CFA: Coupled-hypersphere-based Feature Adaptation For Target-Oriented Anomaly Localization (IEEE 2022)

2023. 01. 31

Presenter : Taewoong Kang
twk@deepnoid.com

Abstract

- Anomaly Localization
- Unlike previous studies, we focused on approximating the distribution of normal features with adaptation
- A learnable patch descriptor that learns and embeds target-oriented features
- Scalable memory bank independent of the target dataset

SPADE

- Utilize Memory bank
- Calculated anomaly scores by computing the Euclidean distance

PaDiM

- Define Memory bank by modeling the normal distribution at each location of the feature map (patch-level approach)
- Calculated anomaly scores by using Mahalanobis distance metric

PatchCore

- Used only mid-level feature maps
- Proposed greedy coreset subsampling

Overview

- Performance of anomaly localization depends on the size of the memory bank
  - -)stored as many normal features as possible in a memory bank
  - -) unfitted features may cause risk of overestimated normality and increase the inference time
- =) novel approach to produce target-oriented features
  - =) we define a novel loss function
  - =) we present a scalable memory bank

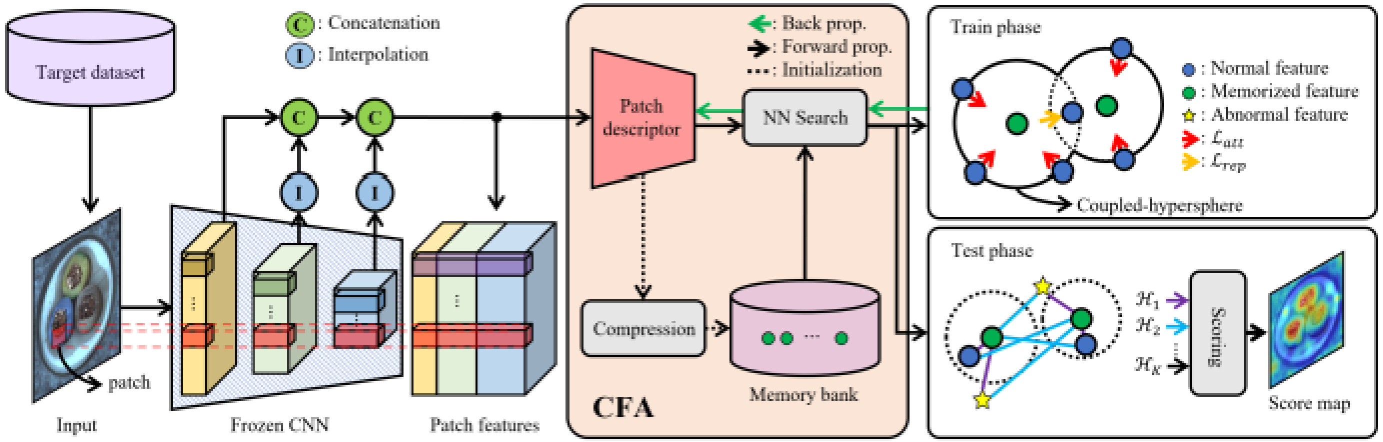


Figure 1. Overall structure of our proposed method (CFA).

Overview

- The feature maps sampled at each depth of CNN are interpolated to have the same resolution and then concatenated
- Patch feature $\mathcal{F} \in \mathbb{R}^{D \times H \times W}$
- Feature $\mathbf{p}_{t \in \{1,\ldots,HW\}} \in \mathbb{R}^{D}$
- Patch descriptor $\phi(\cdot) : \mathbb{R}^{D} \rightarrow \mathbb{R}^{D'}$ (auxiliary network with learnable parameters)
- Memory bank $\mathcal{C}$

