Automated Industrial Anomaly Detection for Eraser Quality Inspection Using Multi-Channel Deep Learning Models

Aritra Das
EMJM in IFRoS, Faculty of Science
Universitat de Girona
Girona, Spain
u6110627@udg.edu

Muhammad Yamin
EMJM in IFRoS, Faculty of Science
Universitat de Girona
Girona, Spain
u6110888@udg.edu.

***Abstract*—Industrial anomaly detection plays a crucial role in automated quality control for manufactured products. This study focuses on automating the inspection process of erasers produced by the MILAN Stationery Company. Each product package contains 15 erasers, and images are captured using a fixed-angle camera system that provides six-channel data, namely AM, GRAY, G, R, W, and an additional NORMAL channel. The objective is to determine whether the printed logo on each eraser is clear and free from defects, which is formulated as a binary classification problem. To address this task, several pretrained convolutional neural network (CNN) models were trained and evaluated, including ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. Experimental results on the MILAN eraser dataset, which consists of 1,381 multi-channel images, demonstrate that the fine-tuned ResNet-50 model achieves the best overall performance. Specifically, the proposed approach attains an AUC-PR of 0.7236 and an F1-score of 0.6376 on the test set, outperforming the other ResNet backbone variants.**

***Index Terms*—Image classification, CNN, Industrial anomaly, MILAN dataset, Transfer learning, Quality control.**

I. Introduction

Industrial anomaly detection plays an important role in modern quality control, where companies must quickly identify defective products before they reach customers [1]. In this project, we focus on detecting defects in erasers produced by MILAN Stationary Company. Their automated imaging system captures packs of 15 erasers across six channels, and the goal is to identify printing defects and shape-related issues from these images. This problem is challenging because real-world labels are noisy, defects can be very small, and the dataset is imbalanced.

Convolutional Neural Networks (CNNs) have become a powerful tool for binary image classification. They can extract features from images automatically and learn patterns that are difficult to detect by hand. CNNs have shown strong performance in many industrial inspection tasks, and they help reduce manual effort while improving reliability [2]. Transfer learning further improves performance by using knowledge learned from large datasets and adapting it to smaller industrial datasets. It allows models to learn faster and generalize better, especially when the available training data is limited or imbalanced [3].

Our key contributions are as follows:

- We found that the binary annotations in the dataset were not always correct. We manually inspected and corrected these labels, which helped reduce class imbalance and improved the overall training quality.
- We fine-tuned a pretrained ResNet-50 model on our corrected dataset. This allowed us to use the advantages of transfer learning and achieve better feature learning for this task.
- We address dataset imbalance by combining class-weighted loss and weighted random sampling during training. This strategy reduces bias toward the majority class and improves learning on minority defect samples.

The rest of this paper is organized as follows. Section II describes our methodology which includes mainly data preprocessing and model description. Section III represents the experimental setup for the proposed methodology. In section IV we presented our findings. Section V briefly discusses the rationale behind the chosen methodology in relation to the obtained results. Finally, section VI summarizes insights and future improvements.

II. Methodology

This section presents the dataset description, preprocessing steps, and the model architecture. Fig. 1 illustrates the proposed methodology adopted in this study.

A. Dataset Analysis

1) Dataset Description: The dataset provided by MILAN contains 1381 samples of erasers captured using a fixed-angle industrial camera system [4]. Each eraser is represented as a 310×310×6 array, formed by stacking six grayscale channels: GRAY, AM, G, R, W, and NORMAL. The dataset includes

