Deep Learning Strategies for Visual Pattern Analysis in Metallic Surfaces

Sandro Bessa de Lima
lima.sandro@ime.eb.br
Military Institute of Engineering (IME)
Rio de Janeiro, RJ, Brazil

Julio Cesar Duarte
duarte@ime.eb.br
Military Institute of Engineering (IME)
Rio de Janeiro, RJ, Brazil

ABSTRACT

Advances in multimedia systems and computer vision have enabled each time more sophisticated methods that are able to analyze complex visual patterns in industrial contexts. The inspection of failures in metallic alloys is important for sectors such as aerospace and defense, yet it faces limitations due to reliance on manual methods and data scarcity. This research presents a systematic approach for selecting convolutional neural network architectures by integrating data augmentation, image segmentation, and self-supervised learning. In particular, it uses the DINO technique to generate interpretable attention maps, enhancing model transparency and performance in defect detection tasks. The method was validated using three architectures, ResNetInceptionV2, DenseNet, and Xception, with DenseNet combined with DINO achieving the best results. Key contributions include the release of a hybrid dataset (real and synthetic), effective strategies for handling class imbalance, and a comparative study of architectures, offering insights for future industrial applications and deep learning research applied to pattern detection on metallic surfaces.

KEYWORDS

Computer vision; Defect detection; Metallic surfaces; Convolutional neural networks; Transfer learning; DINO

1 INTRODUCTION

The production process of metallic alloys is part of various industries, such as aerospace and defense. According to Zuchniak et al. [22], even a small flaw on the metallic surface can compromise a project or cause catastrophes, such as aviation accidents. Moreover, the inspection of these flaws is still an expensive and manual process, often prone to errors and fraud. In this context, technologies based on Artificial Intelligence (AI) have shown promise in overcoming such challenges, introducing greater precision and efficiency in inspection processes.

The inspection of flaws in metallic alloys involves steps that include the identification of defects such as scratches, holes, dents, corrosion, and shearing [9]. However, issues along the production chain, such as reliability limitations, productivity constraints, maintainability concerns, as well as risks to inspectors' lives and fraud in the manual reporting process, affect the effectiveness of inspections [20].

In: Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia'2025). Rio de Janeiro, Brazil. Rio de Janeiro: Brazilian Computer Society, 2025.

ISSN 2966-2753

These limitations can be exacerbated by factors such as a lack of time for proper analyses, adverse environmental conditions, and resource scarcity. In this sense, Machine Learning (ML) and Deep Learning (DL) offer innovative solutions that can automate and optimize these steps, reducing errors and increasing the reliability of results. The inspection of metallic alloys, although important to ensure the safety and reliability of structural components, faces technical and operational challenges that limit its efficiency.

The application of AI, especially through Deep Neural Networks (DNN), allows the exploration of new approaches to overcome these limitations, improving both the precision and automation of the process. Furthermore, the interdisciplinary nature of the research, by combining areas such as computer vision (CV), ML, and synthetic data generation, reinforces its contribution to technological and scientific advancements in the field.

The objective of this work is to propose and validate a methodology based on Advanced Modeling and Processing for fault inspection in metallic alloys, emphasizing the selection of efficient architectures and the application of advanced techniques such as Attention Mechanisms, Transfer Learning, and image segmentation. Experiments validated the proposed method using models based on ResNetInceptionV2, DenseNet, and Xception, with DenseNet achieving superior results when combined with the DINO technique. The results demonstrated improvements in metrics such as accuracy, F1-macro, Loss, and AUC, highlighting the effectiveness of the adopted approach. Furthermore, the application of attention maps to the alpha channel of images enhanced model performance, even in scenarios with limited datasets, demonstrating the relevance of the implemented strategies.

The remainder of this paper is organized as follows. Section 2 provides an overview of deep learning concepts and techniques relevant to the proposed approach. Section 3 reviews related work, highlighting the main contributions and limitations of existing methods. Section 4 describes the proposed method in detail, including both design choices and implementation aspects. Section 5 reports the experimental setup, evaluation metrics, while also discusses the experimental results. Finally, Section 6 presents the conclusions and outlines potential directions for future research.

2 DEEP LEARNING

DL is a field of ML that is inspired by how living beings represent and interpret visual information [14]. This approach is based on DNN architectures, which use multiple layers to learn hierarchical representations and abstract high-level features from raw data. In DL, abstraction is achieved through these multiple layers, where each layer captures increasingly simplified representations of the information or features for the model.

This process is particularly useful in CV tasks, such as image classification, semantic segmentation, and object detection. One of the most commonly used models for CV is the Convolutional Neural Network (CNN). This kind of network uses convolutional and subsampling layers designed to efficiently learn spatial and hierarchical features of images.

After using these layers, many architectures rely on a fully connected Multilayer Perceptron (MLP) responsible for the final classification. Each connection between neurons in different layers is associated with a weight, which is adjusted during the training of the neural network. The training process typically involves the use of optimization algorithms, such as gradient descent, to adjust these weights to minimize the difference between the predicted outputs and the actual values. MLPs are known for their ability to learn complex relationships in data and are widely used in tasks such as classification, regression, and pattern recognition [10].

2.1 Convolutional Neural Netowrks

CNNs are widely used for CV tasks due to their ability to learn spatial features in images. Inspired by the structure of the human brain, these networks use convolutional layers to detect local patterns and *pooling* (subsampling) layers to reduce data dimensionality while retaining relevant information. This makes them particularly effective in large-scale problems, such as image classification and semantic segmentation [1].

Although CNNs are powerful in learning features directly from data, the success of these models in practical applications often depends on the amount and quality of the data available for training [1]. In CV problems, large and diverse datasets, such as *ImageNet*, play a decisive role in enabling CNNs to learn comprehensive and hierarchical representations. However, it is not always feasible to collect and label large volumes of data for all specific applications.

