2025 40th International Conference on Image and Vision Computing New Zealand (IVCNZ) | 979-8-3315-8654-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/IVCNZ67716.2025.11281645

Comparative Study of DINOv2, I-JEPA, and ViT Embeddings for Unsupervised Anomaly Detection

Jean-Marc Spat
HEIA-FR/ HES-SO
Fribourg, Switzerland
jeanmarc.spat@master.hes-so.ch

Houda Chabbi-Drissi
iCoSys
HEIA-FR / HES-SO
Fribourg, Switzerland
0000-0001-7087-8108

***Abstract*—This paper presents a unified, unsupervised framework for Visual Anomaly Detection (VAD) in dynamic, fixed-view scenes, leveraging modern Vision Transformers (ViTs) without additional fine-tuning. We investigate whether embeddings from backbones like a generic ViT, DINOv2, and I-JEPA, combined with a simple clustering approach, are sufficient for identifying anomalies. Our methodology extracts both global (CLS token) and local (patch-level) embeddings, applies clustering (k-Means, HDBSCAN) to model the distribution of normal scenes, and uses a scalable vector database (ChromaDB) for efficient similarity search. The three backbones delivered comparable performance, with the generic ViT showing a small but consistent advantage in balanced accuracy. Although local embeddings provided a 3% gain in balanced accuracy, their sequential processing time is considerably higher. This limitation could be mitigated through parallelization, potentially bringing their efficiency closer to that of global embeddings. In contrast, global embeddings deliver comparable performance while being approximately 256 times faster, enabling near real-time batch processing of 15 s video segments. The k-Means clustering and the proposed retrieval strategy proved most effective, achieving a practical operational trade-off with a balanced accuracy of 82%. These results suggest that ViT embeddings are effective for separating normal and anomalous patterns. Local embeddings appear to capture complementary and pertinent information, and we expect that with modest adjustments in the detection strategy, they could be further leveraged to improve anomaly detection performance.**

***Index Terms*—anomaly detection, embeddings, clustering, novelty detection, DINO, I-JEPA, ViT**

I. Introduction

Visual Anomaly Detection (VAD) seeks to identify rare or unexpected patterns in visual data that deviate from a learned distribution of normality, often in unsupervised settings where only normal samples are available. As discussed in [1], the introduction of Convolutional Neural Networks (CNNs) greatly advanced VAD, but their local receptive fields limit their ability to capture long-range dependencies in complex scenes. Vision Transformers (ViTs) [2] address this by employing self-attention to model both local and global relationships, making them well-suited for anomaly detection tasks [1]. Recent self-supervised learning (SSL) methods such as DINOv2 [3] and I-JEPA [4] further enhance ViT representations by learning robust, semantically meaningful features from large-scale unlabeled data.

In dynamic, fixed-view scenes, normal variations (e.g., changes in lighting, shadows, minor object movements, or typical human activity) must be distinguished from genuine anomalous events (e.g., intrusions, unusual behaviors, or foreign objects). To address this, we employ unsupervised clustering to group embeddings of normal scenes into compact, semantically coherent clusters, allowing the system to capture the natural diversity of normality and improve robustness against false positives. In this work, we investigate whether embeddings from such SSL-based ViTs can be used directly for anomaly detection by simply adding an unsupervised clustering step, without additional fine-tuning or complex adaptation. Our approach extracts global (CLS token) and local (patch-level) embeddings from three ViT backbones (generic ViT, DINO, and I-JEPA), applies clustering (k-Means, HDBSCAN) to model the distribution of normal scenes at multiple semantic levels. To mitigate the scalability issues of traditional memory bank methods in feature-based anomaly detection [1] a scalable vector database (ChromaDB) is used to store all the embeddings, allowing efficient similarity search and anomaly scoring using approximate nearest-neighbor indexing structures such as HNSW.

This minimal pipeline aims to assess how far ViT and modern SSL-based embeddings alone can go in separating normal and anomalous patterns in dynamic, fixed-view scenes. Our approach, inspired by task-agnostic frameworks such as UniFormaly [5], is designed to be generalizable across various scenarios.

