A drift detection prototype for quality control in manufacturing

Dharmil Rajesh Mehta
Fraunhofer IAO, Germany, dharmil-rajesh.mehta@iao.fraunhofer.de

Safa Omri
Fraunhofer IAO, Germany, safa.omri@iao.fraunhofer.de

Richard Zowalla
Fraunhofer IAO, Germany; Hochschule Heilbronn, richard.zowalla@iao.fraunhofer.de

Jens Neuhüttler
Fraunhofer IAO, Germany, jens.neuhuettler@iao.fraunhofer.de

A drift detection prototype for quality control in manufacturing
Research Paper

Dharmil Rajesh Mehta, Safa Omri, Richard Zowalla, and Jens Neuhuettler

Fraunhofer Institute for Industrial Engineering IAO, Research and Innovation Center for Cognitive Service Systems (KODIS), Heilbronn, Germany
{dharmil-rajesh.mehta, safa.omri, richard.zowalla, jens.neuhuettler}@iao.fraunhofer.de

Abstract. Drift detection is a critical factor in sustaining the performance of Computer Vision (CV) models in modern manufacturing, where it ensures the continuous achievement of high efficiency, quality, and innovation. CV models efficiency is more likely to reduce over time due to environmental changes, equipment wear and tear, or alterations in product specifications. Moreover, the collection of labeled data, essential for monitoring and adjusting these models, is often hindered by constraints such as time, cost, and the lack of domain experts for annotation. Addressing this gap, our research introduces an innovative unsupervised drift detection technique capable of identifying shifts in data distribution without the need for labeled data. This method proactively notifies operators of any decline in data quality that breaches a predetermined safety threshold, thereby preserving the operational integrity of CV applications. Our experimental results confirm the method's efficiency in detecting environmental shifts, changes in product specifications, and an increase in the production of defective parts, highlighting its significant potential to enhance quality control in manufacturing processes through reliable drift detection.
Keywords: Drift Detection, Computer Vision, Artificial Intelligence, Concept Drift

1 Introduction

In the dynamic world of manufacturing, where production environments and processes are continuously evolving, the precision and adaptability of machine learning models, especially those based on Computer Vision (CV) play a key role (Omri et al., 2023). These models, integral to various stages of the production process including defect detection, quality control, object recognition, and automation, hinge on the assumption that data distributions remain static over time. Yet, in reality, these distributions are subject to change, a phenomenon known as drift. Drift challenges the foundational assumptions of many machine learning models by altering the statistical properties of the target variable over time, necessitating robust detection mechanisms to ensure models adapt and maintain performance in ever-changing environments (Engstrom et al., 2017; Recht et al., 2019; Hendrycks and Dietterich, 2019).

The integration of Artificial Intelligence (AI), particularly through CV, into manufacturing, signifies a substantial shift towards more efficient, quality-driven, and innovative production methods. However, this shift is not without its challenges. The reliability and effectiveness of CV systems in manufacturing are compromised by fluctuations in environmental conditions, equipment integrity, and product specifications. These variations can subtly, yet significantly, affect the data distribution, leading to a decline in model performance if not promptly identified and addressed. Thus, monitoring CV systems for drift becomes imperative, not only for maintaining operational standards but also for upholding quality control and safety regulations.

Despite the recognized importance of drift detection in maintaining the accuracy of CV models, the field faces several hurdles. Among these is the difficulty of acquiring labeled data in manufacturing settings which is hindered by constraints such as cost, time, and the availability of domain experts. This challenge underscores the need for an unsupervised approach to drift detection, capable of identifying shifts in data distribution without relying on labeled datasets (Breck et al., 2019; Klaise et al., 2020).

Recent advancements in drift detection have proposed more nuanced approaches that consider the conditional distributions of data given specific contexts that are allowed to change over time (Yingying Sun, 2024). Building upon these methodologies, our paper introduces a framework for context-aware drift detection, which leverages machinery from the causal inference domain to develop a comprehensive detection system built upon a foundation of two-sample tests for conditional distributional treatment effects (Cobb et al., 2021).

