Dual-Encoder ViT–SAM Heterogeneous Feature Fusion for Surface Defect Detection of Mechanical Parts

Jianqing Xin[1,*], Yuhao Kong[2], Minwen Guo[3]

[1] Harbin Institute of Technology, Weihai 264200, China

[2] Northwest A&F University, Xianyang 712100, China

[3] Chengdu College of University of Electronic Science and Technology of China, Chengdu 611731, China

Corresponding Author: Jianqing Xin (1309515895@qq.com)

Abstract: Surface defect detection of mechanical parts is challenging because defects are often small, thin, low contrast, and visually similar to manufacturing textures or reflection artifacts [1–4]. This paper presents a dual-encoder framework that combines a lightweight ResNet-based local branch with the Vision Transformer image encoder of the Segment Anything Model (SAM) [10], [15]. A heterogeneous feature fusion module (HFM) aligns local texture features and global semantic features through cross-attention, channel recalibration, and residual preservation. Confidence-based sample filtering, uncertainty-aware multi-task weighting, and staged fine-tuning are used to reduce the effects of label noise and target-domain mismatch [22], [24], [25]. On a production-line dataset containing seven defect categories, the proposed model achieves 91.1% mAP, 88.3% mIoU, and 91.6% recall. It also reaches 58 FPS in a separate NVIDIA GTX 1070 throughput benchmark. These results indicate that the proposed fusion strategy improves detection and segmentation while retaining practical inference efficiency.

Keywords: Surface defect detection; Segment Anything Model; Heterogeneous feature fusion; multi-task learning; Industrial vision

1. Introduction

Industrial manufacturing increasingly requires high-precision inspection under tight production tolerances. Surface defects directly affect assembly reliability, functional consistency, and service life. In practice, scratches, burrs, dents, and indentations may occupy only a small image region and exhibit weak contrast against machining marks or metallic textures. Illumination changes, specular reflection, background clutter, and domain shifts between production stations further complicate detection and segmentation [1], [2], [4].

Earlier inspection systems relied on manually designed edges, texture descriptors, and shallow classifiers. Such pipelines can perform well under controlled imaging conditions but adapt poorly to complex backgrounds and variable defect morphology [1], [2]. Convolutional neural networks (CNNs) learn hierarchical texture and shape representations and remain effective local feature extractors [10]. Nevertheless, finite receptive fields and repeated downsampling can weaken long-range context modeling and boundary recovery for micro-cracks and thin scratches.

Transformers model global dependencies through self-attention [13], while Vision Transformers (ViTs) extend this mechanism to image representation learning [14]. SAM further provides a strong pre-trained segmentation prior through a ViT image encoder [15]. However, large pre-trained models are not uniformly reliable across real applications [17]. Studies of SAM for industrial defects also report sensitivity to weak boundaries, fine structures, and domain-specific textures [18], [19]. Global semantic representations therefore benefit from a complementary local branch rather than being used alone.

Industrial datasets introduce additional training risks. Pixel-level annotation is expensive, ambiguous regions can produce label noise, and station-to-station domain shift limits direct transfer of pre-trained weights [3], [4]. Robust learning under noisy labels and careful adaptation are consequently important [24], [25]. A deployable system must balance accuracy, robustness, and throughput rather than optimize a single metric.

Motivated by these observations, we combine a lightweight ResNet-style CNN branch with the ViT image encoder of SAM. The CNN branch preserves local textures and boundaries, whereas the SAM branch provides long-range semantic context. HFM performs cross-attention followed by Squeeze-and-Excitation channel recalibration [7–9], [15], [20]. A multi-task decoder jointly predicts defect classes, bounding boxes, masks, and boundaries using ideas related to U-Net, anchor-free detection, Dice optimization, and uncertainty-weighted learning [11], [21]–[23].

The main contributions are as follows:

- A coherent dual-encoder architecture that combines SAM global semantics with lightweight CNN texture features for mechanical surface inspection.
- An HFM that explicitly aligns heterogeneous features through cross-attention, residual preservation, and channel recalibration.
- A robust multi-task training strategy with confidence-based filtering and staged fine-tuning for noisy, domain-shifted industrial data.
- A systematic evaluation covering detection, segmentation, convergence, per-class discrimination, throughput, and Grad-CAM interpretation.

