Superpixel-based Anomaly Detection for Irregular Textures with a Focus on Pixel-level Accuracy

Mehdi Rafiei[1], Toby P. Breckon[2] and Alexandros Iosifidis[1]
[1]*DIGIT, Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark*
[2]*Department of Computer Science, Durham University, Durham, UK*
{rafiei, ai}@ece.au.dk, toby.breckon@durham.ac.uk

***Abstract*—Recent anomaly detection methods achieve high performance on commonly used image and pixel-level metrics. However, due to the imbalance in the number of normal and abnormal pixels commonly encountered in anomaly detection problems, commonly adopted pixel-level performance metrics cannot effectively evaluate model performance. This paper proposes a novel approach for anomaly detection within the irregular texture domain, focusing on pixel-level accuracy metrics suitable for such imbalanced problems. The proposed Superpixel-based Coupled-hypersphere-based Feature Adaptation (Sp-CFA) method leverages the intermediate adaptive representation of superpixels to enable superior pixel-level anomaly detection performance. We demonstrate superior performance over the irregular texture classes within the MVTec AD benchmark dataset, KSDD2 dataset, and an X-ray dataset of manufactured fibrous products.**

***Index Terms*—Anomaly detection, Superpixel, Metric selection, Irregular texture**

I. Introduction

The ability to detect anomalies in irregular texture products plays a crucial role in many industries for ensuring product quality, reliability, and safety. In these cases, anomaly detection at the pixel level poses unique challenges due to the complex nature of irregular textures. Unlike objects and regular texture products, irregular textures lack consistent patterns or repetitive structures, making it challenging to define a clear baseline for anomaly detection. Moreover, anomalies in irregular texture products can vary significantly in size, shape, and appearance, further exacerbating the detection task.

In recent years, significant advancements have been made in anomaly detection methods, leveraging statistical analyses and machine learning. In this regard, several deep learning models have been recently proposed for texture surface anomaly detection. Examples of unsupervised deep learning models trained on only non-anomalous (normal) data samples include the autoencoder-based models in [1–4] and the transformer-based model in [5] used to detect anomalies in texture products. A zero-shot method is proposed in [6], which is not trained on normal samples. It works by receiving a sample, extracting features, and assessing whether any region appears differently from others. A domain-generalized textured surface anomaly detection model is proposed in [7]. This model observes normal and abnormal data across multiple source domains and is expected to be generalized to an unseen textured surface of interest. An overview of recent deep models used for anomaly detection on industrial texture products is available in [8].

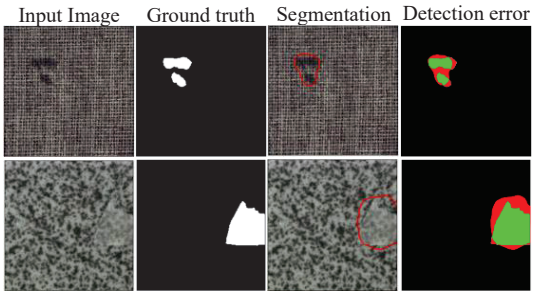

Fig. 1. Anomaly detection examples using CFA [11]. From left to right: input image, ground truth, anomaly segmentation, and detection error (red: false positive or false negative, green: true positive).

Prior works generally consider visual anomaly detection as either a high-level per-image (i.e., image-level binary decision) or low-level per-pixel (i.e., pixel-level segmentation) task. These works often rely on metrics such as Area Under the Receiver Operating Characteristic Curve (AUROC) and F1-score to evaluate the image-level performance and AUROC and Area Under the Per-Region-Overlap curve (AUPRO) [9] to evaluate the pixel-level performance of anomaly detection models. Although these performance metrics are suitable for image-level anomaly detection, they may not be appropriate when used for pixel-level performance evaluation, considering the severe class imbalance between abnormal and normal pixels in almost all anomaly detection problems [10]. As highlighted in [10] Precision-Recall-based metrics, such as the Area Under the Precision-Recall Curve (AUPR) and F1-score, are more suitable as they effectively account for performance disparities resulting from an imbalanced minority (abnormal) class.

By further looking at the qualitative outputs produced by existing methods, it can be seen that the main reason behind their low pixel-level performance on AUPR and F1-score is their lack of precision around anomaly region borders (e.g., Figure 1). To address this problem, in this paper, we propose a new approach for anomaly detection on irregular texture products by exploiting superpixels [12] as an intermediate spatial representation to enhance model precision within the normal-to-abnormal boundary regions and hence more precisely delineate the boundaries of anomalous regions. This is achieved by partitioning the image into superpixels at different levels and incorporating this information into the anomaly detection

process, leading to more accurate localization of anomalies and an overall improvement in pixel-level performance for irregular texture examples.

The key contributions of this work are:

- an extension to the existing region-based paradigm in anomaly detection to consider the use of superpixels as an intermediate spatial representation to facilitate improved pixel-level anomaly detection in normal-to-abnormal boundary regions.
- the primary consideration of AUPR and F1-score as per-pixel anomaly detection performance metrics capable of better delineating relative performance in the presence of significant normal to abnormal class imbalance (as is commonplace in anomaly detection problems).
- demonstrable state of the art performance on the irregular texture classes within the MVTec AD benchmark dataset [13], Kolektor Surface-Defect Dataset 2 (KSDD2) [14], and an X-ray dataset of manufactured fibrous products [15] in comparison to eight leading contemporary anomaly detection methods [11, 16–22].