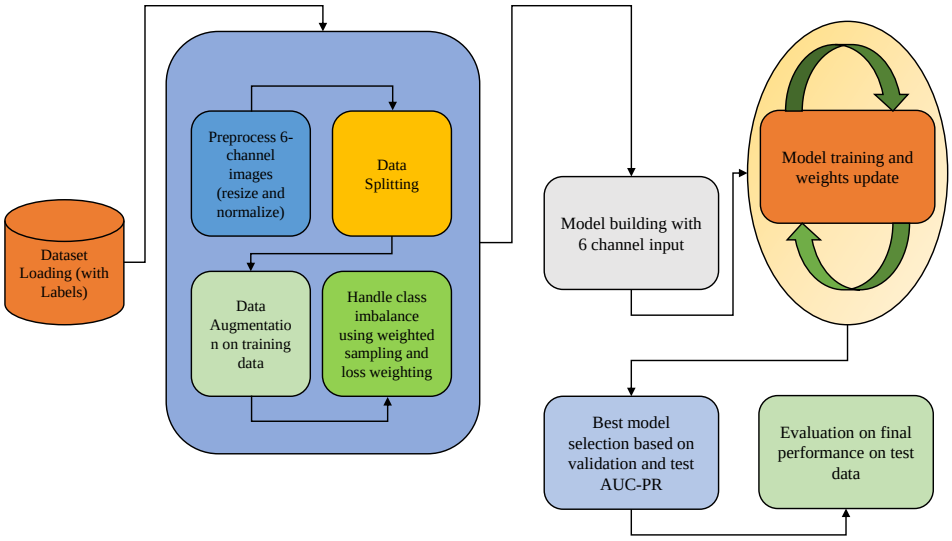


Fig. 1: Proposed methodology diagram of the developed framework.

binary annotations. The images show a variety of printing defects and physical defects. Fig. 2 represents some sample visualization from the dataset.

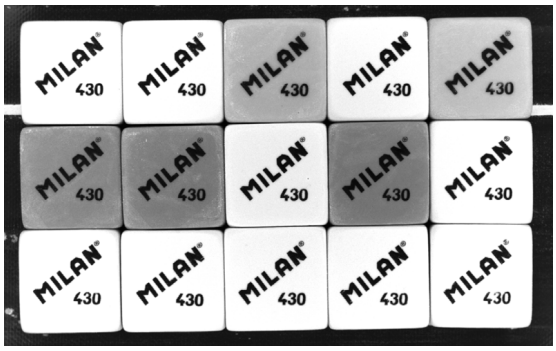


Fig. 2: Data samples visualization.

B. Preprocessing

1) Annotation Loading and Label Standardization: The dataset annotations are loaded from a CSV file containing sample identifiers and defect labels. Column names are cleaned and standardized to ensure consistency across the pipeline. All labels are converted into a binary numerical format, where 0 represents non-defective samples and 1 represents defective samples.

2) Multi-Channel Image Loading: Each sample is stored as a NumPy array consisting of six image channels. During loading, all arrays are converted to floating-point representation. When pixel intensity values exceed unity, normalization is applied by scaling the values to the range $[0, 1]$, which improves numerical stability during training.

3) Stratified Dataset Splitting: The dataset is divided into training and validation subsets using an 80:20 ratio with stratified sampling. This strategy preserves the original class distribution in both subsets, which is essential for reliable evaluation in the presence of class imbalance.

4) Training Data Augmentation: To improve model generalization, data augmentation is applied exclusively to the training set. Each six-channel image is resized to a fixed resolution of 310×310 pixels. Random spatial transformations, including horizontal and vertical flipping, rotation, translation, and scaling, are employed to simulate real-world variability and reduce overfitting. Table I represents the data augmentation pipeline used in this study.

5) Validation Data Preprocessing: For the validation set, only deterministic preprocessing is performed. Images are resized to the same fixed resolution without applying any random transformations. This ensures consistent and unbiased performance evaluation.

6) Class Imbalance Handling: To address class imbalance during training, a weighted random sampling strategy is adopted. Samples belonging to the minority class are assigned higher sampling weights, enabling balanced exposure of both classes during mini-batch formation.

