Handling Drift in Industrial Defect Detection Through MMD-Based Domain Adaptation

Xuban Barberena[a], Fátima A. Saiz[b] and Iñigo Barandiaran[c]
Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), 20009 Donostia, Spain

Keywords: Quality Control, Deep Learning, Artificial Vision, Domain Adaptation, Model Drift, Defect Detection.

Abstract: This study enhances industrial quality control by automating defect detection using artificial vision and deep learning techniques. It addresses the challenge of model drift, where variations in input data distribution affect performance. To tackle this, the paper proposes a simpler, practical approach to unsupervised Domain Adaptation (UDA) for object detection, focusing on industrial applicability. A technique based on the Faster R-CNN architecture and a Maximum Mean Discrepancy (MMD) regularization method for feature alignment is proposed. The study aims to detect data drift using state-of-the-art methods and evaluate the proposed UDA technique's effectiveness in improving surface defect detection. Results show that statistical tests effectively identify variations, enabling timely adaptations. The proposed UDA method achieved mean Average Precision (mAP50) improvements of 3.1% and 6.1% under vibration and noise scenarios, respectively, and a significant 17.8% improvement for conditions with particles, advancing existing methods in the literature.

1 INTRODUCTION

Industrial quality control is crucial for ensuring products meet market standards, particularly in manufacturing where high production quality is essential for competitiveness. Recently, artificial intelligence (AI) and computer vision technologies have been adopted to automate defect detection, with deep learning (DL) proving effective for identifying surface defects (Leiñena et al., 2024).

However, significant challenges arise in dynamic industrial environments that can affect the accuracy and robustness of AI models. DL models rely on inference data being drawn from the same distribution as the training data. In these settings, variations in lighting, sensor noise, movement of components, and other factors can substantially alter image quality. This phenomenon, known as model drift or data drift, occurs when the distribution of inference data deviates from that of the training data, leading to performance degradation in DL models.

Traditional methods to address drift typically involve retraining models with new, often unannotated data, which is costly and time-consuming. To combat model drift, Unsupervised Domain Adaptation (UDA) techniques have emerged, enabling models to adapt to new data distributions without the need for extensive manual data annotation. This adaptability is crucial in industrial settings, where maintaining model performance over time ensures consistent quality control and reduces operational costs. This paper specifically addresses object detection methods, crucial for identifying and localizing defects in images, with a focus on surface defects in metallic components. By combining drift detection techniques with UDA, we aim to not only identify when performance degrades, but also enable the model to adapt and continue performing accurately under changing conditions.

While UDA has been widely studied in other fields, its application to object detection has been limited due to the complexity of this task. Object detection not only requires classifying objects but also accurately localizing them within the image, adding an extra layer of difficulty compared to tasks like image classification. Ensuring robust performance across domains for both classification and localization makes UDA for object detection particularly challenging. Current methods often become overly complex, assuming that sophistication translates to better performance.

However, recent studies suggest that simpler approaches can achieve performance comparable to more complex, state-of-the-art techniques. One promising method involves using regularizers to find data representations that remain invariant across do-

[a] https://orcid.org/0009-0004-9037-4192
[b] https://orcid.org/0000-0001-6065-7029
[c] https://orcid.org/0000-0002-8080-5807

Barberena, X., Saiz, F. A. and Barandiaran, I.
Handling Drift in Industrial Defect Detection Through MMD-Based Domain Adaptation.
DOI: 10.5220/0013170900003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 420-429
ISBN: 978-989-758-728-3; ISSN: 2184-4321

mains (Tzeng et al., 2014). Specifically, the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) loss has been explored to minimize feature distribution distances during training, enhancing alignment and robustness against drift. Despite its application in various contexts, MMD loss has not been extensively utilized in UDA for object detection, particularly within the Faster R-CNN (Ren et al., 2015) architecture.

This paper proposes implementing this method as a simpler alternative to complex techniques, emphasizing practical applicability in industrial environments. This study aims to achieve two main objectives: detecting data drift using state-of-the-art methods and evaluating the effectiveness of the proposed UDA technique to enhance surface defect detection in industrial environments. The simplicity and practicality of this approach make it particularly suitable for real-world industrial applications, where maintaining performance without complex system redesigns is a priority.