II. Related work

The method proposed in this paper exploits superpixels to enhance the ability of the Coupled-hypersphere-based Feature Adaptation (CFA) method [11] for delineating the boundaries of anomalous regions with greater precision. Therefore, in this section, the original CFA method and the Simple Linear Iterative Clustering (SLIC) method [12], used to extract superpixels, are briefly described.

A. Coupled-hypersphere Feature Adaptation

CFA leverages transfer learning to mitigate the bias of pre-trained convolutional neural networks (CNN) by learning patch features from normal samples in the target dataset. It ensures that these patch features have a high density around the memorized features, thus addressing the problem of over-estimating abnormality in pre-trained CNN. As it can be seen in Figure 2, the patch features, denoted as $F \in \mathbb{R}^{D\times H\times W}$, are obtained by inferring samples from the target dataset using a biased CNN. Since the feature maps have different spatial resolutions at each depth of the CNN, they are interpolated and concatenated. Therefore, H and W denote the height and width of the largest feature map, respectively, and D indicates the sum of the dimensions of the sampled feature maps. To convert the patch features into target-oriented features, CFA employs an auxiliary network called the patch descriptor $\phi(\cdot) : \mathbb{R}^{D} \rightarrow \mathbb{R}^{D'}$.

During the training phase, CFA utilizes a memory bank C to store the initial target-oriented features obtained from a training set of only normal samples. These features are stored according to a specific modeling procedure. The key idea behind CFA is to perform contrastive supervision based on coupled-hyperspheres created using the memorized features $\mathbf{c} \in C$ as centers. This procedure involves optimizing the parameters of the patch descriptor $\phi(\cdot)$ by minimizing the Coupled-hypersphere-based Feature Adaptation loss function.

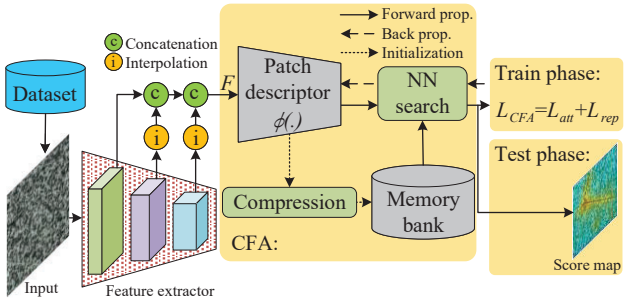

Fig. 2. Overall structure of the CFA model.

This loss comprises two terms: L_{att} and L_{rep}. L_{att} attracts the patch features $\mathbf{p}_t$ towards the coupled-hypersphere with the k-th nearest neighbor ($\mathbf{c}_t^k$) as the center and is defined as:

$$L_{att} = \frac{1}{TK}\sum_{t=1}^{T}\sum_{k=1}^{K}\max\left(0, D\left(\phi(\mathbf{p}_t), \mathbf{c}_t^k\right) - r^2\right), \quad (1)$$

where K represents the number of nearest neighbors matching with $\mathbf{p}_t$, T is the number of patch features obtained from a single sample, and D denotes the predefined distance metric, such as the Euclidean distance. L_{att} encourages the gradual embedding of $\mathbf{p}_t$ closer to the hypersphere created with $\mathbf{c}_k^t$, thereby facilitating feature adaptation.

CFA incorporates hard negative features to enhance the effectiveness of the patch descriptor $\phi(\cdot)$ further. These hard negative features are the K+j-th nearest neighbor $\mathbf{c}_t^j$ of $\mathbf{p}_t$, obtained through the nearest neighbor search within the memory bank C. L_{rep}, the contrastive supervision term, is introduced to repel $\mathbf{p}_t$ from the hypersphere created with $\mathbf{c}_t^j$ as the center. L_{rep} is formulated as:

$$L_{rep} = \frac{1}{TJ}\sum_{t=1}^{T}\sum_{j=1}^{J}\max\left(0, r^2 - D\left(\phi(\mathbf{p}_t), \mathbf{c}_t^j\right) - \alpha\right), \quad (2)$$

where J represents the total number of hard negative features used for contrastive supervision, and α is a hyperparameter controlling the balance between L_{att} and L_{rep}. The total loss function is the combination of these two loss functions:

$$L_{CFA} = L_{att} + L_{rep}. \quad (3)$$

By minimizing L_{CFA}, CFA optimizes the weights of the patch descriptor $\phi(\cdot)$ through transfer learning. This process ensures that the patch features become densely clustered and facilitates the distinction between normal and abnormal features.

Finally, the distance between $\mathbf{p}_t$ and C is calculated using $\min_k D(\phi(\mathbf{p}_t), \mathbf{c}_t^k)$ meaning the minimum distance between $\mathbf{p}_t$ and the memorized features in C. This distance expresses the degree of abnormality of $\mathbf{p}_t$, and it is used to define the anomaly score.