2. Proposed Method

The proposed network contains a lightweight CNN encoder-decoder, a pre-trained SAM image encoder, HFM, and a multi-task prediction head. These components build on residual learning, encoder-decoder segmentation, SAM, channel attention, anchor-free detection, Dice optimization, and multi-task weighting [10, 11, 15], [20–23]. Figure 1 summarizes the architecture.

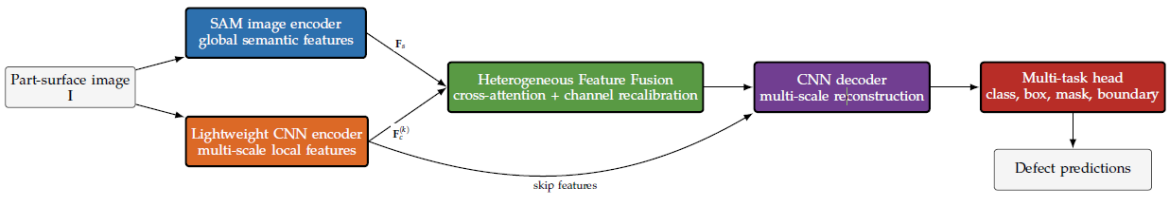

Figure 1: Overall architecture of the proposed dual-encoder surface defect detection network. The SAM branch supplies global semantics, while the CNN branch preserves local textures and boundaries.

For an input image:

$$\mathbf{I} \in \mathbb{R}^{H\times W\times 3} \tag{1}$$

The two encoders produce:

$$\{\boldsymbol{F}_c^{(l)}\}_{l=1}^{L} = E_c(\mathbf{I}), F_s = E_s(\mathbf{I}) \tag{2}$$

At fusion level k, HFM and the decoder compute:

$$\tilde{\boldsymbol{F}}_c^{(k)} = HFM\left(F_c^{(k)}, F_s\right) \tag{3}$$

$$\boldsymbol{F}_{dec} = D\left(\tilde{\boldsymbol{F}}_c^{(k)}, \{\boldsymbol{F}_c^{(l)}\}_{l\neq k}\right) \tag{4}$$

$$(\hat{y}_{det}, \hat{y}_{seg}, \hat{y}_{bnd}) = H(F_{dec}) \tag{5}$$

2.1 CNN Encoder-Decoder Branch

The CNN branch uses residual blocks and a feature hierarchy related to ResNet and FPN [10], [12]. Skip connections follow the U-shaped reconstruction principle [11]. Table 1 gives the feature dimensions.

Table 1: CNN Encoder Feature Configuration.

Stage	Spatial resolution	Channels
0	H × W	3
1	H/4 × W/4	64
2	H/8 × W/8	128
3	H/16 × W/16	256
4	H/32 × W/32	512

The l_{th} encoder stage is represented as:

$$\boldsymbol{F}_c^{(l)} = B\left(\boldsymbol{C}_{3\times 3}^{(l)}\left(\boldsymbol{F}_c^{(l-1)}\right)\right) \tag{6}$$

where $\boldsymbol{C}_{3\times 3}^{(l)}$ is a strided convolution or residual stage and B denotes normalization and nonlinearity. Starting from $G^{(L)} = \tilde{\boldsymbol{F}}_c^{(k)}$, the decoder performs:

$$G^{(l)} = B\left(C_{3\times 3}\left(\left[Up\left(G^{l+1}\right), \boldsymbol{F}_c^{(l)}\right]\right)\right) \tag{7}$$

and produces:

$$F_{dec} = C_{3\times 3}(G^{(0)}) \tag{8}$$

2.2 SAM Image Encoder Branch

SAM contains an image encoder, prompt encoder, and mask decoder [15]. We reuse only its ViT image encoder, whose patch representation follows standard ViT processing [14]. The initial tokens are:

$$X_0 = PatchEmbed(I) + E_{pos} \tag{9}$$

For Transformer block r,

$$X_r' = X_{r-1} + MSA\left(LN(X_{r-1})\right) \tag{10}$$

$$X_r = X_r' + MLP\left(LN(X_r')\right) \tag{11}$$

The final token sequence is reshaped into a two-dimensional map:

$$F_s = Up\left(Reshape\left(X_{L_s}\right)\right) \tag{12}$$

Before fusion, spatial and channel dimensions are aligned as:

$$\overline{F_s} = C_{1\times1}\big(Rescale(F_s, H_k, W_k)\big) \tag{13}$$

The SAM encoder is frozen during the first training stage. It is then partially unfrozen and optimized with a learning rate one order of magnitude lower than the CNN branch. This adaptation reduces catastrophic changes to pre-trained features while addressing industrial domain shift [3], [4], [17].

