Evaluating Vision Transformer Models for Visual Quality Control in Industrial Manufacturing

Miriam Alber[1,3] (✉), Christoph Hönes[2], and Patrick Baier[3]

[1] iteratec GmbH Karlsruhe, Germany miriam.alber@iteratec.com
[2] Hasso Plattner Institute / University of Potsdam, Germany
[3] University of Applied Sciences Karlsruhe, Germany

Abstract. One of the most promising use-cases for machine learning in industrial manufacturing is the early detection of defective products using a quality control system. Such a system can save costs and reduces human errors due to the monotonous nature of visual inspections. Today, a rich body of research exists which employs machine learning methods to identify rare defective products in unbalanced visual quality control datasets. These methods typically rely on two components: A visual backbone to capture the features of the input image and an anomaly detection algorithm that decides if these features are within an expected distribution. With the rise of transformer architecture as visual backbones of choice, there exists now a great variety of different combinations of these two components, ranging all along the trade-off between detection quality and inference time. Facing this variety, practitioners in the field often have to spend a considerable amount of time on researching the right combination for their use-case at hand. Our contribution is to help practitioners with this choice by reviewing and evaluating current vision transformer models together with anomaly detection methods. For this, we chose SotA models of both disciplines, combined them and evaluated them towards the goal of having small, fast and efficient anomaly detection models suitable for industrial manufacturing. We evaluated the results of our experiments on the well-known MVTecAD and BTAD datasets. Moreover, we give guidelines for choosing a suitable model architecture for a quality control system in practice, considering given use-case and hardware constraints.

Keywords: Vision Transformer · Industrial Quality Control · Anomaly Detection

1 Introduction

In industrial manufacturing, early detection of defective products saves material and costs and enhances public trust in the manufacturer. Automating this process increases scalability, saves labour costs and reduces human error due to the monotonous nature of visual inspections [27]. To this end, the possibility of automating this process using machine learning methods has been subject of extensive research [24]. Anomaly detection (AD) in machine learning addresses

the challenge of identifying rare defective products in unbalanced datasets, often utilizing unsupervised or semi-supervised training strategies. While AD refers to image-level classification of samples as either normal or anomalous, anomaly localization (AL) aims to identify anomalies on a more fine-grained level and indicates where the anomalous feature was detected within the image. This provides interpretability of the model's decisions, facilitating human-in-the-loop control by allowing focus on the anomalous regions. Unsupervised AD architectures typically consist of an image encoding backbone and a detection algorithm that identifies if the extracted features are within an expected distribution. Since the publication of *ViT* in 2020 [9], vision transformers have emerged as an alternative to traditional CNN backbones, offering enhanced global dependency capture, making them particularly interesting for AD tasks. Emerging hierarchical transformer architectures promise to solve the problem of their excessive size by keeping their advantages [26]. Many approaches to solve AD problems in existing literature consider monolithic vision transformer but not hierarchical ones [21,31]. We believe hierarchical vision transformer models can be a great benefit for industrial visual quality control regarding their computational and memory demands. Moreover, we want to help practitioners with their choice of the best setup by reviewing and evaluating current vision transformer models together with AD methods. Our contributions can be summarized as follows:

1. First, we provide a comprehensive overview of current state-of-the-art (SotA) hierarchical vision transformer models and approaches for the task of visual quality control.
2. Second, we reproduce two of the most promising AD methods and combine them with different visual backbones to find small, fast and efficient AD models, suitable for industrial manufacturing, and evaluate our results on the well-known MVTecAD and BTAD datasets.
3. At last, we recommend guidelines for choosing a suitable architecture for deploying a quality control system in practice considering given use-case and hardware constraints.

The rest of this paper is structured as follows: In the next section, we review existing work regarding visual backbones, AD and AL. Section 3 describes the setup we chose for our experiments, which are then discussed and analysed in Section 4. Our source code is available on GitHub (`https://github.com/Miwri/vit-ad`).