The remainder of the paper is organized as follows: section II reviews related work in visual anomaly detection based on using embeddings. Section III details our proposed methodology, including embedding extraction, clustering, and the use of a vector database for inference. Section IV describes the experimental setup, the dataset Safe/Unsafe, and evaluation metrics. Section V presents results, comparisons, and ablation studies. Finally, Section VI concludes with a discussion and directions for future work.

II. Related Work

Based on the comprehensive taxonomy of existing approaches given in [1], our work falls under feature-based methods, particularly those that utilize memory banks to capture distributions of normal features. Accordingly, we review the

979-8-3315-8654-6/25/$31.00 ©2025 IEEE

most relevant literature which uses embeddings and memory banks.

In industrial manufacturing context, many approaches have been developed using embeddings and memory banks. They are evaluated on the MVTec AD benchmark achieving excellent results. Among them, PatchCore [6] is a state-of-the-art unsupervised visual anomaly detection method that employs pre-trained CNNs such as ResNet50 to extract mid-level features from intermediate layers. PatchCore and its extension SA-PatchCore [7] that integrates a self-attention module to capture contextual relationships between image regions, store a subset of these features in a memory bank using Coreset Sampling. Anomalies are then identified based on nearest-neighbor distances to the memory of normal training data. Similarly, the dual-memory architecture proposed in [8] (DMAD) combines both normal and anomaly prototypes in a semi-supervised framework, also using CNN-based features, to enhance representation learning in industrial contexts. The memory banks of multi-scale features (MBMF) approach described in [9] also achieved good performance. It builds separate memory banks using multiple spatial scales of CNN-derived features to handle anomalies of different sizes.

While CNN-based approaches have proven effective, recent works have successfully explored ViT embeddings for anomaly detection. While PatchCore-like models successfully detect local structural or appearance-based anomalies, they lack mechanisms for logical anomaly detection which corresponds to component relationships. ComAD [11] addresses this by decomposing the image into semantic components and modeling logical relationships between image regions, obtaining fine-grained anomaly detection. The method leverages features extracted from a pre-trained DINO ViT model. In [10], the authors enhanced DINOv2 representations by introducing register tokens, which isolate high-norm patch tokens to improve out-of-distribution (OOD) generalization and anomaly rejection in classification tasks. While their method is supervised and classifier-based, it highlights the robustness of ViT-derived features, which we leverage in a memory-based unsupervised framework. Similarly, [12] proposes a semantic anomaly detection that uses of DINOv2 embeddings to identify contextually invalid or unusual combinations of familiar visual elements, detecting high-level semantic deviations. UniFormaly [5] introduces a task-agnostic and unified framework for various visual anomaly detection tasks, using a single, off-the-shelf ViT-based model. It introduces Back Patch Masking (BPM) to filter out irrelevant background regions via self-attention maps, and employs a top-k feature matching strategy to unify the anomaly scoring logic across tasks.

To position our approach, prior work has demonstrated strong performance in industrial anomaly detection using embeddings from CNN and various memory bank strategies (e.g., PatchCore-like [6], [7], DMAD [8], MBMF [9]). Our approach leverages both off-the-shelf ViT models and ViTs pre-trained using SSL methods, following recent methods that have highlighted the power of ViT embeddings in classification or semantic contexts [10], [12]. While sharing UniFormaly's [5] goal of generalized, task-agnostic VAD leveraging self-supervised ViT features, our fully unsupervised framework distinguishes itself by explicitly modeling multi-scale normality. This is achieved through the separate unsupervised clustering of global `[CLS]` token and local patch-level ViT embeddings. Furthermore, we employ efficient vector indexing via ChromaDB to enable scalable, retrieval-based anomaly detection.

