Foreground-Aware Knowledge Distillation for Enhanced Damage Detection

Pantelis Menteidis, Christos Papaioannidis, and Ioannis Pitas

Aristotle University of Thessaloniki, Thessaloniki 541 24, Greece {mentpant, cpapaionn, pitas}@csd.auth.gr

Abstract. Damage detection remains a critical challenge, especially within the industrial automation sector, necessitating the development of advanced inspection technologies and their potential applications. Conventional industrial inspection methods are hindered by high costs and operational disruptions, motivating the development of innovative and efficient solutions. This paper introduces a novel, architecture-agnostic deep neural network (DNN) knowledge distillation (KD) method able to enhance vision-based damage detection performance even in challenging industrial environments. Our proposed method integrates foreground knowledge with feature KD to enhance data feature utilization in detection models, effectively minimizing background clutter. The results demonstrate the efficiency of our method in consistently enhancing the student's training process, including up to a 12% increase in mean Average Precision (mAP), across various DNN architectures. Our approach bridges the gap between academic research and real-world industrial current applications, offering a robust solution for damage detection in insulated pipelines.

Keywords: Industrial inspection · Damage detection · Feature knowledge distillation

1 Introduction

Damages in the industrial pipelines lead to inefficiencies and leakages, harming plant productivity operations. Corrosion Under Insulation (CUI), if not properly managed, potentially causes catastrophic failures. While manual inspection is possible, it is often difficult and time-consuming due to the height and complexity of plant environments. The modern industrial automation sector urgently requires advanced vision inspection technologies to enhance the maintenance of sensitive industrial infrastructures. This need is particularly pronounced in refinery industries, where inspecting insulated pipes is essential for maintaining operational flow and energy efficiency. Deep Neural Network (DNN) models have significantly advanced the field of industrial defect detection. They offer superior precision, facilitating the automation of tasks that were previously deemed impossible. Recently, various vision damage detection studies employed DNN models for structural infrastructures, leveraging their enhanced state-of-the-art (SOTA) performance in a variety of tasks and applications [4, 8, 32, 33].

Fig. 1: Insulated pipes images from the PDI dataset [17].

Despite advancements in DNN models, a significant gap persists between academic research and real-world applications, particularly in complex outdoor environments where vision-based analysis algorithms face substantial challenges. DNN models are prone to failure when encountering real-world challenges that were not present in their training datasets. While significant advancements have been made in object detection with SOTA algorithms [10,11,14,16,35], these efforts often fall short of addressing the complexities and challenges encountered in real-world scenarios. Most benchmark datasets used to develop these algorithms are ideally structured and do not accurately reflect the complexities encountered in real-world applications. While these datasets are crucial for advancements in general detection tasks, they fail to address the issue of background clutter prevalent in real-world applications. This clutter introduces additional noise for the models, often resulting in poor performance in practical scenarios. As illustrated in Figure 1, the working conditions in industry data are highly challenging and cluttered. This underscores the necessity to adapt these models for such environments. The work of [17], which introduced the PDI dataset that includes images from various refineries, highlights that SOTA object detection models struggle in these difficult and cluttered environments. This underscores the critical need to adapt these models to such environments.

Motivated by the need for robust real-world solutions in the industry, we propose a novel feature Knowledge Distillation (KD) method specifically tailored for algorithms operating in clutter environments including industrial damage detection in insulated pipes. Our approach leverages feature KD incorporating foreground knowledge during the training phase aiming to improve vision-based industrial damage detection. To address background clutter, we employ a teacher-student framework where the teacher model is trained on preprocessed image data, and the student model is trained on original image data. The student leverages feature KD from the teacher, which processes the same data but with the preprocessing. This approach allows the student model to utilize the foreground knowledge from the teacher, maximizing the amount of relevant information used during its training. The term "relevant information" refers to the

data deemed useful by the model for accurately detecting the object of interest—in this case, the area of the pipe—without being confused by background noise. This approach significantly aids the DNN models in detecting the damages in challenging industrial environments where background clutter often degrades model performance. Notably, the proposed method is architecture-agnostic and can be integrated into various models by modifying only the training phase, making it straightforward to implement and highly versatile for integration purposes.

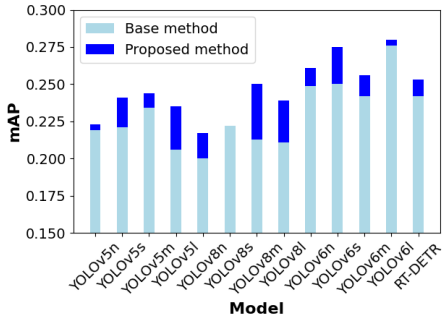

Fig. 2: Results overview.

We demonstrate the superiority of our approach compared to traditional feature KD methods [25, 27, 29, 38, 39] that do not utilize additional foreground information during the training phase. The proposed method is simple yet efficient, significantly improving SOTA object detection DNN models across nearly all benchmark metrics. Extensive experiments incorporating our method into various SOTA object detection models, surpassing the performance of the base training models as shown in Figure 2, highlight its generality. By experimenting with different loss functions and image preprocessing techniques, we optimized model performance, underscoring the importance of this training framework in adapting SOTA DNN algorithms to real-world industrial challenges.