B. Simple Linear Iterative Clustering

SLIC [12] is a superpixel-based image segmentation method that aims to group neighboring pixels with similar characteristics (i.e., color similarity and proximity in the image

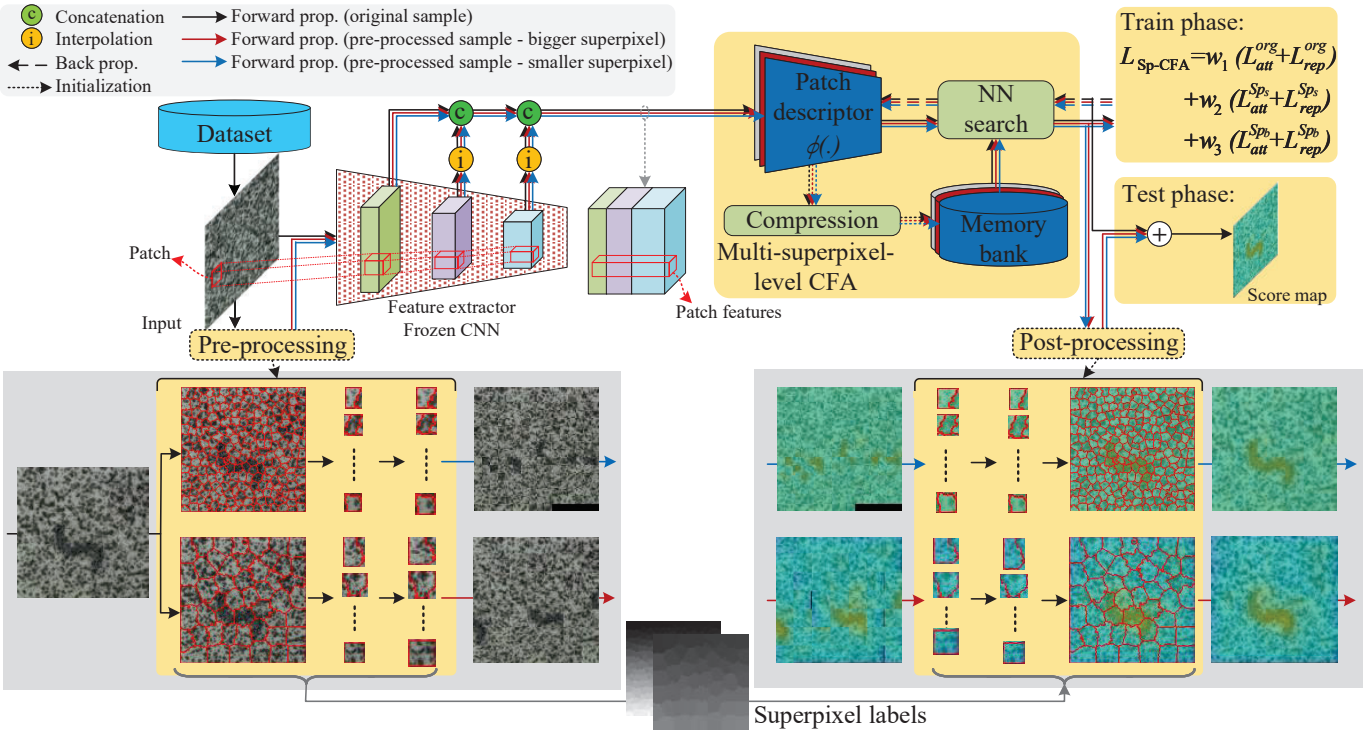

Fig. 3. Overall structure of the proposed Sp-CFA model.

plane) into compact regions. Combining the benefits of K-means clustering with graph-based segmentation allows this to be achieved. Using the regular grid pattern, SLIC generates superpixels by starting with a set of cluster nodes that are constantly updated according to pixel intensity and spatial proximity. This method has been shown to be effective in achieving accurate and efficient image segmentation, making it a popular choice for various computer vision applications.

III. Superpixel-based CFA (Sp-CFA)

As shown in Figure 3, the proposed Sp-CFA model consists of the multi-superpixel-level CFA and two pre-processing and post-processing units. This section describes these units.

A. Multi-superpixel-level CFA

Multi-superpixel-level CFA builds around CFA by exploiting an enriched input. In addition to each original sample, this unit receives two pre-processed inputs. These two additional inputs result from processing the original sample using the pre-processing unit, considering two different superpixel sizes. The rationale behind employing two distinct superpixel sizes is explained in section IV-C. The region sizes of the superpixels, denoted as the region sizes of the bigger superpixel (rs_b) and the region sizes of the smaller superpixel (rs_s), serve as crucial hyperparameters within the model. This leads to three patch descriptor networks interacting with the memory banks. The model is trained in the same way as the original CFA.

Therefore, the total L_{Sp_CFA} loss is formed by three sets of L_{att} and L_{rep} losses, one for the original input and two for the pre-processed inputs (i.e., bigger superpixels (Sp_b) and smaller superpixel (Sp_s)), respectively:

$$L_{Sp\_CFA} = w_1\left(L_{att}^{org}+L_{rep}^{org}\right)+w_2\left(L_{att}^{Sp_s}+L_{rep}^{Sp_s}\right) +w_3\left(L_{att}^{Sp_b}+L_{rep}^{Sp_b}\right). \quad (4)$$

The proposed architecture provides a score map S_t for each input. The two score maps corresponding to the pre-processed inputs are given to the post-processing unit. Subsequently, the score map related to the original input is combined with the two post-processed score maps obtained by the post-processing unit as follows:

$$S_t^{final} = \alpha_1 S_t^{org} + \alpha_2 S_t^{Sp_s} + \alpha_3 S_t^{Sp_b}, \quad (5)$$

where α_1, α_2, and α_3 are hyperparameters chosen using a grid search hyperparameter tuning process.