This paper presents an innovative unsupervised drift detection methodology designed to tackle these challenges. Our method warns the user of a decline in the data quality that could jeopardize the performance of CV models, ensuring that manufacturing processes continue to benefit from the efficiency and innovation that AI brings. Through rigorous experimentation, we demonstrate the capability of our approach in detecting environmental changes, shifts in product specifications, and rise in the production of defective items. In doing so, we offer a valuable tool for the manufacturing industry to leverage the full potential of computer vision technology, enhancing quality control and ensuring the reliability of AI-driven processes.

Our contributions are twofold:

1. we demonstrate the practical applicability of the drift detection method in the industry by identifying environmental changes and variations in product specifications.
2. we offer a platform where users can not only test our method with sample data but also upload their data to assess and compare drift occurrences.

Through rigorous experimentation and analysis, we aim to establish a new paradigm in maintaining the accuracy and reliability of CV systems in the dynamic and ever-changing environment of manufacturing. This endeavor not only addresses an immediate need within the industry but also paves the way for future innovations in the fields of machine learning and computer vision.

The remainder of the paper is structured as follows. In section 2 we discuss the related work in the field of drift detection. In section 3 we discuss the methodology for drift detection. In section 4 we present the experiment, result, and prototype. In section 5 we conclude the paper and give an outlook of the future work.

2 Related Work

The latest research in drift detection, particularly in machine learning (ML) and its industrial applications, focuses on advanced methods for identifying and addressing both concept drift and data drift. These drifts are crucial for understanding because they directly impact the performance and reliability of ML models in dynamic environments. Munirathinam (2020) presented a model for detecting drifts and identifying outliers in IoT sensor data, with a focus on real-time monitoring and early detection. It is used to identify potential outliers and detect both aggressive and progressive drifts, enabling real-time monitoring and early detection. In the field of industrial process modeling, concept drift detection methods have been developed, focusing on regression modeling and addressing the challenges of detecting concept drift in industrial processes. Sun et al. (2020) provides a comprehensive overview of concept drift detection methods for industrial process modeling, addressing the need for more research in this area. In the context of business processes, drift detection is used to identify sudden and gradual changes, with a focus on automated and statistically grounded methods (Maaradji et al., 2015, 2017). Yeshchenko et al. (2019) introduces the Visual Drift Detection tool, which categorizes, drills down, and quantifies process drifts, providing a valuable aid for business process redesign. These methods have been shown to accurately detect typical change patterns and scale up to work for online drift detection. These studies underscore the diverse approaches to understanding and mitigating drift effects but also highlight a common limitation: the heavy reliance on labeled data and the assumption of relatively stable environmental conditions, which are often not present in manufacturing contexts.

The use of drift detection in computer vision has been explored in various studies. Suprem et al. (2020) introduced ODIN, a system that uses adversarial auto-encoders to detect and recover from drift in visual data. Liu and Zhou (2014) proposed a framework that combines concept drift detection with an online semi-supervised boosting method for robust visual tracking. Rabanser et al. (2019) introduced an approach for drift detection using statistical tests and distance-based methods. Liu et al. (2021) proposed a Learned Kernel MMD a kernel-based two-sample test method that improves the performance of drift detection by learning the kernels from deep neural nets. Cobb et al. (2021) introduced Context-Aware MMD a drift detection framework for conditional distribution effects. dos Reis et al. (2016) introduced an approach for drift detection in streaming. The proposed method significantly reduced the insertion and deletion time of the data points for drift detection. Hegedüs et al. (2012) presented an approach for detecting concept drift in large-scale, fully distributed networks, using online learners and an adaptive mechanism. These studies collectively demonstrate the potential of drift detection in improving the accuracy and robustness of computer vision systems. Jin et al. (2023) proposed the different environmental factors that can affect the performance of the CV model in manufacturing.

3 Methodology

In this section, we will explain the methods used to detect drift in high-dimensional data such as images. We first reduce the dimension and then perform statistical hypothesis testing to compare the distribution of features. In addition, we discuss common issues and challenges for a CV system in an industrial setting, which may influence deep learning model performance in production. The framework for drift detection in the experiment is as shown in Figure 1.

3.1 Dimensionality Reduction

Convolutional Neural Networks (CNNs) are one of the most suitable methods for feature extraction for image data. The performance of Efficient Net is much better as compared to other network architectures and even obtained the state-of-the-art performance on ImageNet and the CIFAR-100 datasets (Tan and Le, 2019). The architecture not only increases the accuracy of prediction on the dataset but also penalizes if the network is very computationally heavy. Hence we use Efficient Net B5 for the experiment as a feature extractor.