2.3 Heterogeneous Feature Fusion Module

HFM combines Transformer-style attention [13] with channel recalibration [20]. Its data flow is shown in Figure 2.

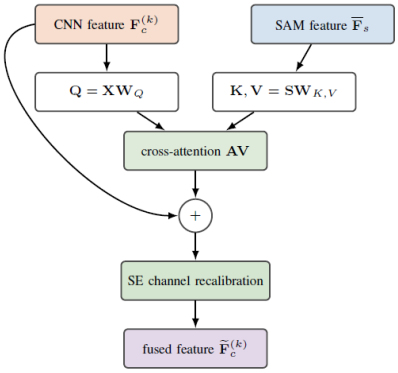

Figure 2: Structure of HFM. Cross-attention aligns the two branches, a residual path preserves local information, and channel recalibration suppresses irrelevant responses.

Flattening the aligned feature maps gives:

$$X = Flatten\big(F_c^{(K)}\big), S = Flatten(\overline{F_s}), X, S \in \mathbb{R}^{N\times C_k} \tag{14}$$

Where $N = H_k W_k$ Projections into a common d-dimensional space are:

$$Q = XW_Q, K = SW_K, V = SW_V \tag{15}$$

The attention and aggregation are:

$$A = softmax\left(\frac{QK^T}{\sqrt{d}}\right) \tag{16}$$

$$Z = AV \tag{17}$$

A residual projection preserves the CNN representation:

$$\widetilde{X} = X + ZW_o, \tilde{F} = Reshape\big(\widetilde{X}\big) \tag{18}$$

Channel recalibration then computes:

$$z = GAP\big(\tilde{F}\big) \tag{19}$$

$$s = \sigma\big(W_2\delta(W_1 z)\big), \tilde{F}_c^{(k)} = s\odot\tilde{F} \tag{20}$$

2.4 Multi-Task Prediction Head and Robust Objective

The detector follows an anchor-free formulation related to FCOS [21]. At location (i, j),

$$p_{ij} = f_{cls}(F_{dec})_{ij}, b_{ij} = f_{reg}(F_{dec})_{ij} \tag{21}$$

The segmentation mask is:

$$\widehat{M} = \sigma\left(C_{1\times1}\big(Up^{(t)}(F_{dec})\big)\right) \tag{22}$$

A parallel single-channel branch predicts the boundary probability map $\hat{B}$.

The total loss combines classification, segmentation, and boundary terms:

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{seg} + \beta\mathcal{L}_{bnd} \quad (23)$$

where $\mathcal{L}_{cls}$ combines binary cross-entropy and Dice loss [23]. We use α = 1.0 and β = 0.5. Uncertainty-aware weighting motivates balancing tasks with different scales [22].

For confidence-based noise filtering, the maximum predicted class probability $q_i = max_c\, p_i, c$ is computed for each sample

In each mini-batch, the lowest-confidence fraction k is excluded from the supervised classification update:

$$S = \{i: q_i \geq Q_k(\{q_j\})\}, \mathcal{L}_{cls} = \frac{1}{|S|}\sum_{i\epsilon s} \ell CE(i) \quad (24)$$

Where Q_k is the kth confidence quantile and k = 10%. This design follows the broader principle of reducing the influence of unreliable labels [24], [25]; it does not assume that every low-confidence sample is mislabeled.

3. Experimental Setup and Results

3.1 Dataset and Preprocessing

The dataset was acquired with CCD cameras on a production line and contains uneven illumination, cluttered backgrounds, specular reflections, and subtle texture differences. These conditions resemble the real-world variability emphasized in industrial anomaly detection surveys and datasets [1–6]. The data include 4,900 training images and 420 test images across seven categories, as shown in Table 2.

Table 2: Dataset Distribution.

Defect type	Training	Test
Scratch	700	60
Dent	700	60
Oil drop	700	60
Indentation	700	60
Rust spot	700	60
Abrasion	700	60
Burr	700	60
Total	4900	420

Images are resized and cropped to 224 × 224 pixels, normalized per channel, and augmented with horizontal flipping, random resized cropping, brightness and contrast variation, and low-amplitude Gaussian noise. The split is fixed across all model comparisons.