To conclude, the main contributions of this work are:

- A comprehensive review of the state of the art in image drift detection has been conducted, with several approaches from different works implemented. This enabled a comparative analysis of these methods, specifically focused on a real industrial use case.
- We propose a simpler yet effective approach to enhance UDA methods for object detection in industrial environments by minimizing the feature distribution discrepancy between source and target domains. This method improves the model's generalization without requiring complex redesigns. A regularizer based on Maximum Mean Discrepancy (MMD) is implemented to align feature distributions.
- Proposed method has been compared with, and shown to outperform, one of the leading state-of-the-art approaches.
- Given the difficulty in obtaining a database containing industrial images with drift, a simulated dataset was created to reflect various scenarios that could arise in an industrial environment.

2 RELATED WORK

2.1 Object Detection

Object detection is a computer vision field focused on identifying and localizing specific objects within images or videos. This involves two primary objectives: classifying the presence of objects and accurately determining their locations, typically represented by bounding boxes. Early methods relied on handcrafted features like Histograms of Oriented Gradients (HOG) (Dalal and Triggs, 2005) and Haar-like features (Lienhart and Maydt, 2002), often combined with classifiers such as Support Vector Machines (SVM) (Malisiewicz et al., 2011) or techniques like Sliding Windows (Sudowe and Leibe, 2011).

However, these methods struggled with complex patterns and variability in real-world images. The introduction of Convolutional Neural Networks (CNNs) revolutionized object detection by automatically learning hierarchical features from raw images, eliminating the need for handcrafted features and significantly improving detection accuracy and robustness. While CNNs excelled in classification tasks, the R-CNN (Region-based Convolutional Neural Networks) (Girshick et al., 2014) architecture marked a significant advancement by introducing a two-stage detection method: generating region proposals (ROIs) in the first stage and classifying them with a CNN in the second. This approach evolved into more sophisticated networks like Faster R-CNN (Ren et al., 2015), which integrates region proposal networks (RPNs) for end-to-end detection, achieving state-of-the-art results.

In contrast, single-stage detection methods streamline the process by integrating detection and localization into one step, improving processing efficiency. Networks such as SSD (Single Shot MultiBox Detector) (Liu et al., 2016) and YOLO (You Only Look Once) (Redmon et al., 2016) divide the input image into a grid of cells, predicting multiple bounding boxes and their associated classification probabilities simultaneously.

2.2 Model Drift

Traditional approaches relied on statistical tests, such as the Kolmogorov-Smirnov test (Massey Jr, 1951) and the Chi-square test (Pearson, 1900), which compare distributions between training and new data. While these statistical tests are still popular for detecting data drift, they struggle with high-dimensional data due to the curse of dimensionality, which makes it harder to detect subtle shifts in distributions (Hastie et al., 2009). As AI models, particularly in computer vision, often work with high-dimensional features, newer methods are required to effectively identify drift in such data.

Recent advances like Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) provide powerful tools for comparing distributions without requiring labelled data, making them ideal for unsupervised drift detection in industrial applications. Similarly, adversarial approaches (Rabanser et al., 2019), where a domain classifier discriminates between source and target data, offer a flexible method for identifying and addressing drift in high-dimensional feature spaces.

Additionally, recent methods such as (Greco et al., 2024) contribute to improving drift detection, while the survey by (Hinder et al., 2023) provides a comprehensive review of the state-of-the-art approaches in this field.

2.3 Unsupervised Domain Adaptation for Object Detection

UDA relies on fully labeled instances in the source domain while having no labels for the target domain. This approach is particularly relevant in real-world scenarios where new data often lacks annotations. UDA has been widely researched for tasks like classification (Saito et al., 2018) and semantic segmentation (Toldo et al., 2020). Unlike image classification, which only requires assigning a label to an entire image, object detection involves both classification and localization, making the task more complex. Domain adaptation for object detection must ensure that both the feature extraction and the bounding box prediction generalize well across domains, adding another layer of difficulty compared to tasks like image classification or segmentation.

The survey (Oza et al., 2023) categorizes existing UDA for object detection methods into different types. Adversarial feature learning aligns learned features across domains by training two competing models: a generator (feature extractor) and a discriminator (domain classifier). The generator minimizes the task loss (e.g., object detection) while trying to confuse the discriminator, which is trained to differentiate domains. By doing so, the generator learns domain-invariant features. DA-Faster (Chen et al., 2018) was one of the first to apply this adversarial approach proposed in (Ganin and Lempitsky, 2015) within the Faster R-CNN framework, pioneering UDA for object detection and influencing many subsequent works (Chen et al., 2021).