III. Methodology

1) Training stage: (red, upper part of Fig. 1). Either global embeddings or local embeddings of a specific patch are used as input. The embeddings are processed by two distinct clustering pipelines: a density-based pipeline using HDBSCAN and a centroid-based pipeline using k-Means. Each pipeline yields its own set of cluster assignments and associated metadata; these are stored in a vector database collection (with approximate nearest-neighbor indexing) together with cluster identifiers. As a result, two independent models are produced—one tied to the HDBSCAN clusters and one tied to the k-Means clusters.

2) Testing stage: (blue, lower part of Fig. 1). A previously unseen embedding is queried against the vector database via similarity search to retrieve its nearest neighbors. For each clustering model, the retrieved neighbors are used to determine the most likely cluster and to compute a distance-based confidence score for non-membership in that cluster. All distance computations were performed using the Euclidean metric, as vector magnitude is often informative for images. A predefined threshold is then applied to each model's confidence score; scores below the threshold are treated as normal for that model, while scores above the threshold are flagged as anomalies. The higher the threshold, the less false negatives will appear.

A. Feature Extraction

Our study performs a comparative analysis of feature representations extracted from three ViT based models: DINOv2, I-JEPA, and a ViT pretrained on ImageNet-21k.

A Vision Transformer (ViT) backbone first pre-processes an input image $Im \in \mathbb{R}^{H \times W \times 3}$ by resizing it to a square resolution and normalizing pixel values, resulting in $I \in \mathbb{R}^{H' \times W' \times 3}$. The image I is then divided into a grid of non-overlapping

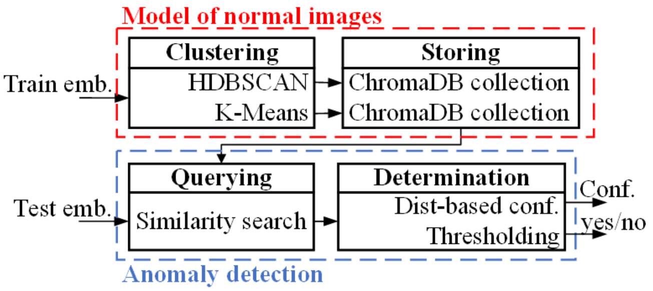

Fig. 1. Two-step pipeline. The red, upper part models the normal situation based on a set of embeddings. The blue, lower part determines if a new, unknown embedding is an anomaly with a confidence value in [0,1].

square patches of size $P \times P$, yielding $N = \frac{H' \times W'}{P^2}$ patches. A special learnable `[CLS]` token is prepended to the patch sequence, it is designed to aggregate global contextual information. The ViT processes each of the resulting $N+1$ tokens to produce a corresponding D-dimensional embedding vector z_i.

To capture different semantic granularity, we extract two types of embeddings: global and local. For DINOv2 and the ImageNet-pretrained ViT, the global representation g corresponds to the `[CLS]` token embedding, in the final transformer layer. It serves as a compact representation of the overall scene, making it effective for detecting scene level anomalies.

Local features, correspond to the N patch embeddings output by the last hidden state of the transformer: z_i. These token wise embeddings retain spatial structure and are well suited for local anomaly localization.

Unlike the others, I-JEPA does not use a `[CLS]` token during pretraining. Instead, we extract patch-level embeddings from its target encoder. The global image representation g is then computed using average pooling across all patch embeddings as widely adopted in ViT-based models.

$$g = \frac{1}{N} \sum_{i=1}^{N} z_i \in \mathbb{R}^D$$

B. Detection strategies

The proposed framework enables two distinct strategies for anomaly detection. Firstly, detection can rely solely on global embeddings, providing a high-level assessment of scene normality; this can be sufficient for easiest case like intrusion. Secondly, it can exclusively utilize local patch embeddings, offering fine-grained anomaly localization; This can be valuable for identifying subtle, fine-grained anomalies within specific regions that might not drastically alter the global scene like a person wearing an unusual item.