3.2 Implementation Details

Training was performed on Windows 10 with PyTorch and CUDA using an AMD Ryzen Threadripper PRO 3975WX CPU and an NVIDIA RTX A5000 GPU. Throughput was measured separately on an NVIDIA GTX 1070 with batch size one. Table 3 lists the configuration. The ResNet-18, SAM, and MobileSAM components follow their original designs [10, 15, 16].

Table 3: Training and Network Configuration.

Item	Setting
CNN backbone	ResNet-18 with lightweight decoder
Global branch	Pre-trained SAM image encoder
Input size	224 × 224
Batch size	32
Optimizer	SGD, momentum 0.9
Learning rate	0.01 (CNN); 0.001 (SAM fine-tuning)
Weight decay	1×10^{-4}
Schedule	×0.1 every 50 epochs
Epochs	100
HFM	projection 256 channels
Filtering ratio	k = 10%
Task heads	Classification, box, mask, boundary

3.3 Evaluation Metrics

Detection is evaluated with average precision and mean average precision, while segmentation uses IoU and mIoU. Precision, recall, and F1 are also reported. These metrics are widely used in detection and segmentation benchmarks [6, 21, 23].

For C classes:

$$AP_c = \int_0^1 P_c(R)dR\,, mAP = \frac{1}{c}\sum_{c=1}^{c} AP_c \tag{25}$$

$$\mathrm{IoU_c} = \frac{|\widehat{\mathrm{M}}_\mathrm{c} \cap \mathrm{M_c}|}{|\widehat{\mathrm{M}}_\mathrm{c} \cup \mathrm{M_c}|}, \mathrm{mIoU} = \frac{1}{\mathrm{C}}\sum_{\mathrm{c}=1}^{\mathrm{C}} \mathrm{IoU_c} \tag{26}$$

Precision, recall, and F1 are:

$$P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+FN}, F1 = \frac{2PR}{P+R} \tag{27}$$

Key experiments were repeated three times and their means are reported. No significance claim is made because per-run variances were not preserved in the source record.

3.4 Training Dynamics

Figure 3 compares the training trajectories of the proposed fusion model and a single-branch baseline. The fusion model converges faster and reaches a higher training plateau for both mAP and mIoU. Because these curves use training metrics, generalization claims are based on the held-out test results in Table 4.

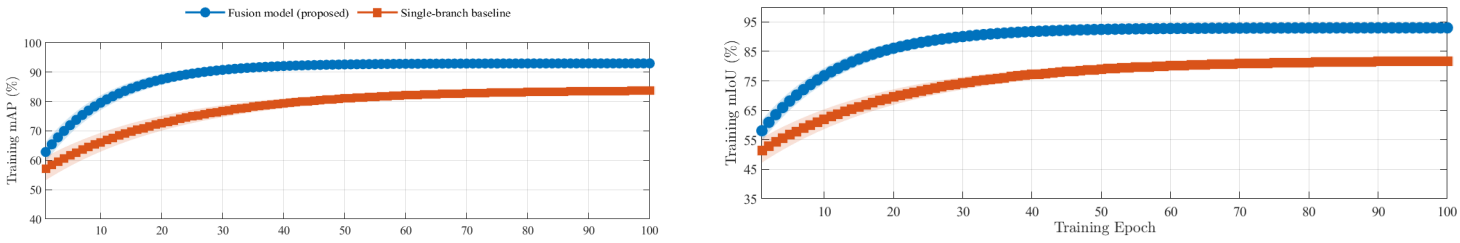

Figure 3: Training trajectories for mAP (left) and mIoU (right). Shaded bands show the variation recorded across repeated runs.

Table 4: Overall Performance Comparison.

Method	mAP (%)	mIoU (%)	Recall (%)	FPS (%)
CNN (ResNet-18)	85.0	79.2	82.3	110
SAM image encoder	90.1	84.6	88.0	65
MobileSAM	88.5	83.6	87.0	62
Proposed fusion	91.1	88.3	91.6	58

3.5 Quantitative Comparison

The proposed model is compared with a ResNet-18 CNN, a SAM image-encoder baseline, and MobileSAM. The baseline identities are grounded in the corresponding source architectures [10, 15, 16]. Table 5 shows that the proposed model achieves the highest mAP, mIoU, and recall. Relative to the strongest single-branch result, mAP improves by 1.0 percentage point and mIoU by 3.7 percentage points. The 58 FPS throughput remains compatible with near-real-time inspection, although all FPS comparisons should be interpreted under the stated GTX 1070 benchmark setting.