In summary, this paper presents the following key contributions:

- We introduce a novel, architecture-agnostic DNN training approach utilizing feature KD, tailored specifically for improving damage detection in insulated pipes across real-world industrial settings. Our method efficiently maximizes the use of essential foreground information to achieve optimal damage detection results while avoiding background clutter.
- Our baseline models exhibit significant performance enhancements, including a 12% increase in mean Average Precision (mAP) along with other considerable improvements in metrics within the scope of damage detection in industrial insulated pipelines.

– Model performance has been optimized through extensive experimentation with various loss functions and image preprocessing techniques, illustrating the framework's ability to adapt SOTA DNN algorithms to practical industrial applications.

The remainder of this paper is organized as follows. Section 2 summarizes the related work. Section 3 presents the methodology of the training framework. Our experiments are detailed in Section 4, including a subsection on our ablation study in Section 4.1. Finally, conclusions and proposed future work are discussed in Section 5.

2 Related Work

Knowledge Distillation. KD, introduced by [7], involves transferring knowledge from a large, pre-trained teacher network to a smaller student network. Initially used for model compression, KD enhances performance through guided training. Feature-based KD methods, introduced by [25], focus on intermediate representations, with subsequent advancements made by [9, 18, 46]. Recent innovations include Mutual Information Maximization KD by [28] and flow-based techniques by [42]. Some works have aimed to improve the attention to the foreground objects via KD [40, 43, 44]. In detection tasks also, [19] proposed a Frequency Attention Module, and [45] developed the Feature-Richness Score method. In [41] and [38], they enhanced detector performance using feature KD, while [14] implemented self-distillation in YOLOv6, facilitating easy integration with high-performance detectors. Channel-wise KD for dense prediction tasks was explored by [29], whereas [39] and [27] addressed low-resolution face recognition using different input resolutions. These studies highlight KD's versatility and effectiveness, motivating our proposed method to enhance DNN model performance in real-world applications with suboptimal inputs.

Damage Detection. Damage detection, an extension of object detection, has significantly advanced in recent years with the advent of deep learning techniques that offer high precision and rapid inference, making them suitable for industrial applications. One-stage methods, such as YOLO [21] and SSD [15], have become prevalent due to their speed, whereas two-stage methods, like R-CNN [6], Fast R-CNN [5], and Faster R-CNN [24], are noted for their higher accuracy but slower performance. The field has seen extensive research and continuous updates to models like YOLO almost annually [2,10,11,14,22,23,34–36]. Recently, transformer-based algorithms have gained prominence in object detection, with models ranging from DETR [3] to real-time variants like RT-DETR [16], showcasing their potential to lead in this domain like in others. Also, DNN models have significantly advanced surface defect detection, enhancing accuracy and efficiency across various industrial applications [1,13,26,31,37]. In UAV inspection, deep learning models like Mask R-CNN [12] and methods integrating synthetic data [30] have improved infrastructure maintenance. Despite these advancements, challenges remain in cluttered backgrounds, particularly for UAV-based inspection.

3 Method

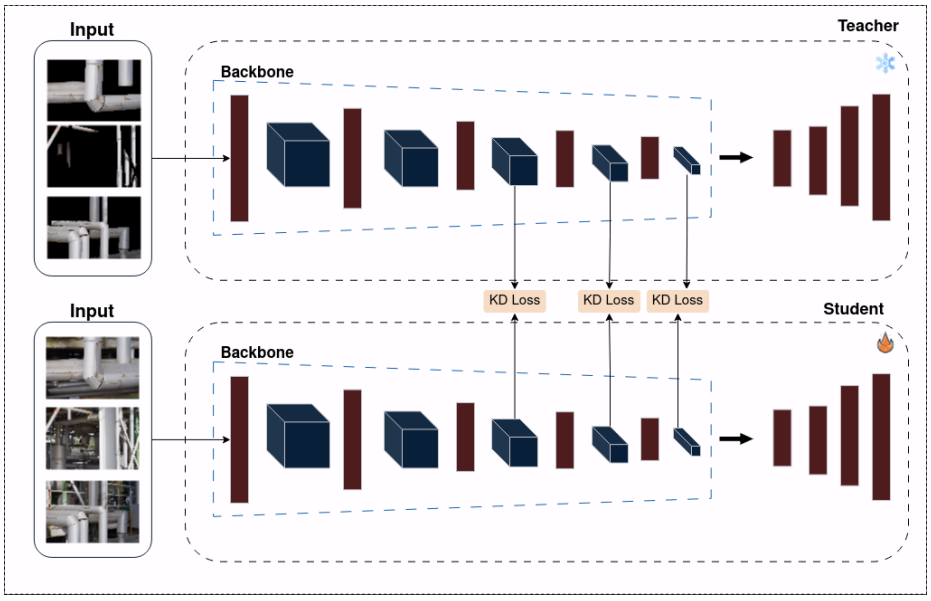

Fig. 3: Architecture of the proposed feature knowledge distillation (KD) method. The teacher model processes images without the cluttered background, while the student model processes the original images. The KD metric loss, applied to the final feature maps, encourages the student to mimic the features of the pretrained teacher on more informative images.

