Contrastive Learning Based Knowledge Distillation for Enhancing Defect Detection

Jing-Ming Guo[*], Lun-Da Yuan, Cian Huang, and Yi-Chong Zeng
[*]National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.
E-mail: jmguo@seed.net.tw Tel: +886-2-27303241

Abstract— **Defect detection is critical for maintaining industrial production quality. This paper introduces pre-trained contrastive learning models to enhance defect detection. We integrate the simCLR model with two methods: fine-tuning and training from scratch. Fine-tuning adjusts pre-trained weights for defect detection tasks, yielding superior performance with reduced loss and higher Top-1 Accuracy. Training from scratch initializes model parameters specifically for anomaly detection, offering a tailored approach. The method employs the reverse knowledge distillation (RD) model with a simCLR-trained backbone, leveraging self-supervised and unsupervised learning exclusively on defect-free samples for practical implementation. Evaluation is conducted on the MVTec Anomaly Detection Dataset, encompassing 15 industrial product categories with high-resolution images. The experiment results demonstrate that fine-tuning achieves optimal performance, maintaining lower loss and higher Top-1 Accuracy.**

I. INTRODUCTION

Defect detection is critical in manufacturing to ensure product quality, reliability, and cost efficiency amidst automated production demands. Unsupervised and self-supervised learning methods have greatly improved defect detection by enhancing visual representation learning. Momentum Contrast (MoCo) improves self-supervised learning with a diverse set of negative samples [1]. SwAV learns visual features through cluster assignments without labeled data [2]. SimCLR shows that contrastive learning with simple designs and large batch sizes is effective [3].Siamese networks excel in one-shot learning and are useful for few-shot defect detection [4]. Knowledge distillation [5] helps smaller models learn from larger ones, improving performance. Bergmann et al. use a student-teacher framework for anomaly detection [6]. In the application of defect detection, recent advancements in deep learning technologies have introduced various approaches, such as Reverse Distillation (RD), CFA, and PatchCore. The RD improves anomaly detection by transferring knowledge from complex models to simpler ones. This technique uses one-class embeddings and distillation to enhance detection accuracy while making the process more efficient and less computationally demanding [7, 8]. CFA enhances anomaly localization by adjusting features with a coupled hypersphere method. This approach fine-tunes feature representations to better detect and localize specific anomalies, improving detection accuracy in complex scenarios [9]. PatchCore improves anomaly detection by using a memory bank to store features from normal samples. This approach helps in recognizing deviations from these patterns and effectively identifies subtle anomalies that might be overlooked otherwise [10].

This paper advocates using contrastive learning pre-trained models to enhance defect detection via knowledge distillation. Key contributions include:

- Recommend the contrastive learning pre-trained models to boost knowledge distillation model performance in defect detection.
- Explore and compare fine-tuning and training from scratch approaches for simCLR models, demonstrating that fine-tuning reduces loss and achieves high Top-1 accuracy.
- Enhance detection performance by employing pre-trained backbone architectures as auxiliary models in the RD model.

The rest of this paper is organized as follows: Section II introduces the methodology. Section III reveals the materials and experiment results. Sections IV and V are, respectively, the discussion and conclusion.

II. METHODOLOGY

The proposed method comprises the simCLR and the defect detection models. First, we introduce the processes of the training and the testing phases. Subsequently, the simCLR

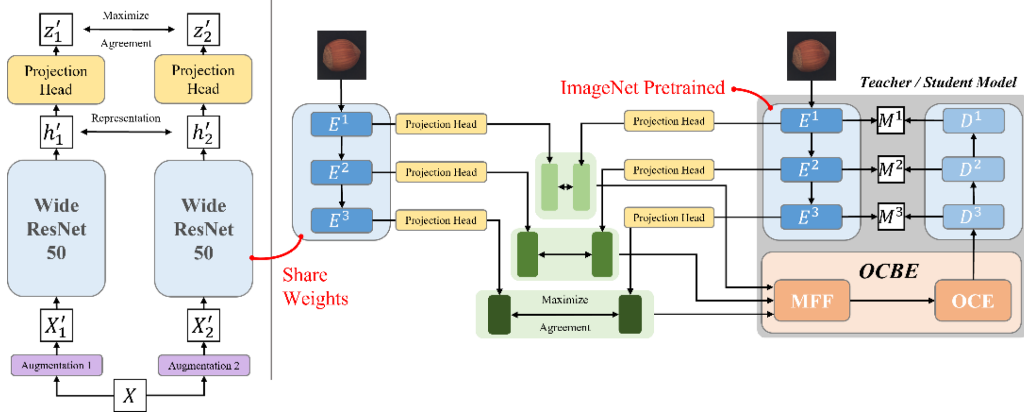

Fig. 1. Architecture flow chart of training phase

and defect detection models are presented in detail. Finally, anomaly scoring is defined for evaluating the performance.