On the other hand, mean-teacher methods (Cai et al., 2019) use a teacher-student model to leverage labeled source data and unlabeled target data, with the teacher providing pseudo-labels and the student improving performance by aligning with the teacher's predictions. Image-to-Image Translation methods have also been explored aiming to translate images from one domain to another and create intermediate images between domains to reduce the gap. For example, (Arruda et al., 2019) employs a strategy based on this to adapt the model from detecting daytime scenes to nighttime scenes. Pseudo-label based self-training methods are also popular as they generate pseudo-labels for unlabeled target data using model predictions and then retrain the model with both labeled source data and these pseudo-labels to boost performance (Kim et al., 2019).

3 MATERIALS AND METHODS

In this section, the materials and methods used in this study are detailed. First, the dataset employed for the experimentation is presented. Next, the process followed for drift detection is described. Finally, the proposed domain adaptation architecture is explained in detail.

3.1 Datasets

In the manufacturing industry, obtaining high-quality data is a persistent challenge due to strict privacy and confidentiality around production processes. This limitation restricts access to diverse datasets and complicates the capture of variations or drift, which are essential for developing robust models.

Collecting drift data in highly optimized manufacturing environments is particularly difficult, as production plants aim to minimize variations and maintain strict control over operational variables. Consequently, while some types of drift may occur, they are infrequent and inadequately documented for creating comprehensive datasets.

Variations in manufacturing can arise from several factors. Changes in lighting due to fluctuations in natural or artificial sources can alter product appearance. Sensor failures or machine vibrations may introduce noise into images, while the movement of parts or cameras can lead to blurriness. Dust and dirt particles in the environment can obstruct visibility and distort image characteristics.

Given the challenges in obtaining real data that captures these variations, synthetic drift conditions have been simulated using the original data. To simulate drift conditions, we employed traditional data augmentation techniques, introducing variations such as changes in brightness, particles, noise, and vibration. These augmentations were carefully chosen to reflect real-world industrial scenarios. Ensuring that the synthetic data mimics true industrial drift conditions is essential for the robustness of the domain adaptation method. Figure 1 shows a sample of the different sets that were simulated from the original images.

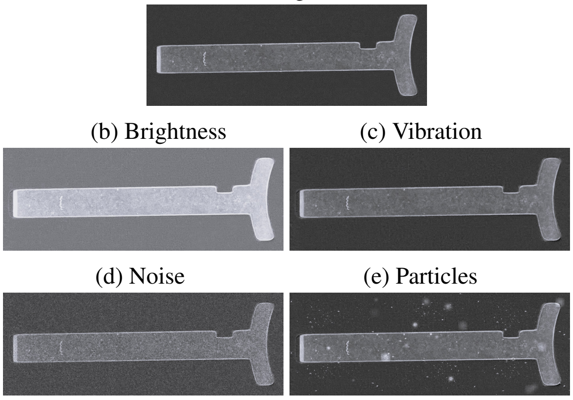
(a) Original (b) Brightness (c) Vibration (d) Noise (e) Particles

Figure 1: Sample of synthetic data for different scenarios based on an original image.

The original dataset comprises 625 images annotated by domain experts, supplemented by an additional 625 unannotated images to form specific sets for each drift scenario. As this study focuses on an unsupervised problem, labels for evaluating domain adaptation methods are not available. To facilitate a fair assessment, the original test images were used to create the drift test sets, ensuring consistent annotations across all sets. This approach prevents potential bias from unbalanced test datasets. The final dataset is composed of 200 images, half of which contain surface defects. This balance between defective and non-defective images ensures that the model can effectively learn to differentiate between the two classes under varying drift conditions, reducing the risk of class imbalance affecting performance. Table 1 shows the described distribution.

Table 1: Datasets distribution. All drift sets contain the same number of samples.

Set	Original Images	Original Labels	Drift Images	Drift Labels
Train	625	625	625	0
Test	200	200	200	200

3.2 Drift Detectors

Figure 2 illustrates a common pipeline for detecting drift in high-dimensional data, proposed in (Rabanser et al., 2019). In this process, both the original reference data Z_{ref} and the test data Z are subjected to the same dimensionality reduction. Afterward, a two-sample statistical test is applied to determine whether drift is present.

3.2.1 Dimension Reduction

Given an input dataset $\mathbb{X} \in \mathbb{R}^{N\times d}$, where N is the number of observations and d is the number of dimensions, the goal is to reduce the dimensionality from d to K, where $K \ll d$. This involves transforming the original dataset X into a new dataset $\hat{X}$ with a reduced dimension, preserving the essential structure and features while simplifying analysis. There exist several approaches to dimensionality reduction. One common type is linear projections such a PCA (Principal Component Analysis), which involve applying a projection or transformation matrix R to the dataset X, such that $\hat{X} = XR$. This linear transformation repositions data points into a lower-dimensional space using linear combinations of the original variables, aiming to preserve the structure and relationships of the original data. Another approach is to use nonlinear projections such as t-SNE (t-Distributed Stochastic Neighbor Embedding).