Our approach involves training the teacher model using preprocessed image data to focus on foreground features, followed by training the student model with the original image data utilizing the distilled foreground knowledge from the teacher model, thereby enhancing its capability to distinguish and recognize foreground objects. During training, the weights of the teacher model are kept frozen while the student model is trained on the dataset $\mathcal{D}$. For each training batch $\{(\mathbf{X_i}, \mathbf{a_i})\}_{i=1}^{C}$, where $\mathbf{X_i} \in \mathbb{R}^{C\times H\times W}$ represents the images, $\mathbf{a_i}$ denotes the detection ground truth labels, C is the batch size, the teacher model processes a corresponding batch $\{(\mathbf{X_{M_i}}, \mathbf{a_i})\}_{i=1}^{C}$ from the preprocessed dataset $\mathcal{D}_M$, where $\mathbf{X_{M_i}} \in \mathbb{R}^{C\times H\times W}$ represents the preprocessed images. The order and transformations applied to these batches are consistent across both models. Feature maps from the last layers of both models are compared using a metric loss, which is the distillation loss, and integrated into the overall loss function of the student model. This feature distillation approach brings the student's feature representations closer to the teacher's foreground feature representations, thereby enhancing the student's performance. The proposed method is illustrated in Figure 3,

where we utilize two DNN models of identical architecture and size. It is a two-stage training process where the teacher model is first trained on the task using $\mathcal{D}_M$.

Method	mAP	$\mathbf{mAP_{50}}$	$\mathbf{mAR_1}$	$\mathbf{mAR_{10}}$	$\mathbf{mAR_{100}}$
Original image	0.25	**0.505**	0.232	0.349	0.419
RB	**0.254**	0.473	**0.244**	0.360	0.430
BB	0.241	0.483	0.237	0.354	0.414
EF	0.245	0.452	0.242	0.351	0.427
SF	0.209	0.403	0.223	0.320	0.400
RB & EF	0.231	0.469	0.223	0.339	0.418
RB & SF	0.247	0.458	0.233	0.361	0.413
BB & EF	0.235	0.455	0.228	**0.388**	**0.429**
BB & SF	0.217	0.423	0.213	0.335	0.409

Table 1: Comparison of different foreground-background preprocessing methods on train and validation set for YOLOv6s [14] on mAP and mAR metrics.

As shown in Table 1, the use of image preprocessing with masks improves the performance of the YOLOv6s [14]. The simplest preprocessing technique involves removing the background and retaining only the pipe in the image, which results in a slight improvement in mAP and mAR_1 metrics. Beyond the Remove Background (RB) technique, we explored additional preprocessing methods. One such method involves applying a Gaussian filter to create a Blurred Background (BB). This technique aims to preserve some background information while introducing a smoothing effect by implementing a convolution function $\mathcal{T}_{\text{bgrnd}}$ with a Gaussian kernel. Additionally, we enhanced the foreground features through a sharpening filter, termed Sharpen Foreground (SF), which applies a Laplacian sharpening filter using the convolution function $\mathcal{T}_{\text{fgrnd}}$. Furthermore, to improve contrast, we employed histogram equalization on the foreground, referred to as Equalize Foreground (EF). We further experimented with combinations of these preprocessing techniques for both background and foreground to evaluate their impact on model performance. Histogram equalization adjusts the intensity distribution of the image to enhance contrast. The transformation is defined as:

$$\mathcal{T}_{\text{fgrnd}} = H(I) = \frac{L-1}{N} \sum_{j=0}^{i} h(j) \tag{1}$$

where $H(\mathbf{X})$ is the histogram-equalized intensity, L is the number of possible intensity levels, N is the total number of pixels, and $h(j)$ is the histogram count for intensity level j. These preprocessing transformations are illustrated in Figure 4. Notably, the combination of BB and EF significantly increased performance, with an improvement in mAR_{10} by nearly 0.04 (+11%) and in mAR_{100} by 0.01 (+2%).

Method	mAP	$\mathbf{mAP_{50}}$	$\mathbf{mAR_1}$	$\mathbf{mAR_{10}}$	$\mathbf{mAR_{100}}$
Original image	**0.25**	**0.505**	**0.232**	0.349	0.419
RB	0.23	0.422	0.219	0.342	**0.422**
BB	0.232	0.453	0.229	0.326	0.398
EF	0.25	0.448	0.245	**0.365**	0.428
SF	0.204	0.389	0.223	0.31	0.381
RB + EF	0.21	0.429	0.197	0.311	0.395
RB + SF	0.232	0.42	0.228	0.33	0.402
BB + EF	0.218	0.414	0.217	0.33	0.419
BB + SF	0.178	0.34	0.204	0.29	0.363

Table 2: Comparison of different foreground-background preprocessing methods on YOLOv6s [14], evaluated on mAP and mAR metrics.

As demonstrated in Table 2, the absence of these masks during the validation phase leads to a decrease in model performance across most metrics. This highlights the importance of maintaining these masks during validation to preserve performance levels. However, this is not trivial at inference time because the masks are not available as they are during training. One way to obtain these masks at inference is to incorporate a segmentation model, such as the one proposed by [20]. However, this approach has two main drawbacks. First, the overall system becomes slower since the detection model has to wait for the output of the segmentation model. Second, any error in the segmentation model's output can propagate and degrade the overall performance of the detector. For instance, if the segmentation model incorrectly classifies a part of the pipe as background, the detection model will never see this part, potentially missing any damage present. The proposed method addresses these issues by exploiting the foreground knowledge only during the training phase.