3.2 Statistical Hypothesis Testing:

To analyze potential drift in the input data, we compare the distribution of the feature vectors within a reference data set with the distribution observed in a test data set, i.e. data collected at a different location or under different environmental conditions. We perform statistical testing using Maximum Mean Discrepancy (MMD) multivariate kernel tests and Kolmogorov-Smirnov Test (KS) with Bonferroni Correction a univariate test and aggregating it with Bonferroni correction.

Multivariate Kernel Two-Sample Tests: MMD is a kernel-based two-sample statistical test that calculates the distance between feature means (Gretton et al., 2012*a*). A feature map Φ maps X to kernel Hilbert space τ.

$$k(X, Y) = \langle \phi(X), \phi(y) \rangle_{\tau} \tag{1}$$

Given a probability P on X, the feature means μ_P maps $\Phi(X)$ to the mean of every coordinate of $\Phi(X)$.

$$\mu_p(\Phi(X))) = \left[E\left[\Phi(X_1)\right], ..., E\left[\Phi(X_m)\right]\right]^T \tag{2}$$

The inner product of feature means of X and Y can also be written as a kernel function such that:

$$\begin{aligned} &\langle \mu_p(\Phi(X)), \mu_q(\Phi(X)) \rangle_{\tau} \\ &= E_{p,q}\left[\langle \Phi(X), \Phi(Y) \rangle_{\tau}\right] \\ &= E_{p,q}\left[k(X, Y)\right] \end{aligned} \tag{3}$$

MMD is the distance between feature means of X, Y that is of μ_p, μ_q:

$$MMD^2(p, q) = \|u_p - u_q\| \tag{4}$$

By using the norm of the inner product, the Equation 3 becomes:

$$
\begin{aligned}
MMD^2(p,q) &= \langle u_p - u_q, u_p - u_q \rangle \\
&= \langle u_p - u_p \rangle - 2 \langle u_p, u_q \rangle \\
&+ \langle u_q, u_q \rangle
\end{aligned} \tag{5}
$$

We calculate an unbiased estimate of the squared MMD using samples from both distributions as follows:

$$
\begin{aligned}
MMD^2(X,Y) = &\frac{1}{m(m-1)} \sum_i \sum_{j \neq i} \kappa(x_i, x_j) - \\
&\frac{1}{mm} \sum_i \sum_j \kappa(x_i, y_j) \\
&+ \frac{1}{m(m-1)} \sum_i \sum_{j \neq i} \kappa(y_i, y_j)
\end{aligned} \tag{6}
$$

P-value is calculated by performing a permutation test on the Gaussian Kernel matrix (Gretton et al., 2012*b*).

$$
\kappa(x,y) = e^{-\frac{1}{\sigma}||x-y||^2} \tag{7}
$$

Multiple Univariate Testing: In MMD, K-Dimensional feature vectors are tested at the same time. Hence, we selected KS Test, where each dimension is tested individually. Z is the maximum absolute difference between two empirical cumulative distribution functions $F_p(z)$ and $F_q(z)$. Multiple p-values make it difficult to decide whether there is a drift in overall data because we don't know which p-value to select. Therefore we correct it using Bonferroni correction. It rejects the null hypothesis when the minimum p-value is less than α/K.

$$
Z = sup\,|F_p(z) - F_q(z)| \tag{8}
$$

Threshold selection: For selecting the probability of rejecting the null hypothesis (p-value) we could either use a higher p-value (10%) or lower p-value (1% or 0.1%). If we use a lower p-value, the machines would be running where the data might not be similar to the reference dataset and the operator might not be notified until the value drops close to 0 which can be very expensive. If we use a higher p-value value the operator will be notified if there is any slightest change in the environment. The cost of rejecting a true null hypothesis in our scenario is less than failing to reject a false null hypothesis. Hence we have used a p-value of 10% for the experiments.

