VMLC: Statistical Process Control for Image Classification in Manufacturing

Philipp Mascha PHILIPP.MASCHA@AMS-OSRAM.COM
Manufacturing IT and Automation,
Osram Automotive,
Schwabmünchen, Germany

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

Through ground-breaking advances in Machine Learning its real-world applications have become commonplace in many areas over the past decade. Deep and complex models are able to solve difficult tasks with super-human precision. But for manufacturing quality control, in theory a ideal match for these methods, the step from proof-of-concept towards live deployment is often not feasible. One major obstacle is the unreliability of Machine Learning predictions when confronted with data diverging from the known characteristics. While overall accuracy is high, wrong results may be returned with no indication of their uncertainty. In manufacturing, where scarce errors mean great damages, additional safety measures are required.

In this work, I present Visual Machine Learning Control (VMLC), an approach developed upon a real world visual quality control system that operates in a high throughput manufacturing line. Instead of applying sole classification or anomaly detection, both is done in combination. A scalar metric derived from an Auto-Encoder reconstruction error measures the compliance of captured images with the training data the system is trained on.

This metric is integrated into the widely used framework of industrial Statistical Process Control, significantly increasing robustness through meaningful control limits and enabling active learning. The system is evaluated on a large dataset of real-world industrial welding images.

Keywords: Computer vision, Manufacturing, Monitoring, Robustness

1. Introduction

In the past decade, Machine Learning (ML) has made revolutionary advances in the fields of computer vision, natural language processing, data mining and many more. Many applications found their way into the consumer market, be it trough adaptive image filters, speech-to-text or translation and a new wave of generative tools like ChatGPT or DALL-E open a whole new world of possibilities. Despite all this, manufacturing still struggles to develop ML assisted tools past the state of proof-of-concept.

Potential is plenty: Many quality controls involve visual checks, often conducted by humans. These are costly, especially in high throughput manufacturing, often rely on individual subjective bias and are both stressful and monotonous for the inspector. Modern classification systems with the promise of super-human performance and real-time error

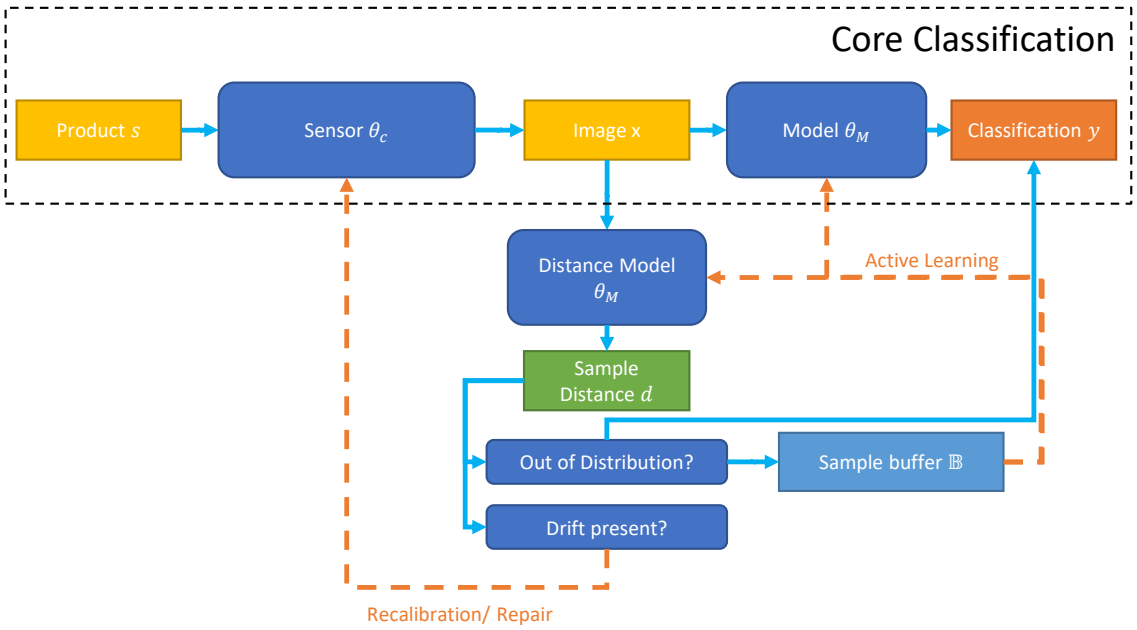


Figure 1: Core classification system with added process control.

detection capabilities are a good fit for these tasks, especially as huge amounts of visual data seems easily obtainable. Many implementations exist, for example in the fields of opto-semiconductor Lin et al. (2019), pharmaceutical Unnikrishnan et al. (2019) or electronics Tabernik et al. (2020) production.