2 Related Work

2.1 Vision Backbone

Vision transformers have become a powerful alternative to CNNs for various computer vision tasks, with many studies highlighting their proficiency in capturing global dependencies, which are essential for AD and AL workloads [20,30,6,21,31]. Monolithical vision transformers such as *ViT* follow the architecture of NLP transformers, offering performance competitive with CNNs like *ResNet* [12] but

demand more training data, memory, and longer inference times. By applying an enhanced training strategy using a teacher model and a distillation token to match it's output, Touvron et al. [25] published *DeiT*, an improved monolithic transformer that is smaller than *ViT*, demands less training data and has a superior performance. However, in light of the limited computational resources in production settings, the light-weight class of hierarchical vision transformers gained interest in recent work [26,28,19,16,33]. The main idea is to address the poor scaling of monolithic transformers to high-resolution images by reducing the image size across layers. This leads to models that need less time and resources for training and inference but perform equal to existing vision transformer models [26]. *HaloNet* [26] is one of the first approaches of using size-reducing layers in combination with attention in the encoder, inspired by the architecture used in *ResNet* [12]. Wang et al. introduced *PVT*, a model with an attention backbone suitable not only for classification but also for dense prediction tasks such as object detection and segmentation [28]. Zhang et al. [33] propose with *NesT* a lightweight yet not overly complex architecture by applying the *ViT* architecture [9] on distinct subsets (blocks) of patches and subsequently reducing groups of four neighbouring blocks into one. Liu et al. developed a similar approach to *PVT* but highlight, that their model has linear instead of quadratic complexity when it comes to scaling with image size [19]. Their *SwinTransformer* model alternates between a window-based and shifted window-based self-attention, which computes attention locally on a fixed number of patches within non-overlapping windows. To enable cross-window connections, non-overlapping neighboring windows from the previous layer are included in the calculations of each block. For dimension reduction, a patch merging layer is added before each stage, which itself can consist of two or more transformer blocks. Li et al. [16] highlight the issue of multi-stage vision transformers with sparse self-attention failing to detect fine-grained inter-region dependencies. They introduce a label-free knowledge distillation strategy, using altered image views for training. Their approach, dubbed *EsViT*, combines view and region-level prediction losses. Their training procedure can be applied to various transformer architectures and pre-trained weights are available for [19,28] and [25]. Li et al. [18] introduced the *EfficientFormer*, an approach that is comparable in inference speed with lightweight CNN implementations such as *MobileNetV2* [23] and thus can be used in edge applications. To achieve this, they observed the main bottlenecks (e.g linear projection layers for patch embeddings, reshape operations) in the existing vision transformers and tried to improve efficiency with a few architecture modifications.

2.2 Anomaly Detection and Localization

There are several different categories of approaches for implementing AD and AL in practice [24]. Reconstruction-based methods use an auto-encoder with the objective to compress the input to a lower dimensional latent space and subsequently reconstruct the original image, using the reconstruction error as anomaly score. They tend to deliver weaker results than other categories [30]. Self-supervised learning methods try to use synthetic data to train a model,

which requires costly data augmentation strategies that often heavily depend on expert knowledge. In contrast, promising representation-based approaches try to model the distribution of normal features and classify defects as samples that are in regions with low probability density or outside of the modeled distribution [24]. Roth et al. [22] achieved excellent detection scores with *PatchCore* by using a memory bank and measuring the distance of extracted features to the nearest feature in the bank. Bae et al. [2] enhance *PatchCore* with position and neighborhood information to improve the detection of global anomalies. Hyun et al. introduce *ReConPatch*, that uses multiple levels of a CNN encoder as base to create patch-level features, also enhanced by neighborhood information [13]. Li et al. introduce *SemiREST*, a model that uses supervised and semi-supervised learning in combination with the *SwinTransformer* architecture [17].