Table 5: Per-class Accuracy of the Proposed Model on Difficult Categories.

Model	Scratch	Abrasion	Dent	Indentation
Proposed fusion	99.2	93.7	95.1	96.3

3.6 Difficult-Class Analysis

The source manuscript listed DACL and "O2U-Net + DACL" results without a verifiable implementation or publication that matched this industrial dataset. To avoid unsupported cross-paper attribution, those rows are not retained. Table V therefore reports only the proposed model's reproduced perclass accuracy for the four difficult categories. The lowest result occurs for abrasion, which remains visually confusable with scratches under weak illumination.

3.7 Qualitative Interpretation

Although t-SNE is useful for exploratory feature projection [26], its output is sensitive to projection settings and is not used here as standalone evidence of class separability. Instead, Grad-CAM [27] is used to inspect spatial attention. Figure 4 shows that the activation maps overlap the defect regions, while the segmentation and boundary branches produce compact predictions. These examples support the intended interaction between semantic attention and boundary refinement, but they do not replace quantitative evaluation.

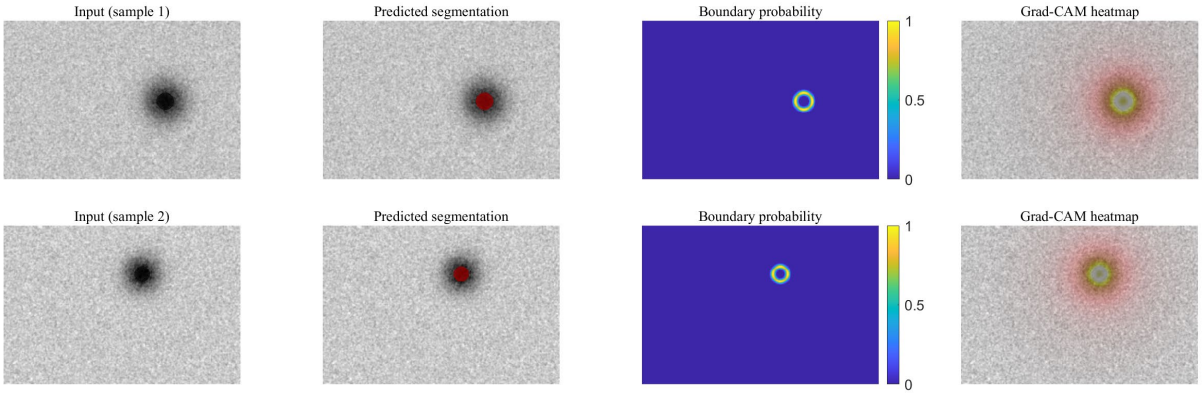

Figure 4: Qualitative results for two samples. From left to right: input image, predicted segmentation, boundary probability, and Grad-CAM heatmap.

5. Conclusion

This paper presented a dual-encoder framework for detecting and segmenting mechanical surface defects. A lightweight CNN branch preserves local texture and boundary information, while the ViT image encoder of SAM supplies global semantic context. HFM aligns the two feature spaces through cross-attention, residual preservation, and channel recalibration. Confidencebased filtering and staged fine-tuning further address label noise and domain mismatch. The model achieved 91.1% mAP, 88.3% mIoU, 91.6% recall, and 58 FPS in the stated throughput benchmark.

The results are consistent with prior findings on SAM, industrial adaptation, attention, and robust training [15, 17–20, 22, 24]. Remaining limitations include performance under severe occlusion, sensitivity to domain shifts not represented in the training set, and the absence of run-level variance records for formal significance testing. Future work should evaluate cross-line generalization and lighter adaptation strategies on additional public and industrial datasets.