But the inherent nature of manufacturing creates obstacles often overlooked by outsiders and academic research:

- The best performing ML systems are often based on a single training run and static in nature. This collides with the dynamic environment of manufacturing, where changes in process, materials or equipment as well as product re-designs may invalidate the deployed model.
- Furthermore, the captured data may drift due to perturbations like mechanical variance, degradation of the light source or dust and other contamination.
- Online learning methods like Reinforcement Learning Arulkumaran et al. (2017) are not applicable since the required exploration actions are not feasible for production.
- Industrial data sets often differ immensely in a few key aspects: Variance between images is very low, class imbalance is extreme (pictures of defects are very scarce or must be produced by hand, while data for faultless products exists by the thousands) and there is a huge risk potential defect classes are not covered by the training data.
- While modern systems yield unprecedented accuracy, when they miss-classify, they often do so with high confidence Carlini and Wagner (2017). This problem has become

more apparent with generative language models that produce false 'facts' or so-called hallucinations Ji et al. (2023). There exists no widely accepted measurement of ML system per-sample reliability, especially not in the visual domain Tran et al. (2022).

This often prevents making the step towards productive deployment. In peculiar high quality manufacturing like the automotive industry where every undetected defect is a huge risk both financially due to possible large-scale recalls and to customer trust. On top, integration in established quality processes like LEAN manufacturing or Statistical Process Control (SPC) is required as well as establishing trust into these unconventional systems by operators, leadership and customers.

In this paper, I will present a system that accommodates all these requirements and peculiarities. It was developed over two years for a ML based quality system currently used in live-production on a high throughput manufacturing line. Furthermore, I will demonstrate its stability trough evaluation on data that contains both Out of Distribution (OOD) samples as well as artificially induced data drift.

2. Related Works

The approach presented in this work aspires to solve the presented challenges by implementing a monitoring metric enabling SPC by detecting both drift and OOD samples

2.1. SPC and Machine Learning

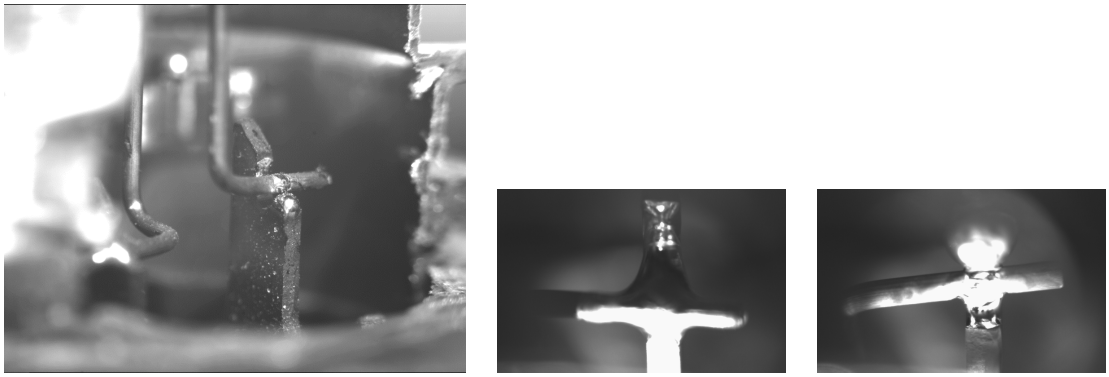
SPC measures process stability and defines control limits when intervention into a process is needed. It requires reliable and measurable metrics that can be evaluated statistically to determine whether a machine, sensor or process is working as expected. Research on this specific intersection is scarce, although the newer paradigms of MLOps work towards this direction Granlund et al. (2021), although with a focus on tooling, deployment and process.

There exists a huge research gap for representing the high dimensionality of images for SPC, especially with respect to ML, as is noted in Tran et al. (2022). A historical gold standard for SPC on image data does not exist Megahed et al. (2011). Newer research suggests methods for monitoring image data based on regions of interest Okhrin et al. (2021) or generating control charts for 3D point clouds Stankus and Castillo-Villar (2019).

2.2. Drift and OOD Detection

Quantifying the dissimilarity between initial training data and samples measured during deployment may fall into the domain of data drift detection Barros and Santos (2018), where measured data diverges from the initial training data. Or it may tackle the problem of OOD or anomaly samples Chalapathy and Chawla (2019). Both have a similar goal: Identifying when the model is extrapolating (or guessing) rather then interpolating from its known data.

For streaming data many approaches exist, *e.g.*based on the Kolmogorov-Smirnov test dos Reis et al. (2016), p-Values Jordaney et al. (2017) or MD3 Sethi and Kantardzic (2017). Their simplicity may be sufficient for a low dimensional time series but are not applicable to the high abstraction and amount of variables contained in images.



(a) (b) (c)

Figure 2: Picture of the welding point (a) as well as examples for a good (b) and defect (c) sample.

Specific, but not exclusive, to the the visual domain encoder decoder based approaches like Auto-Encoders (AEs)Zhou and Paffenroth (2017) or Principal Component Analysis (PCA)Li et al. (2014) promise good results for measuring dissimilarities. These approaches train an unsupervised model $dec(enc(x))$ on the training data with a loss function based on $\mathcal{L}_A E(x) := \mathrm{E}(dec(enc(x)), x)$. Thereby, the dimensionality of $enc(x)$ (often called the latent space) is significantly smaller than of the input x, achieving efficient encoding of the input data. The resulting metric is either based on the reconstruction error Lübbering et al. (2021) the distribution of the latent space Abati et al. (2019); Aytekin et al. (2018) or comparing with known images that have a similar latent distribution Gong et al. (2019).