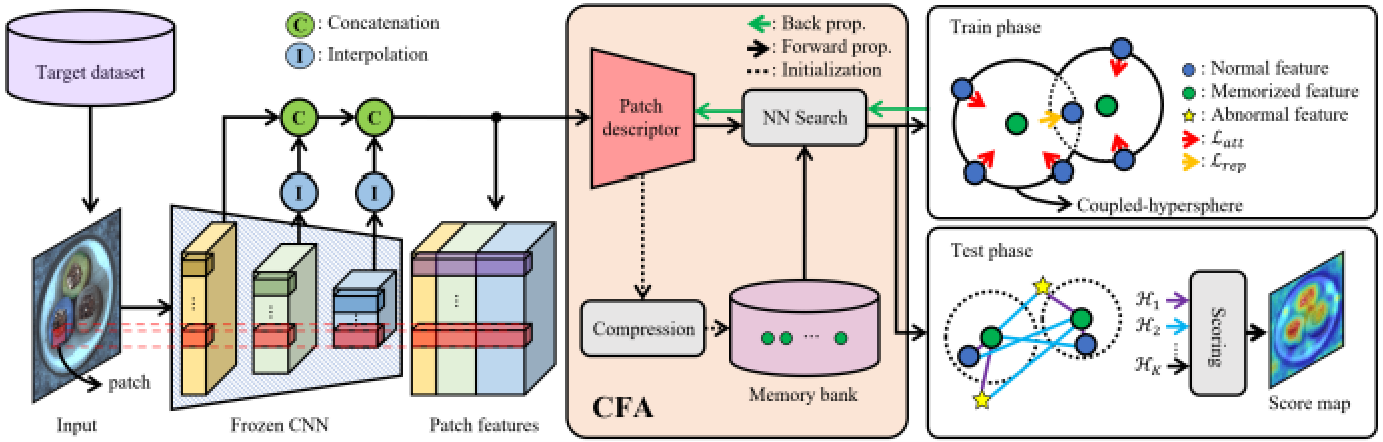


Figure 1. Overall structure of our proposed method (CFA).

CFA

- Coupled-hypersphere-based Feature Adaptation (CFA)
- Extract clustered normal features so that has a high density

- 1. the k-nearest neighbor $\mathbf{c}_t^k$ is searched through the NN search of $\phi(\mathbf{p}_t)$ and $\mathcal{C}$
- 2. CFA supervises $\phi(\cdot)$ so that $\mathbf{p}_t$ is embedded close to $\mathbf{c}_t^k$

- $$\mathcal{L}_{att} = \frac{1}{TK}\sum_{t=1}^{T}\sum_{k=1}^{K}\max\{0, \mathcal{D}(\phi(\mathbf{p}_t), \mathbf{c}_t^k) - r^2\}$$ attracting

- (K : hyperparameter the number of NN
  $T = h \times w$, $\mathcal{D}(\cdot,\cdot)$ is predefined distance metric (Euclidean))

CFA

- To address ambiguous $\phi(\mathbf{p})$ belonging to multiple hyperspheres
- Use hard negative features

- Hard negative features are defined as the K+j−th nearest neighbor $\mathbf{c}_t^j$

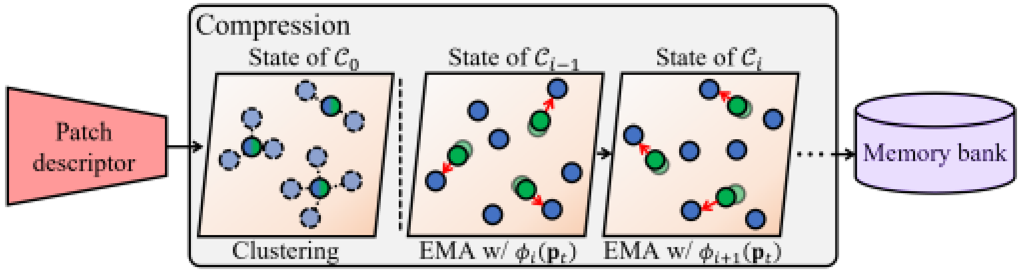
$$\mathcal{L}_{rep} = \frac{1}{TJ}\sum_{t=1}^{T}\sum_{j=1}^{J}\max\{0, r^2 - \mathcal{D}(\phi(\mathbf{p}_t), \mathbf{c}_t^j) - \alpha\}$$

  - hyperparameter J is the total number of hard negative features
  - hyperparameter α is used to control the balance between $\mathcal{L}_{att}$ and $\mathcal{L}_{rep}$

- $\mathcal{L}_{CFA} = \mathcal{L}_{att} + \mathcal{L}_{rep}$

Memory Bank Compression

- 1. an initial memory bank C0 is constructed by K-means clustering to all $\phi_0(\mathbf{p}_{t\in\{1,\ldots,T\}})$
- 2. i-th memory bank of the next state Ci is calculated by EMA (exponential moving average) of $\mathcal{C}_i^{NN}$ and $\mathcal{C}_{i\text{-}1}$
- $\mathcal{C}_i^{NN}$ is the set of nearest patch feature



Algorithm 1 Memory Bank Modeling.