However, in recent years, autoencoders have been used for image dimensionality reduction (Wang et al., 2016). An autoencoder consists of an encoding function $\phi : X \mapsto \mathcal{H}$ that maps the input data to a lower-dimensional latent space $\mathcal{H}$, and a decoding function $\psi : \mathcal{H} \mapsto X$ that attempts to reconstruct the original data from this latent representation. Training involves learning ϕ and ψ to minimize reconstruction error. However, (Rabanser et al., 2019) suggests that even randomly initialized (untrained) autoencoders can be used for dimensionality reduction with the goal of detecting drift. Autoencoders, even when not fully trained, have shown promise for dimensionality reduction in high-dimensional datasets like images. The advantage of using an untrained autoencoder is that it avoids overfitting to specific features in the data, providing a more general reduction technique that can effectively support drift detection across different conditions. In this work, the approach of using an untrained autoencoder is employed to reduce the dimensionality of the data as part of the preprocessing for drift detection techniques.

3.2.2 Two-Sample Tests

Following the dimensionality reduction process using untrained autoencoders, a two-sample test is applied, which is a type of statistical test that compares two independent datasets to determine whether they come from the same distribution or if there are significant differences between them. The null hypothesis H_0 states that the observed and expected (or reference) data come from the same distribution, while the alternative hypothesis H_1 asserts that they do not. If the p-value associated with the statistical test is suffi-

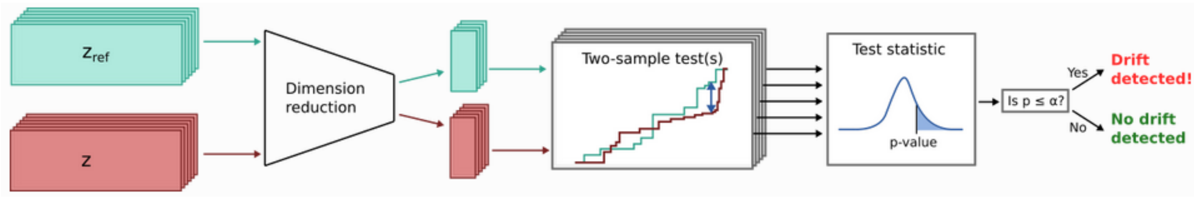

Figure 2: Pipeline for the drift detection process on images.

ciently small (typically less than a predefined significance level, such as 0.05), the null hypothesis can be rejected. This suggests that there is enough evidence to conclude that the observed and expected data do not come from the same distribution, indicating a drift in the data.

There are various tests available, but one of the most commonly used in the literature is the non-parametric univariate Kolmogorov-Smirnov (KS) test (Massey Jr, 1951). This test compares the empirical cumulative distribution functions (ECDFs) of both samples and calculates the maximum distance between them. The KS statistic is defined as:

$$KS = \max_x |F_s(x) - F_t(x)| \tag{1}$$

where F_s and F_t are the ECDFs of the source and target data, respectively.

Another widely recognized test is Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), a statistical measure used to assess the difference between two probability distributions by comparing their mean embeddings in a reproducing kernel Hilbert space (RKHS). Specifically, the MMD is defined as:

$$MMD^2(P,Q) = E_P[k(x,x)] - 2E_{P,Q}[k(x,y)] + E_Q[k(y,y)] \tag{2}$$

where P and Q are the two distributions being compared, x and y are random samples drawn from distributions P and Q, respectively, and $k(x,y)$ is a kernel function that measures the similarity between the transformed features of the samples. The expectation operator E computes the average values of the kernel functions. A smaller MMD value indicates greater similarity between the two distributions, while a larger value suggests a significant difference.

3.2.3 Learned Drift Detectors

The option of reducing dimensionality and applying statistical tests can be suitable in many cases, but in more complex situations, this approach may be insufficient. A more advanced alternative is the use of learned drift detectors. Unlike traditional methods that rely on statistical tests or direct comparison of features between two datasets, learned drift detectors use machine learning techniques to identify patterns and relationships in the data, detecting significant deviations in these patterns.

The article (Lopez-Paz and Oquab, 2016) proposes a classifier for two-sample tests that estimates the accuracy of a binary classifier using the reference and test datasets. If the accuracy is significantly higher than a predefined p-value, the null hypothesis H_0 is not rejected.