7) Tensor Conversion for Model Input: Finally, all preprocessed images are converted into PyTorch tensors using a channel-first format $(6, H, W)$. Corresponding labels are also converted into tensor form, preparing the data for efficient input into the deep learning model.

C. Data augmentation pipeline:

TABLE I: Data augmentation techniques applied during training

Technique	Description
Image Resizing	All images are resized to a fixed resolution of 310×310 pixels to ensure consistent input dimensions.
Random Horizontal Flip	Images are randomly flipped horizontally with a probability of 0.5 to model left–right variations.
Random Vertical Flip	Images are randomly flipped vertically with a probability of 0.5 to improve robustness to orientation changes.
Random Rotation	Images are randomly rotated within $\pm 15^{\circ}$ to handle slight angular misalignments.
Random Translation	Images are randomly translated by up to 10% of image dimensions to simulate positional variation.
Random Scaling	Images are randomly scaled within the range of 0.85 to 1.15 to account for size variability.

D. Model Architecture

The proposed model is based on the ResNet-50 architecture, which is a deep convolutional neural network with residual connections. These residual connections help the network train deeper layers more effectively and improve feature learning [5].

The original ResNet-50 model is designed for three-channel RGB images. In this work, the first convolutional layer is modified to accept six-channel input images. The kernel size, stride, and padding of this layer remain unchanged. The pretrained weights of the RGB channels are duplicated to initialize the additional channels, allowing effective transfer learning.

All remaining convolutional layers of the network are kept unchanged. These layers learn hierarchical feature representations from the multi-channel input images.

The final fully connected layer is replaced with a single-neuron output layer for binary classification. A sigmoid activation function is applied to produce the final defect probability.

III. Experimental setup:

TABLE II: Experimental setup and training hyperparameters

S.No.	Hyperparameter	Value
1	Input image size	310×310
2	Number of input channels	6
3	Batch size	16
4	Number of epochs	50
5	Optimizer	AdamW
6	Learning rate	1×10^{-4}
7	Weight decay	1×10^{-2}
8	Loss function	BCEWithLogitsLoss
9	Positive class weighting	pos_weight $= \frac{N_{neg}}{N_{pos}}$
10	Learning rate scheduler	ReduceLROnPlateau
11	Scheduler monitor	Validation AUC-PR
12	Scheduler patience	10 epochs
13	Scheduler reduction factor	0.1
14	Class imbalance handling	WeightedRandomSampler
15	Evaluation metrics	F1-score, AUC-PR

The experimental setup was designed to ensure stable training and reliable evaluation of the proposed methodology. All input images were resized to 310×310 and processed as six-channel inputs to fully exploit the available multi-channel information. The model was trained for 50 epochs with a batch size of 16.

The AdamW optimizer was employed with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-2} to achieve better convergence and reduce overfitting. Binary Cross-Entropy with Logits loss was used, along with positive class weighting, to address class imbalance in the dataset. In addition, a WeightedRandomSampler was applied during training to further mitigate the imbalance issue.

A ReduceLROnPlateau learning rate scheduler was utilized and monitored based on the validation AUC-PR metric. The scheduler reduced the learning rate when performance stagnated, with a patience of 10 epochs and a reduction factor of 0.1. Model performance was evaluated using F1-score and AUC-PR, which are suitable metrics for imbalanced classification tasks. Overall, this experimental configuration ensures robust training and effective performance evaluation of the proposed model. Table II reprsents the experimental set up followed in this study.