3.1 Data preprocessing

We use a dataset $\mathcal{D} = \{(\mathbf{X_i}, \mathbf{a_i}, \mathbf{M_i})\}_{i=1}^{N}$, $\mathbf{M_i}$ are the ground truth masks of the pipeline, and N is the total number of images. Using the ground truth masks $\mathbf{M_i}$, we create a new dataset $\mathcal{D}_M = \{(\mathbf{X_{M_i}}, \mathbf{a_i})\}_{i=1}^{N}$. To create $\mathcal{D}_M$, we use the ground truth masks $\mathbf{M}$ to separate the background and the foreground and apply different transformations to each. The mask $\mathbf{M}$ contains values of 1 where the pipeline is present and 0 elsewhere. Using this mask, we extract the foreground and background of the image and apply distinct transformations to each. Afterward, we combine the transformed foreground and background to reconstruct the new image, which is then added to the preprocessed dataset $\mathcal{D}_M$. The ground truth labels $\mathbf{a_i}$ remain unchanged.

Formally, in our framework, given an input image $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ where C, H, and W denote the number of channels, height, and width of $\mathbf{X}$, respectively,

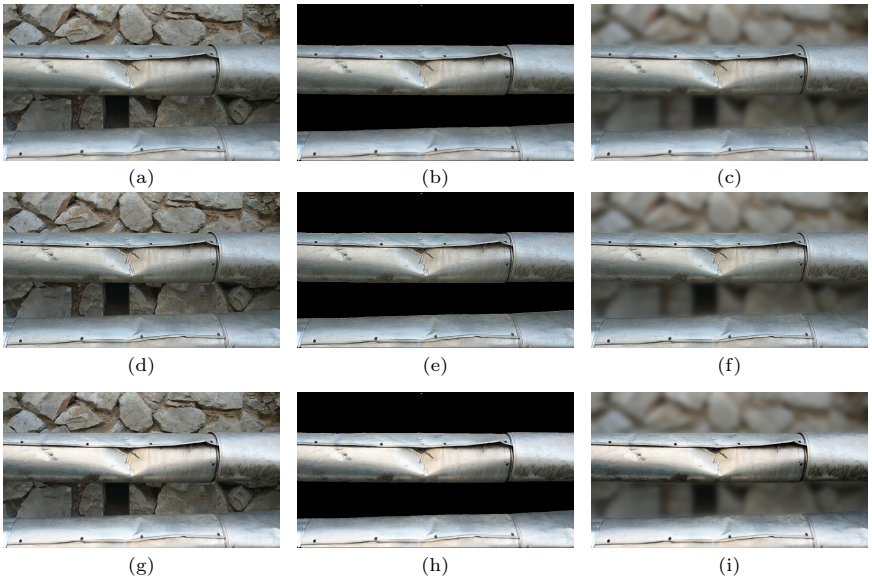

Fig. 4: Image preprocessing: (a) Original image, (b) Remove Background (RB), (c) Blurred Background (BB), (d) Sharpen Foreground (SF), (e) Remove Background & Sharpen Foreground (RB & SF), (f) Blurred Background & Sharpen Foreground (BB & SF), (g) Equalize Foreground (EF), (h) Remove Background & Equalize Foreground (RB & EF), (i) Blurred Background & Equalize Foreground (BB & EF).

the preprocessed image $\mathbf{X_M}$ is obtained by:

$$\mathbf{X}_{\text{foreground}} = \mathcal{T}_{\text{fgrnd}}(\mathbf{X}) \odot \mathbf{M} \tag{2}$$

$$\mathbf{X}_{\text{background}} = \mathcal{T}_{\text{bgrnd}}(\mathbf{X}) \odot \neg\mathbf{M} \tag{3}$$

$$\mathbf{X}_M = \mathbf{X}_{\text{background}} + \mathbf{X}_{\text{foreground}} \tag{4}$$

where $\odot$ denotes element-wise multiplication, $\mathcal{T}_{\text{fgrnd}}$ is the foreground transformation function, and $\mathcal{T}_{\text{bgrnd}}$ is the background transformation function. The $\neg\mathbf{M}$ is the binary complement of $\mathbf{M}$, having a value of 1 for the background and 0 for the pipeline. We experimented with various $\mathcal{T}_{\text{fgrnd}}$ and $\mathcal{T}_{\text{bgrnd}}$ functions to identify the optimal combination, as discussed in Subsection 4.1.

3.2 Feature KD with different inputs

The latest SOTA DNN detection models typically consist of three primary components: the backbone $B(\cdot)$, the neck $N(\cdot)$, and the detection head $H(\cdot)$. The overall flow of the detection process can be represented as $\hat{\mathbf{Y}} = F(\mathbf{X}) = H \circ N \circ B(\mathbf{X})$, where $F(\mathbf{X})$ denotes the model's output.

The backbone network $B(\cdot)$ extracts feature maps from an input image $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$. The final output is a feature map $\mathbf{F}^L$ from the last layer L:

$$\mathbf{F}^L = B(\mathbf{X}) = f^L \circ f^{L-1} \circ \cdots \circ f^1(\mathbf{X}) \tag{5}$$

where $\mathbf{F}^L \in \mathbb{R}^{C_L\times H_L\times W_L}$ and f^l represents the transformation function at layer l.

To capture multi-scale information, we extract feature maps from a sequence of intermediate layers. Let $\mathcal{L} = \{L, L-1, \ldots, L-K+1\}$ be the ordered set of layers, where K is the number of layers considered. Consequently, the backbone output is an ordered sequence of feature maps $\left(\mathbf{F}^l\right)_{l=L-K+1}^{L} = B(\mathbf{X})$. This approach allows the model to utilize multi-scale information effectively, enhancing its capability to detect objects of varying scales and improve overall performance.