Another relevant method is the Learned Kernel (Liu et al., 2020), which detects drift adaptively in data streams. This approach learns a weighted kernel that captures differences between two data distributions by measuring the discrepancy between samples. It is an extension of the Maximum Mean Discrepancy (MMD), where the kernel is learned. The function of the learnable kernel is defined as follows:

$$k(z,z^{ref}) = (1-\varepsilon)k_a(\Phi(z),\Phi(z^{ref})) + \varepsilon k_b(z,z^{ref}), \tag{3}$$

where Φ is a learnable projection, k_a and k_b are kernel characteristics, and $\varepsilon > 0$ is a small constant.

3.3 Faster R-CNN

Faster R-CNN is a state-of-the-art object detection framework that improves upon its predecessors by integrating a Region Proposal Network (RPN) to generate high-quality region proposals. The architecture consists of two main components: the RPN, which is responsible for proposing candidate object bounding boxes, and a Fast R-CNN detector that classifies these proposals and refines their coordinates. By sharing convolutional features between the RPN and the detection network, Faster R-CNN achieves significant improvements in both speed and accuracy, making it well-suited for real-time applications.

The loss function used in Faster R-CNN comprises two main components: the classification loss and the bounding box regression loss, both calculated in the RPN and detection network components. The total loss L is defined as follows:

$$L_{\text{Faster R-CNN}} = L_{\text{cls}}^{\text{RPN}} + L_{\text{reg}}^{\text{RPN}} + L_{\text{cls}}^{\text{Det}} + L_{\text{reg}}^{\text{Det}}, \tag{4}$$

where L_{cls} is the classification loss, typically computed using softmax cross-entropy for the predicted class labels, and L_{reg} is the bounding box regression loss, which measures the accuracy of the predicted bounding box coordinates. The bounding box regression loss is often formulated as a smooth L1 loss.

Faster R-CNN serves as the backbone architecture in the majority of works that incorporate domain adaptation modules for object detection, due to its flexible and modular design, which facilitates the integration of adaptation techniques aimed at improving performance across different domains.

3.4 MMD-Based Domain Adaptation for Improved Object Detection Under Drift in Industrial Applications

The MMD regularizer is used to align the learned features so that they are as similar as possible between the source and target domains. The idea is that if the model performs well on the original data, ensuring that the feature distributions of the last convolutional layer are aligned will allow the model to transfer its knowledge to the new domain, making it domain-invariant. This strategy operates on the premise that minimizing the discrepancy between feature distributions across domains enhances the model's ability to generalize in diverse conditions without requiring complex redesigns, thereby facilitating a simpler implementation in industrial environments.

The MMD regularizer adds a penalty to the loss function objective to reduce model complexity and improve its generalization ability. This is achieved by imposing additional constraints during training, encouraging the model to learn more robust and generalizable representations, while minimizing the discrepancy between the feature distributions of the source and target domains.

The loss computed in the regularizer module is then added to the total loss of Faster R-CNN, which is calculated based on the original images:

$$L_{\text{tot}} = L_{\text{Faster R-CNN}}(X_{\text{s}}, y) + \lambda MMD^2(X_{\text{s}}, X_{\text{t}}), \quad (5)$$

where $L_{\text{Faster R-CNN}}(X_{\text{s}}, y)$ represents the total loss of the Faster R-CNN architecture on the source labeled data (X_{s}) and the ground truth labels (y). $MMD^2(X_{\text{s}}, X_{\text{t}})$ denotes the distance between the source data (X_{s}) and the target data (X_{t}). Finally, λ is a new hyperparameter that controls the intensity of the domain confusion.

This process is illustrated in Figure 3. The diagram represents the proposed method and highlights the implementation of the regularizer in the last convolutional layer of the backbone, as these features are the most abstract and representative of the content.

During training, for each mini-batch, $MMD^2(Xs, Xt)$ loss is computed between the feature maps of the last convolutional layer for X_s and X_t, while the $L_{\text{Faster R-CNN}}(X_{\text{s}}, y)$ loss is computed only on the labeled source data X_s Both losses are then combined using the hyperparameter λ, which balances the supervised learning and domain alignment. The network weights are then updated based on the total loss using the SGD optimizer.

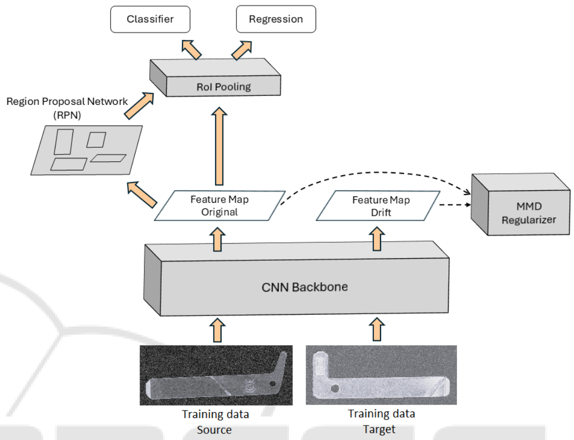

Figure 3: Proposed Faster R-CNN architecture with MMD regulizer.

