FSLC: Fast Scoring with Learnable Coreset for Zero-shot Industrial Anomaly Detection

Songtao Ni, Yuxin Li, Xu Zhao
School of Automation and Intelligent Sensing,
Shanghai JiaoTong University, China

BMVC 2025

Introduction

- **Background:**
  In some contexts of industrial defect inspection, training data of any kind is unavailable. Consequently, the system must learn to identify anomalies solely from the test samples themselves as displayed in Figure 1.
- **Motivation:**
  - The absence of annotated training data results in a lack of prior knowledge.
  - Algorithms deployed in industrial settings must satisfy demanding criteria for both inference speed and predictive accuracy.

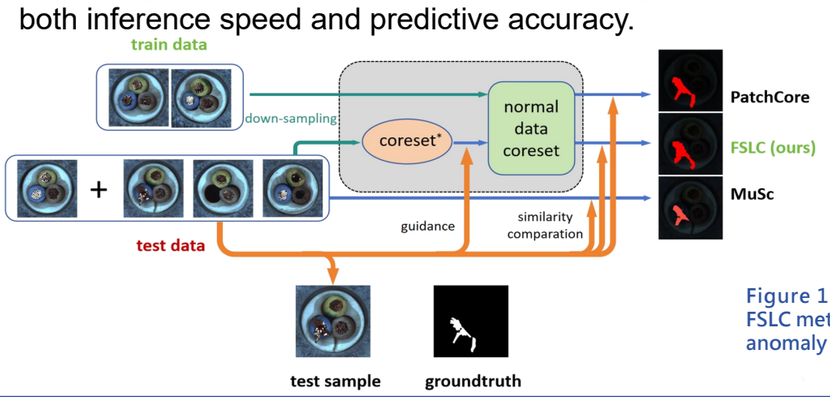
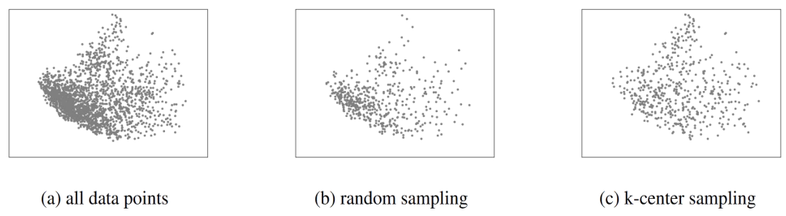


Figure 1: Comparison between our FSLC method and other unsupervised anomaly detection methods.

Main Idea

- **Assumption:**
  Within a batch of data from an industrial line, normal patterns are statistically dominant, occurring in most instances, while any given anomaly is sparse, being confined to a very small fraction of images.
- **Design Concept:**
  - A coreset of sparse down-sampled features provides neighborhood-based coverage that approximates the full feature distribution in the feature space as shown in ***Figure 2***.
  - An effective anomaly score can be derived by incorporating prior assumptions about the data distribution.



(a) all data points (b) random sampling (c) k-center sampling

Figure 2: Visualization of the differences between random sampling and k-center sampling.

Method

1. Feature Extraction:

- ***Vision Transformer***:
  Image encoder from pretrained CLIP network from OpenAI
- ***Multi-scale***:
  Output features from different layers

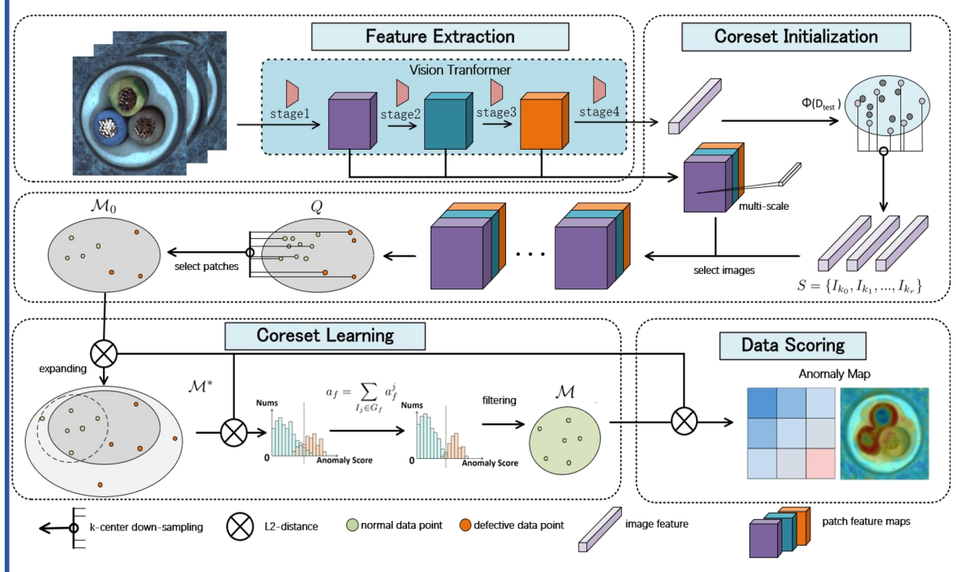
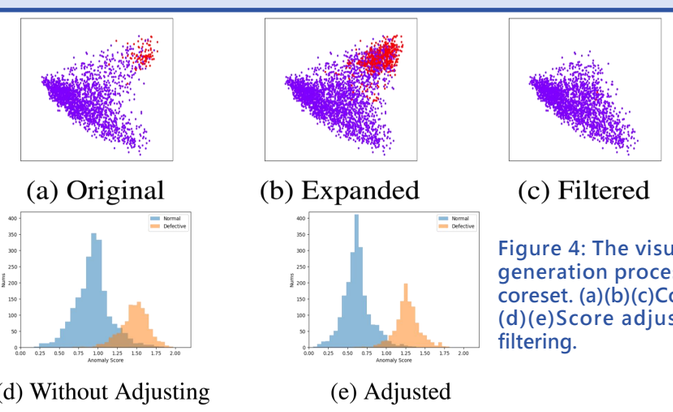