B. Pre-processing unit

The pre-processing unit receives an original sample and gives two pre-processed versions to the multi-superpixel-level CFA unit and the superpixel labels to the post-processing unit.

Considering two selected superpixel sizes, we have two sets of superpixels for each original sample. For each set, we go through every superpixel and select the rectangular patch that covers that superpixel in the input image. Subsequently, each selected patch is reshaped to a square and resized uniformly via bicubic interpolation. Finally, all the square patches are concatenated to form a new sample having the same size as the original sample. We used the SLIC implementation provided

in the OpenCV library to find the superpixels. Instead of the number of superpixels, this algorithm receives the approximate region size of superpixels as input. Therefore, we do not have precise control over the number of superpixels, and there is a chance that the concatenated patches do not cover the whole new sample, leading to the inclusion of zero patches for matching the size of the new sample to that of the original sample (see an example in Figure 3).

Therefore, the pre-processed outputs are made of square patches, each covering a superpixel, while it breaks the overall spatial relation outside of the superpixels. Utilizing this one-superpixel square patch strategy, in conjunction with the CNN layers in the feature extractor and the patch descriptors $\phi(\cdot)$, ensures a focused analysis of each superpixel. This focused attention is important for improving the model's accuracy. Furthermore, the inherent randomness and irregularity found in irregular texture products ensure the preservation of essential information, even without the overall spatial relation outside the superpixels.

C. Post-processing unit

The post-processing unit receives two score maps related to pre-processed inputs of the multi-superpixel-level CFA unit and gives two post-processed score maps using the received superpixel labels from the pre-processing unit.

Considering the transformations applied by the pre-processing unit, which include reshaping, resizing, and relocating of the superpixels, the pixels in the pre-processed samples and, consequently, in the generated score maps are not located at their original positions. To combine the generated score maps, we need to ensure that the values at the same pixel positions correspond to the same pixel in the original sample.

The post-processing unit receives two score maps and two superpixel labels to accomplish this. The process is the reverse of the one applied in the pre-processing unit. Firstly, knowing the number of superpixels using the superpixel labels, each map is divided into square patches related to the superpixels. Subsequently, using the superpixel labels, the squares are resized, reshaped and, finally, the exact pixels corresponding to the superpixels are selected. In the end, the selected score map pixels related to a superpixel are concatenated to shape the post-processed score maps. These new maps can then be summed up with the one corresponding to the original input.

IV. Evaluation

In addition to the proposal of a new method, our research contributes to addressing the choice of suitable performance evaluation metrics for pixel-level anomaly detection methods. By adopting the AUPR and F1-score metrics, we compare the performance of the proposed method with that of eight leading contemporary anomaly detection methods, considering the class imbalance and emphasizing the detection of anomalies.

A. Experimental Setup

To evaluate the effectiveness of the proposed method, we performed experiments on a subset of the MVTec AD dataset

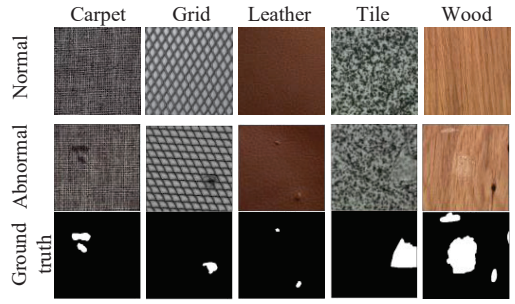

Fig. 4. Normal, abnormal, and ground truth samples from MVTec AD dataset.

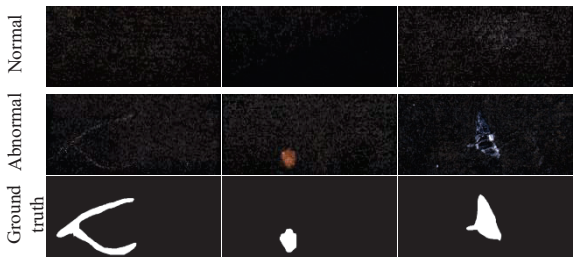

Fig. 5. Normal, abnormal, and ground truth samples from KSDD2 dataset.

comprising of five different texture products covering a range of irregular textures, the KSDD2 dataset, and an X-ray dataset consisting of fibrous products representing a real-world scenario for anomaly detection. The latter allows us to assess the performance of our approach in an industrial context and to validate its effectiveness in identifying anomalies in practical settings.

- **MVTec AD** [13]: This dataset includes 5,354 industrial samples. It is divided into 15 subclasses, of which 5 are texture products (Carpet, Grid, Leather, Tile, and Wood). The subclass Grid is not an irregular texture product; however, we decided to include it in the experiments to evaluate the performance on a regular texture product. A normal and abnormal sample of each class, along with their ground truth, are shown in Figure 4.
- **KSDD2** [14]: This dataset is specifically created for detecting surface defects. It consists of different defect types like scratches, small spots, and imperfections on the surface. The dataset is designed for supervised, weakly-supervised, or unsupervised anomaly detection; therefore, it contains a training set with 246 abnormal and 2,085 normal samples. However, we only used the normal samples during the training phase. Additionally, the test set has 110 abnormal and 894 normal samples. A few normal and abnormal samples with their ground truth are shown in Figure 5.
- **X-ray** [15]: To represent a real-world scenario, assess the performance of our approach in an industrial context, and validate its effectiveness in identifying anomalies in practical settings, the X-ray dataset consisting of fibrous products is used with corresponding segmentation masks. A few normal and abnormal samples, along with their ground truth, are shown in Figure 6.