4 RESULTS

This section details the experiments conducted to evaluate state-of-the-art models for drift detection in images and to compare the performance of the proposed methods against the DA-Faster architecture, which is a leading approach in object detection. The experiments aim to assess how well these methods can detect drift and enhance model performance in various scenarios. Furthermore, visualization techniques will be introduced to provide deeper insights into the results and to illustrate the impact of the proposed methods on the overall detection process.

4.1 Drift Detection

In this study, we compared four drift detection methods: the Kolmogorov-Smirnov test, MMD, a classifier-based detector, and the Learned Kernel method. Each method has different strengths: statistical tests like KS are simple and efficient, while methods like the Learned Kernel offer more advanced detection capabilities in high-dimensional and evolving data streams, making them more suitable for complex industrial applications.

The input image dimension is 128×320, and the dimensionality reduction is performed using an untrained autoencoder or feature extractor consisting of three convolutional layers with 64, 128, and 512 filters, respectively, each using a kernel size of 4×4, a stride of 2, and ReLU activation. The output of the convolutional layers is flattened and passed through a dense layer that reduces the representation to a single feature. This compact representation is specifically designed to enable the application of two-sample test methods for drift detection.

Table 2 presents the results for each drift detection method across the generated datasets. It also highlights the lowest execution times for each dataset, indicating the fastest method.

Table 2: Results for each detector of *drift* in the different sets.

	Original	Brightness	Vibration	Noise	Particles
KS					
p-value	0.45	0.03	0.03	0.04	0.03
Drift?	No	Yes	Yes	Yes	Yes
Time	1.791	1.808	1.804	1.874	1.817
MMD					
p-value	0.57	0.01	0.04	0.03	0.04
Drift?	No	Yes	Yes	Yes	Yes
Time	2.189	2.204	2.133	2.135	2.153
Classifier					
p-value	0.81	0.00	0.01	0.01	0.01
Drift?	No	Yes	Yes	Yes	Yes
Time	15.891	15.667	15.642	15.576	15.856
Learned Kernel					
p-value	0.89	0.00	0.01	0.00	0.01
Drift?	No	Yes	Yes	Yes	Yes
Time	1.712	1.604	1.571	1.507	1.740

A lower p-value indicates stronger evidence against the null hypothesis that the datasets come from the same distribution. Therefore, in terms of drift detection effectiveness, the Learned Kernel method consistently outperformed all other techniques across the datasets, achieving the lowest p-values. This demonstrates its capacity to adapt to drift conditions effectively, making it the most reliable option for pseudo real-time industrial applications. Additionally, the runtime of each method was evaluated, confirming that all are generally suitable for production requirements, except for the classifier, which requires more time to process the data.

Additionally, the robustness of detections was evaluated in pseudo-real-time by processing images as small data streams, typical in dynamic industrial scenarios. The evaluation of drift detection in pseudo-real time, illustrated in Figure 4, is particularly crucial in industrial settings where timely identification of changes can prevent defects and maintain product quality.

In the figure, each point represents a sample from the data stream, with green points indicating original samples that are consistent with the training data and red points representing samples that exhibit drift. The horizontal red line marks the p-value threshold for detecting drift. In this case, the model used is the Learned Kernel, which effectively identifies changes by ensuring that the green points remain above the line, indicating that there is no drift. Conversely, the red points fall below the threshold, demonstrating the model's ability to successfully detect deviations in pseudo-real-time.

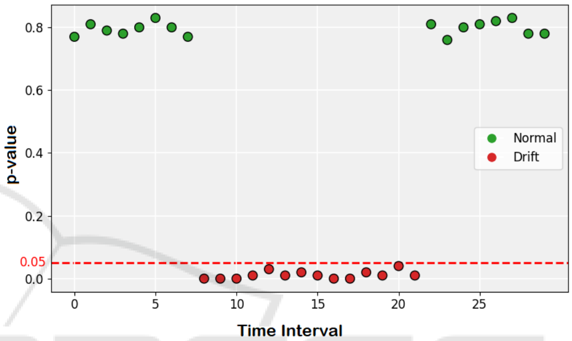

Figure 4: Drift detection in pseudo-real time using the Learned Kernel method in data streams. The green points represent original samples and red points indicates samples with drift.