The teacher model is trained using $\mathbf{X}_M$, where the student model is trained using both $\mathbf{X}$ and $\mathbf{X}_M$. During student training, the feature maps are given by:

$$\{\mathbf{F}_S^l\}_{l=L-K}^{L} = B_S(\mathbf{X}) \quad \text{and} \quad \{\mathbf{F}_T^l\}_{l=L-K}^{L} = B_T(\mathbf{X}_M) \tag{6}$$

where B_S and B_T denote the backbones of the student and teacher models, respectively.

To optimize the student model, we measure the cosine distance between the feature maps of the student $\mathbf{F}_S^i$ and the teacher $\mathbf{F}_T^i$:

$$d(\mathbf{F}_S^i, \mathbf{F}_T^i) = 1 - \frac{\text{flatten}(\mathbf{F}_S^i) \cdot \text{flatten}(\mathbf{F}_T^i)^T}{\|\text{flatten}(\mathbf{F}_S^i)\|_2 \|\text{flatten}(\mathbf{F}_T^i)\|_2} \tag{7}$$

where $\text{flatten} : \mathbb{R}^{C_i\times H_i\times W_i} \to \mathbb{R}^{C_iH_iW_i}$ is the vectorization function, and $\|\cdot\|_2$ is the l_2 norm.

The distillation loss $\mathcal{L}_{\text{KD}}$ and the total loss $\mathcal{L}_{\text{total}}$ are defined as follows:

$$\mathcal{L}_{\text{KD}} = \sum_{i=1}^{K} d(\mathbf{F}_S^i, \mathbf{F}_T^i), \tag{8}$$

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \alpha\mathcal{L}_{\text{KD}} \tag{9}$$

where $\mathcal{L}$ is the original task-specific loss (e.g., classification or detection loss) and α is a weight hyperparameter balancing the two loss components.

4 Experiments

In our experimental evaluation, we use a subset of the PDI dataset [17], which consists of 939 images. We use 752 images for the training phase and 187 images for the validation phase. We selected a subset because the PDI dataset [17] does not include pipe segmentation masks, so we had to annotate them manually. For experimental purposes, we annotated a subset of the dataset. The images in the PDI dataset [17] have varying resolutions, ranging from 1920x1080 to 9504x6336, due to the different camera sensors used. This variation in resolution presents

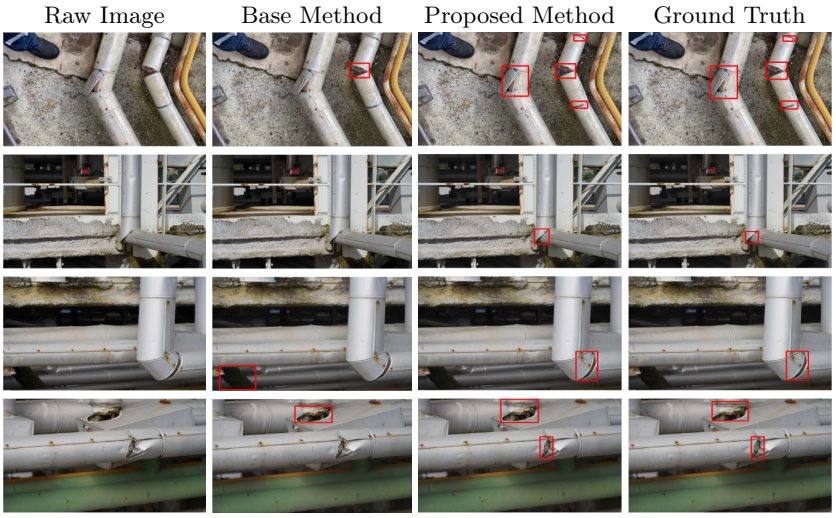

Fig. 5: Comparison of YOLOv6s [14] results. Each column presents the results for the raw image, base training method, proposed method, and ground truth.

additional challenges. The dataset comprises images from generic, anonymized European refinery imagery, adding to its diversity and complexity.

The experiments were conducted on a machine running Ubuntu 16.04.4 LTS, equipped with an NVIDIA GeForce RTX 2080ti GPU and an Intel(R) Core(TM) i7-6900K CPU @ 3.20GHz. The software environment included Python 3.8.18 and CUDA 11.3. The training batch size was set to 16, and the image size was 640 x 640 pixels. All models were trained for 100 epochs and were pre-trained on the COCO dataset. We used the default fine-tuning configuration for each model.

In this paper, we use several metrics to evaluate the performance of our framework. Specifically, we employ COCO metrics for detection, which include mAP at different intersection over union (IoU) thresholds. Additionally, we measure mean average recall (mAR) under different scenarios, such as the maximum number of objects in an image. The primary metric we focus on is mAP, which is the standard benchmark metric for object detection tasks. However, we also report other metrics to comprehensively assess the benefits and performance improvements of our method.