3.3 Environmental Influences on Image Data Quality

In industrial CV system, the quality of the captured image is important for accurately monitoring product quality via a deployed deep learning model. In the following subsec-

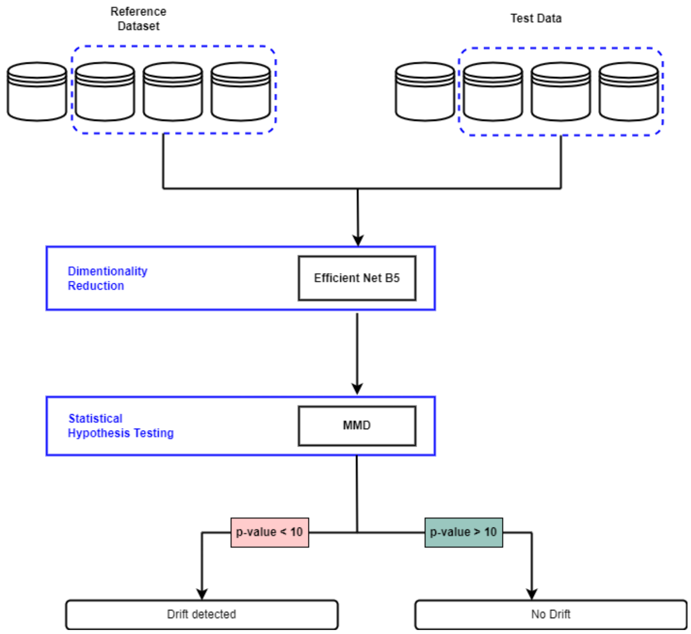


Figure 1. Drift Detection Framework

tions, we will discuss common environmental influences that can affect image quality and for which a drift detection system needs to be robust. These effects are depicted in Figure 2. Addressing these issues is critical in maintaining the reliability and effectiveness of the CV system in monitoring product quality and ensuring smooth operations along production lines.

Blurry Images: A common problem is the occurrence of `blurred images`, which can be caused by a variety of factors. Poor focusing, where the camera fails to adjust its lens accurately to capture a clear image, is a primary issue (Gedraite and Hadad, 2011). In addition, slower shutter speeds can cause motion blur, especially when monitoring fast-moving production lines or moving objects. Finally, the presence of dirt or damage to the lens can obstruct the light path, causing distortion and blurriness in the captured images. The blurry images are generated using the cv2 Gaussian Blur function using a kernel size of 5x5.

Images in Motion: When products are in motion (e.g. on fast-moving product lines), or when the camera itself is in motion, images can suffer from `motion blur`. This phenomenon occurs when the camera's exposure time is not synchronized with the speed of moving objects, resulting in a smear effect in the captured images that affects the clarity and detail of the captured images. To simulate images in motion we have used a kernel size of 5x5 and the motion direction is 45°.

Changes in Lighting Conditions: Variations of `lightning conditions` pose a significant challenge to the consistency and reliability of image acquisition. Sudden changes in the lightning environment, whether due to natural light variations or artificial sources, can result in images with increased or decreased brightness. These variations can greatly affect the accuracy of image analysis, affecting their ability to effectively detect defects or anomalies in the captured data. Increased brightness in images can lead to overexposure, washing out important details, and distorting the visual information. Reduced brightness can obscure fine features and reduce contrast between objects. Overexposed images have 60% more brightness than the normal images and underexposed images have 20% less brightness.

Change of Camera: Cameras may be upgraded, replaced, or modified for a variety of reasons, ranging from technological advances to equipment failure. However, such changes can introduce variability in the characteristics and performance of the imaging devices, potentially resulting in images that differ from those contained in the original training dataset. This variability can pose a challenge to CV systems trained on specific images, as the newly captured images may have different resolutions, color profiles, or different focal lengths. The reference dataset has an image size of 300x300 and the images generated by the new camera are of 512x512 size.