Gaussian Mixture Models (GMM) are based on the assumption, that the underlying distribution of normal features is more complex than a single Gaussian distribution and thus learns a set of distributions [5]. At inference time, a GMM estimates the probability that a feature belongs to the set of learned distributions [21]. Zong et al. [34] applied an auto-encoder model for AD on non-image data, integrating a GMM for both dense prediction and regularization to avoid local minima in training. Zhang et al. [32] developed a three-component model for AD in high-resolution images, featuring class-specific training and evaluation. This model includes a patch embedding creator, a GMM for density prediction, and a multi-layer perceptron (MLP) for location prediction, jointly trained end-to-end. This approach yields a smaller model compared to pre-trained alternatives. Mishra et al. [21] and Choi et al. [6] advanced auto-encoder networks for AD by integrating them with a GMM and using a vision transformer encoder, respectively. Mishra et al.'s *VT-ADL* model involves end-to-end training, feeding encoder output patch embeddings to a GMM for Gaussian calculations and a CNN decoder for reconstruction. They use mean squared error, log-likelihood and structural similarity (SSIM) losses as training objective and for the anomaly map generation. Choi et al. [6] adopt a similar methodology but employ a variational auto-encoder and a pre-trained transformer, emphasizing a workflow from image collection to expert validation. Both approaches underscore transformers' superiority in capturing global dependencies over CNNs. Fan et al. [10] integrate GMMs with variational auto-encoders for pinpointing anomalies in surveillance video streams, employing a model that samples from multiple distributions to accommodate complex feature spaces.

Normalizing Flow models (NF) are located in two categories, generative models and representation based approaches and are an efficient alternative to GMMs for estimating complex distributions. They have seen increasing popularity for AD due to their fast inference and strong downstream performance. They estimate the exact likelihood of features by following any arbitrary distribution and computing the likelihood by using the KL divergence between a prior and a base distribution [11,8]. Gudovskiy et al. [11] introduce *CFLOW-AD*, an AD model pairing a pre-trained *ResNet* encoder with a conditional NF, training independently for each class with fixed position embeddings. Yu et al. [31] critique fixed

position embeddings for complex datasets, introducing *FastFlow*, an AD model with a 2D NF that eliminates the need for positional encoding by preserving spatial structures. *FastFlow* uses a *ResNet* or a vision transformer encoder with selective embedding stages. The authors report AUROC scores on, among others, the MVTecAD and BTAD dataset, achieving speed gains over the previous models in [11] and [22]. Lei et al. [15] improve results with pyramid NFs and end-to-end trained 1×1 CNNs, while Kim et al. [14] enhance *FastFlow* and *CFLOW-AD* stability and performance with a novel training approach.

3 Experimental Setup

3.1 Backbone Architectures

Because of their performance advantages, our analysis primarily focuses on transformer-based architectures. Given the importance of hardware efficiency in practical applications, the lightweight and efficient hierarchical transformers are of particular interest in our study. For a comprehensive perspective, we also evaluate a classical *ResNet-50* model as a baseline. We chose this *ResNet* variant since it is a common choice when making a trade-off between the quality of representations and computational cost [11,31]. The *DeiT* architecture was chosen as an example of a monolithic vision transformer that closely follows the architecture of the original *ViT* proposed in [9], but achieves better performance by using knowledge distillation [25]. We used the largest variant *DeiT-base* in our experiments since it has the highest performance and was also used in [31]. Moreover, we selected the *EsViT* approach to investigate the performance of hierarchical transformer versions. We used the variant based on the *Swin-T* architecture since it has a relatively small number of parameters which is comparable to *ResNet-50*. A window size of 14 achieved the best results in [16], hence we adopted this choice for our experiments. *EfficientFormer* is another efficient transformer suitable for settings with limited computational resources. To have a comparable parameter size to *ResNet-50* and make a good trade-off between efficiency and performance, we picked the medium-sized model variant *L3*. A comparison of the number of parameters of the different image encoder models is shown in Table 1.