There also exist approaches that supervise abstract representations produced by the classification model itself Lee et al. (2018); Papernot and McDaniel (2018); Jiang et al. (2018) or regularize the model to produce an output that better reflects uncertainty Du et al. (2022). A comparison of selected methods exists in Mascha (2023), proving the advantages of AE based approaches on industrial data.

3. Motivation and System Description

This research was primarily conducted on a camera system used in a high throughput manufacturing environment. Its purpose is to verify the quality of laser welding conducted between an electrode and its contact fin by detecting weld beads as well as disconnected or malformed components. This visual system is a retro-fit for an existing manufacturing more than 10 years of age. Due to these circumstance, image quality varies greatly and lighting is not always optimal.

Initial implementation used classical computer vision tools like edge detection and feature matching. But due to the high variance induced by light reflexes on the shiny surface and complex defect patterns performance was lacking, resulting in a high false-positive rate as well as undetected defects.

Detection was greatly improved by replacing the old, classical system with a 50 layer deep ResNet He et al. (2016). It was trained with approximately 14700 images labeled by a process engineer, with only around 4% of the samples containing defect). After going live in production, problems arose due to images that were not classified correctly. These were either defects that were not represented in the training set, like two weld beads being present at once. Or systematic drift was introduced into the system, for example a shift of the image center caused by mechanical variations.

This created the need for a system that could:

- Detect anomalies that are not represented in the training data, so called OOD samples.
- Measure the general drift between the data gathered during production and the one used for training.
- Indicate how reliable the current classification of the model is to prevent undetected defects to be shipped to the customer.
- Identify samples beneficial for further classifier training from the huge amount of available data.

The last point is especially inherent to manufacturing data where most images have low variance between each other. Additionally a scalar quantification of each image would be desirable to integrate the system into standard SPC tools required in modern Lean-based manufacturing Womack et al. (1990). These encompass a so called control chart, where an operator can easily see whether the process is still operating as expected or a control limit is breached, requiring manual intervention.

4. Solution

To accommodate all these requirements, I present VMLC, illustrated in figure 1. On top of the usual classification through the classification model $M(x;\theta_M)$ a control system is placed. It consists of a distance model $M_d(x;\theta_d)$ that maps the image x to a scalar value d. A low value of d represents high confidence of M that the sample is similar to those used for its training, thus implying correct classification through generalization.

When the value of d is too high to be confidently labeled, the output of M is invalidated to prevent a potential mislabeling. Furthermore, x is added to a sample buffer $\mathbb{B}$ to be labeled and used for future retraining in an Active Learning Ren et al. (2021) manner. Additionally, the overall distribution of d over the last samples is compared with the expected value to detect whether the process is still in control. When diverging too much, a control action is triggered to remove the cause of the deviation.

4.1. Distance Metric

To generate a scalar control metric on a process involving visual data controllable its high dimensionality has to be reduced. To do so, there exist a multitude of approaches which usually fall into two classes: Either model centric approaches that work on the output of the classification model $M(x;\theta_M)$ or data centric that compare the distribution of input data between the training samples $\mathbb{S}^T$ and the images taken during production $\mathbb{S}^P$.

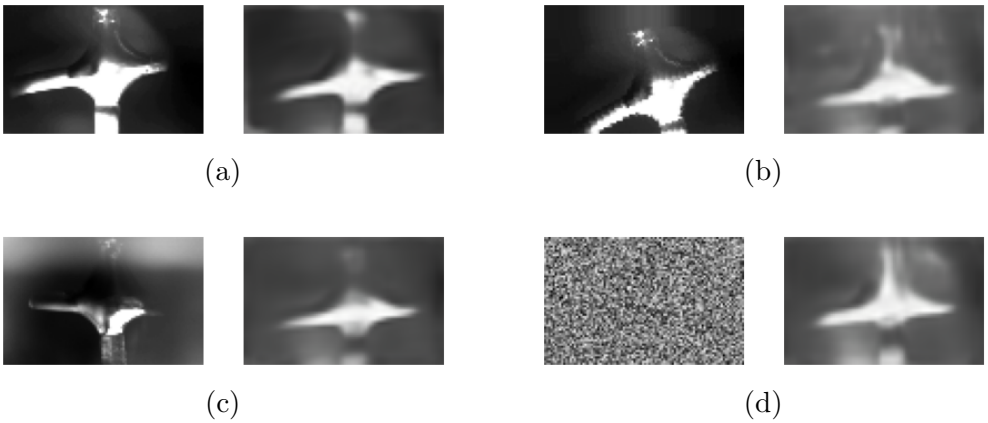


Figure 3: Reconstruction of an (a) similar, (b) transformed, (c) OOD sample as well as (d) random noise.