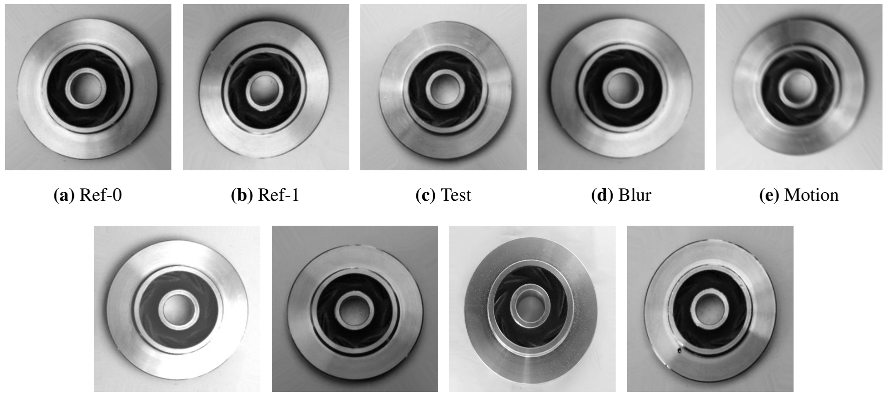
Manufacturing of only Defective Parts: The scenario where only defective parts are being produced is not a case of data drift, because the CV model is trained to catch these data and label them as `def`. However, it is economically more favorable to detect this case as drifted data and alert the manufacturers as soon as possible so that they can find the underlying cause.

Images without drift: While deploying drift detection in production most of the time the production process would be working fine under normal circumstances. Hence it is of utmost priority to detect normal running conditions in manufacturing and not send false alarms to operators.

4 Results and Prototype

4.1 Experiments Setup

To observe the performance of drift detection on the industrial use case and verify that it can detect the environmental influences described above we have performed experiments on a submersible pump impeller for quality inspection (Dabhi et al., 2020). The functionality of the impeller is to transfer the energy from the motor to the fluid being pumped. The dataset consists of the top view of the impeller which is made by the casting process. The dataset consists of 8648 images, which are either labeled `ok` or `def` (defect). The class distribution and the datasets used for testing in each experiment are explained in Table 1.



(a) Ref-0 **(b)** Ref-1 **(c)** Test **(d)** Blur **(e)** Motion

(f) Bright **(g)** Dark **(h)** Camera **(i)** Defect

Figure 2. Sample Images. (a) Ref-0: Image from train dataset. (b) Ref-1: Image from train dataset. (c) Test: Image from test dataset. (d) Blur: Gaussian Blur Kernel Size 5x5. (e) Motion: 45° Kernel Size 5x5. (f) Bright: 60% more brightness. (g) Dark: 20% less brightness. (h) Camera: Resolution of 512x512. (i) Defect: Manufacturing Defects

Reference Dataset: For quality control, the best-performing CV models that were trained on these datasets are used in production. The full training dataset may be very large and unavailable during production; to address this issue, we create ten reference datasets by sampling 100 images from train-ok `Ref-1`. In addition, the majority of parts produced during manufacturing will be of label `ok`; however, the proportion of `ok` and `def` in the training dataset is forty-five percent. Therefore, we generate ten reference datasets using train-ok dataset `Ref-1`.

Test Dataset: To ensure that the drift detection works properly and does not give false alarms, we use test datasets for this purpose. The test dataset is ideal since it contains images that are not shown to the model during the training step, but the probability density of the features must be similar to the probability density of the reference dataset. Hence we generate the `Test Samples` dataset by randomly sampling 100 images from test-ok `T-1`. Similarly, we generate images with the environmental influences i.e. `Blurry Images, Images in Motion, Change in Lighting Conditions` by adding noise in the `Test Samples` dataset using the OpenCV library. To test the `Change of Camera` scenario we generated a dataset by sampling 100 images from new-camera-ok `NC-1`. We generate `Parts with defects` dataset by sampling 100 images from test-def `T-2`.

4.2 Statistical Results

In this section, we want to demonstrate that the drift detection method can be deployed in a CV system to ensure that the CV model is deployed in a safe environment. It can trigger an alarm and notify the users when there is any change in the environment or

Table 1. Dataset

Identifier	Dataset	No. of Images	Class	Image Size	Experiment
Ref-1	train-ok	2875	ok	300 x 300	Reference Data
Ref-2	train-def	3758	def	300 x 300	Not used
T-1	test-ok	262	ok	300 x 300	Blurry Images, Images in Motion, Changes in Lightning Conditions
T-2	test-def	453	def	300 x 300	Manufacturing of only Defective Parts
NC-1	new-camera-ok	519	ok	512 x 512	Change of Camera
NC-2	new-camera-def	781	def	512 x 512	Not used

when the machine produces more faulty products. Table 2 shows the drift score using pre-trained Efficient Net B5 as a feature extractor and using MMD and KS Test as a statistical test for drift detection. Test Samples when compared with Reference Datasets should have a p-value of more than 10 as they are from similar distribution. KS Test gave false positive results for the Ref-3 Dataset. KS Test also found it difficult to detect environmental changes such as increased intensities, blurred images, and faulty parts. MMD outperformed KS Test in both scenarios by accurately detecting test samples and erroneous samples. Hence, we have developed the prototype using MMD as a hypothesis-testing method.