From a technical perspective, each patch is independently modeled using the same processing pipeline illustrated in Fig. 1. They all yield a confidence score indicating the likelihood of a localized anomaly. These scores are then aggregated into a global decision through majority voting across all patches: if $\#patches_{unsafe} \geq \#patches_{safe}$, the image is labeled unsafe; otherwise, it is labeled safe. This design accounts for the sensitivity of individual patches due to clustering effects, while the attention mechanism ensures that information from anomalous patches can propagate to the rest of the image representation.

C. Unsupervised Clustering and Anomaly Scoring

The method applies unsupervised clustering separately to the two types of embeddings introduced in the previous section: the global embeddings g^j, derived from the `[CLS]` token of the j-ith input image and representing the entire image, and its local patch embeddings z_i^j, corresponding to individual spatial regions within the image.

Each set of embeddings $\{g^j\}$ and $\{z_i^j\}$ is clustered independently using two algorithms HDBSCAN and k-Means.

1) HDBSCAN: builds a cluster hierarchy from local density estimates and extracts the most stable flat clusters. The hyperparameter $min\_cluster\_size$ sets the smallest group of points that can be considered a cluster, controlling the granularity of the results. The algorithm is inherently capable of identifying and labeling isolated points as noise, as well as computing membership strengths for new point. We can view the inverse of the strength $1 - strength$ as the anomaly confidence. By applying a threshold on the latter, one can adjust the strictness of the model in controlling false negatives.

2) k-Means: partitions a dataset into exactly k clusters by minimizing the within-cluster sum of squared distances between points and their assigned cluster centroids. To determine the optimal number of clusters, the silhouette score can be employed as an evaluation metric. This measure quantifies the quality of clustering by comparing the cohesion of each point with its own cluster to its separation from other clusters, with higher scores indicating more distinct and well-formed clusters. The value ranges from -1.0 to 1.0. A score higher than 0.5 is considered ”reasonable”, and higher than 0.7 ”strong”. For inference, k-Means is not designed to detect noise points. Authors in [15] introduced the k-NNN method, which extends classical k-NN by incorporating structural information from the neighborhood of each training point. Their key insight is that anomalies may manifest as deviations in low-variance directions of the embedding space. To capture this, they analyze not only the k-nearest neighbors of a test point, but also the eigenstructure of each neighbor’s local neighborhood. This approach improves detection sensitivity and aligns well with the notion that anomalies distort local geometry rather than global position alone. We also make use of the neighbours of neighbours principle to compare the distance to the closest trainpoint with the local density of normal datapoints (Fig. 2): For a given embedding p we compute the distance d to the nearest neighbor nn and the average distance of nn to its m nearest neighbors of the same cluster $nnn_{1...m}$. That value is treated as the local density *loc_dens*. In our experiments, we chose $m = 5$. The resulting anomaly confidence can then be defined as:

$$\text{confidence} = \begin{cases} \min\left(1, \frac{\text{d}}{\text{loc\_dens}}\right), & \text{if loc\_dens} \neq 0, \\ 1, & \text{otherwise.} \end{cases}$$

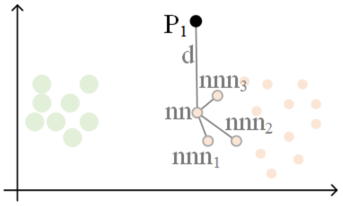

Fig. 2. Illustration of neighbours of neighbour retrieval. The embedding vector p retrieves nn of the red cluster. The local density is computed from $nnn_{1..3}$.

D. *Persistence of Embedding Vectors*

Since the k-Means method requires vector neighbour retrieval and distance computation for inference, both global and local embeddings are persisted in ChromaDB, each tagged with its assigned cluster ID for k-Means. To ensure optimized multi-scale retrieval and anomaly scoring during inference, all global embeddings $\{g^j : j = 1..Q\}$, corresponding to the Q images in the dataset, are stored together in a single, dedicated collection. Besides, each patch index $i : 1..N$ has its own dedicated collection storing all corresponding local embeddings $\{z_i^j : j = 1..Q\}$. This results in N separate collections specifically for local patch embeddings, which is crucial for maintaining spatial granularity and enabling position-aware anomaly detection alongside the global scene representation. This is done for each embedding type: DINOv2, ViT and I-JEPA.