In this work the distance metric is specified as the reconstruction error of an AE. Due to the massive reduction in information, AEs normally only reconstruct data that is similar to their set (see figure 3). Even samples that look similar but stem from another setup (the OOD sample) are not reconstructed properly. This can be exploited by using the reconstruction error E as metric $M_d(s)$ for the dissimilarity between a sample s and the training set $\mathbb{S}^T$.

To incentive the encoder to learn the same representation as the classifier, the layers of the classification model before pooling are used as the encoder part of the AE. This is similar to approaches like Zhu and Zhang (2019), although here the Auto-Encoder is based on the classifier and not the other way around. After training M on the labeled dataset, its parameters θ_M are frozen and θ_d is learned on the same data. In evaluation this provided no improvements over using a standalone AE but decreased training time and memory consumption.

Another boost to performance is achieved by applying the Structural Similarity Index (SSIM) instead of Mean Squared Error (MSE) to calculate the reconstruction error, as is suggested by Bergmann et al. (2018). SSIM is a lot more sensitive towards smaller features inside the image since it sums over regions of interest instead of the per-pixel error. This is especially beneficial for industrial data, where small details are often more important than large features.

4.2. General Algorithm

Algorithm 1 describes the complete process for calculating the distance metric and enforcing the control actions Retrain and Repair. Each iteration is one measurement Observe made by the camera, which is parameterized by θ_c.

Algorithm 1: Main algorithmic loop of VMLC.

$\mathbb{B} \leftarrow \emptyset$
$\delta_i \leftarrow 0$
$\mathbb{D} \leftarrow \{x \in \mathbb{S}^T : M_d(x; \theta_d)\}$
$\mu, \sigma \leftarrow \mathcal{N}(\mathbb{D})_{N_s}$
$UCL \leftarrow \mu + 3\sigma$
$SL \leftarrow \eta_{t_{ood}}(\mathbb{D})$
for $s_i \in \mathbb{S}^P$ **do**
  $x \leftarrow Observe(s_i, \theta_c)$
  $d \leftarrow M_d(x; \theta_d)$
  /* Check if OOD sample is discovered. */
  if $d > SL$ **then**
    $y \leftarrow Oracle(x)$
    $\mathbb{B} \leftarrow \mathbb{B} \cup \{(x, y)\}$
  else
    $y \leftarrow M(x; \theta_M)$
  end
  $\overline{d} \leftarrow \frac{1}{N_s} \sum_{n=0}^{N_s} M_d(s_{i-n}; \theta_d)$
  /* Invoke active learning if required. */
  if $\overline{d} > UCL \wedge \delta_i < |s|_{min} \ \vee \ |\mathbb{B}| = |\mathbb{B}|_{max}$ **then**
    $\mathbb{S}^T \leftarrow \mathbb{S}^T \cup \mathbb{B}$
    $\theta_M \leftarrow Retrain(\theta_M, \mathbb{S}^{\mathbb{T}}, \mathbb{M})$
    $\theta_d \leftarrow Retrain(\theta_d, \mathbb{S}^{\mathbb{T}}, \mathbb{M})$
    $\mathbb{B} \leftarrow \emptyset$
    $\mathbb{D} \leftarrow \{x \in \mathbb{S}^T : M_d(x; \theta_d)\}$
    $\mu, \sigma \leftarrow \mathcal{N}(\mathbb{D})_{N_s}$
    $UCL \leftarrow \mu + 3\sigma$
    $SL \leftarrow \eta_{t_{ood}}(\mathbb{D})$
  end
  /* Check if repair is needed. */
  if $\overline{d} > UCL \ \wedge \ i \geq |s|_{min}$ **then**
    $\theta_c \leftarrow Repair()$
    $\delta_i \leftarrow 0$
  end
  $\delta_i \leftarrow \delta_i + 1$
  return (x, y)
end

4.2.1. Process Control

The common approach towards SPC imposes control limits on the control metric d. These are created by calculating mean μ and variance σ^2 of a normal distribution $\mathcal{N}$ measured over known in-control samples ($\mathbb{S}^T$ in this case) by applying the Central Limit Theorem (CLT) on batches of N_s data points. These so-called lower control limit (LCL) and upper control limit (UCL) are calculated as multiples of σ from the mean. A distance of 3σ is common. Since this implementation is based on a distance metric only an UCL is specified.

As soon as a limit is breached investigation into the process is requested. This either implies a correction of the process itself, for example by repairing the preceding machine, or necessitates re-calibration of the sensor. Since these actions often require to stop the involved machine and therefore halt production, it is desirable to keep such interventions to a minimum.

4.2.2. Specification Limit and Active Learning

Another type of control action may occur when singular measurements are so far from the expected values that they suggest some kind of error, be it a faulty measurement, unknown defects or an unknown product. Products with such measurements should be considered in breach of specification and not to be processed further. In ML terms, we speak of an OOD sample or anomaly.

In VMLC this is represented by the Specification Limit (SL). It is specified by the t_{ood}-th percentile η over all measured samples $\mathbb{D}$. Any rejected sample is added to the sample buffer $\mathbb{B}$ to be used for investigation or improvement of the system or process. In case of a ML system, these improvements often require a retraining of the model: Either training data is not yet able to sufficiently cover the amount of variance present in the application. Or new variance is introduced into the system, for example by a product re-design, change in pre-material or overall process.