In this context, Transfer Learning (TL) techniques emerge as an efficient solution. For instance, pre-trained models on large datasets, such as *ImageNet*, can be reused as a starting point for new tasks. These models already possess learned feature hierarchies, such as edges, textures, and shapes, which can be adapted to solve specific problems with less data available.

For example, in a CV application aimed at industrial object detection, a pre-trained CNN model can be fine-tuned to recognize unique patterns associated with the industrial environment, significantly reducing training time and data requirements. As such, transfer learning not only enhances the effectiveness of CNNs but also broadens their applicability in resource-constrained scenarios, establishing itself as an indispensable technique in the field of ML.

2.2 Distillation with No Labels

DINO, which stands for *Distillation with No Labels*, is a self-supervised learning method for CV developed by *Facebook AI Research*. Its goal is to extract high-quality semantic representations without using manual labels. The central premise is a *teacher–student* architecture where two networks receive distinct views of the same image; the *teacher* network is updated via an exponential moving average of the *student* network's weights, inspired by the BYOL method [11].

One of the most remarkable findings of DINO is that, even without explicit supervision of classes or masks, the learned representations exhibit attention maps that outline objects in the scene. Touvron et al. [18] demonstrated that in Vision Transformers (ViT) pre-trained with DINO, intermediate layers already generate semantic masks that can be directly used in segmentation tasks, without the need for additional refinement.

In the context of DINO, especially when applied to ViTs, attention heads extract rich semantic representations. Each input image is divided into small patches, treated as input tokens for the transformer architecture. The multiple attention heads operate in parallel and learn to identify spatial and semantic relationships among the patches. Surprisingly, even without explicit supervision, some attention heads spontaneously learn to focus on relevant regions of the image, such as the outline of an object or its central region, while others may capture complementary structures, such as the background or edges.

Caron et al. [7] demonstrated that, following DINO pre-training, certain attention heads produce attention maps that closely resemble segmentation masks. This behavior happens since the model is trained to generate consistent embeddings across different views, such as crops and augmentations of the same image, which encourages the network to identify stable and invariant semantic patterns.

For instance, Figure 1 illustrates the behavior of DINO's third head, highlighting the generation of interpretable attention maps.

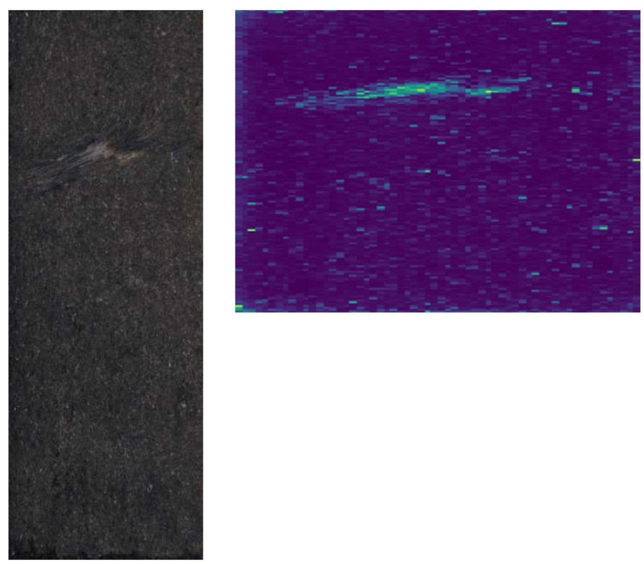


Figure 1: Original Image (Left) and DINO's Attention Map 3rd Head (Right)

3 RELATED WORK

This section presents a list of the articles reviewed, providing an overview of the types of research, neural network architectures, dataset types, DL strategies, evaluation metrics, and IoT-related approaches in the most recent and relevant studies conducted in the field of defect detection on metallic surfaces.

Keeping this in mind, a Systematic Literature Review (SLR) was conducted to identify and synthesize research on the topic, following a structured protocol with defined research questions, inclusion and exclusion criteria, and a systematic search strategy. Studies were evaluated for relevance and methodological rigor to ensure a comprehensive understanding of the state of the art. The studies most relevant to this research are introduced in the subsequent discussion.

When analyzing technological applications aimed at defect detection on metallic surfaces, various scenarios were identified within the collection of articles. Seventy-five percent of the articles focus on corrosion. Few studies address *data segmentation* (30%), while the vast majority of research utilizes private datasets (75%). There is potential for further exploration of TL (15%), DA (25%), and IoT (25%). The most commonly used performance metrics are accuracy (35%), precision (50%), recall (25%), and F1-macro (25%). Figure 2 presents the main scenarios identified in the research, providing an overview of the topic.

This list encompasses approaches that include the application of various CNN models, data segmentation techniques, AD methods (*data augmentation*), and evaluation metrics specific to the industry.

The following sections provide descriptions of the neural network architectures, the datasets used, and summaries of each of the enumerated studies.

Atha and Jahanshahi [3] tested the ZF-Net, VGG16, *Corrosion7*, *Corrosion5*, and VGG15 CNN architectures for corrosion detection on metallic surfaces using a proprietary dataset of 926 images. VGG16 with fine-tuning achieved the highest performance metrics: recall 98.25%, precision 98.70%, and F1-score 98.47%. Other models showed competitive F1-scores ranging from 96.57% to 96.89%.

Xu et al. [19] employed techniques like *Faster-RCNN*, ResNet50, and VGG16 for corrosion detection on coated metal plates using a proprietary dataset of 538 images captured over time to track corrosion progression. *Faster-RCNN* achieved the best results, with MaP above 94 and an error rate under 10%, demonstrating high precision in identifying corrosion areas. These findings underscore its effectiveness for industrial inspection applications.