The statistical information of these datasets is shown in Table I.

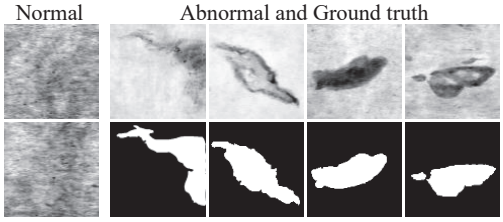

Fig. 6. Normal, abnormal, and ground truth samples from the X-ray dataset.

TABLE I
STATISTICAL OVERVIEW OF MVTEC AD AND X-RAY DATASETS.

		# Train	# Test Norm.	# Test Abn.	# Defect region	Image length	Gray-scale
MVTec	Carpet	280	28	89	97	1024	
	Grid	264	21	57	170	1024	✓
	Leather	245	32	92	99	1024	
	Tile	230	33	84	86	840	
	Wood	247	19	60	168	1024	
KSDD2		2085	894	110	119	230×630	
X-ray		176	60	242	266	42-318	✓

All CNNs used in the experiments were pre-trained on ImageNet. We extracted feature maps from intermediate layers to obtain multi-scale features from the pre-trained CNNs. The hyperparameters related to CFA are set to the values used in the original paper [11]. For all datasets, we used $w_1 = w_2 = w_3 = 1$. A grid search strategy was adopted to search for the α_1, α_2, α_3 values, and the region sizes of the superpixels (rs_s and rs_b). To accomplish this task, the model's performance is thoroughly evaluated across all combinations of the selected values for the hyperparameters, utilizing the validation set. The optimal values selected through this evaluation process are presented in Table II. All experiments are executed five times, and the average results are reported.

Regarding pixel-level performance assessment, we report the results using four metrics, i.e., pixel-level AUROC (P-AUROC), pixel-level AUPRO (P-AUPRO), pixel-level AUPR (P-AUPR), and pixel-level F1-score (P-F1-score). As this study focuses on pixel-level performance in anomaly detection problems, it should be noted that it is possible to obtain lower image-level performance by improving the pixel-level metrics. Therefore, we also report performance using two image-level metrics, i.e., image-level AUROC (I-AUROC), and image-level F1-score (I-F1-score).

B. Quantitative Results

The quantitative performance assessment of the proposed Sp-CFA in comparison with eight other state-of-the-art methods (i.e., Patch-core [16], CFA [11], CFLOW-AD [17], Patch Distribution Modeling (PaDiM) [18], Deep Features Modeling (DFM) [19], EfficientAD [20], Reverse Distillation [21], and FastFlow [22]), is presented in this section[1].

Table III shows the pixel-level results over all five texture classes in the MVTec AD dataset and the average pixel and image-level performance over the classes. First, it can be seen that although the proposed Sp-CFA method improves the mean pixel-level performance based on the AUROC and AUPRO metrics, previous methods already reported good results using these metrics. However, we assert that using these two metrics is not sensible, considering the highly imbalanced normal and abnormal distributed pixels over the dataset. Therefore, AUPR and F1-score are more suitable for the task, and the results in Table III show considerably better performance of Sp-CFA based on these metrics.

Additionally, although the performance of Sp-CFA on the Grid class is promising, it is still similar to the performance achieved by all other methods. This is because Grid is a regular texture product. Our model has the best performance on irregular texture products, which can be seen based on the fact that our method provided significantly higher performance over all other classes (irregular texture classes).

The image-level results on the MVTec AD dataset and all the methods are shown in the last two rows of Table III. It can be seen that although adding superpixels to the CFA method slightly reduced the performance in comparison to the original CFA method, the achieved performance is still high and acceptable. When the task at hand targets image-level performance, some existing methods can be better choices compared to Sp-CFA. However, when considering pixel-level performance, Sp-CFA achieves a much better performance with acceptable image-level performance.

Results on KSDD2 and X-ray datasets measured using pixel and image-level performance metrics are reported in Tables IV and V, respectively. It can be seen that for both datasets, the proposed Sp-CFA method reached the best performance in three out of four pixel-level metrics. The image-level performance is also high, even higher than that of CFA, although it is not the best.

C. Ablation Study

To assess the impact of different pre-trained CNN feature extractors on pixel-level performance, three backbones (i.e., VGG19 [23], ResNet18 [24], and Wide-ResNet50 [25]) were used for the proposed method. The average pixel-level performance on the texture classes of the MVTec AD dataset is shown in Table VI. The results show that Wide-ResNet50 leads to the best performance for all metrics.

To assess the impact of using one or two additional inputs to the Sp-CFA model, the average pixel-level performance on the texture classes of the MVTec AD dataset is shown in Table VII. It can be seen that adding the input corresponding

TABLE II
SELECTED HYPER PARAMETERS.

		α_1	α_2	α_3	rs_s	rs_b
MVTec	Carpet	0.25	0	0.75	10	-
	Grid	0.1	0.1	0.8	10	35
	Leather	0.1	0	0.9	15	-
	Tile	0.15	0.2	0.65	10	30
	Wood	0.15	0.1	0.75	10	40
KSDD2		0.1	0.35	0.55	10	30
X-ray		0.25	0.35	0.4	20	40

[1]We used the Anomalib library https://github.com/openvinotoolkit/anomalib to generate the results for these methods.