An HDBSCAN object from the *hdbscan Python library* already implements the indexing and retrieval functions. Similar to storing in ChromaDB, all global embeddings $\{g^j : j = 1..Q\}$ will be contained within a model, alongside with N separate models for each patch index $i : 1..N$. Each of the N local models will contain all corresponding local embeddings $\{z_i^j : j = 1..Q\}$. The model can be persisted by storing it on the disk and loaded in memory when needed.

IV. Experimental Setup

A. *Safe/Unsafe Behaviours Dataset*

We are particularly interested in the Video dataset for Safe and Unsafe Behaviours [13] released with the work described in [14], as it captures complex, dynamic, real-world industrial scenes that are highly relevant for evaluating anomaly detection methods. This dataset contains 691 video clips recorded from two IP surveillance cameras in a workplace, mounted about 4 m above the ground, with two different fields of view: walkway (Fig. 3 Left) and forklift way (Fig. 3 Right). The videos are shot with full HD resolution (1920×1080) pixels at 24 fps. We can see fixed machines and equipment, walking and working people and moving forklift vehicles. They were shot during the day, and the scene is illuminated by sunlight. The authors identified 4 different unsafe behaviours classes namely:

1) Safe Walkway Violation, when a person walks outside of the green path marked on the floor
2) Unauthorized Intervention, when a person works on a machine without wearing a green intervention vest
3) Opened Panel Cover, when an employee leaves the panel connected to the machine open after an intervention
4) Carrying Overload with Forklift, when 3 blocks or more are carried with a forklift

Only the last class was shot by the second camera. Each of those unsafe classes have their safe counterpart, namely Safe Walkway, Authorized Intervention, Closed Panel Cover, Safe Carrying. The classes are imbalanced and have an average duration of 7.7 s. The anomalies are typically small and localized on an unsafe frame.

B. *Dataset preparation*

Unlike our unsupervised method, Unsafe-Net [14] requires extensive labeled data and supervised training, making it less scalable to unseen scenarios or novel anomalies. For our needs, we preprocessed the video clips of the dataset, in the following steps:

- `Field of view`: we dropped the videos of the second camera (forklift) to keep most of the videos of a single field of view. This shortens the dataset and removes the classes Carrying Overload with Forklift and Safe Carrying.
- `Video trimming`: the safe clips are kept intact since all frames contain safe behaviours. The unsafe clips have been manually trimmed (typically start and/or end of the video) such that the sequences only contain unsafe frames.
- `Image extraction`: we extracted a frame every 0.23 s (4.17 fps). We made this choice because both an anomaly or a normal situation always last the duration of the clip. In addition, there are no fast moving objects. Therefore, more fps would result in a lot of highly similar images.
- `Train-Test splitting`: the trainset contains 70% of the safe images, and 30% are contained in the testset. The latter also contains all the unsafe images. The testset is slightly imbalanced between safe and unsafe samples. Table I shows a summary of the number of images kept from the dataset.
- `Model-specific preprocessing`: for each embedding model, a dedicated image processor converts raw images into the precise tensor representation required by the model (color-space conversion, orientation correction, resizing and cropping, scaling pixel intensities, normalization).
- `Embedding`: each image is then associated to a global embedding $\{g^j : j = 1..Q\}$ and each of the 256 patches of size 14×14 pixels is associated to a local embedding $\{z_i^j : j = 1..Q\}$ using the following models:
  - ViT, $vit\_huge\_patch14\_224.orig\_in21k$, embedding size 1280
  - DINOv2, $facebook/dinov2-large$, embedding size 1024
  - I-JEPA, $facebook/ijepa\_vith14\_1k$, embedding size 1280