Table 3 presents a comparison between the base training method, a simple feature distillation method using Cosine Similarity (CS) loss, and our proposed method across several SOTA object detection models. We utilized CS in the simple feature distillation to highlight the significance of using different inputs for the teacher and the student in our framework. The results show that our proposed method consistently outperforms both the baseline and the simple feature distillation method across almost all models. This demonstrates the architecture-agnostic capability of our framework. Notably, even transformer-based architec-

tures exhibit improvements with our method compared to the baseline training. However, these architectures show a significant performance decrease when simple feature distillation is applied. Overall, this table illustrates that our framework effectively enhances the performance of nearly every detector.

	mAP		**Comparison**		
Model	Base	Feature-KD	Proposed	vs Base	vs Feature-KD
YOLOv5n	0.219	0.207	**0.223**	+0.004 (+1.8%)	+0.016 (+7.7%)
YOLOv5s	0.221	**0.245**	0.241	+0.020 (+9%)	-0.004 (-1.6%)
YOLOv5m	0.234	0.215	**0.244**	+0.010 (+4.2%)	+0.029 (+13.4%)
YOLOv5l	0.206	0.234	**0.235**	+0.029 (+14%)	+0.001 (+0.4%)
YOLOv8n	0.200	**0.218**	0.217	+0.017 (+8.5%)	-0.001 (-0.4%)
YOLOv8s	**0.224**	0.229	0.222	-0.002 (-0.8%)	-0.007 (-3%)
YOLOv8m	0.213	0.236	**0.250**	+0.037 (+17.3%)	+0.014 (+5.9%)
YOLOv8l	0.211	0.209	**0.239**	+0.028 (+13.2%)	+0.030 (+14.3%)
YOLOv6n	0.249	0.242	**0.261**	+0.012 (+4.8%)	+0.019 (+7.8%)
YOLOv6s	0.250	0.240	**0.275**	+0.025 (+10%)	+0.035 (+14.5%)
YOLOv6m	0.242	**0.275**	0.256	+0.014 (+5.7%)	-0.019 (-6.9%)
YOLOv6l	0.276	0.277	**0.280**	+0.004 (+1.4%)	+0.003 (+1%)
RT-DETR	0.242	0.175	**0.253**	+0.011 (+4.5%)	+0.078 (+44.5%)

Table 3: Comparison of the base training method and feature distillation with the proposed method, including improvements across various model architectures.

In Figure 5, we can see the inference results of YOLOv6s [14] with the base training method and our proposed training method. The significant improvement in recall and precision achieved by our training framework is evident. Our proposed method closely matches the ground truth, whereas the baseline model struggles with detection and often produces false positives showing the importance of our training framework and how it can help the detection models in those types of environments.

In Table 4 we compare our best results which are with the RB & SF for teacher input, determined through an ablation study discussed later in this paper in Subsection 4.1. Unlike our method, these traditional approaches do not use different inputs for the teacher and student models. In these methods, the teacher is pretrained on the task with the original images, and the distillation process follows the classical feature KD method [25], where both the student and the teacher have the same input. Additionally, it demonstrates a significant performance increase compared to base training. Specifically, our method improves the mAP of the SOTA baseline detector YOLOv6s [14] by 0.031 (+12%), mAP_{50} by 0.045 (+9%), mAR_1 by 0.02 (+8%), mAR_{10} by 0.038 (+10%), and mAR_{100} by 0.029 (+7%). These results show that our proposed training framework with KD method significantly outperforms the base training approach.

Method	mAP	mAP_{50}	mAR_1	mAR_{10}	mAR_{100}
Base training	0.25	0.505	0.232	0.349	0.419
Feature KD-CS [27]	0.24	0.505	0.215	0.348	0.403
Feature KD-MMD [39]	0.246	0.445	0.223	0.376	0.432
Feature KD-CWD [29]	0.235	0.432	0.225	0.325	0.39
Feature KD-MSE [25, 38]	0.255	0.504	0.227	0.365	0.426
KD-KL [7, 14, 46]	0.225	0.449	0.227	0.343	0.377
Proposed Method	**0.281**	**0.55**	**0.252**	**0.387**	**0.438**

Table 4: Performance comparison of various feature knowledge distillation (KD) methods using different loss functions on YOLOv6s [14]. Metrics include mAP and mAR, highlighting the effectiveness of the proposed method.

4.1 Ablation Study

We conducted an ablation study to evaluate the impact of various preprocessing techniques on the teacher model's input and to assess different distance metrics for distillation loss. We aimed to identify the most effective preprocessing methods for both the background and foreground, as well as to determine the optimal distance metric for our KD loss. The distance metrics used were sourced from existing literature on feature distillation. Our objective was to identify the best loss function and the corresponding α hyperparameter, which balances the KD loss within the overall loss function of the detection model. This study was crucial for optimizing the model's performance in the specific task of damage detection in insulated pipelines.

Method	mAP	mAP_{50}	mAR_1	mAR_{10}	mAR_{100}
MSE, $\alpha = 0.1$	0.261	0.493	0.244	0.364	0.431
MSE, $\alpha = 0.5$	0.271	0.501	0.246	0.366	**0.438**
MSE, $\alpha = 1$	0.265	0.516	0.248	0.361	0.419
MMD, $\alpha = 0.1$	0.261	0.502	**0.252**	0.361	0.433
MMD, $\alpha = 0.5$	0.261	0.497	0.243	0.371	0.425
MMD, $\alpha = 1$	0.25	0.467	0.235	0.353	0.42
CWD, $\alpha = 0.1$	0.263	0.494	0.244	0.364	0.43
CWD, $\alpha = 0.5$	0.259	0.505	0.229	0.363	0.427
CWD, $\alpha = 1$	0.252	0.482	0.238	0.344	0.423
CS, $\alpha = 0.1$	0.256	0.488	0.242	0.355	0.422
CS, $\alpha = 0.5$	**0.275**	**0.531**	0.248	**0.369**	0.419
CS, $\alpha = 1$	0.266	0.509	0.244	0.366	0.423

Table 5: Comparison of proposed method with different losses for YOLOv6s [14] on mAP and mAR metrics.