Yun et al. [21] applied DCNN technology for defect classification on steel surfaces using a dataset of 657 samples from real production lines, representing various defect types. A new algorithm, the multi-discriminant regularized local descriptor, was introduced to enhance classification capabilities. Model performance was assessed

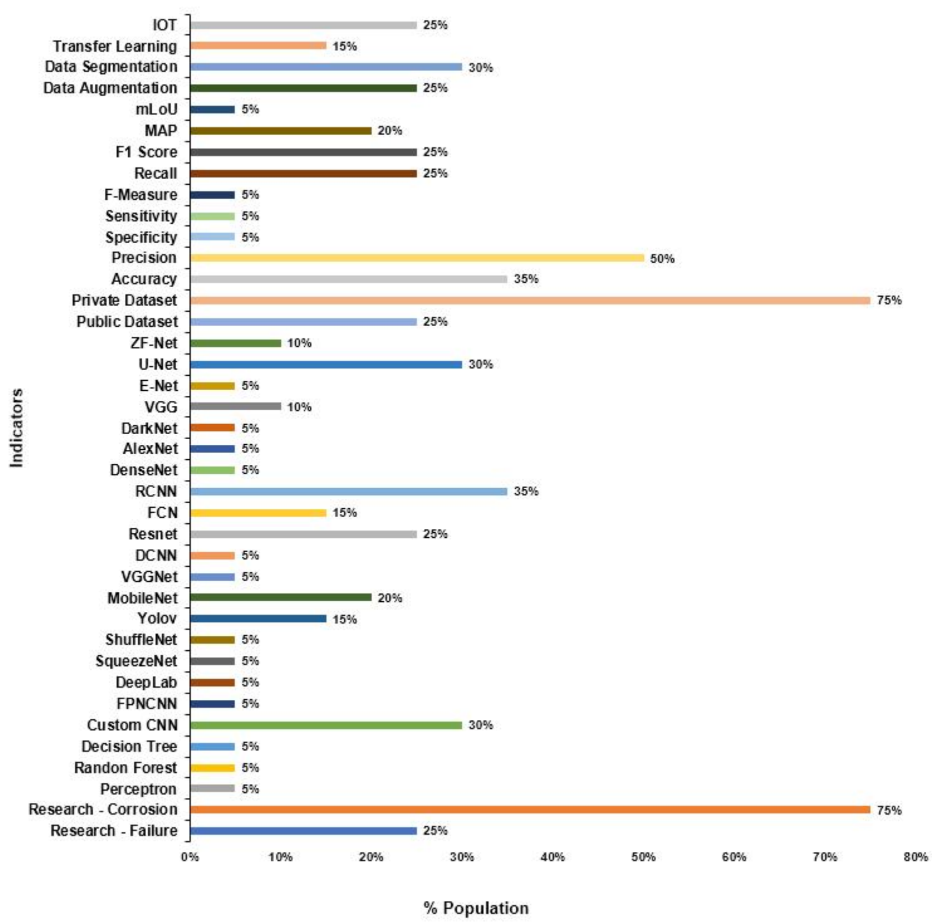


Figure 2: Overview of the most recent and relevant research. Reproduced from [5], originally published by the author in *Revista de Informática Teórica e Aplicada* (RITA), 32(2):64–82, 2025.

with precision, recall, and F1-macro metrics, demonstrating the method's effectiveness in identifying defects on metallic surfaces.

Duy et al. [8] utilized models like *DenseNet*, U-Net, E-Net, FCN, and RCNN for corrosion detection, emphasizing IoT applications such as drone-based image capture. A dataset of 270 training images with augmentation and 180 testing images was employed. Semantic segmentation with PSPNet, analyzing scenes at the pixel level, yielded high precision in corrosion area segmentation. Performance was evaluated using the Intersection over Union (IoU) metric, showcasing the effectiveness of the proposed methods.

Katsamenis et al. [12] applied FCN, U-Net, and Mask R-CNN (trained on the COCO dataset) for corrosion detection on metallic surfaces using a real, expertly annotated dataset from the *H2020 PANOPTIS* project. The approach projected corrosion regions at the pixel level to identify defect semantics. Models were evaluated using accuracy and F1-macro metrics, demonstrating their effectiveness in classifying corroded areas and advancing automated inspection methodologies.

Tian et al. [17] utilized R-CNN, Faster R-CNN, FCN, and U-Net for corrosion detection on metallic surfaces, using a manually labeled dataset of 116 images standardized by ISO 8501-1. The dataset, sourced from various devices, underwent refinement through HSI color space conversion and image cropping to focus on corrosion regions. Labeling was conducted with the *LabelImg* tool for accuracy, and models were assessed using accuracy and F1-macro metrics, demonstrating their effectiveness in detecting and classifying corroded areas.

Bastian et al. [4] examined pipeline image classification with varying corrosion levels using Custom-CNN, ZFNet, and VGGNet. The dataset, expanded to 142,400 images via augmentation, included pipelines classified into four categories: no corrosion, low, medium, and high-level corrosion. Custom-CNN outperformed other models in recall, precision, and F1-score metrics, proving to be the most effective solution. The approach also showcased potential for IoT applications, such as aerial robot-based inspections.

Zuchniak et al. [22] utilized ResNet50, CustomCNN, and advanced knowledge distillation techniques (Single, Ensemble, Averaging Student, Geometric Student, Multi-output Student) for corrosion analysis on metallic surfaces. Images from the DAIS system were labeled by experts into mild and moderate corrosion categories. A method for aggregating knowledge from multiple ML models was developed to enhance performance with evaluation metrics like recall, precision, accuracy, and F1-macro, demonstrating the effectiveness of the approach in classifying corrosion levels and advancing ML techniques for industrial applications.