Table 2. Drift Score on the Datasets

Reference	Test Sample		Blurry Images		Images in Motion		Bright Images		Dark Images		Different Camera		Parts with defects	
	MMD	KS	MMD	KS	MMD	KS	MMD	KS	MMD	KS	MMD	KS	MMD	KS
Ref-0	44	44	0	0	1	0	0	6	0	0	3	1	0	23
Ref-1	39	27	0	0	2	0	0	2	0	0	1	5	0	10
Ref-2	73	33	0	0	2	0	0	4	0	0	0	0	0	3
Ref-3	77	8	1	0	3	0	0	19	0	0	1	1	0	1
Ref-4	77	100	1	2	2	0	0	9	0	0	0	0	0	2
Ref-5	98	100	2	4	9	0	0	36	0	0	0	2	0	2
Ref-6	95	72	1	1	8	0	0	11	0	0	0	2	0	2
Ref-7	57	85	0	80	1	0	0	8	0	0	0	0	0	0
Ref-8	99	100	2	3	8	0	0	47	0	0	0	1	0	0
Ref-9	78	100	1	4	3	0	0	13	0	0	0	2	0	1

4.3 Prototype

In the progression of our research, we engineered an interactive prototype to facilitate real-world application and evaluation of our unsupervised drift detection methodology. This prototype represents a bridge between theoretical research findings and practical, hands-on experimentation, enabling users to directly engage with the drift detection mechanism within a controlled environment. The design and implementation details, alongside the functionalities of this prototype, are articulated below:

(i)Architectural Framework: The prototype's architecture comprises a React-based frontend interface and a FastAPI-driven backend, implemented in Python. This architectural choice was governed by the need for a responsive user interface that could seamlessly interact with a robust and efficient backend processing unit. React, renowned for its efficient update and rendering capabilities, provides an intuitive interface for users to interact with the prototype. Concurrently, FastAPI offers a high-performance backend solution with asynchronous support, crucial for handling the computationally intensive tasks associated with drift detection in high-dimensional data.
(ii) Functional Capabilities: The prototype is designed to perform a dual role: firstly, as a demonstration tool, allowing users to replicate the drift detection experiments highlighted in this study; and secondly, as an analysis platform, enabling users to upload their datasets to evaluate the drift detection capabilities in scenarios specific to their operational environments. This functionality is pivotal for validating the adaptability and effectiveness of the proposed drift detection methodology across various manufacturing settings and conditions. The users have the possibility to run different tests as shown in the experiment and compare the drift scores. If the user has some data available and wants to perform drift detection on them he can even do so by uploading them on the platform. We are planning to publish the code on our platform as open source with a business-friendly OSS license in the near future.

5 Conclusion and Future Work

In this paper, we have addressed the critical need for robust drift detection in the manufacturing sector, paving the way for more resilient and adaptable CV systems. By tackling the challenges of unsupervised drift detection, we contribute to the broader discourse on maintaining the accuracy and reliability of AI applications in dynamic environments, ensuring that the manufacturing industry can continue to reap the benefits of technological advancement without compromise. We have demonstrated the capability of drift detection on a small dataset for a particular use case of quality control in manufacturing.

Our work suggests several open questions that might offer promising paths for future work, we aim to broaden the scope of our investigation to encompass a variety of domains beyond CV, thus expanding the applicability and impact of our findings. A key focus of our future work will involve comprehensive research into the development of an advanced drift detection framework. This framework will not only identify instances of data drift but will also possess the capability to categorize the nature of such drifts accurately. By integrating a 'human-in-the-loop' approach, we envision a system where drift detection is not only automated but is also enhanced by human expertise, enabling precise classification of concept drift types. This dual approach promises to refine the detection process further, ensuring more nuanced and actionable insights for maintaining the integrity of AI-driven systems in the manufacturing industry and beyond.

References

Breck, E., Polyzotis, N., Roy, S., Whang, S. E. and Zinkevich, M. A. (2019), Data validation for machine learning, *in* 'Conference on Machine Learning and Systems'.