4.2 Domain Adaptation

A specific model was trained for each drift scenario using the same configuration and hyperparameters. Table 3 presents the mAP50 metric for our model and the state-of-the-art DA-Faster model across each dataset, along with initial results from the original Faster R-CNN model for reference.

Table 3: mAP50 of each model in the different test sets.

Test Set	Faster R-CNN	DA-Faster	Ours
Original	84.2	-	-
Brightness	81.2	79.8	81.7
Vibration	76.6	76.0	79.7
Noise	74.1	79.0	80.2
Particles	45.6	42.1	63.4

The initial model shows strong performance on the original dataset, achieving 84.2% in the mAP50 metric. However, in the various drift scenarios, its performance is significantly affected. When adapting with DA-Faster, an improvement is only observed in the noisy scenario, with a +4.9% increase compared to the initial model.

In contrast, our proposed method consistently improved the mAP50 metric across all drift scenarios. Notably, it achieved enhancements of +6.1% in noisy conditions and +17.8% in the presence of particles, showcasing its ability to adapt effectively where DA-Faster struggled, particularly in challenging environments.

Figure 6 shows examples of the detection results before and after applying the proposed adaptation method.

To further understand the effects of our domain adaptation approach, we employed visualization techniques such as t-SNE and Grad-CAM. The t-SNE visualizations (Figure 5) indicate that successful adaptation leads to more aligned features, reflecting the model's ability to generalize across domains. Additionally, Grad-CAM visualizations (Figure 7) illustrate how the model's focus shifts correctly to relevant areas after adaptation, enhancing its detection capabilities in the presence of noise.

Figure 5: t-SNE on the features of the last convolutional layer.

On the other hand, Grad-CAM (Gradient-weighted Class Activation Mapping) was used to generate heatmaps that highlight the most important regions in an image for the model's predictions. Unlike t-SNE, which offers a global feature distribution view, Grad-CAM provides a localized interpretation of how specific regions influence the network's decision.

Figure 7 illustrates a component example where the original model correctly focuses on defects. However, with added noise, the focus shifts incorrectly. After adaptation, the model successfully refocuses on the relevant areas for the new domain.

5 CONCLUSIONS

This study aimed to detect data drift using state-of-the-art methods and evaluate the viability of UDA for object detection techniques, including a proposed MMD-based regularization method for the Faster R-CNN architecture, for industrial deployment. The proposed approach focuses on improving the model's generalization ability across varying conditions without requiring complex redesigns or deep expertise in advanced methods.

The results show that statistical tests were effective in detecting drift in images simulating industrial conditions, making them suitable for production environments by enabling timely adaptation. In terms of DA, the proposed method outperformed DA-Faster in scenarios with noise, vibration and particles, achieving a mAP50 improvement of 3.1%, 6.1% and 17.8%, respectively. In contrast, DA-Faster only showed marginal improvements and struggled to match performance in other scenarios.

Controlling drift is crucial in industrial settings. While effective detection methods exist, UDA for object detection remains complex. Current UDA methods, though promising, are not always sufficient for ensuring optimal performance across all industrial scenarios. While the methods used here have shown positive results, challenges may arise in real-world applications, especially when dealing with more subtle forms of drift, such as gradual changes in sensor calibration or material properties. In such cases, the methods employed may not perform as effectively, and further adaptation or refinement could be necessary to meet production-level requirements.

Moving forward, it would be beneficial to test the proposed approach on public datasets. However, there are limited publicly available datasets featuring industrial images with drift for UDA methods. Additionally, future work could explore more complex and diverse variations in datasets to better evaluate generalization, investigate the application of the proposed regularizer to other detection networks to assess its generalization capacity, and conduct comparative studies of domain adaptation techniques applied to the same problem to provide broader insights.

ACKNOWLEDGEMENTS

This research has been supported by the Elkartek Programme, Basque Government (Spain) (SMART-EYE, KK-2023/00021 and BEREZIA, (KK-2023/00012).