Ahuja S [2] employed U-Net technology combined with the residual U-Net method and Bidirectional Conv-LSTM to detect pitting corrosion on metallic surfaces. Using a proprietary dataset, the approach demonstrated effectiveness in industrial scenarios, achieving high accuracy, precision, specificity, sensitivity, and F1-macro metrics. The results demonstrated its potential for practical applications in inspection and automation within production processes.

Rahman et al. [15] utilized CNN technology and semantic segmentation for corrosion detection on metallic surfaces, with a dataset collected by inspection engineers to ensure quality. Image labeling was performed using an unsupervised texture-based segmentation method integrated with an RGB-feature classifier, enhancing precision in identifying corroded areas. The model's performance, assessed through accuracy, precision, recall, F1-macro, and AUC-ROC metrics, demonstrated its effectiveness in classifying and segmenting corroded surfaces.

Ramani et al. [16] employed DeepLabV3, ResNet34, U-Net, FPNCNN, and MobileNet-v1 for corrosion detection and segmentation on metallic surfaces using a dataset of 237 images, with 102 labeled corroded regions via the *LabelMe* tool. The approach emphasized DeepLabV3's semantic segmentation capabilities to identify corrosion efficiently. Performance metrics, including accuracy, recall, precision, and mIoU, showcased the models' effectiveness and their contribution to advancing automated inspection techniques for metallic surfaces.

Li et al. [13] used YOLOv4 and an enhanced YOLO-attention model for detecting welding defects in wire arc additive manufacturing (WAAM). A dataset of 760 images with 8,583 objects, captured under industrial and experimental conditions, ensured diversity for training. The model achieved a mean average precision (MAP) of 94.5% and processed at least 42 frames per second, demonstrating high efficiency and potential for real-time automated inspections in industrial environments.

The proposal presented in this research advances existing studies by integrating a systematic approach to the selection of CNN architectures. Additionally, the combination of ML techniques—such as DA, TL, and knowledge distillation—is applied in a coordinated manner to optimize the task of defect detection in metallic alloys. Knowledge distillation, although rarely explored in this context, is employed in this research to enhance model generalization and capture more relevant information through the use of attention maps.

4 METHOD

The proposed solution follows a three-step approach. First, a pre-trained model, originally trained on a large and general-purpose dataset like ImageNet, is used to perform TL, allowing the model to adapt to the specific domain of defect detection in metallic alloys. Next, the dataset is expanded using DA techniques, enhancing variability and robustness in the training set. Finally, the model is fine-tuned for defect recognition through a knowledge distillation method named DINO, which offers attention maps to capture and highlight relevant semantic patterns on metallic surfaces.

To facilitate data extraction, DA techniques such as rotation, brightness adjustment, and image size standardization are applied to mitigate potential distortions in the collected images. It is worth noting that the images used in this study were pre-labeled as either "defect" or "no defect" based on prior manual annotation.

The main steps of the proposed solution, illustrated in Figure 3, are as follows: data collection; preprocessing, which includes generating attention heads and applying them to the alpha channel when applicable; architecture search, testing ResNetInceptionV2, DenseNet, and Xception; training the model through training and validation data; feature extraction; defect pattern classification. Finally, based on the results, we identify the architectures that performed best with the applied method.

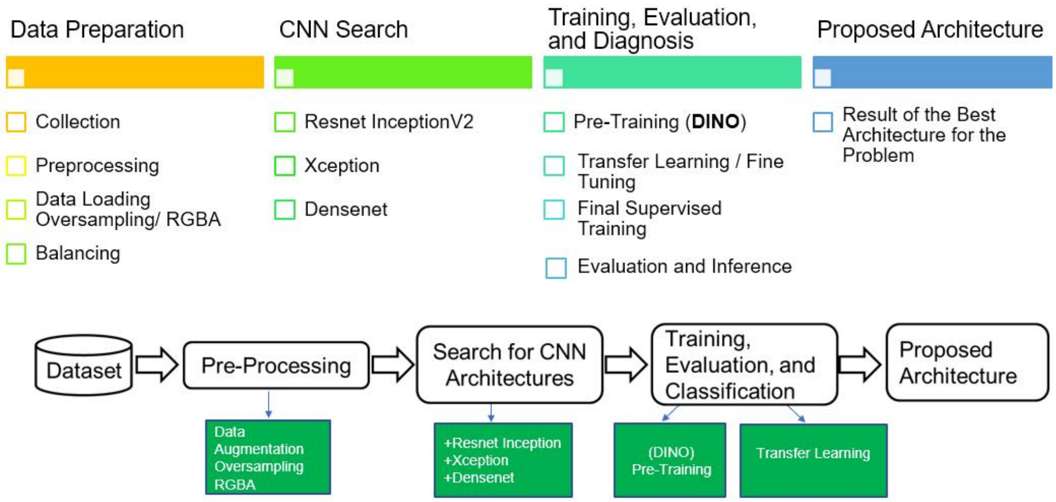


Figure 3: Outline of the proposed solution

Figure 4 presents the flow of the proposed method, which combines self-supervised learning techniques with DINO and visual attention for image classification. The process begins with dataset preprocessing, followed by DA and the selection of an image example. Next, the image is processed by the DINO model, which generates attention maps for the tested attention heads, which are used as inputs to the classifier to perform binary classification, distinguishing images labeled as "defect" from those labeled as "without defect."

The DINO method was selected due to its robustness and efficiency in learning discriminative visual representations without labels. Unlike contrastive methods such as SimCLR, BYOL, or MoCo, DINO employs a teacher–student self-distillation mechanism that enhances feature stability and transferability, making it particularly suitable for tasks involving defect detection on metallic surfaces with limited annotated data.

For example, in this case, the classifier based on ResNetInceptionV2 is responsible for making the final prediction.