Require:Patch descriptor ϕ, dataset $\mathcal{X}$, EMA parameter β
Initialization: $\mathcal{C}_0 \leftarrow \text{KMeans}\phi_0(\mathbf{p})$
for $i \in \{1,\ldots,|\mathcal{X}|\}$ **do**
  $\mathcal{C}_i^{NN} \leftarrow \{\}$
  for $j \in \{1,\ldots,|\mathcal{C}|\}$ **do**
    $Y \leftarrow (\phi_i(\mathbf{p}) \cup \mathcal{C}_i^{NN}) \cap (\mathcal{C}_i^{NN})^c$
    $\mathcal{C}_i^{NN} \cup \arg\min_{y\in Y} \left\| y - \mathcal{C}_{i-1}^j \right\|_2$
  end for
  $\mathcal{C}_i \leftarrow (1-\beta)\cdot\mathcal{C}_{i-1} + \beta\cdot\mathcal{C}_i^{NN}$
end for
$\mathcal{C} \leftarrow \mathcal{C}_{|\mathcal{X}|}$
return $\mathcal{C}$

Memory Bank Compression

- Effective adaptation
- C of the proposed method is not affected by $|\mathcal{X}|$

Table 1. Complexity estimates of memory bank modeling and memory bank size.

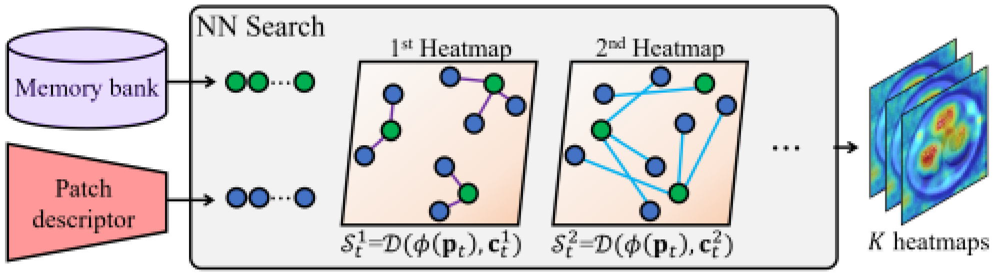
Methods	Modeling	Memory Bank
SPADE	$\mathcal{O}(\|\mathcal{X}\|HWD)$	$\mathcal{G} \in \mathbb{R}^{\|\mathcal{X}\| \times H \times W \times D}$
PaDiM	$\mathcal{O}(\|\mathcal{X}\|HWD^2)$	$\mathcal{N}(\mu, \mathbf{\Sigma}) \in \mathbb{R}^{H \times W \times D^2}$
PatchCore	$\mathcal{O}(\|\mathcal{X}\|HWD')$	$\mathcal{M} \in \mathbb{R}^{\|\mathcal{X}\| \times \gamma(H \times W) \times D'}$
Ours	$\mathcal{O}(HWD')$	$\mathcal{C} \in \mathbb{R}^{\gamma(H \times W \times D)}$

Scoring Function

- Define the anomaly score naively using $\mathcal{D}(\phi(\mathbf{p}_t), \mathbf{c}_t^k)$

$$\mathcal{S}_t = \min_k \mathcal{D}(\phi(\mathbf{p}_t), \mathbf{c}_t^k)$$

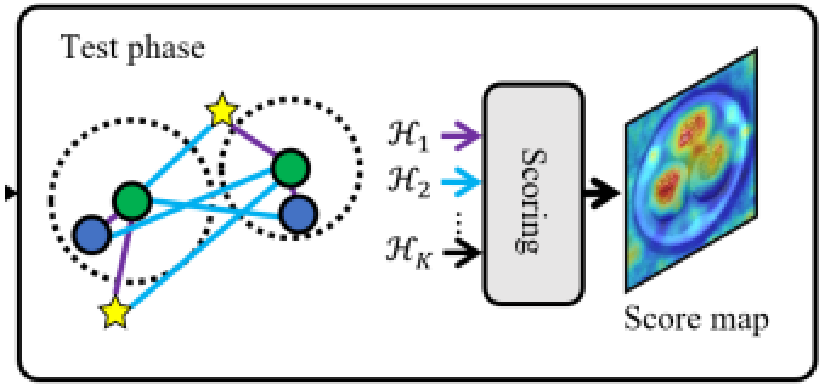
- But, it can be uncertain which memorized features will match $\phi(\mathbf{p}_t)$
  - (underestimates the nomality)
- Novel scoring function that considers the certainty of $\phi(\mathbf{p}_t)$
- $$\mathcal{A}_t = \frac{e^{-\mathcal{S}_t}}{\sum_{k=1}^{K} e^{-\mathcal{D}(\phi(\mathbf{p}_t), \mathbf{c}_t^k)}} \cdot \mathcal{S}_t$$
- Softmin to measure how close the nearest c is compared to the other c