The first case usually arises in the early stages of deployment where the amount of images taken during production vastly outnumbers the limited training data. Thus, when the OOD-Buffer $\mathbb{B}$ reaches a certain size $|\mathbb{B}|_{max}$ a iterative retraining is triggered. It is expected for the buffer to be filled slower the longer the application runs and the more the training set grows as outliers become more and more uncommon.

The second case may arise at any point in time during deployment. It should imply a steep incline in measured d-value, thus triggering a re-calibration. This might either be recognized by the person responsible for the manual re-calibration, launching the retraining in response. Alternatively, the need for retraining can be detected automatically when many re-calibrations are triggered in a short amount of time. To capture this, a minimum required sample amount $|s|_{min}$ since the last re-calibration δ_i is specified.

The learning itself is conducted by expanding the known set $\mathbb{S}^T$ with the OOD-Buffer $\mathbb{B}$. New samples are labeled by an instance called the oracle, which is usually a domain expert. The combined data is automatically split into a training and evaluation set during training to guarantee good generalization. Those sets are further used as training and calibration set for the distance model $M_d(x;\theta_d)$, which must also be retrained to accommodate the new data supporting $M(x;\theta_M)$.

p_d [0.,0.,0.] [**0.2**,0.,0.] [0.,**0.2**,0.] [0.,0.,**0.2**] [**0.1**,**0.1**,**0.1**]

Figure 4: Images with artificial drift through transformations.

Any retraining might be conducted asynchronously if the model performance is not degraded too much, especially for the iterative case. Since retraining implies a great effort and possible downtime, it should be invoked as little as possible.

It should be noted that other sample selection methods like learn loss Yoo and Kweon (2019) or TypiClust Hacohen et al. (2022) for Active Learning exist. While these may theoretical be better performance wise, they do not provide any comprehensible dependency from the measured metric, which is problematic in terms of system acceptance.

5. Evaluation and Discussion

This evaluation was conducted on the welding data set described in section 3. Each sample consists of an image with a binary label (either 'Defect' or 'Ok') assigned by a domain expert. Images where scaled to a size of 60x94 to accelerate training without measurable impact to performance. The approximately 14700 samples were split into a dedicated training $\mathbb{S}^T$ and test set $\mathbb{S}^P$ for evaluation by a 1:1 ratio. All experiments where conducted and median averaged over a 5-fold cross validation of the training set with a further 4:1 split to provide evaluation $\mathbb{S}^E \subset \mathbb{S}^T$ and training samples $(\mathbb{S}^T \setminus \mathbb{S}^E)$ for training only. Experiments where additionally conducted on MNIST. All samples containing the digits 0, 1 and 2 were removed to use them as OOD samples.

5.1. Implementation Details

For the classification model, a 50-layer ResNet based on He et al. (2016) was used. The decoder consisted of a combination of residual blocks containing convolutional and batch normalization layers Ioffe and Szegedy (2015) alternating with up-sampling layers used in image up-scaling networks Li et al. (2018) to mirror the encoder network. The latent space consisted of 100 dimensions, as this size appeared to be the cut-off point for good results.

Each classifier was trained for 30 and the AE for 40 epochs total. Retraining was done by invoking 10 additional epochs on the classifier, separating training and evaluation data from the sample buffer in the same 4:1 ratio.

5.2. Handling Data Drift

Handling of sensor drift was tested by introducing artificial drift into the dataset. Both shift and rotation where applied to each image according to a distortion vector p_d (see figure 4). Its values where modified at random points in time by adding a random offset. The amount of distortion present was quantified by calculating $\|p_d\|$. When a repair was requested by the respective metric, its values were reset to 0.

	Method	n_r	Accuracy			$\overline{p_d}$
			Averaged	OK	Defect	
Weld	No intervention	0	0.635	0.764	0.537	0.145
	Fixed Interval	14	0.930	0.992	0.869	0.024
	Classifier output	14	**0.962**	**0.997**	**0.926**	**0.005**
	Fixed Interval	**4**	0.849	0.945	0.753	0.069
	VMLC	**4**	**0.962**	**0.997**	0.925	0.006
	Always repair	147	0.965	0.999	0.931	0.003
	Baseline (no drift)	-	0.971	0.999	0.944	0.000
MNIST	No intervention	0	0.762	-	-	0.139
	Fixed Interval	18	0.971	-	-	0.028
	Classifier output	18	**0.993**	-	-	**0.005**
	Fixed Interval	**4**	0.929	-	-	0.055
	VMLC	**4**	0.982	-	-	0.027
	Always repair	138	0.993	-	-	0.003
	Baseline (no drift)	-	0.994	-	-	0.000

Figure 5: Performance when handling artificial drift on $\mathbb{S}^P$.

Method	Welding			MNIST	
	OOD det.	Without Rejected	Retrained	OOD det.	Retrained
No sampling	-	0.796	0.796	-	0.696
Random sampling	0.506	-	0.896	0.315	**0.953**
Classifier output	0.808	0.890	0.918	0.429	0.942
VMLC	**0.973**	**0.938**	**0.926**	**0.674**	**0.953**

Figure 6: Out of Distribution sample detection and class balanced classification accuracy when confronted with OOD data.