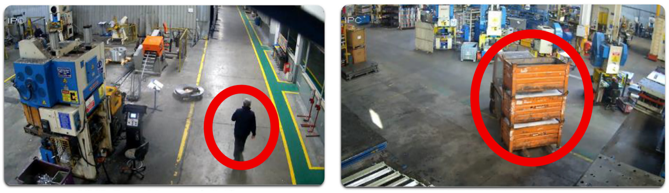

Fig. 3. Example of unsafe images from dataset [13]. Left: Safe Walkway Violation (camera 1). Right: Carrying Overload with Forklift (camera 2)

TABLE I
DATASET SPLITTING SUMMARY

Split	% of Safe	# Safe	# Unsafe	# Total
Train	72%	2,536	0	2,536
Test	28%	994	1,050	2,044

C. Clustering patameters

HDBSCAN clustering is mostly influenced by three parameters. $min\_cluster\_size$ which controls granularity, $min\_samples$ which controls how conservative the algorithm is in forming clusters and $metric$ that can adapt to the data's geometry. During our preprocessing of the data, the image extraction of the videos results in about 40 images of a same video. Since we want the clustering to abstract each video but rather cluster more general meaning, there should be at least 40 datapoints per cluster, so $min\_cluster\_size = min\_samples = 40$. The euclidean $metric$ is the default value we used.

k-Means clustering strongly depends on the parameter $n_{clusters}$ (or k), which influences how many clusters the algorithm will be forced to find. Since we want to cluster in an unsupervised way, we are not aware of the best number of clusters. Multiple candidate values of k were automatically and dynamically tested for each token, and clustering quality was assessed using the silhouette score. The testing range for k was constrained from one cluster up to half the number of videos to preserve the abstract interpretation of a typical situation rather than producing clusters for each individual video. Given 113 videos, all integer k in the interval $[0 : 50]$ were valuated.

D. Evaluation Metrics

The models yield True when the image is predicted as an anomaly or False when it is predicted as a normal behaviour, based on a threshold. The threshold has been choosen to maximise precision, recall and specificity, with more importance to recall. A True Positive is a real anomaly, a True Negative is a real normal behaviour, a False Positive is an actual normal behaviour that was predicted as abnormal, and a False Negative is an actual anomaly that was predicted as a normal behaviour.

Based on that, we will compute:

- Recall: of all actual anomalies, how many were detected. This metric is critical and should be prioritized since missed anomalies can have severe consequences.
- Precision: of the images flagged as anomalies, how many are actual anomalies. It is more acceptable that is metrics is lower compared to recall because false alarms can be tolerated.
- False Positive Rate: of all actual normal behaviours, how many were flagged as abnormal.
- Specificity: of all actual normals, how many were flagged normal.
- F1-Score: single value balance of precision and recall. This accounts only for the positive class.
- Balanced Accuracy: accuracy that give equal weight to both classes.

TABLE II
MODEL PERFORMANCE METRICS FOR GLOBAL EMBEDDINGS. UPPER ROWS FOR KMEANS AND LOWER ROWS FOR HDBSCAN.

Emb.	Thresh.	Prec.	Rec.	FPR	Spec.	F1	B. Acc.
ViT	0.77	**0.80**	0.80	**0.21**	**0.79**	**0.80**	**0.79**
	0.90	0.51	1.00	1.00	0.00	0.68	0.50
DINOv2	0.76	0.78	0.79	0.23	0.77	0.79	0.78
	0.20	0.57	0.82	0.66	0.34	0.67	0.58
I-JEPA	**0.75**	0.78	**0.82**	0.24	0.76	**0.80**	**0.79**
	0.90	0.51	1.00	1.00	0.00	0.68	0.50

TABLE III
MODEL PERFORMANCE METRICS FOR LOCAL EMBEDDINGS. UPPER ROWS FOR KMEANS AND LOWER ROWS FOR HDBSCAN.