DEEP NOID

Scoring Function

- Heatmaps are generated from naïve anomaly scores
- k-th heatmap $\mathcal{H}^k$ = $\{\mathcal{D}(\phi(\mathbf{p}_t), \mathbf{c}_t^k) | 1 \leq t \leq T\}$ is generated and rearranged so that heatmap has spatial information
- Then certainty score is calculated at all pixel locations to obtain the final output of CFA
- For the same resolution as the input samples, A is properly interpolated, and the Gaussian smoothing of $\sigma = 4$ is applied as post-processing



DEEP NOID

Qualitative Results

- Red : anomaly score
- Dotted circle : the area of abnormal features
- Triangle : memorized features

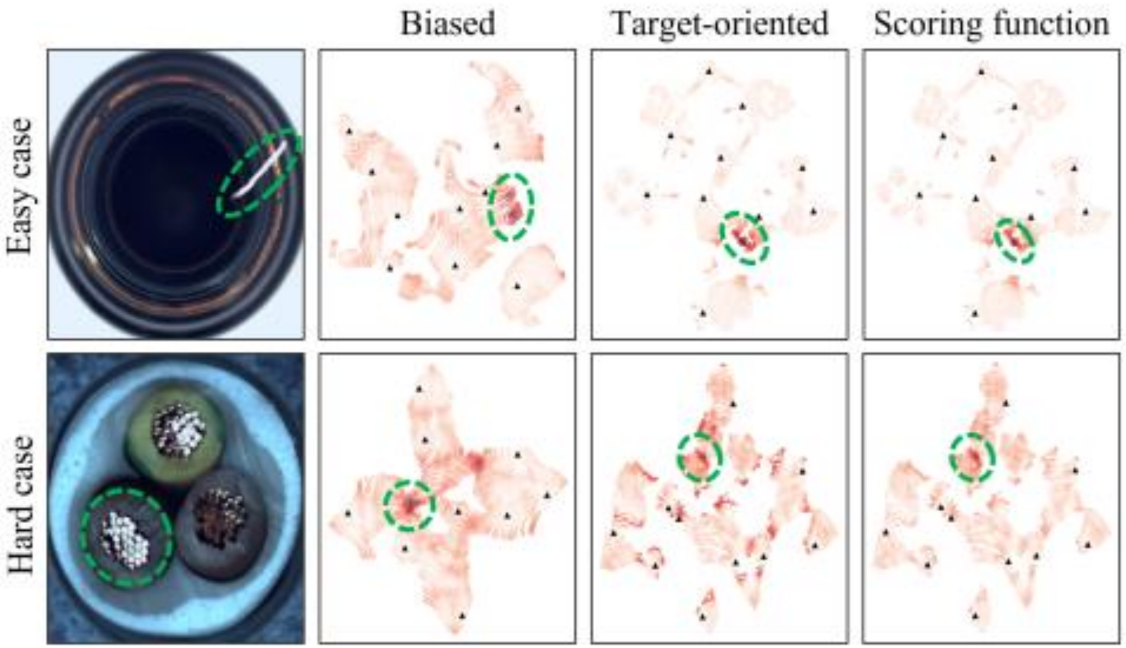


Figure 3. Visualization of anomaly score of each patch feature.

Experiments

- All the experiments were performed on MVTec AD benchmark
- Evaluation metric -〉 AUROC (I-AUROC for detection, P-AUROC for localization), AUPRO (PRO)
- All CNNs were pretrained with ImageNet

Set up

- Extract feature maps corresponding to {C2, C3, C4}
- Patch descripter-〉 1x1 CoordConv, initialized to He's initializer, optimized with AdamW, amsgrad
- Batch size 4
- r and α were set to 1e-5, 1e-1

Results

- CFA++ : ensembles the results when using cropped images and using only resized samples
- CFA++ shows 0.4% better I-AUROC than PatchCore

Table 2. Image/Pixel-level AUROC (%) of anomaly localization methods on MVTec AD dataset.