Figure 3: Overview of FSLC. Following the extraction of muti-scale local features, an initialization and a subsequent learning process are employed to form a coreset. This coreset is then utilized to reassess anomaly scores for the data.

2. Coreset Initialization:

- ***Stepwise Down-sampling***:
  The procedure begins with the sparse sampling of r images. Following this, an initial coreset $\mathcal{M}_0$ is formed by sampling from the aggregate of all local features of these images.



(a) Original (b) Expanded (c) Filtered

(d) Without Adjusting (e) Adjusted

Figure 4: The visualization of the generation process involving the coreset. (a)(b)(c)Coreset evolution. (d)(e)Score adjusting in coreset filtering.

3. Coreset Learning:

- ***Expanding***:
  Add several other features into $\mathcal{M}_0$ to get $\mathcal{M}^*$ (***Figure 4(a)*** to ***Figure 4(b)***)
- ***Filtering***:
  Remove features with high anomaly scores from $\mathcal{M}^*$ to get $\mathcal{M}$ (***Figure 4(b)*** to ***Figure 4(c)***)
  Use tricks of score-adjusting to separate different features better in ***Figure 4(d)(e)***

$$a_f^j = \min_{n=1}^{H\times W} \| f - f_j^n \|_2 \, , a_f^* = \sum_{j=1}^{N} a_f^j = \sum_{j=1}^{N} \min_{n=1}^{H\times W} \| f - f_j^n \|_2$$

$$G_f = \{I_k | a_f^k < a_f^j, I_k \in D_{test}, \forall I_j \in D_{test} - G_f, |G_f| = K\},$$

$$a_f = \sum_{I_j \in G_f} a_f^j = \sum_{I_j \in G_f} \min_{n=1}^{H\times W} \| f - f_j^n \|_2 \, .$$

4. Data Scoring:

- ***Pixel Scores***
  Nearest distance in coreset
- ***Image Scores***
  Evaluate by integrating max-N scores

$$s(f) = \min_{f' \in \mathcal{M}} \| f - f' \|_2$$

$$S(I_i) = s(f^*) \cdot w(I_i) \, ,$$

$$where \;\; f^* = \arg\max_{f \in I_i} s(f) \, ,$$

$$and \;\; w(I_i) = 1 - \frac{\exp(s(f^*))}{\sum_{f \in \mathcal{N}_b(f^*)} \exp(s(f))} \, .$$