References

[1] J. Liu, G. Xie, J. Wang, S. Li, C. Wang, F. Zheng, and Y. Jin, "Deep industrial image anomaly detection: A survey," Machine Intelligence Research, vol. 21, no. 1, pp. 104–135, 2024. https://doi.org/10.1007/s11633-023-1459-z

[2] Z. Li, Y. Yan, X. Wang, Y. Ge, and L. Meng, "A survey of deep learning for industrial visual anomaly detection," Artificial Intelligence Review, vol. 58, no. 9, Art. no. 279, 2025. https://doi.org/10.1007/s10462-025-11287-7

[3] X. Tao, X. Gong, X. Zhang, S. Yan, and C. Adak, "Deep learning for unsupervised anomaly localization in industrial images: A survey," IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1–21, 2022. https://doi.org/10.1109/TIM.2022.3196436

[4] Z. Zhang, Z. Zhao, X. Zhang, C. Sun, and X. Chen, "Industrial anomaly detection with domain shift: A real-world dataset and masked multi-scale reconstruction," Computers in Industry, vol. 151, Art. no. 103990, 2023. https://doi.org/10.1016/j.compind.2023.103990

[5] C. Wang, W. Zhu, B.-B. Gao, Z. Gan, J. Zhang, Z. Gu, S. Qian, M. Chen, and L. Ma, "Real-IAD: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2024, pp. 22883–22892. https://doi.org/10.1109/CVPR52733.2024.02159

[6] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for

unsupervised anomaly detection," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9584–9592. https://doi.org/10.1109/CVPR.2019.00982

[7] S. Li, C. Wu, and N. Xiong, "Hybrid architecture based on CNN and Transformer for strip steel surface defect classification," Electronics, vol. 11, no. 8, Art. no. 1200, 2022. https://doi.org/10.3390/ electronics11081200

[8] L. Zhao, Y. Zheng, T. Peng, and E. Zheng, "Metal surface defect detection based on a Transformer with multi-scale mask feature fusion," Sensors, vol. 23, no. 23, Art. no. 9381, 2023. https://doi.org/10.3390/s23239381

[9] B. Tang, Z.-K. Song, W. Sun, and X.-D. Wang, "An end-to-end steel surface defect detection approach via Swin Transformer," IET Image Processing, vol. 17, no. 5, pp. 1334–1345, 2023. https://doi.org/10.1049/ipr2.12715

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. https://doi.org/10.1109/CVPR. 2016.90

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, LNCS 9351, 2015, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4 28

[12] T.-Y. Lin, P. Doll´ar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117–2125. https://doi.org/10.1109/CVPR.2017.106

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems 30, 2017, pp. 5998–6008. https://papers.nips.cc/paper/7181-attention-is-all-you-need

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in International Conference on Learning Representations (ICLR), 2021.

[15] A. Kirillov, E. Mintun, N. Ravi, et al., "Segment Anything," in Proc. IEEE/CVF International Conf. Computer Vision (ICCV), 2023, pp. 4015–4026.

[16] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, "Faster Segment Anything: Towards lightweight SAM for mobile applications,"

[17] W. Ji, J. Li, Q. Bi, T. Liu, W. Li, and L. Cheng, "Segment Anything is not always perfect: An investigation of SAM on different real-world applications," Machine Intelligence Research, vol. 21, no. 4, pp. 617–630, 2024. https://doi.org/10.1007/s11633-023-1385-0

[18] B. Hu, B. Gao, C. Tan, T. Wu, and S. Z. Li, "Segment Anything in defect detection,"

[19] K. Song, W. Cui, H. Yu, X. Li, and Y. Yan, "SAM era: Can it segment any industrial surface defects?" Computers, Materials & Continua, vol. 78, no. 3, pp. 3953–3969, 2024. https://doi.org/10.32604/cmc.2024.048451

[20] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

[21] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional onestage object detection," in Proc. IEEE/CVF International Conf. Computer Vision (ICCV), 2019, pp. 9626–9635. https://doi.org/10.1109/ICCV.2019.00972

[22] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7482–7491. https://doi.org/10.1109/CVPR.2018.00781

[23] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in Proc. Fourth International Conf. 3D Vision (3DV), 2016, pp. 565–571.

https://doi.org/10.1109/3DV.2016.79

[24] B. Han, Q. Yao, X. Yu, et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in Advances in Neural Information Processing Systems 31, 2018, pp. 8527–8537. NeurIPS paper page.

[25] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in Advances in Neural Information Processing Systems 31, 2018, pp. 8778–8788. NeurIPS paper page.

[26] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579–2605, 2008. https://www.jmlr.org/papers/v9/vandermaaten08a.html

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. IEEE International Conf. Computer Vision (ICCV), 2017, pp. 618–626. https://doi.org/10.1109/ICCV.2017.74