WEAKLY SUPERVISED DEFECT LOCALIZATION WITH RESIDUAL FEATURES SUPPLEMENTARY MATERIALS

1. DETAILS OF EXPERIMENTAL SETUP

1.1. Dataset

We evaluated the proposed method on the Road Damage Dataset 2020 (RDD2020) [1], the RDD2022 [2] and the Visual Anomaly (VisA) dataset [3].

RDD2020. RDD2020 is a large-scale dataset with 26,620 images of roads in India, Japan, and the Czech Republic, captured using smartphones mounted on vehicles. The images are horizontal and show four types of road damage: longitudinal cracks (D00), transverse cracks (D10), alligator cracks (D20), and potholes (D40). Each damage type is annotated with bounding boxes. Since the test set annotations for RDD2020 were not released owing to their use in the Crowdsensing-based Road Damage Detection Challenge (CRDDC) 2020 [4], we split the original training set into 70% for training and 30% for testing. Table 1 details the training and test sets used in our experiments.

RDD2022. RDD2022 extends RDD2020 by adding new data from Norway, the United States, and China. The images are both top-down and horizontal views, showing the same four types of road damage as RDD2020. Similar to RDD2020, we split the original training set into 70% for training and 30% for testing, because the test set annotations were not released. Table 2 shows the number of images with each label in the training and test sets.

VisA. The VisA dataset is designed for anomaly detection and segmentation tasks and used as an image anomaly detection benchmark. The dataset is divided into 12 subsets, each corresponding to different objects. We used the 2-class high-shot setup on VisA, which is provided in [3]. In the 2-class high-shot setup, for each object, 60% and 40% of normal and anomalous images are assigned to training and test sets, respectively.

1.2. Implementation Details

1.2.1. Proposed method

We introduce residual feature learning (RFL) and the reference image selection (RIS) module to the ResNet50-feature pyramid network (FPN) [5]. In order to produce score maps, we convert P3, P4, and P5 feature maps obtained from the FPN by convolution operations with a 3×3 kernel, 1×1 kernel, and sigmoid function. We up-sample the score maps to the same input size by bilinear interpolation, because the height and width of the P3 feature map are 1/8 of the input image size. We use a ResNet50 [6] model pretrained on ImageNet [7], which is publicly available. The input image size is also the same, resized to a fixed size of 512×512.

For training the proposed method in RDD2020/2022, we train for 10,000 iterations with a batch size of 96. We use stochastic gradient descent [8] for the parameter optimization. Our hyperparameter settings are as follows: initial learning rate: 0.002 for the backbone and 0.02 for the FPN; weight decay: 0.0005; momentum: 0.9. The cosine annealing strategy [9] is adopted to adjust the learning rate. We use the warm-up technique [10], with a warm-up step of 100 and warm-up multiplier of 0.1. We also use data augmentation techniques including HSV color transformation, image scaling, flipping, rotation, and shifting. Data augmentation is implemented by using the Albumentations library [11]. We use the PyTorch framework [12] for all our experiments. For training the proposed method in VisA, we train for 1,000 iterations and use the Adam optimizer with learning rate 0.00005. Other settings are the same as for RDD2020/2022.

We used two NVIDIA RTX A6000 GPUs to train our model. It takes about 7 h to train our model on RDD2020. The average inference time per image was 59 ms. This duration for our method encompasses the time needed for feature extraction and similarity calculation for the selection of reference images.

1.2.2. Baseline Methods

We compare our method with several baselines, including WeCLIP [13], SeCo [14], and multiple instance learning (MIL) [15]. Below are the details of these baselines:

WeCLIP[13]: WeCLIP is weakly supervised semantic segmentation (WSSS) method based on the frozen CLIP backbone. WeCLIP has a frozen CLIP CAM refinement module and improves the quality of pseudo labels. We use their official implementation and modify the prompts for the defect detection and localization task. In the original WeCLIP, the prompt input to the CLIP text encoder is based on CLIP-ES[16]. Specifically, the prompt `'a clean origami {}.'` is selected. In RDD2020/2022, specifically, we select `'a clean origami {}.'` for the

Table 1. Details of the RDD2020 dataset split into training and test sets. The table shows the number of images with each label.