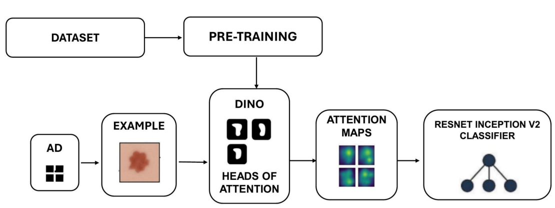


Figure 4: Preprocessing and classification

This model delivers reasonable performance, as shown in Section 5, without requiring extensive hyperparameter tuning or testing multiple pre-trained models. However, due to its simpler design, it struggles to capture more complex patterns and exhibits a higher loss. This highlights the importance of this work, as more advanced models, such as the hybrid ResNet Inception V2, can demonstrate superior performance and warrant further investigation.

CNNs serve as the foundation for this research, given their well-established advantages in CV tasks involving digital image processing. The process begins with the training phase, during which images of defects in metallic alloys are analyzed to learn morphological characteristics and relevant features that distinguish between the classes "defect" and "no defect."

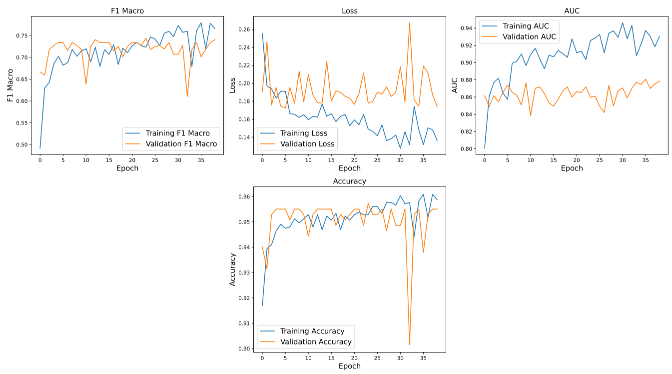
Following training, a specialized model is obtained to classify images of defects on metallic surfaces. This model features a modified final decision layer, replacing the standard CNN layer with a ReLU activation function, to facilitate classification and improve backpropagation efficiency. The final layer receives feature vectors and outputs predictions indicating the presence or absence of defects in the alloy image.

An experimental study was conducted to evaluate and compare the performance of three CNN architectures: Inception ResNet V2, DenseNet, and Xception. This phase aims to select the architecture that generated the best model by evaluating accuracy, loss, F1-macro, and AUC. Evaluations were performed considering both resource-rich environments (cloud computing) and resource-constrained hardware typical of embedded systems.

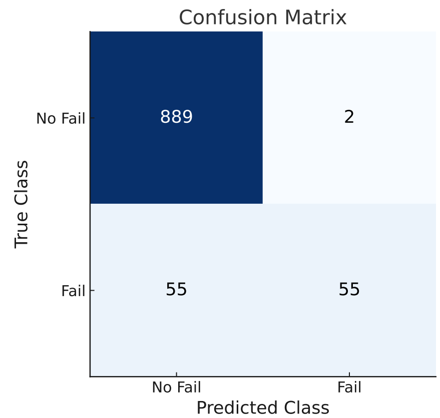
5 EXPERIMENTS

5.1 Dataset

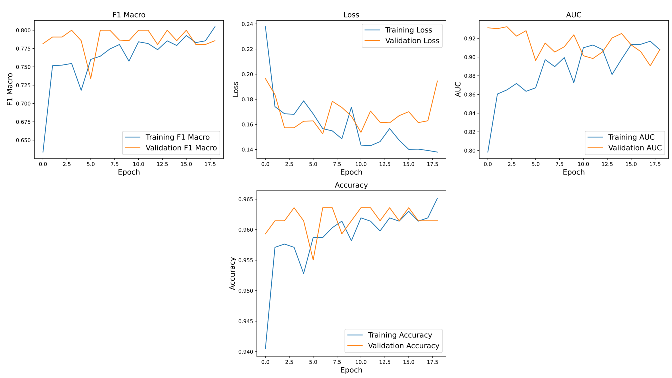
The KolektorSDD2 (*Kolektor Surface-Defect Dataset 2*) is a dataset created for surface defect detection, developed by Božič et al. [6] in the context of supervised learning for defect detection in real industrial scenarios. It is widely used for developing and evaluating computer vision models aimed at defect detection in industrial settings, and comprises over 3,000 images, with 356 containing visible defects and 2,979 without defects. The images have approximate dimensions of 230 × 630 pixels and include various defect types such as *scratches*, *small stains*, and *surface imperfections*.



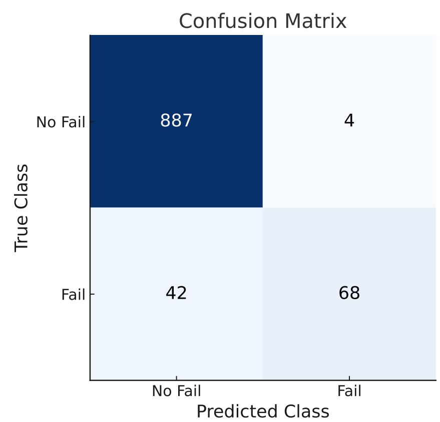
(a) Metrics - No DINO



(b) Confusion Matrix - No DINO



(c) Metrics - DINO - Resnet - Head 3



(d) Confusion Matrix - DINO - Resnet - Head 3

Figure 5: Comparative experiments: without DINO vs. with DINO

The dataset is divided into a training set, containing 246 positive images (with defects) and 2,085 negative images (without defects), and a test set, comprising 110 positive images and 894 negative images.