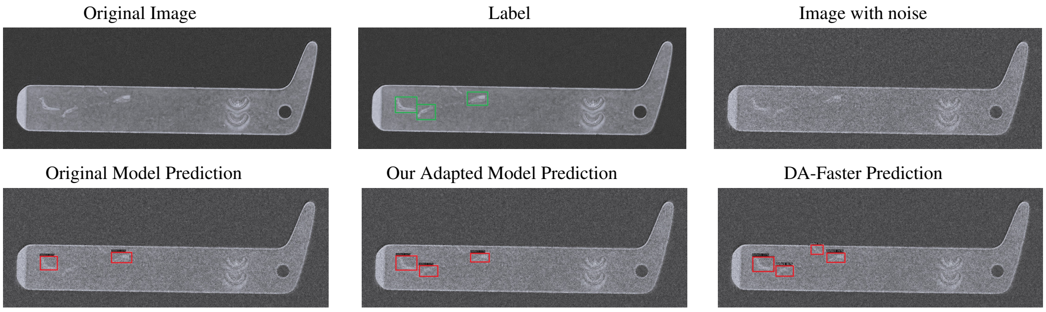

Figure 6: Example of the predictions made by different models on a component with noise.

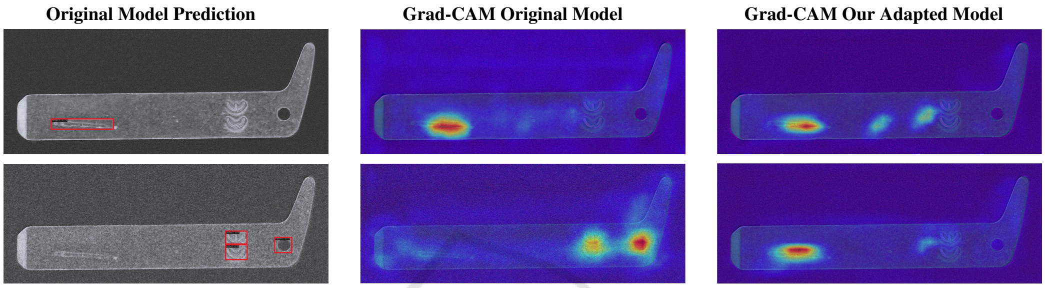

Figure 7: Example of Grad-CAM applied to an image affected by noise. The first row shows the original image, while the second row displays the same image with added noise.

REFERENCES

Arruda, V. F., Paixao, T. M., Berriel, R. F., De Souza, A. F., Badue, C., Sebe, N., and Oliveira-Santos, T. (2019). Cross-domain car detection using unsupervised image-to-image translation: From day to night. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Cai, Q., Pan, Y., Ngo, C.-W., Tian, X., Duan, L., and Yao, T. (2019). Exploring object relation in mean teacher for cross-domain detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11457–11466.

Chen, Y., Li, W., Sakaridis, C., Dai, D., and Van Gool, L. (2018). Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348.

Chen, Y., Wang, H., Li, W., Sakaridis, C., Dai, D., and Van Gool, L. (2021). Scale-aware domain adaptive faster r-cnn. *International Journal of Computer Vision*, 129(7):2223–2243.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee.

Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Greco, S., Vacchetti, B., Apiletti, D., and Cerquitelli, T. (2024). Unsupervised concept drift detection from deep learning representations in real-time. *arXiv preprint arXiv:2406.17813*.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hinder, F., Vaquet, V., and Hammer, B. (2023). One or two things we know about concept drift–a survey on monitoring evolving environments. *arXiv preprint arXiv:2310.15826*.

Kim, S., Choi, J., Kim, T., and Kim, C. (2019). Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6092–6101.

Leiñena, J., Saiz, F. A., and Barandiaran, I. (2024). Latent diffusion models to enhance the performance of visual defect segmentation networks in steel surface inspection. *Sensors*, 24(18):6016.

Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *Proceed-*

ings. international conference on image processing, volume 1, pages I–I. IEEE.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A., and Sutherland, D. J. (2020). Learning deep kernels for non-parametric two-sample tests. In *International conference on machine learning*, pages 6316–6326. PMLR.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer.

Lopez-Paz, D. and Oquab, M. (2016). Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*.

Malisiewicz, T., Gupta, A., and Efros, A. A. (2011). Ensemble of exemplar-svms for object detection and beyond. In *2011 International conference on computer vision*, pages 89–96. IEEE.

Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Oza, P., Sindagi, V. A., Sharmini, V. V., and Patel, V. M. (2023). Unsupervised domain adaptation of object detectors: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Pearson, K. (1900). X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.

Rabanser, S., Günnemann, S., and Lipton, Z. (2019). Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. (2018). Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732.

Sudowe, P. and Leibe, B. (2011). Efficient use of geometric constraints for sliding-window object detection in video. In *International Conference on Computer Vision Systems*, pages 11–20. Springer.

Toldo, M., Maracani, A., Michieli, U., and Zanuttigh, P. (2020). Unsupervised domain adaptation in semantic segmentation: a review. *Technologies*, 8(2):35.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Wang, Y., Yao, H., and Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242.