	Num of images	D00	D10	D20	D40	Normal
Training	14,728	3,236	1,817	4,583	2,128	6,240
Test	6,313	1,434	783	2,018	946	2,606

Table 2. Details of the RDD2022 dataset split into training and test sets. The table shows the number of images with each label for the training and test sets.

	Num of images	D00	D10	D20	D40	Normal
Training	26,866	9,453	5,410	5,913	2,582	10,235
Test	11,519	4,095	2,299	2,499	1,092	4,383

background category. Here, {} corresponds to 'ground', 'land', 'grass', 'tree', 'building', 'wall', 'sky', 'lake', 'water', 'river', 'sea', 'railway', 'railroad', 'keyboard', 'helmet', 'cloud', 'house', 'mountain', 'ocean', 'road', 'rock', 'street', 'valley', 'bridge', 'sign'. Additionally, we select '{} on road.' for the foreground category. Here, {} represents the names of each damage category (longitudinal cracks, transverse cracks, alligator cracks, and potholes). In the VisA experiments, we use the prompts from WinCLIP[17]. For the foreground category, we select the following prompts: 'damaged {}.', 'broken {}.', '{} with flaw.','{} with defect.','{} with damage.' For the background category, we select the following prompts: '{}.', 'flawless {}.', 'perfect {}.', 'unblemished {}.', '{} without flaw.', '{} without defect.', '{} without damage.' Here, {} represents each object category. The remaining settings follow the training setup used in the PASCAL VOC 2012 experiment as described in the original paper [13] for RDD2020/2022 and VisA experiments. We utilize the frozen CLIP backbone with the ViT-16-base architecture. The batch size is set to 4, and the maximum number of iterations is set to 30,000. We employ the AdamW optimizer [18] with a learning rate of $2e^{-3}$ and a weight decay of $1e^{-3}$. During inference, we use multi-scale testing with scales {0.75, 1.0}. DenseCRF [19] is applied as a post-processing method to refine the predictions.

SeCo[14]: SeCo is one of the SOTA methods in WSSS that tackles the co-occurrence problem by designing image decomposition and contrastive representation. We use the official implementation of SeCo. The training setup follows the PASCAL VOC 2012 experiment settings as described in the original paper [14] for RDD2020/2022 and VisA experiments. Encoders in the dual-teacher single-student framework use ViT-B as the backbone, initialized with pre-trained weights on ImageNet. For inference, we also follow the PASCAL VOC 2012 experiment setup, using multi-scale testing with scales {1.0, 1.25, 1.5} and DenseCRF [19] to refine the predictions.

MIL[15]: We implemented MIL by removing the RFL and RIS modules from our proposed method. Similar to our proposed method, MIL is based on ResNet50-FPN and generates score maps using the P3, P4, and P5 feature maps obtained from the FPN. The score maps are produced through convolution operations with a 3×3 kernel, 1×1 kernel, and sigmoid function. Global max pooling is applied to the score maps to obtain image-level scores. Binary cross-entropy loss is calculated from the obtained image-level scores and image-level labels. The training and inference settings are the same as those of the proposed method.

2. DETAILS OF EVALUATION METRICS

In this section, we detail the evaluation metrics used in our RDD2020/2022 experiments. We use mean average precision (mAP) with bounding boxes as a localization metric, as RDD2020/2022 are annotated with bounding boxes. Below, we explain how to define precision and recall using bounding boxes and predicted score maps. Our method and the baseline methods output score maps $\boldsymbol{S}$, which have elements of score at each position. However, the ground truth label for the damage is given as a bounding box in RDD2020/RDD2022. To quantitatively evaluate localization for damage regions, we assign a detection score to the bounding box. In this study, the maximum score within the correct bounding box is set to the detection score for each damage, and the maximum score outside the correct bounding box is calculated as the detection score for false detection evaluation. We thus calculated the average precision (AP) using these scores.

3. FULL QUANTITATIVE RESULTS

This section provides a detailed quantitative evaluation, expanding on the average metrics presented in the main paper due to space constraints. Table 3 compares the proposed method and baseline methods in terms of I-AUROC, P-AUPR, and PRO for all objects in the VisA dataset. Table 4 and Table 5 show the I-AUROC and AP for each class in the RDD2020 and RDD2022 datasets, respectively.