A. Overall pipeline

First, we train the simCLR model [3], which serves as an auxiliary model for the Teacher Encoder (denoted as E^1、E^2、E^3) in anomaly detection via RD from the One-Class Embedding model. The proposed method enhances the performance of the Student Decoder (denoted as D^1、D^2、D^3) to normal images by training the teacher-student model. Fig. 1 shows the architecture flow chart of the training phase.

During the testing phase in Fig. 2, we remove the auxiliary model while retaining the trained mapping model named Projection Head. Then, defect images are inputted into the model sequentially. As the decoder learns the capability to reconstruct normal image features, encountering abnormal defects results in a significant difference between the feature representations outputted by the encoder and those reconstructed by the decoder. We present the feature extraction of each feature layer in the form of anomaly score maps (denoted as M^1、M^2、M^3) and finally fuse the differences of each layer to form the results of anomaly detection.

B. simCLR

The training methodology of simCLR uses contrastive learning. Given an input image x, data augmentation produces transformed images x'_1 and x'_2. These images are processed by a neural network, resulting in feature representations h'_1 and h'_2. These representations undergo non-linear transformation via a projection head, yielding outputs z'_1 and z'_2. Since both outputs originate from the same image x, the goal is to minimize the differences between z'_1 and z'_2 using a loss function, optimizing the model to reduce divergence.

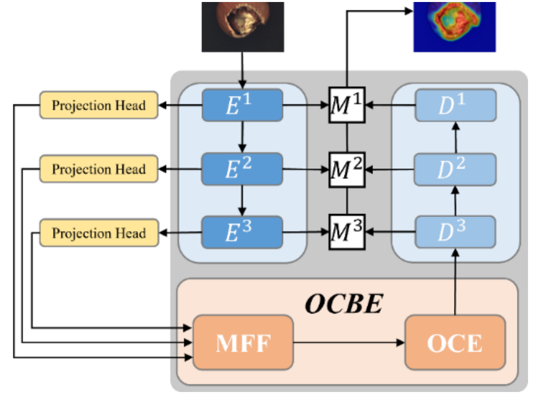

Fig. 2. Architecture flow chart of testing phase

The proposed method uses normal samples of the MVTec dataset as a benchmark. Data augmentation strategies include Random Resized Crop, Random Horizontal Flip, Random Color Jitter, Random Grayscale, and Gaussian Blur, which are applied to inputting RGB images. The backbone of the SimCLR model is Wide ResNet-50, whose projection head consists of two fully connected layers and a ReLU activation function shown in Fig. 1.

C. Defect Detection Model

The Defect Detection Teacher-Student model has two main parts: an auxiliary model on the left and an RD model [8] on the right. The left side uses a pre-trained simCLR model [3] based on the Wide ResNet-50 architecture, with weights shared from simCLR. This model includes a trainable projection head that helps with defect detection during testing. The right side features the RD model, which functions as a teacher-student system. It includes a pre-trained Wide ResNet-50 Teacher Encoder from ImageNet, an OCBE (One-Class Bottleneck Embedding) module, and a Student Decoder trained during the training phase. Both models process normal images, with the Wide ResNet-50 models