TABLE III
PIXEL-LEVEL AND AVERAGE IMAGE-LEVEL RESULTS ON TEXTURE CLASSES IN MVTEC AD DATASET.

		Patch-core	CFA	CFLOW	PaDim	DFM	EfficientAD	Reverse Distillation	FastFlow	Sp-CFA (Ours)
P-AUROC	Carpet	98.8	98.1	98.8	98.8	98.4	94.3	**99.0**	98.4	96.7
	Grid	96.9	95.3	97.4	96.7	92.6	93.7	**98.6**	97.4	97.1
	Leather	99.0	98.9	**99.5**	98.9	98.2	97.5	99.3	99.4	**99.5**
	Tile	95.2	98.5	95.9	94.9	94.3	90.7	93.2	95.0	**98.6**
	Wood	92.8	97.3	95.0	93.9	90.5	87.4	93.9	95.2	**97.6**
	Average	96.6	97.6	97.3	96.6	94.8	92.7	96.8	97.1	**97.9**
P-AUPRO	Carpet	92.8	92.1	93.8	95.3	90.7	88.8	**96.7**	94.5	91.7
	Grid	86.6	86.1	90.2	89.0	79.6	89.0	**95.7**	92.3	91.3
	Leather	96.7	95.3	98.4	98.0	96.0	97.3	97.7	**98.4**	**98.4**
	Tile	85.3	90.6	90.4	86.3	85.2	82.6	81.1	89.6	**94.6**
	Wood	86.2	89.1	88.6	91.6	83.6	79.6	90.6	**93.6**	92.3
	Average	89.5	90.6	92.3	92.0	87.0	87.5	92.4	**93.7**	**93.7**
P-AUPR	Carpet	56.7	53.7	56.5	54.4	50.1	62.5	62.9	54.0	**64.1**
	Grid	26.8	19.3	29.8	26.0	12.5	**51.4**	35.6	28.9	21.9
	Leather	38.3	33.7	50.9	37.9	25.0	54.7	48.6	53.2	**55.3**
	Tile	48.4	84.5	57.0	46.3	46.8	74.4	39.1	61.9	**87.8**
	Wood	42.7	68.2	41.1	40.6	27.7	53.8	36.5	53.5	**73.0**
	Average	42.6	51.9	47.1	41.0	32.4	59.4	44.5	50.3	**60.4**
P-F1-score	Carpet	55.6	58.9	62.0	53.2	51.5	61.9	61.6	55.6	**62.0**
	Grid	33.0	27.7	39.4	34.0	22.0	**49.9**	43.4	41.1	30.3
	Leather	38.7	36.0	55.1	35.9	30.4	55.6	50.4	55.7	**57.4**
	Tile	59.5	77.2	65.4	58.2	55.6	74.0	53.0	62.7	**81.1**
	Wood	45.3	61.7	50.6	43.6	34.0	54.4	46.2	57.4	**65.6**
	Average	46.4	52.3	54.5	45.0	38.7	59.2	50.9	54.5	**59.3**
I-AUROC	Average	98.7	98.3	97.0	98.8	93.5	**99.2**	98.5	96.8	98.2
I-F1-score	Average	98.3	98.1	97.6	97.5	95.1	**98.4**	97.7	94.1	97.9

TABLE IV
PIXEL- AND IMAGE-LEVEL RESULTS ON KSDD2 DATASET.

	Patch-core	CFA	CFLOW	PaDim	DFM	EfficientAD	Reverse Distillation	FastFlow	Sp-CFA (Ours)
P-AUROC	98.3	98.1	98.1	97.5	**98.6**	97.9	98.4	92.0	**98.6**
P-AUPRO	93.0	92.8	92.9	**94.2**	93.3	93.8	94.0	85.6	92.8
P-AUPR	48.1	28.8	49.5	40.3	35.7	46.6	41.4	48.7	**51.2**
P-F1-score	50.6	35.3	52.8	43.6	42.2	50.4	46.2	49.7	**54.4**
I-AUROC	**95.2**	94.1	91.0	90.3	93.0	93.9	92.5	91.9	94.0
I-F1-score	82.5	73.3	76.8	65.1	73.9	78.3	72.0	**84.6**	77.8

TABLE V
PIXEL- AND IMAGE-LEVEL RESULTS ON X-RAY DATASET.

	Patch-core	CFA	CFLOW	PaDim	DFM	EfficientAD	Reverse Distillation	FastFlow	Sp-CFA (Ours)
P-AUROC	90.5	85.8	87.8	88.8	88.4	85.2	82.4	86.0	**90.7**
P-AUPRO	**94.0**	66.2	68.8	74.2	72.5	69.9	61.5	67.8	76.9
P-AUPR	63.6	64.5	49.9	57.6	60.8	57.8	41.5	48.5	**65.5**
P-F1-score	60.9	58.5	55.3	58.4	58.7	46.1	47.8	54.9	**61.5**
I-AUROC	**1**	98.8	96.4	97.3	**1**	99.6	98.6	99.6	99.7
I-F1-score	**1**	97.7	97.4	97.9	**1**	99.8	98.0	99.4	97.5

TABLE VI
AVERAGE PIXEL-LEVEL PERFORMANCE OF THE PROPOSED METHOD WITH VARIOUS PRE-TRAINED CNN FEATURE EXTRACTORS ON MVTEC AD DATASET.