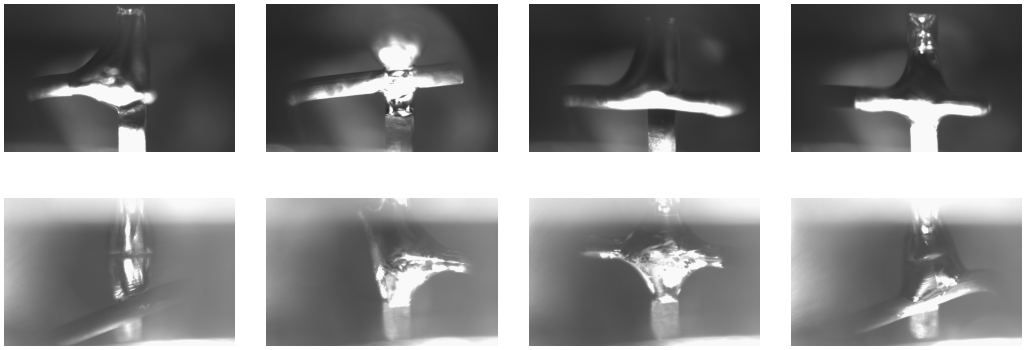


Figure 7: Images taken from two different camera setups.

Table 5 illustrates the change in classifier performance without interventions, with scheduled repairs after a fixed amount of samples and when applying the M_d model used in VMLC. To have a fair comparison, the scheduled interval was set to result in the same amount of repairs n_r required by M_d. Additionally, a uncertainty based drift detector using only the classification model output $1. - max(M)$ was used and an UCL value calculated.

Since the industrial data is highly imbalanced accuracy was class balanced by averaging over both classes. This was done for all subsequent experiments. To speed up training and reflect real life restrictions, samples where grouped into batches of size $N_s = 50$.

The improvement through VMLC is apparent: The system is able to operate close to the theoretical maximum (when repairing after each batch) while keeping repairs to a minimum. While the classification output based approach delivers the same or slightly better performance, it requests three to four times more (potentially unnecessary) interventions on average. For a manufacturing environment these would soon result into a loss of trust from the operators, which are called to action for no apparent reason. Thus, the reconstruction error based method proofs far superior on both data sets.

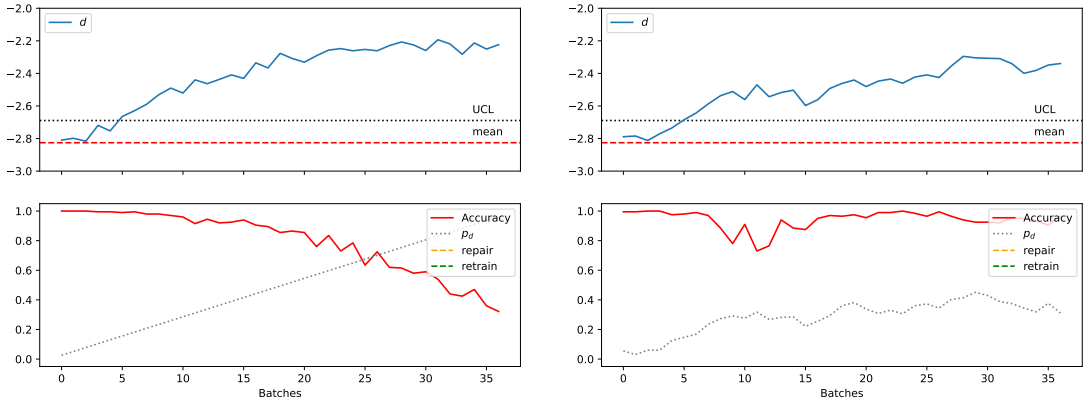
5.3. Active Learning on OOD Samples

To evaluate the self-correcting capabilities through active learning, the methods behavior was tested when confronted with a significant change in data manifestation. To do so, a set $\mathbb{S}^P \cup \mathbb{S}^O$ was used where $\mathbb{S}^O$ contained around 7500 images captured from a second camera setup with different lighting and positioning. These pictures are similar in character but contrast strongly in quality and orientation (see figure 7). For MNIST $\mathbb{S}^O$ was composed of images with the yet unknown class labels 0, 1 and 2.