5.2 Reproducibility of the Experiments

The experiments conducted in this work involved training convolutional neural network models using the DINO technique and data augmentation (DA) strategies applied to industrial context images. To ensure that these experiments can be replicated by other researchers, all relevant details were documented, including hardware and software environment configurations, training and initialization parameters, and evaluation procedures. This establishes a foundation that allows for the reproduction of the results, which are presented in Table 1

5.3 Results

The strategies employed are described along with the best results obtained from the conducted experiments, emphasizing various CNN architectures (ResNetInceptionV2, DenseNet, and Xception) and machine learning techniques such as DA and TL. The analysis begins with the results of experiments conducted without the use of DINO, followed by those incorporating DINO, where attention heads ranging from [0, 1, 2, 3, 4, 5] are considered. Subsequently, experiments utilizing (Oversampling + DINO) techniques are discussed, and finally, the results of experiments employing attention maps generated by the attention heads (DINO) and applied to the alpha channel (RGBA) are presented.

The results of all experiments are concisely summarized in Figure 5. The evaluated strategies are defined as follows:

A. **Results without DINO** - Evaluation of the architecture using the convolutional neural networks ResnetInceptionV2, without the DINO distillation technique.

Table 1: Conditions required to replicate the experiments performed.

Item	Description
Computational environment	Google Colab – Server running *Ubuntu* 22.04.4 LTS (64-bit), Intel(R) Xeon(R) CPU @2.20GHz, NVIDIA L4 GPU (compatible with *CUDA* 12.2)
Main libraries and versions	Python 3.12.11; TensorFlow 2.12.0; NumPy 2.2.0; *Scikit-learn* 0.24; Matplotlib; Seaborn; Glob; imageio; Keras; CV2; Pandas (among other libraries)
Data partitioning strategy	Dataset divided into training (80%) and testing (20%) subsets, using stratified partitioning to preserve the class distribution in each subset
Early stopping criteria	Training was interrupted when the validation metric did not improve after 15 consecutive epochs (patience = 15)
Monitored metric (*val_f1_score*)	F1-Score computed on the validation set at each epoch, used to monitor model performance and trigger early stopping
Random seed	*random_state*=42, ensuring reproducible data splits
Stratified split	*stratify*=y, preserving class balance between training and testing sets
Model compiler	Optimizer=Adam; loss='*binary_cross_entropy*'; *metrics*=['*binary_accuracy*', 'AUC', *f1_metric*]
Time and Memory	Average training time per epoch was approximately six minutes, and GPU memory usage remained below 20 GB

Table 2: Comparison of Performance Metrics between Models

Item	DINO	Strategy	Head	TP	FN	FP	TN	F1 macro	Loss	AUC	Accuracy
A.a	X	ResnetInceptionV2	No	899	2	55	55	0,8143	0,15	0,93	0,9421
B.a	✓	Resnet InceptionV2	3	887	4	42	68	0,8619	0,145	0,91	0,9540
B.a	✓	Resnet InceptionV2	4	355	536	17	93	0,4062	0,065	0,995	0,4476
B.a	✓	**Densenet**	3	887	4	30	80	0,9030	0,15	0,97	0,9650
B.a	✓	Densenet	4	268	623	12	98	0,3458	0,03	0,995	0,3656
B.a	✓	Densenet	5	885	6	33	77	0,8879	0,18	0,935	0,9610
B.a	✓	**Xception**	3	884	7	28	82	0,9019	0,09	0,985	0,9650
B.a	✓	Xception	4	250	641	13	97	0,3320	0,01	0,995	0,3467
B.b	✓	Oversampling + Resnet InceptionV2	3	878	13	35	75	0,8654	0,10	0,99	0,9520
B.b	✓	Oversampling + Resnet InceptionV2	4	492	399	21	89	0,5804	0,05	0,995	0,4992
B.b	✓	Oversampling + Densenet	0	887	4	38	72	0,8750	0,09	0,985	0,9570
B.b	✓	**Oversampling + Densenet**	3	883	8	25	85	0,9094	0,07	0,995	0,9670
B.b	✓	Oversampling + Densenet	4	270	621	10	100	0,3497	0,01	1,00	0,3696
B.b	✓	Oversampling + Densenet	5	877	14	28	82	0,8867	0,09	0,995	0,9570
B.b	✓	Oversampling + Xception	3	873	18	34	76	0,8583	0,07	0,995	0,9481
B.b	✓	Oversampling + Xception	4	138	753	8	102	0,2388	0,01	0,999	0,2398
B.b	✓	Oversampling + Xception	5	882	9	35	75	0.8744	0,06	0,995	0,9560
B.c	✓	RGBA + Resnet Inception V2	5	861	30	31	79	0,8435	0,13	0,962	0,9391

Note: Highlighted entries in the table are either shown in **bold** or with a light green background for emphasis.

(a) **No Head** Without applying attention maps, using the ResNetInceptionV2 convolutional neural network.
  (i) **ResnetInceptionV2 - No Head**

B. **Results with DINO** - Evaluation of the architecture using the convolutional neural networks ResnetInceptionV2, Densenet, and Xception, applying the DINO distillation technique.

(a) **Head** Application of the oversampling technique to mitigate imbalanced classes using the convolutional neural networks ResnetInceptionV2, Densenet, Xception, applying only the DINO distillation technique.
  (i) **Resnet Inception V2 — Head 3**
  (ii) **Resnet Inception V2 — Head 4**
  (iii) **Densenet — Head 3**
  (iv) **Densenet — Head 4**
  (v) **Densenet — Head 5**
  (vi) **Xception — Head 3**
  (vii) **Xception — Head 4**

(b) **Oversampling** Application of the oversampling technique to mitigate imbalanced classes using the convolutional neural networks ResnetInceptionV2, Densenet, Xception, applying the DINO distillation technique
  (i) **Resnet Inception V2 — Head 3**
  (ii) **Resnet Inception V2 — Head 4**

(iii) **Densenet — Head 0**
(iv) **Densenet — Head 3**
(v) **Densenet — Head 4**
(vi) **Densenet — Head 5**
(vii) **Xception — Head 3**
(viii) **Xception — Head 4**
(ix) **Xception — Head 5**

(c) **RGBA Experiments** Application of the attention head in the fourth channel (alpha), aiming to maximize the features for the classifier, using the convolutional neural networks ResnetInceptionV2, Densenet, Xception, applying the DINO distillation technique

(i) **Resnet Inception V2 — Head 5.**

5.4 Analysis of the Results

The metrics used to evaluate the effectiveness of the generated models were accuracy, loss, AUC, and F1-macro. For each tested model, the respective training and validation curves were generated for the mentioned metrics, as well as confusion matrices that detail the performance in different scenarios.