Backbone	P-AUROC	P-AUPRO	P-AUPR	P-F1-score
VGG19	95.6	91.6	58.1	57.4
ResNet18	97.4	93.1	59.9	58.9
WRN50-2	**97.9**	**93.7**	**60.4**	**59.3**

TABLE VII
AVERAGE PIXEL-LEVEL PERFORMANCE OF THE PROPOSED METHOD WITH VARIOUS SUPERPIXEL LEVELS ON MVTEC AD DATASET.

rs_s	rs_b	P-AUROC	P-AUPRO	P-AUPR	P-F1-score
✓		97.8	**93.9**	60.3	59.2
	✓	97.5	92.8	58.8	55.9
✓	✓	**97.9**	93.7	**60.4**	**59.3**

to smaller superpixels to the original CFA led to better performance compared to adding the input corresponding to bigger superpixels. Adding the two inputs corresponding to smaller and bigger superpixels led to the best overall performance, by reaching the best results based on AUROC, AUPR, and F1-score metrics. The enhancement observed by introducing the second superpixel is marginal compared to the improvement achieved by incorporating the smaller superpixel

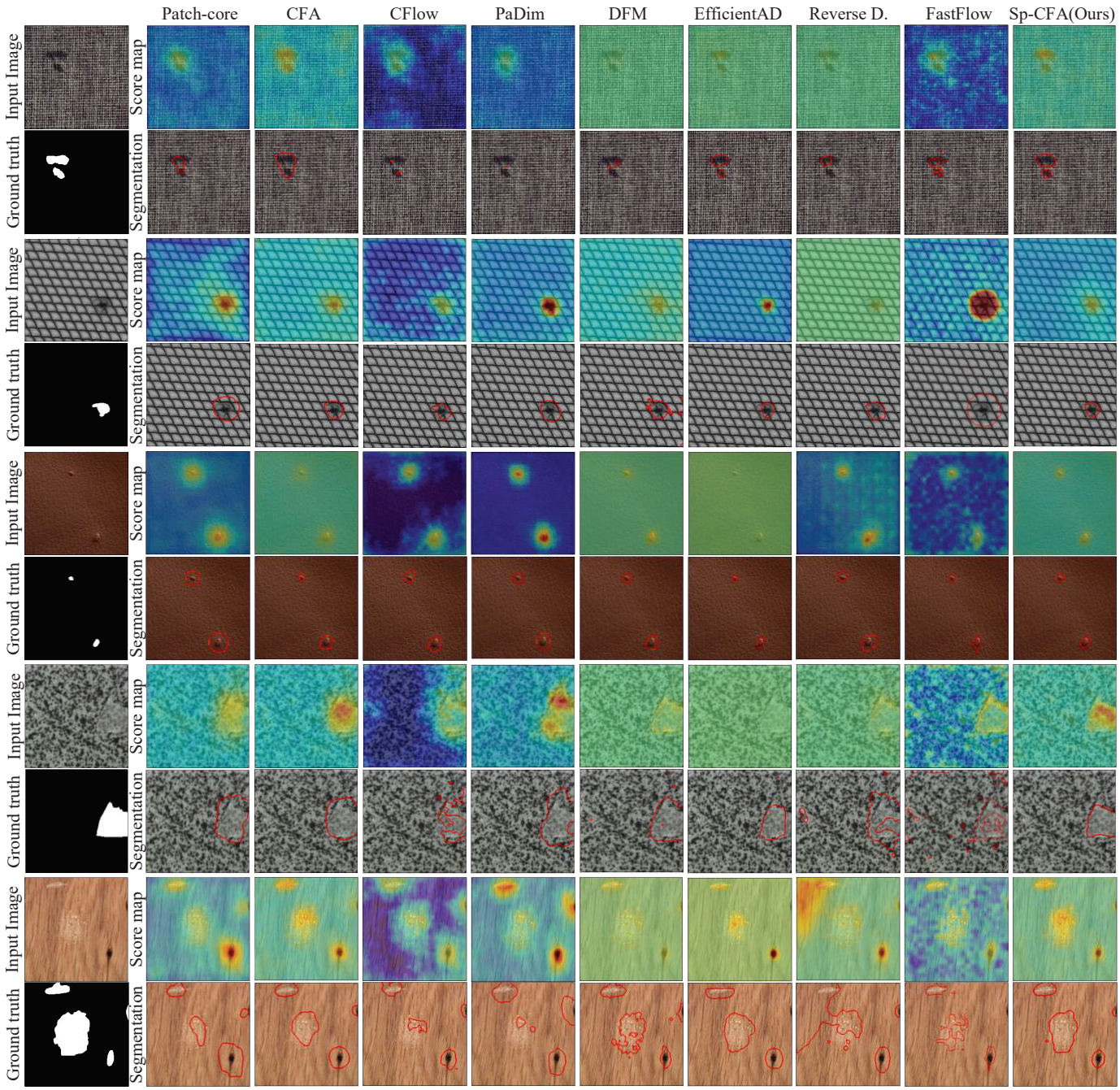

Fig. 7. Visualization of results of anomaly localization for texture classes in MVTec AD dataset.

alone. Therefore, although it is conceivable that adding a third superpixel size might yield further performance gains, the improvement would likely be minimal. Consequently, we opted for simplicity, utilizing only two sizes and avoiding unnecessary complexity for such marginal enhancements.