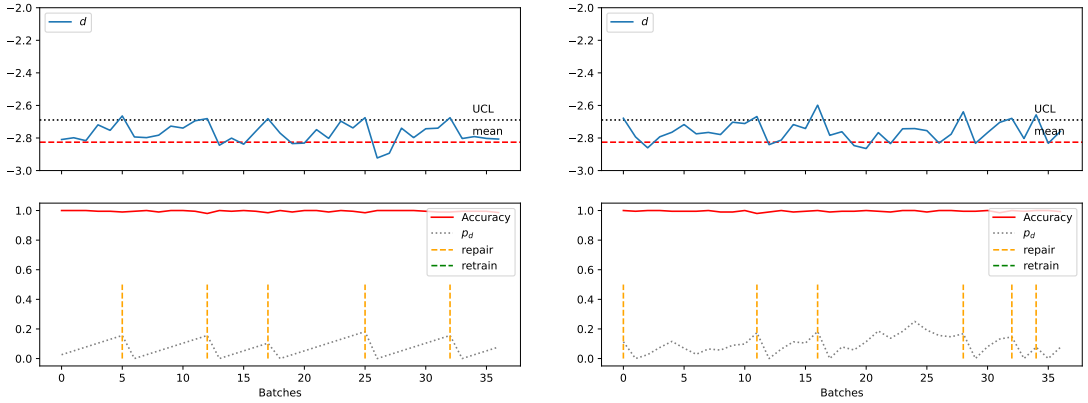
Samples where rejected when $d > \eta_{t_{ood}}(\mathbb{D})$ with $t_{ood} = 0.97$. The first 500 samples rejected where added to $\mathbb{S}^T$ for retraining and performance was measured on the remaining samples $\mathbb{S}^P \cup \mathbb{S}^O \setminus \mathbb{S}^T$ afterwards. Welding data was also evaluated when omitting all rejected samples from the set.

The results displayed in table 6 show clear improvements for the welding data. The AE based approach finds OOD samples more accurately and improves the accuracy of the resulting model. Also, performance is impacted the least when continuing operations under the presence of OOD data. Although selection on MNIST was also better, the small amount

Runs with artificial drift where no control action is taken:



Runs where drift is mitigated after reaching the UCL:



Runs where OOD samples are encountered after a fixed amount of time (black line):

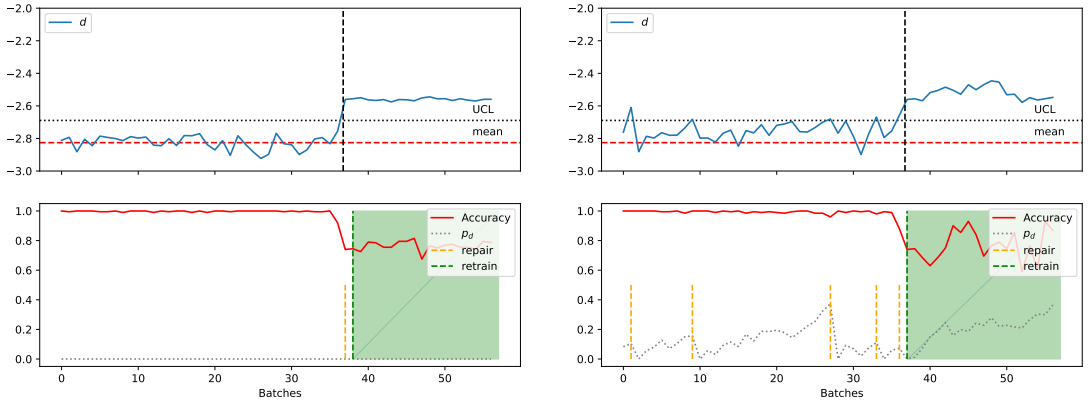


Figure 8: Plots over full runs. The green area indicates that the process is considered out of specification with $|s|_{min} = 1$ and needs fixing through retraining or other means. Samples where batched with $N_s = 200$.

of novelty samples collected at random where already sufficient to train an adequate model for all classes.

5.4. Overall Performance

Figure 8 visualizes the method in a simulated productive environment. The upper plot shows the information an operator would see, while the lower shows the underlying performance and drift metrics. When active, drift is mitigated before performance is decreased while drastic measures like retraining are only requested when necessary.

In general, VMLC is capable to handle all challenges introduced in section 1 without over-reacting to natural variance in the image data. The overall higher performance on welding data compared to MNIST shows that it is better suited towards an industrial environment, but still may be beneficial to other domain.

Its real-world implementation already operates in live production for over a year. Through multiple training iterations it reduced the amount of miss-classifications by multiple orders of magnitude and improved overall trust in the 'alien' and 'unconventional' application of ML.

6. Conclusion

Making a visual classification system viable for production requires more then just a good classifier. Various attributes of visual manufacturing data render a naive approach towards deployment insufficient. In this work, I presented VMLC as an overarching system to tackle these peculiarities. By combining the worlds of ML driven computer vision and industrial SPC through an supervising anomaly detection model, the underlying classification is able to run stable under apparent data drift and OOD data.

Through various tests on real-world industrial data the system proves to be more than capable to handle the requirements imposed by this broad but unique domain. Its similarity with processes well known by the plant staff increases acceptance and makes it simpler to integrate into tools already present. Thus, the described system has the potential to deliver a general and easy-to-implement guideline for productive deployment of visual ML quality controls.

Future research should focus on evaluating VMLC with additional data sets and further classification tasks like semantic segmentation. Additionally, influences on the supervising models capabilities by artificial data augmentation methods should be considered. A comparison with different, independent sample selection methods for active learning or alternative metrics for drift quantification may also further improve overall performance.