The experimental process was conducted in well-defined stages, each with specific strategies to evaluate the models. Initially, the models were trained using only the neural network architectures — ResNet Inception V2, DenseNet, and Xception — combined with the Transfer Learning technique, using "ImageNet," without the application of other additional MML approaches. This initial experiment served as a baseline for comparison in subsequent stages. To prevent the model from continuing to train after the stabilization of learning, an *EarlyStopping* strategy was applied.

Next, proprietary datasets were created using the attention maps generated by each attention head of DINO. Each neural network architecture generated five datasets, one for each attention head. With these datasets, the models were retrained following the proposed architectures, and the results were collected for analysis. Additionally, a strategy for class balancing was implemented, using oversampling combined with the attention maps generated by each DINO head.

Subsequently, the models were retrained with an additional approach: the use of the attention map generated by each attention head, applied to the alpha channel of the images — red, blue, green, alpha (RGBA). This approach demonstrated improvements in the results, especially in scenarios with limited datasets and without class balancing strategies.

The results obtained were fully recorded, and Table 2 presents only the best results or those that deviated from the standard, as observed in the experiments with head 04.

Among the results, the model that combined DINO with oversampling and DenseNet with Head 3 stood out, delivering the best overall performance. This model demonstrated balanced behavior, achieving a high number of correct classifications for the majority class ("no defect") while significantly reducing errors in the minority class ("defect").

Another relevant point is that the models that applied *oversampling* generally presented better results for the F1-macro metric. The DenseNet architecture also demonstrated superior performance compared to other tested architectures in most of the experiments conducted. Additionally, it was identified that the fourth attention head showed a consistent bias for correctly classifying the minority class, penalizing the majority class.

Finally, experiments conducted with datasets containing attention maps generated by the attention heads (*0, 1, 2, 3, 4, 5*), combined with the attention maps applied to the alpha channel of the image, showed improvements or similar performance compared to models that used only the attention map in the dataset or did not apply DINO. These results were positive even in scenarios with limited datasets and without class balancing strategies, as presented in Table 2.

6 CONCLUSIONS

The increasing digitalization of manufacturing industries is accelerating the integration of AI into their processes, particularly for anomaly detection and visual inspections on metallic surfaces. Within this context, this research contributes to the development of innovative ML and CV-based solutions to address challenges such as data scarcity and class imbalance—critical aspects in sectors like aerospace, defense, and oil and gas, where structural integrity is vital for safety and reliability.

This work proposed a systematic method to strengthen and optimize the detection of defects in metallic alloys. The study investigated major CNN topologies and techniques applied to defect detection, emphasizing TL, DA, and self-distillation. Experiments conducted with ResNetInceptionV2, DenseNet, and Xception validated the approach, with DenseNet achieving the best results when combined with the DINO technique and the third attention head.

The results demonstrated consistent improvements in accuracy, F1-macro, AUC, and mAP metrics, confirming the effectiveness of the proposed strategies. The integration of attention maps into the alpha channel of images enhanced model performance, even under data limitations, evidencing the contribution of the attention mechanism to better feature representation and model interpretability.

Among the main contributions, the application of the DINO framework stands out for generating interpretable attention maps that identify relevant regions in the images, improving model generalization, and mitigating the effects of class imbalance. The creation of a diversified *dataset*, combining real and synthetic data, represents another relevant contribution, providing a valuable foundation for future research. Complementary strategies, including DA, semantic segmentation, and TL, were decisive in achieving robustness and adaptability across the tested architectures.

For future work, it is recommended to extend the proposed approach to other metallic surfaces and industrial contexts, exploring the combination of self-distillation with newer architectures such as Transformers. Further analyses could investigate the use of attention mechanisms in different image channels, including multispectral and 3D data, to broaden applicability. Additionally, deploying the method in embedded systems—such as drones and IoT devices—may enhance automation in aerospace and oil and gas sectors, provided computational efficiency is maintained for operation in resource-constrained environments.

REFERENCES

[1] A. Abraham. 2005. Adaptation of Fuzzy Inference System Using Neural Learning. In *Fuzzy Systems Engineering: Theory and Practice*, Nadia Nedjah and Luiza de Macedo Mourelle (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 53–83. https://doi.org/10.1007/11339366_3

[2] Ravulakollu K. Ahuja S, Shukla M. 2021. Optimized deep learning framework for detecting pitting corrosion based on image segmentation. *International Journal of Performability Engineering* Volume 17 (2021), Pages 627 – 637.

[3] Deegan J Atha and Mohammad R Jahanshahi. 2018. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Structural Health Monitoring* 17, 5 (2018), 1110–1128. https://doi.org/10.1177/1475921717737051 arXiv:https://doi.org/10.1177/1475921717737051

[4] Blossom Treesa Bastian, Jaspreeth N, S. Kumar Ranjith, and C.V. Jiji. 2019. Visual inspection and characterization of external corrosion in pipelines using deep neural network. *NDT & E International* 107 (2019), 102134. https://doi.org/10.1016/j.ndteint.2019.102134

[5] Sandro Bessa, Julio Duarte, and Marcos Santos. 2025. A Bibliometric Study on the Applications of Neural Networks in Metal Surface Defect Inspection. *Revista de Informática Teórica e Aplicada (RITA)* 32, 2 (2025), 64–82. https://doi.org/10.22456/2175-2745.142111 Available online: https://seer.ufrgs.br/index.php/rita/article/view/142111.