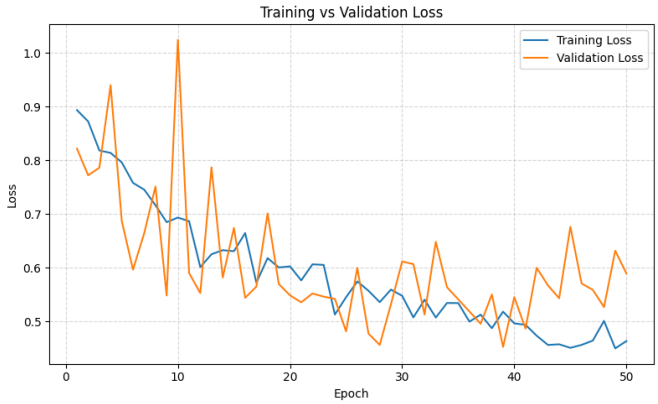
IV. Results

The training and validation curves show the learning behavior of the proposed ResNet50 model. In the AUC-PR curve, the training performance increases steadily over epochs. The validation AUC-PR also improves and follows a similar trend, with small fluctuations. This indicates stable learning and good generalization. In the loss curve, the training loss decreases consistently as epochs increase. The validation loss also shows an overall decreasing trend with some variations. These curves suggest that the model converges well without severe overfitting. Fig. 3 represents training and validation performance curves of the proposed model.

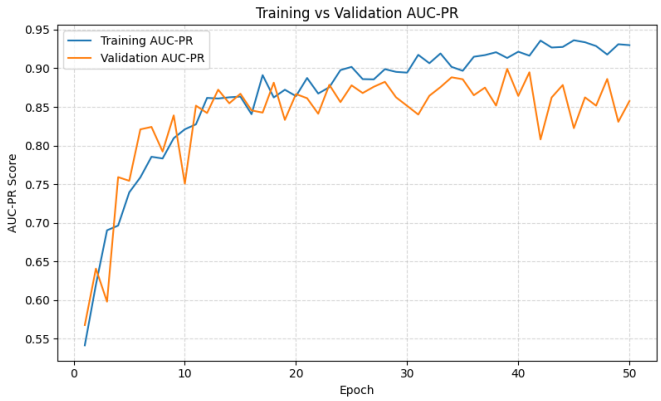
TABLE III: Performance comparison of the proposed model with and without data augmentation

Setting	Train	Val	Test
	AUC-PR		
With Augmentation	**0.913**	**0.899**	**0.7236**
Without Augmentation	1.000	0.831	0.5380
	F1-score		
With Augmentation	**0.804**	**0.802**	**0.6376**
Without Augmentation	0.999	0.752	0.5160

As shown in table III, a comparative analysis was conducted to evaluate the effect of data augmentation on the performance of the proposed ResNet50 model. When data augmentation was applied, the model showed better generalization, with higher validation and test AUC-PR and F1-score compared to the non-augmented setting. Without augmentation, the model achieved almost perfect training performance, indicating overfitting. However, its validation and test results dropped significantly. These results clearly show that data augmentation



(a) Epochs vs Loss



(b) Epochs vs AUC-PR

Fig. 3: Training and validation performance curves of the proposed model. (a) Epoch vs Loss. (b) Epoch vs AUC-PR.

helps the ResNet50 model learn more robust and transferable features, leading to improved performance on unseen data.

TABLE IV: Performance comparison of the proposed ResNet50 model using different optimizers

Optimizer	Train	Val	Test
AUC-PR			
ResNet50 + AdamW	**0.913**	**0.899**	**0.7236**
ResNet50 + Adam	0.900	0.870	0.6630
ResNet50 + SGD	0.560	0.457	0.3340
F1-score			
ResNet50 + AdamW	**0.804**	**0.802**	**0.6376**
ResNet50 + Adam	0.816	0.776	0.6200
ResNet50 + SGD	0.666	0.566	0.3070

An experimental evaluation was carried out to investigate the impact of different optimizers on the performance of the proposed ResNet50 model (shown in table IV). Among all the tested optimizers, AdamW achieved the best overall performance, yielding the highest validation and test AUC-PR and F1-scores. The Adam optimizer also demonstrated strong results but showed slightly weaker generalization compared to AdamW. In contrast, SGD exhibited substantially poorer performance across the training, validation, and test sets. These findings suggest that adaptive optimizers, particularly AdamW, are more effective for training the proposed ResNet50 model.