	Backbone			GMM	NF
	Pa	PE	FM	Pa	Pa
ResNet **50**	28M	[512,1024,2048]	[28,14,7]	525M	115M
DeiT **B**	87M	768	14	118M	31M
EsViT **T**	27M	768	7	118M	31M
EffFormer **L3**	31M	512	7	53M	14M

Table 1: Model configurations with parameter sizes (Pa) in million parameters, patch embedding size (PE) and size of the embedding feature maps (FM). The sizes of GMM and NF are dependent on the image backbone.

3.2 Anomaly Detection Architectures

Based on the feature vector produced by the vision backbone a classifier needs to distinguish normal examples from defective anomalies. We focused our experiments on GMMs, which were one of the first approaches using a vision transformer backbone and NFs, which showed promising performance in AD tasks (as discussed in Section 2.2).

The performance of a GMM usually improves with a higher number of Gaussians but the use of a fully-connected MLP for every component can make them prohibitively expensive. Figure 1 illustrates how we used image-processing backbones together with a GMM to perform AD in the case of *DeiT*. The procedure for the other transformer backbones is analogous with differing numbers for the embedding size. With *ResNet*, we followed the approach in [11] and trained two GMMs for the output of the last two blocks of *ResNet-50* to capture global and local dependencies. Based on the patch embeddings we trained three MLPs rep-

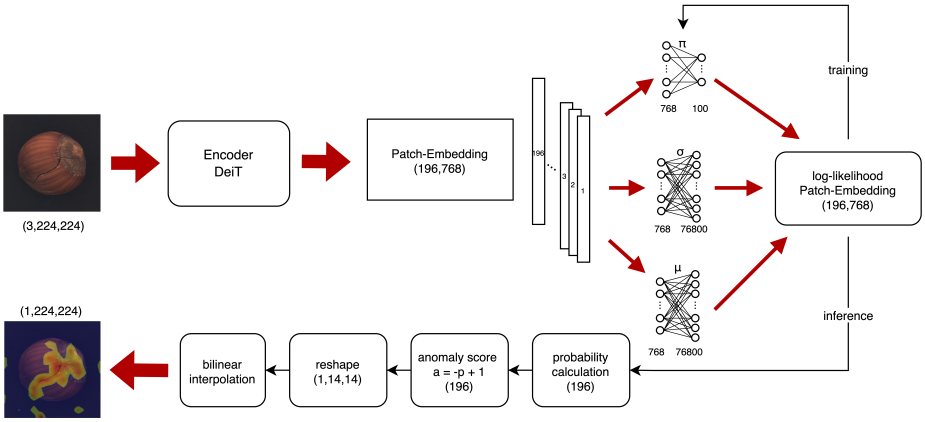

Fig. 1: Gaussian Mixture Model in combination with *DeiT* encoder.

resenting the means (μ), standard deviations (σ) and weights (π) of the Gaussian Mixture components to capture the distribution of the normal examples. The size of the MLPs depends on the number of Gaussians. During training we minimized the negative log-likelihood, while for inference we calculated an anomaly score for every patch as follows: The log-likelihood is normalized to a value between zero and one using min-max normalization over the batch to obtain a pseudo-probability p. Our anomaly score is defined as $a = 1 - p$. To obtain a 2D anomaly map we reshaped the 196 patches to a 14×14 matrix and used bilinear interpolation to project the anomaly map to our original image size of 224×224. An image is deemed anomalous if its highest patch anomaly score surpasses a specific threshold, which is empirically determined using the validation set, as suggested by You et al. [30].