Cobb, O., Looveren, A. V. and Klaise, J. (2021), Sequential multivariate change detection with calibrated and memoryless false detection rates, *in* 'International Conference on Artificial Intelligence and Statistics'.

Dabhi, R., Kantesaria, N., Vaghasia, P., Hirpara, J. and Bhoraniya, R. (2020), 'Casting product image data for quality inspection', *Kaggle Datasets* .
URL: *https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product*

dos Reis, D. M., Flach, P., Matwin, S. and Batista, G. (2016), Fast unsupervised online drift detection using incremental kolmogorov-smirnov test, *in* 'Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '16, Association for Computing Machinery, New York, NY, USA.

Engstrom, L., Tsipras, D., Schmidt, L. and Madry, A. (2017), 'A rotation and a translation suffice: Fooling cnns with simple transformations', *CoRR* **abs/1712.02779**.

Gedraite, E. and Hadad, M. (2011), Investigation on the effect of a gaussian blur in image filtering and segmentation, pp. 393–396.

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

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. and Smola, A. (2012*b*), 'A kernel two-sample test', *Journal of Machine Learning Research* **13**(25), 723–773.

Hegedüs, I., Nyers, L. and Ormándi, R. (2012), 'Detecting concept drift in fully distributed environments', *2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics* pp. 183–188.

Hendrycks, D. and Dietterich, T. G. (2019), 'Benchmarking neural network robustness to common corruptions and perturbations', *CoRR* **abs/1903.12261**.

Jin, M., Rajesh, M. D., Xing, W., Schniertshauer, J., Neuhüttler, J. and Zimmermann, F. (2023), 'Implementation and validation of non-semantic out-of-distribution detection on image data in manufacturing'.

Klaise, J., Looveren, A. V., Cox, C., Vacanti, G. and Coca, A. (2020), 'Monitoring and explainability of models in production', *ArXiv* **abs/2007.06299**.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A. and Sutherland, D. J. (2021), 'Learning deep kernels for non-parametric two-sample tests'.

Liu, Y. and Zhou, Y. (2014), Online detection of concept drift in visual tracking, *in* 'International Conference on Neural Information Processing'.

Maaradji, A., Dumas, M., Rosa, M. L. and Ostovar, A. (2015), Fast and accurate business process drift detection, *in* 'International Conference on Business Process Management'.

Maaradji, A., Dumas, M., Rosa, M. L. and Ostovar, A. (2017), 'Detecting sudden and gradual drifts in business processes from execution traces', *IEEE Transactions on Knowledge and Data Engineering* **29**, 2140–2154.

Munirathinam, S. (2020), Drift detection analytics for iot sensors, *in* 'IEEE International Symposium on Multimedia'.

Omri, S., Dharmil, M. and Jens, N. (2023), 'Leveraging computer vision for sustainable manufacturing: Potentials, challenges and future perspectives', *AHFE (2023) International Conference, USA* **abs/2007.06299**.

Rabanser, S., Günnemann, S. and Lipton, Z. C. (2019), 'Failing loudly: An empirical study of methods for detecting dataset shift'.

Recht, B., Roelofs, R., Schmidt, L. and Shankar, V. (2019), 'Do imagenet classifiers generalize to imagenet?', *CoRR* **abs/1902.10811**.

Sun, Z., Tang, J., fei Qiao, J. and Cui, C. (2020), 'Review of concept drift detection method for industrial process modeling', *2020 39th Chinese Control Conference (CCC)* pp. 5754–5759.

Suprem, A., Arulraj, J., Pu, C. and Ferreira, J. E. (2020), 'Odin', *Proceedings of the VLDB Endowment* **13**, 2453 – 2465.

Tan, M. and Le, Q. V. (2019), 'Efficientnet: Rethinking model scaling for convolutional neural networks', *CoRR* **abs/1905.11946**.

Yeshchenko, A., Ciccio, C. D., Mendling, J. and Polyvyanyy, A. (2019), Comprehensive process drift analysis with the visual drift detection tool, *in* 'ER Forum/Posters/Demos'.

Yingying Sun, Jusheng Mi, C. J. (2024), 'Entropy based concept drift detection in information systems', *Knowledge-Based Systems* **290**.