TABLE V: Performance comparison of the proposed ResNet50 model using different learning rates

Learning Rate	Train	Val
AUC-PR		
1×10^{-2}	0.462	0.576
1×10^{-3}	0.510	0.526
1×10^{-4}	**0.913**	**0.899**
1×10^{-5}	0.882	0.830
F1-score		
1×10^{-2}	0.659	0.547
1×10^{-3}	0.651	0.547
1×10^{-4}	**0.804**	**0.802**
1×10^{-5}	0.789	0.746

TABLE VI: Performance comparison of the proposed ResNet50 model using different weight decay rates

Weight Decay	Train	Val
AUC-PR		
1×10^{-2}	**0.913**	**0.899**
1×10^{-3}	0.868	0.880
F1-score		
1×10^{-2}	**0.804**	**0.802**
1×10^{-3}	0.777	0.755

In table V, a comparative analysis was performed to study the effect of different learning rates on the proposed ResNet50 model. A high learning rate (1×10^{-2}) led to poor training and validation performance, indicating unstable learning. The learning rate of (1×10^{-3}) showed slight improvement but remained suboptimal. The best results were obtained with a learning rate of (1×10^{-4}), which achieved the highest validation AUC-PR and F1-score. Further reducing the learning rate to (1×10^{-5}) caused a decline in performance. These results indicate that an appropriate learning rate is critical for stable training and good generalization.

An ablation study was conducted to assess the impact of different weight decay rates on the proposed ResNet50 model (shown in table VI). A weight decay of (1×10^{-2}) achieved the best training and validation performance, resulting in higher AUC-PR and F1-scores and demonstrating improved generalization. When the weight decay was reduced to (1×10^{-3}), both training and validation performance declined, indicating weaker regularization and less stable learning. Overall, selecting an appropriate weight decay value is crucial for controlling overfitting and enhancing validation performance.

A model-to-model comparison was conducted using AUC-PR and F1-score on training, validation, and test sets, which is given in table VII. ResNet18 and ResNet34 achieved strong training performance but showed lower generalization on the test set. ResNet50 provided the best validation and test AUC-PR with stable F1-scores. ResNet101 showed similar valida-

TABLE VII: Model-to-model performance comparison on training, validation, and test sets

Model	Train AUC-PR	Val AUC-PR	Test AUC-PR	Train F1	Val F1	Test F1
ResNet18	0.926	0.876	0.7050	0.840	0.796	0.5928
ResNet34	0.925	0.873	0.7158	0.850	0.776	0.6000
ResNet50	**0.913**	**0.899**	**0.7236**	**0.804**	**0.802**	**0.6376**
ResNet101	0.892	0.873	0.6617	0.816	0.806	0.6180
ResNet152	0.908	0.870	0.7156	0.826	0.750	0.6419

tion performance but lower test AUC-PR. ResNet152 achieved good test F1-score but weaker validation performance. Overall, ResNet50 offered the most balanced and reliable performance among all models.

TABLE VIII: Validation classification performance of different ResNet architectures

Model	Class	Precision	Recall	F1-score
ResNet18	No Defect	0.87	0.89	0.88
	Defect	0.81	0.79	0.80
ResNet34	No Defect	0.90	0.78	0.84
	Defect	0.71	0.86	0.77
ResNet50	**No Defect**	**0.88**	**0.90**	**0.89**
	Defect	**0.83**	**0.80**	**0.81**
ResNet101	No Defect	0.90	0.85	0.87
	Defect	0.77	0.85	0.81
ResNet152	No Defect	0.82	0.95	0.88
	Defect	0.90	0.66	0.76

A comparative analysis was performed on different ResNet architectures using validation results, which is given in table VIII. ResNet18 showed stable and balanced performance for both classes. ResNet34 improved recall for the Defect class but reduced precision for the No Defect class. ResNet50 achieved the most balanced results, with high precision, recall, and F1-score for both classes. ResNet101 showed good recall for the Defect class but a slight drop in precision. ResNet152 achieved very high recall for the No Defect class but struggled to detect Defect samples. Overall, ResNet50 provided the best trade-off between accuracy and class balance.