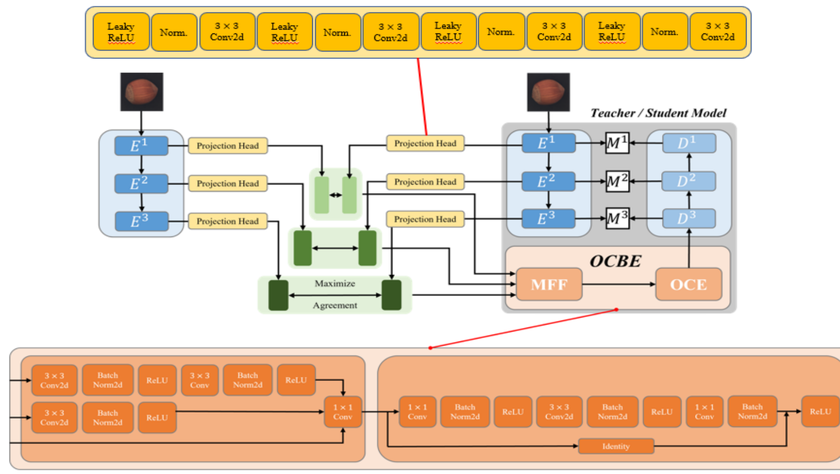

Fig. 3. Defect detection knowledge distillation model architecture – OCBE module and Projection head

pre-trained to enhance feature similarity, following principles similar to simCLR and contrastive learning.

Subsequently, feature representations flow into the OCBE module of the RD model for Multi-scale Feature Fusion (MFF) and dimensionality reduction through an OCE submodule. The Student Decoder reconstructs features from high to low levels, following reverse knowledge distillation principles. Cosine similarity continuously evaluates feature alignment between the Teacher Encoder and Student Decoder layers to ensure effective knowledge transfer.

Overall, the Defect Detection Knowledge Distillation model integrates two pre-trained Wide ResNet-50 backbones, three trainable projection heads, an OCBE module, and a Student Decoder, as shown in Fig 3. The left auxiliary model employs parameters derived from simCLR's Wide ResNet-50, while the RD model uses ImageNet pre-trained parameters.

D. Detection Mechanism

Before the testing phase, our method retains the trained projection head models and removes the pre-trained models used by the auxiliary model, as shown in Fig. 2. During the testing phase, defect images replace normal ones. In the teacher-student model, defect images undergo initial processing by the Teacher Encoder by transforming each defect image into a high-dimensional embedding for precise cosine similarity comparison. Afterward, this embedding is passed to the Student Decoder, which reconstructs a normal image based on this information.

Feature extraction occurs layer by layer, capturing feature maps from each block. Those maps are inputted into the respective projection head models to fine-tune the auxiliary

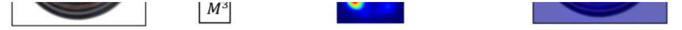

Fig. 4. Detection mechanism

model through convolution operations with trained parameters. Fed feature maps from each layer into the OCBE module, and the MFF module adjusts these maps through zero or multiple convolution operations to standardize the size and dimensionality. A point-wise convolution adapts the overall channel count before the fused feature map proceeds to the OCE module for dimensionality reduction.

The Student Decoder restores the feature map, starting from high-level deep features and progressing to lower-level layers, capturing representations from each block. Next, cosine similarity between the restored feature maps from the Student Decoder and those extracted by the Teacher Encoder is calculated. This comparison generates anomaly score maps (Anomaly Maps) for each layer, indicating differences between encoding and decoding layers. Fig.4 depicts the scaled Anomaly Maps superposed on an original image highlighting defect-affected regions.

III. Materials and Experiment Results

This paper utilized the MVTec AD dataset, a benchmark for anomaly detection. MVTec AD simulates real-world industrial scenarios, a mix of normal and naturally occurring defect samples. Each category includes training and test sets with annotations. The training set contains only normal images, and the test set includes various defect types with annotations. Fig.5 shows normal images, defect images, and labeled data.