D. Qualitative Results

Figure 7 includes qualitative results of anomaly localization for randomly selected samples from texture classes in the MVTec AD dataset obtained by our proposed method and the eight competing state-of-the-art methods. The figure shows the input image, ground truth, the predicted score maps for each method and the corresponding segmented abnormal areas. It can be seen that the segmented abnormal areas obtained by the proposed method are more precise to the borderline of defects. Therefore, the consistent effectiveness of the proposed method in accurately localizing abnormal areas on all irregular texture products compared to other methods, even in challenging cases, shows its high qualitative performance.

V. Conclusion

We introduced a superpixel-based method for anomaly detection in irregular texture products, focusing on pixel-level metrics. The proposed method significantly improves anomaly detection performance by leveraging the power of superpixels in Coupled-hypersphere-based Feature Adaptation (CFA) and addressing the challenge of class imbalance. We demonstrated superior performance in identifying abnormal pixels through experiments on the MVTec AD dataset, KSDD2 dataset, and an X-ray dataset of manufactured fibrous products, providing enhanced anomaly detection capabilities. This research has important implications for industries dealing with irregular texture products, offering a practical solution to improve inspection processes and reduce the risk of defective products.

Acknowledgments

The research leading to the results of this paper received funding from the Innovation Fund Denmark as part of MADE FAST.

References

[1] A. Bionda, L. Frittoli, and G. Boracchi, "Deep autoencoders for anomaly detection in textured images using CW-SSIM," *Image Analysis and Processing*, pp. 669–680, 2022.

[2] X. Tao, S. Yan, X. Gong, and C. Adak, "Learning multi-resolution features for unsupervised anomaly localization on industrial textured surfaces," *IEEE Transactions on Artificial Intelligence*, pp. 1–13, 2022.

[3] H. Üzen, M. Türkoğlu, B. Yanikoglu, and D. Hanbay, "Swin-MFINet: Swin transformer based multi-feature integration network for detection of pixel-level surface defects," *Expert Systems with Applications*, vol. 209, p. 118269, 2022.

[4] P. A. Adey, S. Akçay, M. J. Bordewich, and T. P. Breckon, "Autoencoders without reconstruction for textural anomaly detection," *International Joint Conference on Neural Networks*, pp. 1–8, 2021.

[5] X. Tao, C. Adak, P.-J. Chun, S. Yan, and H. Liu, "ViTALnet: Anomaly on industrial textured surfaces with hybrid transformer," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.

[6] T. Aota, L. T. T. Tong, and T. Okatani, "Zero-shot versus many-shot: Unsupervised texture anomaly detection," *IEEE Winter Conference on Applications of Computer Vision*, pp. 5564–5572, 2023.

[7] S.-F. Chen, Y.-M. Liu, C.-C. Liu, T. P.-C. Chen, and Y.-C. F. Wang, "Domain-generalized textured surface anomaly detection," *IEEE International Conference on Multimedia and Expo*, pp. 01–06, 2022.

[8] M. Prunella, R. M. Scardigno, D. Buongiorno, A. Brunetti, N. Longo, R. Carli, M. Dotoli, and V. Bevilacqua, "Deep learning for automatic vision-based recognition of industrial surface defects: A survey," *IEEE Access*, vol. 11, pp. 43370–43423, 2023.

[9] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4183–4192, 2020.

[10] M. Rafiei, T. P. Breckon, and A. Iosifidis, "On pixel-level performance assessment in anomaly detection," *arXiv preprint arXiv:2310.16435*, 2023.

[11] S. Lee, S. Lee, and B. C. Song, "CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization," *IEEE Access*, vol. 10, pp. 78446–78454, 2022.

[12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[13] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9592–9600, 2019.

[14] J. Božič, D. Tabernik, and D. Skočaj, "Mixed supervision for surface-defect detection: From weakly to fully supervised learning," *Computers in Industry*, vol. 129, p. 103459, 2021.

[15] M. Rafiei, D. T. Tran, and A. Iosifidis, "Recognition of defective mineral wool using pruned resnet models," *arXiv preprint arXiv:2211.00466*, 2022.

[16] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328, 2022.

[17] D. Gudovskiy, S. Ishizaka, and K. Kozuka, "CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows," *IEEE Winter Conference on Applications of Computer Vision*, pp. 98–107, 2022.

[18] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: a patch distribution modeling framework for anomaly detection and localization," *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event*, pp. 475–489, 2021.

[19] N. A. Ahuja, I. Ndiour, T. Kalyanpur, and O. Tickoo, "Probabilistic modeling of deep features for out-of-distribution and adversarial detection," *arXiv preprint arXiv:1909.11786*, 2019.

[20] K. Batzner, L. Heckler, and R. König, "EfficientAD: Accurate visual anomaly detection at millisecond-level latencies," *arXiv preprint arXiv:2303.14535*, 2023.

[21] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9737–9746, 2022.

[22] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu, "FastFlow: Unsupervised anomaly detection and localization via 2d normalizing flows," *arXiv preprint arXiv:2111.07677*, 2021.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[25] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.