Model		SPADE	Patch SVDD	PaDiM	CutPaste	CFLOW	PatchCore	CFA	CFA++
I-AUROC	Textures	96.6	94.5	95.3	98.4	98.7	99.0	99.6	**99.8**
	Objects	96.0	90.8	95.3	94.1	98.0	99.1	99.2	**99.4**
	All	96.2	92.1	95.3	95.5	98.3	99.1	99.3	**99.5**
P-AUROC	Textures	92.9	93.7	95.3	96.9	**98.5**	97.5	97.2	97.5
	Objects	97.6	96.7	95.3	97.8	98.7	98.3	98.6	**98.9**
	All	96.0	95.7	97.5	97.5	**98.6**	98.2	98.2	98.5

Results

- The proposed method guarantees good performance in both localization and detection
- Also, achieves outstanding performance despite using a memory bank with a smaller spatial complexity compared to SPADE, PaDiM, PatchCore

Table 3. Image-level AUROC (%) and Pixel-level AUPRO (%) of anomaly localization methods on RD-MVTec AD dataset.

Model		Textures	Objects	All
VAE (ResNet18)	I-AUROC	54.7	65.8	62.1
	P-AUPRO	23.1	30.2	27.8
CFA++ (ResNet18)	I-AUROC	**98.6**	**95.5**	**96.5**
	P-AUPRO	**81.1**	**82.2**	**81.8**
SPADE (WRN50-2)	I-AUROC	84.6	88.2	87.2
	P-AUPRO	75.6	65.8	69.0
PaDiM (WRN50-2)	I-AUROC	92.4	92.1	92.1
	P-AUPRO	77.8	70.8	73.1
CFA++ (WRN50-2)	I-AUROC	**99.7**	**98.3**	**98.7**
	P-AUPRO	**82.2**	**83.7**	**83.2**

Results

Table 4. Performance comparison of image-level AUROC (%) on each class of MVTec AD dataset. Red, **blue**, and **bold** stand for the first, second, and third places

Class	SPADE	Patch SVDD	PaDiM	CutPaste	CFLOW	PatchCore	CFA	CFA++
Bottle	-	98.6	-	98.2	**100**	**100.0**	**100.0**	**100.0**
Cable	-	90.3	-	81.2	97.6	**99.5**	**99.8**	**99.8**
Capsule	-	76.7	-	**98.2**	97.7	**98.1**	97.3	**99.2**
Carpet	-	92.9	-	93.9	**98.7**	**98.7**	97.3	**99.5**
Grid	-	94.6	-	**100.0**	**99.6**	98.2	99.2	**99.9**
Hazelnut	-	92.0	-	98.3	**100.0**	**100.0**	**100.0**	**100.0**
Leather	-	90.9	-	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Metal nut	-	94.0	-	99.9	99.3	**100.0**	**100.0**	**100.0**
Pill	-	86.1	-	94.9	**96.8**	96.6	**97.9**	**97.9**
Screw	-	81.3	-	88.7	91.9	**98.1**	**97.3**	**97.3**
Tile	-	97.8	-	94.6	**99.9**	98.7	**99.4**	**100.0**
Toothbrush	-	**100.0**	-	99.4	99.7	**100.0**	**100.0**	**100.0**
Transistor	-	91.5	-	96.1	95.2	**100.0**	**100.0**	**100.0**
Wood	-	96.5	-	99.1	99.1	**99.2**	**99.7**	**99.7**
Zipper	-	97.9	-	**99.9**	98.5	99.4	**99.6**	**99.6**
Average	96.2	92.1	95.3	95.5	98.3	**99.1**	**99.3**	**99.5**

Ablation Study

- Normality was underestimated due to biased features -> low performance
- Normal features are more densely clustered -> better performance
- Clustered more discriminatively -> best performance

- ResNet18 uses relatively small feature dimensions which may increase ambiguity, so Lrep effectively solved the problem

Table 5. Image/Pixel-level AUROC (%) of the proposed method according to $\mathcal{L}_{att}$ and $\mathcal{L}_{rep}$ on MVTec AD dataset.