In this study, a separate model was trained for each preprocessing method and used as the teacher. The results indicated that the best performance was achieved when the teacher model was trained with a combination of RB and SF preprocessing. This combination enabled the model to learn more effectively, resulting in improved precision and recall. The only metric where RB and SF were not the best was mAR_{100}, where it was surpassed by the BB and EF combination. However, in all other metrics, the RB and SF combination performed significantly better. Consequently, we selected this preprocessing method for our best-performing model.

Furthermore, we identified the most effective loss function for our approach by evaluating several well-known loss functions previously employed in other distillation works. We also fine-tuned the hyperparameter α to determine the optimal combination. Table 5 provides a comprehensive overview of these experiments, including all relevant metrics. Overall, the best combination was found to be the CS loss with $\alpha = 0.5$. This is further illustrated in Figure 6. For the mAR_1 metric, the Maximum Mean Discrepancy (MMD) loss function at $\alpha = 0.1$ yielded the best performance, while the Mean Squared Error (MSE) loss function at $\alpha = 0.5$ was most effective for the mAR_{100} metric. Our best model, trained with a teacher using the RB and SF preprocessing and employing the CS metric as the distillation loss with the hyperparameter $\alpha = 0.5$, surpasses nearly all detection metrics. This combination demonstrates superior overall performance in both precision and recall.

Method	**mAP**	**mAP_{50}**	**mAR_1**	**mAR_{10}**	**mAR_{100}**
RB	0.275	0.531	0.248	0.369	0.419
BB	0.26	0.487	0.244	0.368	0.435
SF	0.251	0.486	0.226	0.364	0.42
EF	0.244	0.489	0.229	0.362	0.417
BB & SF	0.269	0.503	0.25	0.37	0.43
BB & EF	0.245	0.493	0.227	0.364	**0.443**
RB & SF	**0.281**	**0.55**	**0.252**	**0.387**	0.438
RB & EF	0.257	0.509	0.24	0.369	0.424

Table 6: Comparison of different foreground-background preprocessing for teacher inputs for YOLOv6s [14] on mAP and mAR metrics.

5 Conclusion

In this paper, we addressed the critical need for advanced vision-based inspection technologies in industrial environments, with a specific focus on detecting damage in insulated pipes within refineries. We proposed a novel, architecture-agnostic DNN training methodology based on KD to enhance the performance of SOTA object detection models in these challenging environments where the

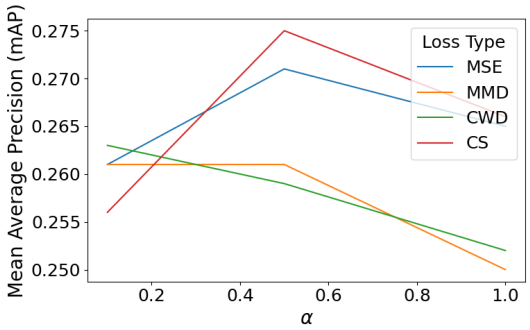

Fig. 6: Comparison of mAP metric for different losses and different α.

base algorithms fail. Our approach leverages feature KD, incorporating foreground knowledge during the training phase, which significantly boosts model performance and effectively addresses the clutter noise in industrial environments.

Extensive experiments validated our framework, demonstrating substantial improvements in key metrics, including up to a 12% increase in mAP. Our approach demonstrates both versatility and efficacy across multiple DNN architectures, highlighting its broad applicability and independence from specific network architectures. Additionally, since our approach is only applied during the training phase and does not impact model deployment, it is straightforward to implement in industrial settings. By experimenting with various loss functions and image preprocessing techniques, we further optimized model performance and identified the optimal parameters for the damage detection task in insulated pipes.

We believe that this work can inspire further research to address real-world vision-based challenges, such as background clutter. Future work will explore applying this method to other tasks, such as anomaly detection and binary semantic segmentation, with the potential for performance enhancements in these areas for industrial applications. This research represents a significant step toward bridging the gap between academic advancements in DNN models and their practical deployment in industrial environments, thereby ensuring more reliable and efficient inspections.

Acknowledgements

This work has received funding from the European Union's Horizon re- search and innovation programme under grant agreement number 101070604 (SIMAR).