References

Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6): 26–38, 2017.

Caglar Aytekin, Xingyang Ni, Francesco Cricri, and Emre Aksu. Clustering and unsupervised anomaly detection with l 2 normalized deep auto-encoder representations. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2018.

Roberto Souto Maior Barros and Silas Garrido T. Carvalho Santos. A large-scale comparison of concept drift detectors. *Information Sciences*, 451-452:348 – 370, 2018. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2018.04.014. URL http://www.sciencedirect.com/science/article/pii/S0020025518302743.

Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

Denis Moreira dos Reis, Peter Flach, Stan Matwin, and Gustavo Batista. 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*, pages 1545–1554, 2016.

Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. VOS: learning what you don't know by virtual outlier synthesis. *CoRR*, abs/2202.01197, 2022. URL https://arxiv.org/abs/2202.01197.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Tuomas Granlund, Vlad Stirbu, and Tommi Mikkonen. Towards regulatory-compliant mlops: Oravizio's journey from a machine learning experiment to a deployed certified medical product. *SN computer Science*, 2(5):342, 2021.

Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. *ArXiv*, abs/2202.02794, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016. URL https://arxiv.org/abs/1603.05027.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

Heinrich Jiang, Been Kim, Melody Y Guan, and Maya R Gupta. To trust or not to trust a classifier. In *NeurIPS*, pages 5546–5557, 2018.

Roberto Jordaney, Kumar Sharad, Santanu K. Dash, Zhi Wang, Davide Papini, Ilia Nouretdinov, and Lorenzo Cavallaro. Transcend: Detecting concept drift in malware classification models. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 625–642, Vancouver, BC, August 2017. USENIX Association. ISBN 978-1-931971-40-9. URL https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/jordaney.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Di Li, Lie-Quan Liang, and Wu-Jie Zhang. Defect inspection and extraction of the mobile phone cover glass based on the principal components analysis. *The International Journal of Advanced Manufacturing Technology*, 73(9):1605–1614, 2014.

Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. Multi-scale residual network for image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Hui Lin, Bin Li, Xinggang Wang, Yufeng Shu, and Shuanglong Niu. Automated defect inspection of led chip using deep convolutional neural network. *Journal of Intelligent Manufacturing*, 30(6):2525–2534, 2019.

Max Lübbering, Michael Gebauer, Rajkumar Ramamurthy, Christian Bauckhage, and Rafet Sifa. Decoupling autoencoders for robust one-vs-rest classification. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, 2021. doi: 10.1109/DSAA53316.2021.9564136.

Philipp Mascha. A comparison of model confidence metrics on visual manufacturing quality data. In Massimo Tistarelli, Shiv Ram Dubey, Satish Kumar Singh, and Xiaoyi Jiang, editors, *Computer Vision and Machine Intelligence*, pages 165–177, Singapore, 2023. Springer Nature Singapore. ISBN 978-981-19-7867-8.

Fadel Megahed, William Woodall, and Jaime Camelio. A review and perspective on control charting with image data. *Journal of Quality Technology*, 43:83–98, 04 2011. doi: 10.1080/00224065.2011.11917848.

Yarema Okhrin, Wolfgang Schmid, and Ivan Semeniuk. New approaches for monitoring image data. *IEEE Transactions on Image Processing*, 30:921–933, 2021. doi: 10.1109/TIP.2020.3039389.

Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning, 2021.

Tegjyot Singh Sethi and Mehmed Kantardzic. On the reliable detection of concept drift from streaming unlabeled data. *Expert Systems with Applications*, 82:77–99, 2017.

Sue E. Stankus and Krystel K. Castillo-Villar. An improved multivariate generalised likelihood ratio control chart for the monitoring of point clouds from 3d laser scanners. *International Journal of Production Research*, 57(8):2344–2355, 2019. doi: 10.1080/00207543.2018.1518600. URL https://doi.org/10.1080/00207543.2018.1518600.

Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skočaj. Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*, 31 (3):759–776, 2020.

Phuong Hanh Tran, Adel Ahmadi Nadi, Thi Hien Nguyen, Kim Duc Tran, and Kim Phuc Tran. *Application of Machine Learning in Statistical Process Control Charts: A Survey and Perspective*, pages 7–42. Springer International Publishing, Cham, 2022. ISBN 978-3-030-83819-5. doi: 10.1007/978-3-030-83819-5_2. URL https://doi.org/10.1007/978-3-030-83819-5_2.

Saritha Unnikrishnan, John Donovan, Russell Macpherson, and David Tormey. Machine learning for automated quality evaluation in pharmaceutical manufacturing of emulsions. *Journal of Pharmaceutical Innovation*, pages 1–12, 2019.

James P Womack, Daniel Roos, and Daniel T Jones. *The machine that changed the world*. Scribner, New York, NY, October 1990.

Donggeun Yoo and In-So Kweon. Learning loss for active learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 93–102, 2019.

Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.

Qiuyu Zhu and Ruixin Zhang. A classification supervised auto-encoder based on predefined evenly-distributed class centroids. *arXiv preprint arXiv:1902.00220*, 2019.