Emb.	Thresh.	Prec.	Rec.	FPR	Spec.	F1	B. Acc.
ViT	0.77	**0.83**	**0.81**	**0.17**	**0.83**	**0.82**	**0.82**
	0.90	0.51	1.00	1.00	0.00	0.68	0.50
DINOv2	0.85	0.77	0.80	0.26	0.74	0.78	0.77
	0.90	0.51	1.00	1.00	0.00	0.68	0.50
I-JEPA	0.76	0.81	0.80	0.20	0.80	0.80	0.80
	0.70	0.54	0.99	0.90	0.10	0.70	0.54

V. RESULTS AND DISCUSSION

Table II shows the metric values for the chosen threshold for the global embedding method. The columns represent the metrics and the main rows represent each embedding type. Each main row is divided into upper values for kMeans and lower values for HDBSCAN. Table III is to be read the same and shows the local embedding method results.

For the local models, k-means required on average 1 h 2 min 2 s (± 2 min 51 s), and HDBSCAN required 32 min 57 s (± 14 min 23 s) to process the 2 044 test samples in batches. In contrast, the global models required only 1/256 of the corresponding local model runtimes, that is, approximately 15 s for k-Means and 8 s for HDBSCAN. This factor of 256 arises from the number of patches, since the local models compute each patch sequentially.

Here are our outcomes across three axes of comparison:

1) Global vs Local Embeddings: Both global and local embeddings reach roughly the same quality in detection of anomalies as balanced accuracy is around 80% for both, but the patch-based approach yields a small consistent advantage (+3% in balanced accuracy). Currently, this modest gain comes at a high computational cost, as 256 patch embeddings must be extracted and scored per image. Executed sequentially, this is about 256 times more expensive than computing a single global embedding, though parallelization could mitigate this overhead. The global embedding approach is capable of performing near real-time batch processing, handling 15 s segments of images within each corresponding 15 s interval.

2) k-Means vs HDBSCAN: In our experiments HDBSCAN did not produce a useful separation between safe and anomalous images: it labeled most inputs as anomalous under the current thresholding scheme, which makes it unsuitable in its current configuration. By contrast, the k-Means pipeline paired with a K-NNN variant retrieval strategy achieved practical operating points, reaching recalls up to 81% while maintaining false positive rates at 17%, which offers a useful balance between detection sensitivity and false alarms.

3) ViT vs DINOv2 vs I-JEPA: The three embedding backbones deliver comparable performance. ViT shows a small but consistent advantage, outperforming the others by roughly 3–5% in balanced accuracy across comparable thresholds.

Finally, while the supervised, closed-set Unsafe-Net system [14], which combines YOLOv4 for spatial object detection with ConvLSTM for temporal action recognition, achieves a mean classification accuracy of 95.81% on known-class recognition, our method is designed for unsupervised detection. Although our current results are less competitive, the approach has the advantage of being simpler and potentially more adaptable, as it does not depend on prior knowledge of anomaly types and could be extended to a wider range of scenarios.

VI. Conclusion

In this work, we presented a unified, unsupervised framework for visual anomaly detection VAD and evaluated for dynamic, fixed-view scenes. We conducted a rigorous comparative study between different backbones, ViT, I-JEPA, and DINOv2, evaluating their performance by independently modeling global and local normality distributions. Our approach effectively adapts to a wide range of anomaly scenarios. Experiments showed that local embeddings yield a small but consistent advantage over global embeddings, improving balanced accuracy by +3% (best case: ViT local, F1 = 82%, balanced accuracy = 82%) and enabling fine-grained, per-patch localization. In contrast, global embeddings deliver comparable detection performance (balanced accuracy = 79%) while incurring lower per-image cost (256× faster), making them well-suited for near-real-time batch processing (e.g., handling 15 s image segments).

Our evaluation showed that the k-Means + K-NNN variant retrieval strategy achieved the best operational trade-off between sensitivity and false alarms, reaching recalls up to 81% while keeping false positive rates around 17%, and producing the top F1 and balanced-accuracy scores (F1 = 82%, balanced accuracy = 82% for ViT local).

A primary direction for future work is to evaluate our pipeline on a wider range of established VAD datasets. We acknowledge that the present evaluation was conducted on a reduced dataset; however, its value lies in the dynamic nature of the scenes and in the fact that only precise situations are labeled as anomalous, making the detection task particularly challenging and realistic.