VisA. As shown in Table 3, the proposed method demonstrates competitive performance with baseline methods in terms of I-AUROC. For the localization evaluation metric, the proposed method outperforms baseline methods in P-AUPR for 5 out of 12 objects. Furthermore, in terms of PRO, the proposed method achieves the highest performance for 6 out of 12 objects.

RDD2020. As shown in Table 4, the proposed method achieved the highest AP across all classes. Additionally, the proposed method demonstrated the highest or second-highest performance in terms of I-AUROC for all classes.

RDD2022. As shown in Table 5, the proposed method outperforms baseline methods in terms of both AP and I-AUROC for all classes.

4. THE EFFECT OF THE NUMBER OF REFERENCE IMAGES

The proposed method requires the selection of K_{ref} reference images to obtain residual features. We investigate the effect of the number of reference images, K_{ref}. Table 6 shows the performance comparison with different K_{ref} values on the RDD2022 and VisA datasets. With $K_{ref} = 3$, the proposed method achieves the highest I-AUROC of 94.5 on the RDD2022 dataset. The mAP on RDD2022 is 66.1, which represents the second-highest performance. For the VisA dataset, the proposed method attains the highest I-AUROC of 99.1 and the highest P-AUPR of 21.5. The PRO is 53.0, which is the second-highest performance. Based on these results, we chose $K_{ref} = 3$ for our main experiment.

Table 3. Comparison with baseline methods on I-AUROC, P-AUPR, and PRO metrics for different objects in the VisA dataset. Bold numbers represent the highest values for each metric.

Metric	I-AUROC				P-AUPR				PRO			
Method	WeCLIP	SeCo	MIL	Ours	WeCLIP	SeCo	MIL	Ours	WeCLIP	SeCo	MIL	Ours
Object												
candle	96.2	97.7	97.7	**99.4**	5.9	2.1	7.9	**11.7**	69.0	53.1	64.7	**69.3**
capsules	93.7	97.3	**99.2**	98.1	28.4	12.5	**51.0**	50.1	50.7	55.1	**66.9**	58.3
cashew	98.3	**99.9**	99.6	99.8	6.2	5.9	1.9	**8.2**	**59.1**	46.4	13.2	46.0
chewinggum	99.3	**99.9**	**99.9**	99.8	**56.2**	12.1	36.9	52.5	42.3	38.2	**56.0**	53.7
fryum	97.8	98.8	**100.0**	99.8	**19.3**	19.2	7.8	12.4	43.0	57.2	54.8	**62.5**
macaroni1	91.3	99.1	**99.8**	**99.8**	**7.3**	2.7	3.4	6.9	**68.2**	29.0	66.2	65.4
macaroni2	78.1	**98.8**	95.1	95.3	6.2	**37.4**	4.0	6.4	33.8	63.5	59.5	**65.5**
pcb1	98.5	97.3	99.7	**99.8**	**15.9**	14.3	2.9	9.2	**61.8**	38.3	19.4	17.8
pcb2	84.3	95.3	**99.7**	99.3	7.1	8.2	1.3	**27.7**	16.9	37.1	13.1	**39.9**
pcb3	74.5	97.1	**99.7**	98.7	8.8	2.9	0.0	**21.6**	16.1	24.6	0.0	**45.0**
pcb4	98.8	**100.0**	99.6	**100.0**	22.5	17.4	13.9	**35.2**	52.5	**53.6**	40.0	50.2
pipe_fryum	99.9	99.7	99.9	**100.0**	15.7	**32.2**	15.3	16.2	54.7	52.1	57.1	**61.9**
Mean	92.6	98.4	**99.2**	99.1	16.6	13.9	12.2	**21.5**	47.3	45.7	42.6	**53.0**

Table 4. Comparison of methods on different classes on RDD2020. Bold numbers represent the highest values for each metric.

Class	D00		D10		D20		D40		Mean	
Metric	I-AUROC	AP	I-AUROC	AP	I-AUROC	AP	I-AUROC	AP	I-AUROC	AP
Method										
WeCLIP	75.6	26.1	88.2	39.6	91.6	45.5	90.5	48.4	86.5	39.9
SeCo	83.4	40.0	90.6	40.5	93.8	52.4	91.2	54.1	89.7	46.8
MIL	**85.3**	48.3	90.2	40.9	**95.4**	77.6	93.9	57.5	91.2	56.1
Ours	85.1	**51.5**	**91.4**	**44.4**	95.2	**80.5**	**94.4**	**61.6**	**91.5**	**59.5**

Table 5. Comparison of methods on different classes on RDD2022. Bold numbers represent the highest values for each metric.