[6] Jakob Božič, Domen Tabernik, and Danijel Skočaj. 2021. Mixed supervision for surface-defect detection: from weakly to fully supervised learning. https://paperswithcode.com/dataset/kolektorsdd2. Accessed: [27/05/2025].

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Montreal, Canada, 9650–9659.

[8] Le Dinh Duy, Ngo Tuan Anh, Ngo Tung Son, Nguyen Viet Tung, Nguyen Ba Duong, and Muhammad Hassan Raza Khan. 2020. Deep Learning in Semantic Segmentation of Rust in Images. In *Proceedings of the 2020 9th International Conference on Software and Computer Applications* (s.n ed.) (Langkawi, Malaysia) *(ICSCA '20, s.n)*. Association for Computing Machinery, New York, NY, USA, 129–132. https://doi.org/10.1145/3384544.3384606

[9] Isack Farady, Chih-Yang Lin, Fityanul Akhyar, R. Roshini, and John Alex. 2021. Evaluation of Data Augmentation on Surface Defect Detection. In *Proceedings of the IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*. IEEE, Tainan, Taiwan, 1–2. https://doi.org/10.1109/ICCE-TW52618.2021.9603212

[10] M.W Gardner and S.R Dorling. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment* 32, 14 (1998), 2627–2636. https://doi.org/10.1016/S1352-2310(97)00447-0

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. 2020. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. arXiv preprint arXiv:2006.07733. https://arxiv.org/abs/2006.07733

[12] Iason Katsamenis, Eftychios Protopapadakis, Anastasios Doulamis, Nikolaos Doulamis, and Athanasios Voulodimos. 2020. Pixel Level Corrosion Detection on Metal Constructions by Fusion of Deep Learning Semantic and Contour Segmentation. In *Advances in Visual Computing*, George Bebis, Zhaozheng Yin, Edward Kim, Jan Bender, Kartic Subr, Bum Chul Kwon, Jian Zhao, Denis Kalkofen, and George Baciu (Eds.). Springer International Publishing, Cham, 160–169. https://doi.org/10.1007/978-3-030-64556-4_13

[13] Wenhao Li, Haiou Zhang, Guilan Wang, Gang Xiong, Meihua Zhao, Guokuan Li, and Runsheng Li. 2023. Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing. *Robotics and Computer-Integrated Manufacturing* 80 (2023), 102470. https://doi.org/10.1016/j.rcim.2022.102470

[14] Will Nash, Tom Drummond, and Nick Birbilis. 2019. Deep Learning AI for Corrosion Detection. In *CORROSION 2019 (NACE CORROSION, Vol. All Days)*. NACE International, OnePetro, Richardson, Texas, NACE–2019–13267. arXiv:https://onepetro.org/NACECORR/proceedings-pdf/CORR19/All-CORR19/NACE-2019-13267/1139758/nace-2019-13267.pdf

[15] Atiqur Rahman, Zheng Wu, and Rony Kalfarisi. 2021. Semantic Deep Learning Integrated with RGB Feature-Based Rule Optimization for Facility Surface Corrosion Detection and Evaluation. *Journal of Computing in Civil Engineering* 35, 6 (08 2021), 04021018. https://doi.org/10.1061/(ASCE)CP.1943âĹŠ5487.0000982

[16] V. Ramani, L. Zhang, and K. Kuang. 2021. Probabilistic assessment of time to cracking of concrete cover due to corrosion using semantic segmentation of imaging probe sensor data. *Automation in Construction* 132 (2021), 103961. https://doi.org/10.1016/j.autcon.2021.103961

[17] Zhiren Tian, Guifeng Zhang, Yongli Liao, Ruihai Li, and Fanqi Huang. 2019. Corrosion Identification of Fittings Based on Computer Vision. In *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. IEEE, Dublin, Ireland, 592–597. https://doi.org/10.1109/AIAM48774.2019.00123

[18] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. arXiv preprint arXiv:2012.12877. https://arxiv.org/abs/2012.12877 arXiv:2012.12877.

[19] Xuebing Xu, Yuanhang Wang, Jun Wu, and Yanzhi Wang. 2020. Intelligent Corrosion Detection and Rating Based on Faster Region-Based Convolutional Neural Network. In *2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai)*. IEEE, Shanghai, China, 1–5. https://doi.org/10.1109/PHM-Shanghai49105.2020.9281005

[20] Jong Pil Yun, Woosang Crino Shin, Gyogwon Koo, Min Su Kim, Chungki Lee, and Sang Jun Lee. 2020. Automated defect inspection system for metal surfaces based on deep learning and data augmentation. *Journal of Manufacturing Systems* 55 (2020), 317–324. https://doi.org/10.1016/j.jmsy.2020.03.009

[21] Jong Pil Yun, Woosang Crino Shin, Gyogwon Koo, Min Su Kim, Chungki Lee, and Sang Jun Lee. 2020. Automated defect inspection system for metal surfaces based on deep learning and data augmentation. *Journal of Manufacturing Systems* 55 (2020), 317–324. https://doi.org/10.1016/j.jmsy.2020.03.009

[22] Paweł Zuchniak, Witold Dzwinel, Tomasz Majerz, Andrzej Pasternak, and Kamil Dragan. 2021. Corrosion Detection on Aircraft Fuselage with Multi-teacher Knowledge Distillation. In *Computational Science – ICCS 2021*, Maciej Paszynski, Dieter Kranzlmüller, Valeria V. Krzhizhanovskaya, and Jack Dongarra (Eds.). Lecture Notes in Computer Science, Vol. 12747. Springer, Cham, 318–332. https://doi.org/10.1007/978-3-030-77970-2_25