Normalizing Flows were chosen as models in view of limited compute. Our setup of a NF model combined with a *DeiT* image-processing backbone is visualized in Figure 2. The setup is analogous for other transformer backbones. For *ResNet*, we used the output of the last three blocks and averaged the results, similar to the GMM approach. The patch embeddings produced by the vision

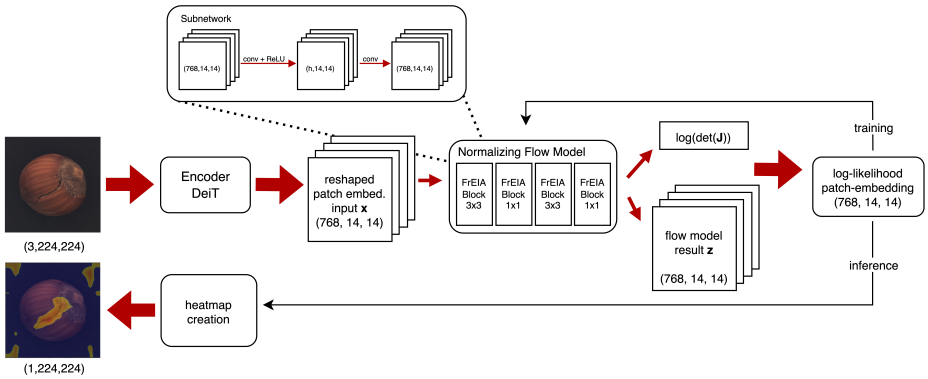

Fig. 2: NF model with the *DeiT* encoder. Each *FrEIA* Block [1] has a subnetwork with hidden dimension h, calculated with the hidden ratio from table 2.

backbone are passed through the NF model consisting of 20 flow steps for the transformer-based methods and eight for *ResNet-50*, due to hardware limitations. Further, we chose a hidden ratio $h = 0.16$. Each flow step consists of a subnetwork of alternating 3×3 and 1×1 convolutional layers. These hyperparameters follow the recommendations of [31]. For the other hyperparameters see Table 2. The flow steps were realized by *AllInOneBlock*s of the *FrEIA* framework [1]. The NF model outputs the log determinant of the Jacobian and the transformed patches z. During training those two outputs are used for the loss objective. At inference, $\log|\det \boldsymbol{J}|$ and z are used to calculate the likelihood which is then used as anomaly map for localization, by upsampling the patches with bilinear interpolation. For more details on the NF architecture see [31].

3.3 Datasets

We evaluate our methods on two datasets that are as close to real-life as possible.

The BTAD dataset consists of three different types of real-world industrial products and has a total of 2830 RGB images with a resolution of between 600×600 and 1600×1600 [21]. Each product category is split into a train set, consisting of only normal images and a test set, which includes normal and anomalous images. In total there are 1799 training images, the rest are test images. The proportion of anomalous images in the test set varies from 9% to 87% depending on the class. A ground-truth mask for each anomalous image is

provided that can be used for AL. There is no further distinction between body and surface defect anomaly classes in the labels.

The MVTecAD dataset consists of 15 different product categories or textures and has a total of 5354 images [3]. Three classes are provided as grayscale images which is common in some real-life settings. The other classes are RGB images. The resolution varies between 700×700 and 1024×1024 and some classes have more examples than others. Each product category is split into a training set with only normal instances and a test set that also contains anomalies. In total there are 3629 training and 1725 test examples. The proportion of abnormal images is, in most cases, about two to three times higher than the normal ones in the test set. The abnormal data is classified into different types of anomalies, for example in crack, hole, cut and print on the *hazelnut* class. A ground-truth mask on pixel-level is given for each abnormal image.

3.4 Implementation Details

Since both datasets do not provide a separate validation set for hyperparameter tuning and model selection we split the training set into 80% train and 20% validation data in both cases. We applied Min-Max scaling to every image. The minimum and maximum values were computed separately for every channel on the training set. We scaled the input images to an image size of 224×224 pixels which is the default training size of most transformer models [9].