Class	D00		D10		D20		D40		Mean	
Metric	I-AUROC	AP	I-AUROC	AP	I-AUROC	AP	I-AUROC	AP	I-AUROC	AP
Method										
WeCLIP	76.3	44.9	89.0	42.3	90.2	46.9	89.5	39.7	86.3	43.4
SeCo	88.9	54.3	89.2	34.2	94.7	50.7	91.1	48.2	91.0	46.9
MIL	91.7	66.0	94.2	57.2	96.1	74.7	92.8	53.8	93.7	62.9
Ours	**92.2**	**67.3**	**94.7**	**60.9**	**96.2**	**78.4**	**94.8**	**57.9**	**94.5**	**66.1**

Table 6. Performance comparison with different K_{ref} values on RDD2022 and VisA datasets. The best values are highlighted in bold, and the second-best values are underlined.

Dataset	RDD2022		VisA		
K_{ref}	I-AUROC	mAP	I-AUROC	P-AUPR	PRO
1	**94.5**	65.3	97.9	19.1	49.9
3	**94.5**	66.1	**99.1**	**21.5**	53.0
5	94.2	65.1	98.8	21.1	**53.2**
8	94.0	**66.6**	98.7	18.1	47.0

5. REFERENCES

[1] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, and Yoshihide Sekimoto, “RDD2020: an annotated image dataset for automatic road damage detection using deep learning,” *Data in Brief*, vol. 36, pp. 107133, 2021.

[2] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, and Yoshihide Sekimoto, “RDD2022: A multi-national image dataset for automatic road damage detection,” *CoRR*, vol. abs/2209.08538, 2022.

[3] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer, “Spot-the-difference self-supervised pre-training for anomaly detection and segmentation,” in *European Conference on Computer Vision*. Springer, 2022, pp. 392–408.

[4] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Hiroshi Omata, Takehiro Kashiyama, and Yoshihide Sekimoto, “Global road damage detection: State-of-the-art solutions,” in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 5533–5539.

[5] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie, “Feature pyramid networks for object detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 936–944, IEEE Computer Society.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 770–778, IEEE Computer Society.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. 2009, pp. 248–255, IEEE Computer Society.

[8] Sebastian Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016.

[9] Ilya Loshchilov and Frank Hutter, “SGDR: stochastic gradient descent with warm restarts,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017, OpenReview.net.

[10] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He, “Accurate, large minibatch SGD: training imagenet in 1 hour,” *CoRR*, vol. abs/1706.02677, 2017.

[11] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin, “Albumentations: Fast and flexible image augmentations,” *Inf.*, vol. 11, no. 2, pp. 125, 2020.

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, Eds., 2019, pp. 8024–8035.

[13] Bingfeng Zhang, Siyue Yu, Yunchao Wei, Yao Zhao, and Jimin Xiao, “Frozen clip: A strong backbone for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3796–3806.

[14] Zhiwei Yang, Kexue Fu, Minghong Duan, Linhao Qu, Shuo Wang, and Zhijian Song, “Separate and conquer: Decoupling co-occurrence via decomposition and representation for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3606–3615.

[15] Yuta Shirakawa, Satoshi Ito, Reiko Noda, Naoto Yoshitani, Masahide Wake, and Honoka Takano, “Automating daily inspection for expressways using anomaly detection model,” in *PHM Society Asia-Pacific Conference*, 2023, vol. 4.

[16] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He, “Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15305–15314.

[17] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer,

“Winclip: Zero-/few-shot anomaly classification and segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19606–19616.

[18] Ilya Loshchilov, Frank Hutter, et al., “Fixing weight decay regularization in adam,” *arXiv preprint arXiv:1711.05101*, vol. 5, 2017.

[19] Philipp Krähenbühl and Vladlen Koltun, “Parameter learning and convergent inference for dense random fields,” in *International conference on machine learning*. PMLR, 2013, pp. 513–521.