As a next step, we are especially interested in leveraging the attention matrices of Vision Transformers (ViTs) to combine global and local embeddings for anomaly scoring. Local embeddings appear to capture complementary and pertinent information, and we expect that with some adjustments in the scoring strategy, they could be further leveraged to improve anomaly detection performance. This approach is motivated by our observation that certain patches are highly sensitive. We therefore propose grouping them into more stable regions rather than using each patch independently, using the attention matrices. This pipeline adjustment aims to capture both coarse scene-level deviations and fine-grained local irregularities, resulting in a more robust and comprehensive anomaly detection model.

Additionally, we are interested in extending our approach to video anomaly detection by leveraging V-JEPA2 [16] embeddings, which inherently encode motion information. This extension would enable modeling of temporal dynamics beyond static images, potentially improving detection of anomalies involving movement.

References

[1] M. Ben Ammar, A. Mendoza, N. Belkhir, A. Manzanera, and G. Franchi, ”Foundation Models and Transformers for Anomaly Detection: A Survey”, 2025, arXiv:2507.15905.

[2] A. Dosovitskiy, et al. . ”An image is worth 16x16 words: Transformers for image recognition at scale”, 2020. In proceedings of International Conference on Learning Representations, arXiv:2010.11929.

[3] M. Oquab, et al.: ”DINOv2: Learning robust visual features without supervision”, 2024, arXiv:2304.07193v2.

[4] M. Assran, et al., ”Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture,” 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 15619-15629, doi: 10.1109/CVPR52729.2023.01499.

[5] Y. Lee, H. Lim, S. Jang, and H. Yoon, ”UniFormaly: Towards Task-Agnostic Unified Framework for Visual Anomaly Detection”, 2023, arXiv:2307.12540v2.

[6] K. Roth, et al., ”Towards Total Recall in Industrial Anomaly Detection,” IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 14298-14308, doi: 10.1109/CVPR52688.2022.01392.

[7] K. Ishida, et al., ”SA-PatchCore: Anomaly Detection in Dataset With Co-Occurrence Relationships Using Self-Attention,” in IEEE Access, vol. 11, pp. 3232-3240, 2023, doi: 10.1109/ACCESS.2023.3234745.

[8] J. Hu, et al., ”DMAD: Dual Memory Bank for Real-World Anomaly Detection”, 2024, https://arxiv.org/abs/2403.12362.

[9] Y. Sun, H. Wang, Y. Hu, H. Jiang and B. Yin, ”MBMF: Constructing memory banks of multi-scale features for anomaly detection”, 2023, IET Computer Vision, 18(3) pp. 355-369, doi:10.1049/cvi2.12258.

[10] S. Yellapragada, et al. ”Leveraging Registers in Vision Transformers for Robust Adaptation”, (2025), arXiv:2501.04784v1.

[11] T. Liu, and al., ”Component-aware anomaly detection framework for adjustable and logical industrial visual inspection”, Advanced Engineering Informatics, Volume 58, 2023, 102161, ISSN 1474-0346

[12] M. P. Ronecker, et al. ”Vision Foundation Model Embedding-Based Semantic Anomaly Detection”, 2025, arXiv:2505.07998.

[13] O. Önal and E. Dandıl, ”Video dataset for the detection of safe and unsafe behaviours in workplaces”, Data in Brief, vol. 56, 2024,

[14] O. Önal and E. Dandıl, ”Unsafe-Net: YOLO v4 and ConvLSTM based computer vision system for real-time detection of unsafe behaviours in workplace”, Multimedia Tools and Applications, 2024, https://doi.org/10.1007/s11042-024-19276-8

[15] O. Nizan and A. Tal, ”k-NNN: Nearest Neighbors of Neighbors for Anomaly Detection”, 2023, arXiv:2305.17695v1

[16] M. Assran, et al. ”V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning”, 2025, arXiv:2506.09985.