References

1. Apostolopoulos, I.D., Tzani, M.A.: Industrial object and defect recognition utilizing multilevel feature extraction from industrial scenes with deep learning approach. Journal of Ambient Intelligence and Humanized Computing **14**(8), 10263–10276 (2023)
2. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
4. Feng, D., Feng, M.Q.: Computer vision for shm of civil infrastructure: From dynamic response measurement to damage detection–a review. Engineering Structures **156**, 105–117 (2018)
5. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
6. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
7. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
8. Jeong, D.: Road damage detection using yolo with smartphone images. In: 2020 IEEE international conference on big data (big data). pp. 5559–5562. IEEE (2020)
9. Ji, M., Heo, B., Park, S.: Show, attend and distill: Knowledge distillation via attention-based feature matching. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 7945–7952 (2021)
10. Jocher, G.: Ultralytics yolov5 (2020). `https://doi.org/10.5281/zenodo.3908559`, `https://github.com/ultralytics/yolov5`
11. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics yolov8 (2023), `https://github.com/ultralytics/ultralytics`
12. Lemos, R., Cabral, R., Ribeiro, D., Santos, R., Alves, V., Dias, A.: Automatic detection of corrosion in large-scale industrial buildings based on artificial intelligence and unmanned aerial vehicles. Applied Sciences **13**(3), 1386 (2023)
13. Li, C., Yan, H., Qian, X., Zhu, S., Zhu, P., Liao, C., Tian, H., Li, X., Wang, X., Li, X.: A domain adaptation yolov5 model for industrial defect inspection. Measurement **213**, 112725 (2023)
14. Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., et al.: Yolov6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976 (2022)
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016)
16. Lv, W., Xu, S., Zhao, Y., Wang, G., Wei, J., Cui, C., Du, Y., Dang, Q., Liu, Y.: Detrs beat yolos on real-time object detection. arXiv preprint arXiv:2304.08069 (2023)
17. Mentesidis, P., Papaioannidis, C., Pitas, I.: ADVANCING INDUSTRIAL INSPECTION: A DATASET FOR AUTOMATED DAMAGE DETECTION IN INSULATED PIPES (2024). `https://doi.org/10.5281/zenodo.12622101`, `https://doi.org/10.5281/zenodo.12622101`

18. Passalis, N., Tefas, A.: Learning deep representations with probabilistic knowledge transfer. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 268–284 (2018)
19. Pham, C., Nguyen, V.A., Le, T., Phung, D., Carneiro, G., Do, T.T.: Frequency attention for knowledge distillation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2277–2286 (2024)
20. Psarras, D., Papaioannidis, C., Mygdalis, V., Pitas, I.: A unified dnn-based system for industrial pipeline segmentation. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7785–7789. IEEE (2024)
21. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
22. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
23. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
24. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
25. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
26. Saberironaghi, A., Ren, J., El-Gindy, M.: Defect detection methods for industrial products using deep learning techniques: A review. Algorithms **16**(2), 95 (2023)
27. Shin, S., Lee, J., Lee, J., Yu, Y., Lee, K.: Teaching where to look: Attention similarity knowledge distillation for low resolution face recognition. In: European Conference on Computer Vision. pp. 631–647. Springer (2022)
28. Shrivastava, A., Qi, Y., Ordonez, V.: Estimating and maximizing mutual information for knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 48–57 (2023)
29. Shu, C., Liu, Y., Gao, J., Yan, Z., Shen, C.: Channel-wise knowledge distillation for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5311–5320 (2021)
30. Spahić, R., Poolla, K., Hepsø, V., Lundteigen, M.A.: Image-based and risk-informed detection of subsea pipeline damage. Discover Artificial Intelligence **3**(1), 23 (2023)
31. Usamentiaga, R., Lema, D.G., Pedrayes, O.D., Garcia, D.F.: Automated surface defect detection in metals: A comparative review of object detection and semantic segmentation using deep learning. IEEE Transactions on Industry Applications **58**(3), 4203–4213 (2022)
32. Vundekode, N.R., Kalapatapu, P., Pasupuleti, V.D.K.: A study on vision based method for damage detection in structures. In: European Workshop on Structural Health Monitoring. pp. 96–105. Springer (2020)
33. Wan, F., Sun, C., He, H., Lei, G., Xu, L., Xiao, T.: Yolo-lrdd: A lightweight method for road damage detection based on improved yolov5s. EURASIP Journal on Advances in Signal Processing **2022**(1), 98 (2022)
34. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G.: Yolov10: Real-time end-to-end object detection. arXiv preprint arXiv:2405.14458 (2024)
35. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696 (2022)

36. Wang, C.Y., Liao, H.Y.M.: YOLOv9: Learning what you want to learn using programmable gradient information (2024)
37. Wang, J., Dai, H., Chen, T., Liu, H., Zhang, X., Zhong, Q., Lu, R.: Toward surface defect detection in electronics manufacturing by an accurate and lightweight yolo-style object detector. Scientific Reports **13**(1), 7062 (2023)
38. Wang, L., Li, X., Liao, Y., Jiang, Z., Wu, J., Wang, F., Qian, C., Liu, S.: Head: Hetero-assists distillation for heterogeneous object detectors. In: European Conference on Computer Vision. pp. 314–331. Springer (2022)
39. Wang, M., Liu, R., Hajime, N., Narishige, A., Uchida, H., Matsunami, T.: Improved knowledge distillation for training fast low resolution face recognition model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
40. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4933–4942 (2019)
41. Yang, C., Ochal, M., Storkey, A., Crowley, E.J.: Prediction-guided distillation for dense object detection. In: European Conference on Computer Vision. pp. 123–138. Springer (2022)
42. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4133–4141 (2017)
43. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
44. Zhang, L., Ma, K.: Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In: International Conference on Learning Representations (2020)
45. Zhixing, D., Zhang, R., Chang, M., Liu, S., Chen, T., Chen, Y., et al.: Distilling object detectors with feature richness. Advances in Neural Information Processing Systems **34**, 5213–5224 (2021)
46. Zhou, Z., Zhuge, C., Guan, X., Liu, W.: Channel distillation: Channel-wise attention for knowledge distillation. arXiv preprint arXiv:2006.01683 (2020)