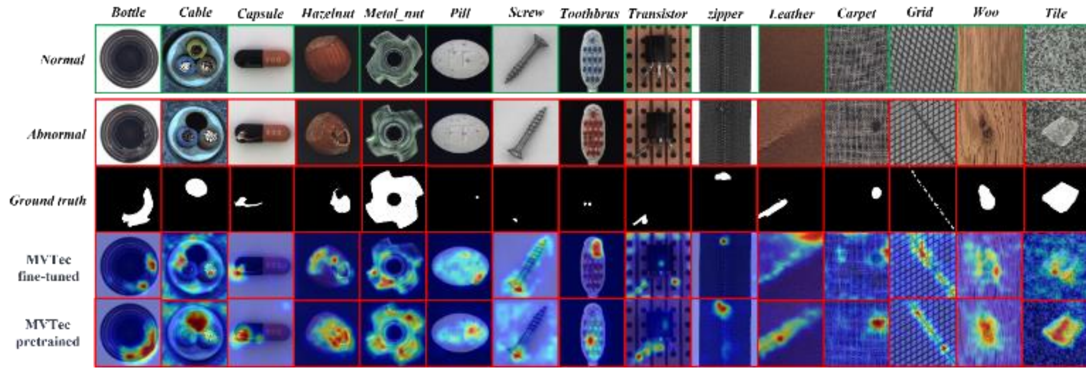

Fig. 5. MVTec dataset (1st and 2nd rows), labeled data (3rd rows), and visual results (4th and 5th rows)

A. *simCLR and RD model*

In this experiment, we investigated two main aspects. First, we compared using an unsupervised learning (contrastive learning) model versus a supervised learning model as the teacher encoder in the RD model. Second, we evaluated the performance of the contrastive learning model when fine-tuned or trained from scratch within the defect detection teacher-student framework. Additionally, we assessed the training outcomes and detection performance of the defect detection model.

The results show that using the contrastive learning model to replace the RD model's teacher encoder was not as effective as a supervised learning approach with an ImageNet pre-trained backbone. However, 60% of the categories achieved AUC values above 0.9, indicating that the unsupervised pre-trained model is still effective for defect detection. Interestingly, training the backbone architecture from scratch produced better results. Table I lists the test results for both strategies, showing that the From Scratch method outperformed the Fine Tune approach.

B. *Performance Evaluation*

In the third experiment, we used the pre-trained backbone from the first experiment as an auxiliary model for fine-tuning the RD model's features. After fine-tuning, we removed the auxiliary model and kept only the trained mapping model. The results in Table II show improvements across categories. Specifically, training the backbone from scratch led to an average AUC increase from 83.07% to 90.51%. Further fine-tuning the backbone improved this to 91.86%, and training from scratch further increased it to 92.56%. Two key observations from the experiments are,

- Training the backbone architecture from scratch generally outperformed the fine-tuning approach in the knowledge distillation model.
- Using the pre-trained backbone architecture as an auxiliary model in subsequent knowledge distillation models was more effective than directly replacing the original teacher encode.

Additionally, the comparative analysis against the original RD model and existing defect detection methodologies, our proposed method achieved the highest AUC in 7 out of 15 categories, specifically Carpet, Grid, Leather, Tile, and Wood as listed in Table III. The fourth and fifth rows of Fig.5 display the visual results of defect detection using the fine-tuned and the pre-trained methods, respectively.

IV. Discussion

According to Table II, that the proposed method is weak in Cable, Metal nut, and Transistor due to issues with data augmentation techniques used during the backbone architecture training in the first experiment. As detailed in [3], simCLR employed augmentation methods Random Color Jitter reduces sensitivity to color variations, affecting the Cable category where color differences can lead to misclassifications as normal features.

Similarly, Random Horizontal Flip, which usually improves image recognition robustness, can cause misclassifications in the Metal nut category by treating rotated nuts as defects. In the Transistor category, variations like misplaced transistors (e.g., upside-down) are also seen as defects, which is problematic when no transistor is present.

It's important to explore alternative data augmentation techniques that enhance model performance without affecting defect detection accuracy. Such strategies can improve model stability and generalization, leading to more reliable defect detection across various datasets and scenarios.

Table I. Comparison of the AUC of each category and the original RD model in different pre-trained backbone architectures

	Bottle	**Cable**	**Capsule**	**Carpet**	**Grid**	**Hazelnut**	**Leather**	**Metal nut**
RD (Fine Tune)	0.8870	0.4300	0.9060	0.8480	0.9670	0.9100	0.8490	0.8360
RD (From Scratch)	**0.9030**	**0.8450**	**0.9200**	**0.8970**	**0.9780**	**0.9630**	**0.9860**	**0.8930**
	Pill	**Serew**	**Tile**	**Toothbrush**	**Transistor**	**Wood**	**Zipper**	
RD (Fine Tune)	0.8760	0.9570	**0.7800**	0.9230	0.5520	0.8450	0.8940	
RD (From Scratch)	**0.9080**	**0.9660**	0.7750	**0.9800**	**0.7370**	**0.8550**	**0.9700**	