The validation PR curves show the performance of different ResNet backbones in Fig. 4. All models maintain high precision at low recall values, indicating reliable predictions for confident samples. As recall increases, precision gradually decreases for all models, which is expected in imbalanced classification tasks. ResNet50 shows the best overall curve, maintaining higher precision across a wider recall range and achieving the highest AUC-PR of 0.90. ResNet18, ResNet34, ResNet101, and ResNet152 show similar trends with slightly lower AUC-PR values around 0.87–0.88. The sharper drop in precision at high recall for deeper models indicates reduced stability in detecting harder samples. Overall, ResNet50 provides the most balanced and robust validation performance among all architectures.

A. Confusion matrix:

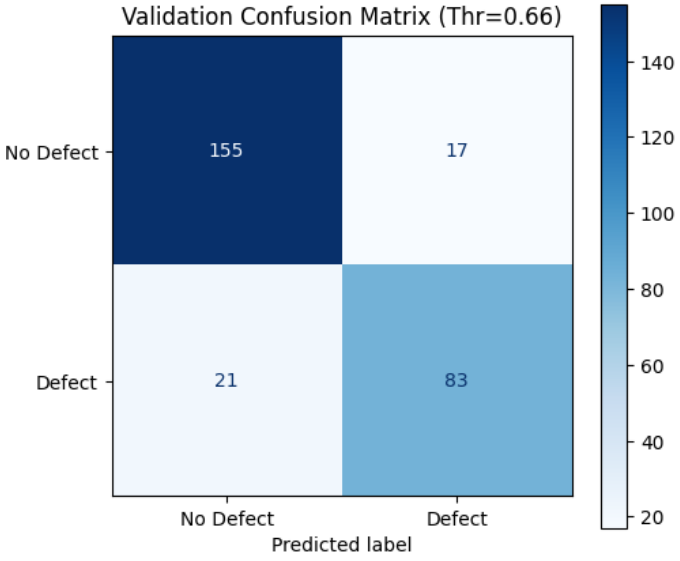


Fig. 5: Confusion matrix on validation dataset for Resnet-50.

The confusion matrix shows how the ResNet-50 model performs on the validation dataset in Fig. 5. It compares the true labels with the predicted labels at a decision threshold of 0.66.

The model correctly classified 155 non-defective samples as non-defective, while 17 non-defective samples were incorrectly predicted as defective. This indicates that the model performs well in identifying normal products and produces relatively few false alarms.

For defective samples, the model correctly detected 83 defects, but 21 defective samples were misclassified as non-defective. This shows that most defects are successfully detected, although a small number are missed.

Overall, the results indicate good classification performance with a reasonable balance between false positives and false negatives. The chosen threshold helps improve defect detection while maintaining stable performance on non-defective samples, making the model suitable for automated quality inspection.

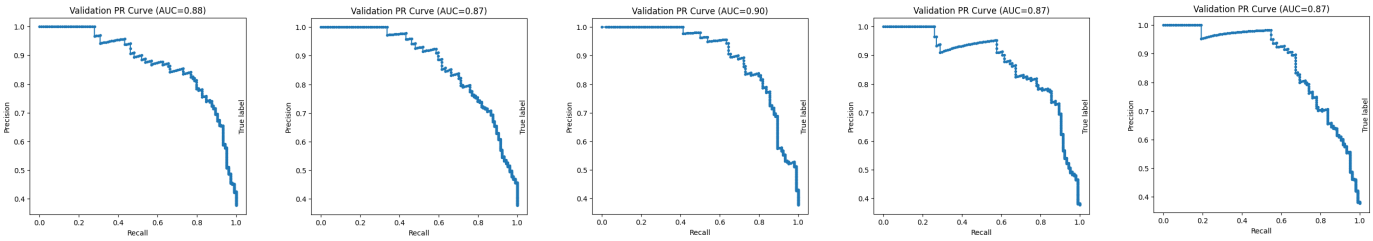


Fig. 4: Validation AUC-PR curves of different ResNet backbones (left to right): ResNet18 (0.88), ResNet34 (0.87), ResNet50 (0.90), ResNet101 (0.87), ResNet152 (0.87).

