-based anomaly detection technique: algorithm, implementation and effectiveness," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 10, pp. 1942–1954, 2006.

[35] J. Mendoza-Bernal, A. González-Vidal, and A. F. Skarmeta, "A convolutional neural network approach for image-based anomaly detection in smart agriculture," *Expert Systems with Applications*, vol. 247, p. 123210, 2024.

[36] A. Alloqmani, Y. B. Abushark, A. I. Khan, and F. Alsolami, "Deep learning based anomaly detection in images: insights, challenges and recommendations," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, 2021.

[37] G. Xie, J. Wang, J. Liu, J. Lyu, Y. Liu, C. Wang, F. Zheng, and Y. Jin, "Im-iad: Industrial image anomaly detection benchmark in manufacturing," *IEEE Transactions on Cybernetics*, 2024.

[38] M. Maggipinto, A. Beghi, and G. A. Susto, "A deep learning-based approach to anomaly detection with 2-dimensional data in manufacturing," in *2019 IEEE 17th international conference on industrial informatics (INDIN)*, vol. 1. IEEE, 2019, pp. 187–192.

[39] Y. Jiang, W. Wang, and C. Zhao, "A machine vision-based realtime anomaly detection method for industrial products using deep learning," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 4842–4847.

[40] Y. Tan, B. Jin, A. Nettekoven, Y. Chen, Y. Yue, U. Topcu, and A. Sangiovanni-Vincentelli, "An encoder-decoder based approach for anomaly detection with application in additive manufacturing," in *2019 18th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2019, pp. 1008–1015.

[41] Y. Kim, T. Lee, Y. Hyun, E. Coatanea, S. Mika, J. Mo, and Y. Yoo, "Self-supervised representation learning anomaly detection methodology based on boosting algorithms enhanced by data augmentation using stylegan for manufacturing imbalanced data," *Computers in Industry*, vol. 153, p. 104024, 2023.

[42] A. Bougaham, M. El Adoui, I. Linden, and B. Frénay, "Composite score for anomaly detection in imbalanced real-world industrial dataset," *Machine Learning*, vol. 113, no. 7, pp. 4381–4406, 2024.

[43] B. Reidy, D. Duggan, B. Glasauer, P. Su, and R. Zand, "Application of machine learning for quality risk factor analysis of electronic assemblies," in *2023 24th International Symposium on Quality Electronic Design (ISQED)*, 2023, pp. 1–6.

[44] M. Haselmann, D. P. Gruber, and P. Tabatabai, "Anomaly detection using deep learning based image completion," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2018, pp. 1237–1242.

[45] R. Harik, F. E. Kalach, J. Samaha, D. Clark, D. Sander, P. Samaha, L. Burns, I. Yousif, V. Gadow, T. Tarekegne *et al.*, "Analog and multimodal manufacturing datasets acquired on the future factories platform," *arXiv preprint arXiv:2401.15544*, 2024.

[46] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 111–119, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:215745726

[47] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen *et al.*, "Simple open-vocabulary object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 728–755.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. Citeseer, 2014.

[49] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.