Table II. Comparison of AUC of each category in different pre-trained backbone architectures

	Bottle	**Cable**	**Capsule**	**Carpet**	**Grid**	**Hazelnut**	**Leather**	**Metal_nut**
RD[8]	0.9870	0.9740	0.9870	0.9890	0.9930	0.9890	0.9940	0.9730
CFA[9]	**0.9884**	**0.9897**	**0.9911**	0.9928	0.9812	0.9885	0.9937	**0.9915**
PatchCore[10]	0.9860	0.9840	0.9880	0.9900	0.9870	0.9870	0.9930	0.9840
Proposed method (From Scratch)	0.9523	0.8651	0.9210	**0.9930**	**0.9951**	**0.9892**	**0.9949**	0.7588
	Pill	**Serew**	**Tile**	**Toothbrush**	**Transistor**	**Wood**	**Zipper**	
RD[8]	0.9820	**0.9960**	0.9560	**0.9910**	0.9250	0.9530	0.9820	
CFA[9]	**0.9893**	0.9891	0.9521	0.9896	**0.9806**	0.9153	**0.9902**	
PatchCore[10]	0.9740	0.9940	0.9540	0.9870	0.9630	0.9500	0.9880	
Proposed method (From Scratch)	0.9215	0.9278	**0.9586**	**0.9910**	0.7420	**0.9532**	0.9552	

Table III. Comparison with other current modeling methods

	Carpet	**Grid**	**Leather**	**Tile**	**Wood**	**Average**
RD[8]	0.9890	0.9930	0.9940	0.9560	0.9530	0.9770
CFA[9]	0.9928	0.9812	0.9937	0.9521	0.9153	0.9670
PatchCore[10]	0.9900	0.9870	0.9930	0.9540	0.9500	0.9748
Proposed method (From Scratch)	**0.9930**	**0.9951**	**0.9949**	**0.9586**	**0.9532**	**0.9790**

V. CONCLUSION

This paper presents a method to enhance defect detection in knowledge distillation models using a contrastive learning pre-trained model. We evaluated the simCLR model with two strategies, namely Fine Tune and From Scratch. Our results showed that Fine Tune achieved lower loss and higher Top-1 Accuracy than From Scratch. We improved the Reverse Knowledge Distillation model by integrating the simCLR-trained backbone as an auxiliary model during training, which was removed for testing. The Fine Tune approach achieved an average AUC of 91.86%, while the From Scratch approach achieved 92.56%. This unsupervised method, which only requires unlabeled normal samples, is well-suited for industrial applications. Future research may explore multi-layer feature fusion and diverse data augmentation strategies for simCLR to further improve performance.

REFERENCES

[1] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," in arXiv preprint arXiv:2003.04297, 2020.

[2] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, "Unsupervised Learning of Visual Features by Contrasting

Cluster Assignments," in arXiv preprint arXiv:2006.09882, 2020.

[3] T. Chen, S. Kornblith, M. Norouzi, & G. Hinton, "A simple framework for contrastive learning of visual representations," *International conference on machine learning*, pp. 1597-1607, 2020, PMLR.

[4] X. Chen, and K. He, "Exploring simple siamese representation learning," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.15750-15758, 2021.

[5] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in arXiv preprint arXiv:1503.02531, 2015.

[6] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4183–4192, 2020.

[7] H. Deng and X. Li., "Anomaly detection via reverse distillation from one-class embedding," *in Proceedings of the IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 9737–9746, 2022.

[8] T. D. Tien, A. T. Nguyen, N. H. Tran, T. D. Huy, S. Duong, m, C. D. T. Nguyen, and S. Q. H. Truong, "Revisiting Reverse Distillation for Anomaly Detection, " *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24511–24520, 2023.

[9] S. Lee, S. Lee, B.-C. Song, "CFA: coupled-hypersphere based feature adaptation for target-oriented anomaly localization," *in IEEE Access*, no. 10, pp. 78446–78454, 2022.

[10] K. Roth, L. Pemula, J. Zepeda, B. Scholkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14298–14308, 2022.