Backbone	$\mathcal{L}_{att}$	$\mathcal{L}_{rep}$	I-AUROC	P-AUROC
ResNet18			83.7	92.4
	✓		97.8	97.8
	✓	✓	**98.9**	**98.1**
WRN50-2			85.9	94.0
	✓		99.1	98.3
	✓	✓	**99.5**	**98.5**

Ablation Study

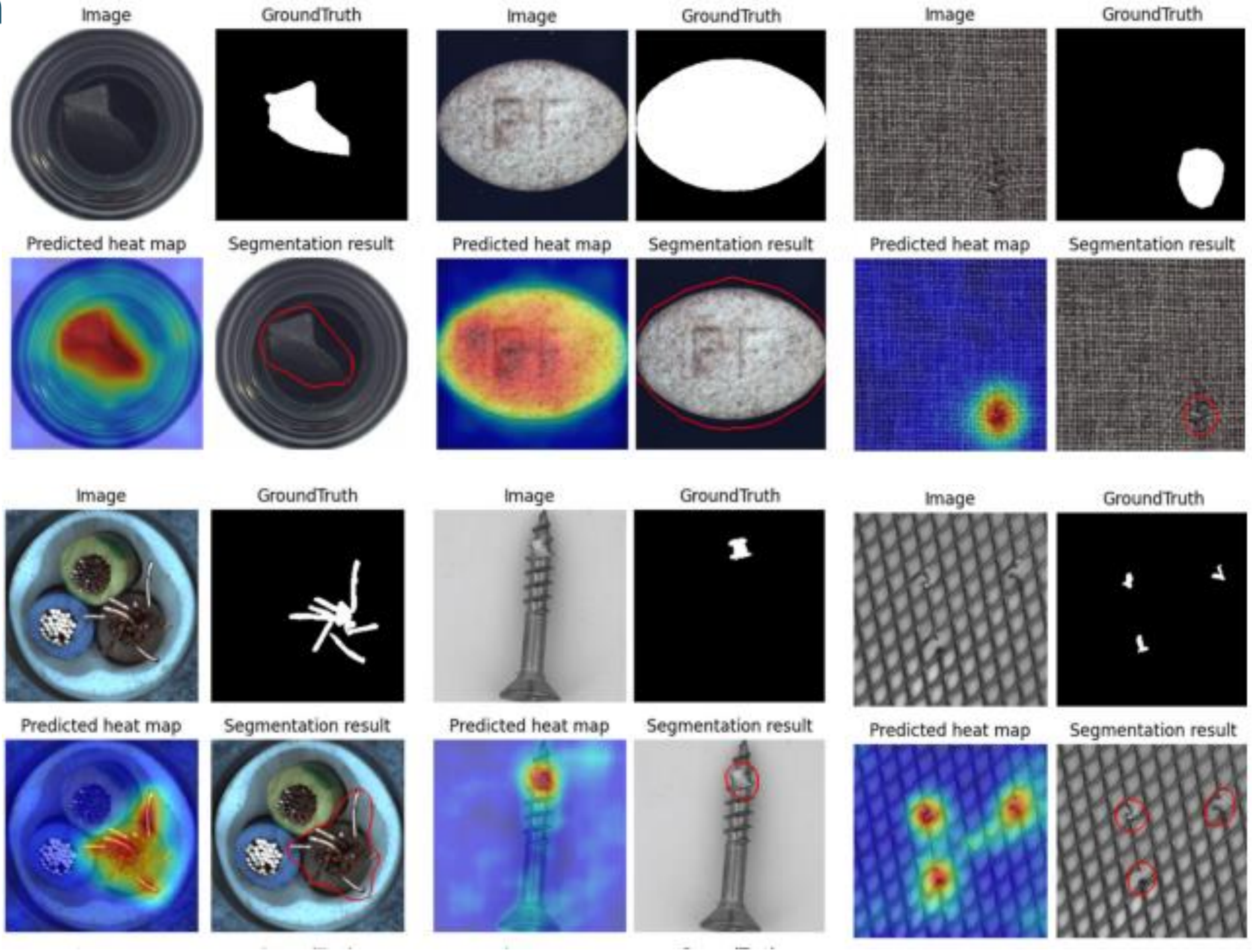
- Feature reduction ratio γ_d
- Patch reduction ratio γ_c
- CFA was compressed as $\frac{\gamma_d \gamma_c}{|\mathcal{X}|}$

Table 6. Pixel-level AUROC (%) of the proposed method with additional memory bank compression on MVTec AD dataset.

Backbone	γ_d	γ_c	P-AUROC	Throughput
WRN50-2	1	1	**98.45**	48
	1/2	1/2	98.44	93 (1.9x)
	1/4	1/4	98.44	132 (2.8x)
	1/8	1/8	98.36	**172** (3.6x)

DEEP NOID

Visualization



Conclusion

- Solve bias problem with CFA (to obtain target-oriented features)
- Lightweight memory bank

Q & A

감사합니다