V. Discussion

An ablation study was conducted to analyze the contribution of different components and hyperparameters to the performance of the proposed ResNet50 model. Each experiment was performed by modifying one factor at a time while keeping the remaining settings unchanged.

First, the effect of data augmentation was evaluated. Without augmentation, the model achieved almost perfect training performance but showed a large drop in validation and test results, indicating severe overfitting. When data augmentation was applied, both validation and test AUC-PR and F1-scores improved significantly. This confirms that augmentation is essential for improving generalization.

Next, different optimizers were analyzed. AdamW achieved the best performance across all metrics, followed by Adam. SGD showed poor results and unstable learning behavior. This demonstrates that adaptive optimizers, especially AdamW, are more effective for training the proposed model.

The impact of learning rate was also examined. A learning rate of 1×10^{-4} produced the best validation AUC-PR and F1-score. Higher learning rates led to unstable training, while lower learning rates reduced performance. This shows that proper learning rate selection is critical for stable convergence.

Weight decay was evaluated to study its regularization effect. A weight decay of 1×10^{-2} resulted in better validation performance compared to 1×10^{-3}. This indicates that stronger regularization helps control overfitting in the proposed model.

Finally, different ResNet backbones were compared to justify the model choice. Although deeper models achieved competitive results, ResNet50 provided the best balance between validation and test performance. It also maintained stable learning with lower complexity compared to deeper variants. Overall, the ablation results confirm that the chosen configuration of ResNet50, along with data augmentation, AdamW optimizer, and carefully tuned hyperparameters, leads to optimal performance for the task.

VI. Conclusion

This study proposed an automated industrial anomaly detection framework for eraser quality inspection using multi-channel image data. Several pretrained CNN models were evaluated, and a detailed ablation study was conducted to analyze the impact of different components and hyperparameters. The results showed that data augmentation significantly improves model generalization by reducing overfitting. The choice of optimizer and learning rate was also found to be critical, where AdamW combined with a learning rate of 1×10^{-4} produced stable and consistent performance. Weight decay further helped in controlling overfitting during training.

Among all evaluated backbone networks, ResNet50 achieved the best balance between performance and model complexity. It outperformed both shallower and deeper ResNet variants in terms of validation and test AUC-PR and F1-score. Overall, the results demonstrate that the proposed ResNet50-based approach is effective and reliable for industrial eraser defect detection. Future work will focus on larger datasets, real-time deployment, and further optimization for practical industrial applications.

References

[1] Jiaqi Liu, Guoyang Xie, Jinbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *Machine Intelligence Research*, 21(1):104–135, 2024.

[2] Benjamin Staar, Michael Lütjen, and Michael Freitag. Anomaly detection with convolutional neural networks for industrial surface inspection. *Procedia CIRP*, 79:484–489, 2019.

[3] Ângela Semitela, Miguel Pereira, António Completo, Nuno Lau, and José P Santos. Improving industrial quality control: A transfer learning approach to surface defect detection. *Sensors*, 25(2):527, 2025.

[4] MILAN - Google Drive — drive.google.com. https://drive.google.com/drive/folders/1fjHdoNhW1p0_m5X2GaFXPQBo6LWcACOq. [Accessed 31-12-2025].

[5] Brett Koonce. Resnet 50. In *Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization*, pages 63–72. Springer, 2021.