Experiment

- **Comparison with other AD methods on MVTec AD (AUROC, pixel / image-level)**

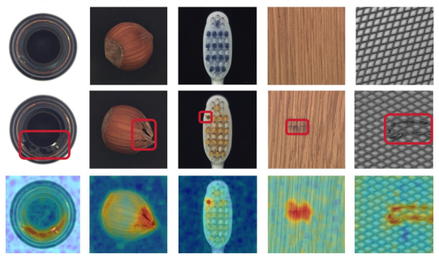
category	PatchCore (k=4)	WinCLIP	WinCLIP+ (k=4)	FastRecon (k=4)	AnomalyCLIP	MuSc	FSLC
Bottle	98.60 / 99.60	89.50 / 99.20	97.80 / 99.30	98.50 / 99.44	90.40 / 89.30	98.60 / 99.90	**99.13 / 100.00**
Cable	**97.90** / 97.40	77.00 / 90.90	94.90 / 88.40	96.12 / 93.79	78.90 / 69.80	96.30 / 99.00	93.88 / **99.74**
Capsule	97.70 / 66.30	86.90 / 82.30	96.20 / 77.30	98.96 / 90.07	95.80 / 87.20	98.90 / **96.70**	**98.98** / 65.14
Carpet	99.00 / 99.00	95.40 / **100.00**	99.30 / **100.00**	99.15 / 99.90	98.80 / **100.00**	**99.50** / 99.90	99.20 / 99.96
Grid	70.60 / 63.00	82.20 / 98.80	98.00 / 99.60	86.32 / 88.81	97.30 / 97.00	**98.40** / 98.70	98.28 / **99.62**
Hazelnut	97.00 / 92.80	94.30 / 93.90	98.80 / 98.40	98.59 / 99.32	97.10 / 97.20	**99.40** / 99.60	98.46 / **99.96**
Leather	96.90 / **100.00**	96.70 / **100.00**	99.30 / **100.00**	99.20 / **100.00**	98.60 / 99.80	99.70 / **100.00**	**99.82** / **100.00**
Metal_Nut	97.00 / 94.70	61.00 / 97.10	92.90 / 99.50	**98.72** / 99.12	74.40 / 93.60	86.00 / 96.30	81.05 / **99.95**
Pill	96.90 / 89.00	80.00 / 79.10	97.10 / 92.80	**98.32** / 93.48	92.00 / 81.80	97.60 / **96.40**	98.16 / 94.76
Screw	92.10 / 54.10	89.60 / 83.30	96.00 / **87.90**	97.10 / 62.46	97.50 / 81.10	**98.90** / 83.50	96.54 / 57.90
Tile	96.00 / **100.00**	77.60 / **100.00**	96.60 / 99.90	96.73 / **100.00**	94.60 / **100.00**	**98.10** / **100.00**	98.07 / **100.00**
Toothbrush	98.80 / 95.20	86.90 / 87.50	98.40 / 96.70	99.04 / 93.61	91.90 / 84.70	99.50 / **100.00**	**99.65** / 96.94
Transistor	**95.00** / 98.40	74.70 / 88.00	88.50 / 85.70	94.18 / 97.29	71.00 / 92.80	92.00 / **99.10**	94.89 / 95.58
Wood	93.10 / 97.40	93.40 / 99.40	95.40 / **99.80**	94.94 / 99.29	96.50 / 96.80	97.40 / 98.50	**98.02** / 98.87
Zipper	98.30 / 95.50	91.60 / 91.50	94.20 / 94.50	**98.89** / 96.95	91.40 / 98.50	98.40 / **99.90**	94.27 / 94.49

- **Time Cost Compared to MuSc (On *Leather* Category)**

MuSc	feature extraction	scoring
time / ms	190.29	1531.08

FSLC	feature extraction	coreset expanding	coreset filtering	scoring
time / ms	170.41	7.46	6.45	2.44

- **Visualization results**

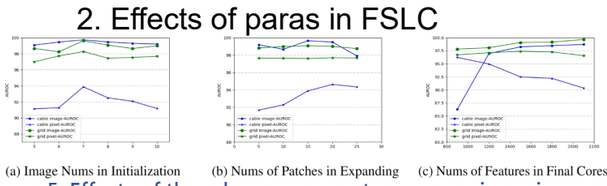


- **Ablation Experiment**

1. Effects of steps in FSLC

	MVTec image	MVTec pixel	VisA image	VisA pixel
FSLC(ours)	93.52→	96.55→	78.95→	95.91→
random image sampling	93.13 ↓	92.33 ↓	79.92 ↑	94.12 ↓
random patch sampling	91.16 ↓	83.21 ↓	75.30 ↓	82.23 ↓
w/o coreset expanding	93.88 ↑	96.11 ↓	72.70 ↓	89.89 ↓
w/o score adjusting	90.42 ↓	93.42 ↓	65.60 ↓	93.91 ↓

2. Effects of paras in FSLC



(a) Image Nums in Initialization (b) Nums of Patches in Expanding (c) Nums of Features in Final Coreset

Figure 5: Effects of three hyper-parameters comparisons in anomaly classification and segmentation AUROC(%) on some categories in MVTec AD dataset.

Conclusion

- **Contribution:**
  - We propose an algorithm suitable for zero-shot unsupervised industrial anomaly detection scenarios, which learns knowledge from test data and continuously improves performance as the test data increases.
  - We propose a coreset learning method that relies solely on unlabeled data to realize the expression of normal data.
  - Our algorithm achieves about 15 times speedup over MuSc on datasets like MVTec AD while maintaining comparable AUROC performance.
- **Prospect:**
  - Weaker assumptions about data distribution.
  - Adaptation to massive data fed in batches.
  - Further improvement of defect segmentation performance.
  - Adaptation to rotational invariance of certain categories.