Hyperparameters were optimized once for each model using the *hazelnut* class of the MVTecAD dataset and the best configuration according to validation loss is adopted for all other experiments. We chose this class because it has the largest amount of training samples in the dataset. A summary of the considered hyperparameters and the best values found for them are shown in Table 2. Note that the batch size of the GMM is relatively small due to hardware limitations. We trained each model separately on every object class for a maximum of 500 epochs using early stopping based on the validation loss with a patience of 30 epochs. Only the parameters of the AD models were updated, the image-backbones were pre-trained on ImageNet1k [7] and kept frozen during training. For evaluation the model checkpoint with the best validation loss was selected. All models were trained on a single NVIDIA GeForce RTX 2080 Ti TURBO GPU with 11GB VRAM.

AD TYPE	ENCODER	BATCH SIZE	LEARN RATE	WEIGHT DECAY	NUMBER OF GAUSSIANS	FLOW STEPS	HIDDEN RATIO
GMM	ResNet50	4	1e-4	1e-4	50	-	-
	Other	8	1e-4	1e-4	100	-	-
NF	ResNet50	16	1e-4	1e-5	-	8	0.16
	EsViT	32	1e-4	1e-5	-	20	0.16
	Other	32	1e-3	1e-5	-	20	0.16

Table 2: Hyperparameter configuration of the different models architectures.

3.5 Metrics

For the evaluation of our approach, we use the AUROC since it is a common metric for visual quality control [31,2,22,13]. For evaluating AD we consider the AUROC on image and for AL on pixel level.

As a second metric we use the Per-Region-Overlap (PRO), which measures the overlap between ground truth and predicted anomalies on a pixel level. To calculate the PRO-score one needs to set a threshold at which level to classify a pixel as anomalous. This is usually done based on the ROC considering a trade-off between acceptable levels of False Positive Rates (FPR) and TPRs. This can depend on the use case, considering whether it is costly to allow a high FPR (e.g. if a false positive results in an entire production batch being discarded). All pixels with anomaly values below the threshold are set to zero, while the rest keep their original anomaly score. We calculate the area under the PRO-curve up to a maximum false positive rate of 30% following the procedures of [4] and [21] for the same datasets. We use the PRO-score to evaluate AL.

Furthermore, we report the precision recall area under curve (PRAUC) as supplement to the AUROC for a more detailed view of the overall performance and to compare results with other works that also use this score. PRAUC is only used to measure image-level AD performance.

3.6 Experiments

Our first goal was to assess if similar outcomes can be achieved as the *VT-ADL* architecture by Mishra et al. [21]. While they trained a vision transformer with a GMM and a CNN from scratch on the BTAD and MVTecAD datasets, we used a pre-trained *DeiT* model as a frozen image backbone and trained only a GMM with the likelihood loss. Chosen for its enhanced performance over *ViT*, our *DeiT* configuration includes twelve layers and twelve heads, versus the six layers and eight heads used by *VT-ADL*. We used an image resolution of 224×224 instead of 500×500 to match the requirements of our pre-trained image encoders and employed a smaller GMM with only 100 instead of 150 Gaussians according to our empirical hyperparameter search. Additionally, we attempted to reproduce Yu et al.'s [31] promising *FastFlow* results for MVTecAD, despite the absence of their source code. We followed their experimental details. While the authors use the 7th *DeiT* encoder block's embedding we additionally evaluate a model version which uses *DeiT*'s last layer. For compatability with our pre-trained backbones we scale the images to a size of 224×224 instead of 384×384 pixels.

A second goal was to study the behavior of GMMs and NFs on a more fine-grained level to find out if there are differences in the performance depending on the object class. For this experiment we used our setup with a pre-trained *DeiT* image encoder and trained and evaluated the 15 classes of the MVTecAD dataset separately. For the NF model we investigated two different versions, NF i11 which uses the features from the last layer of the image encoder and NF i7 which uses the feature maps of the 7th encoder layer. For the sake of completeness we also compared the performance with the reported results from [21] and from [31].

	VT-ADL		GMM		IMAGE	PIXEL	NF i11		IMAGE	PIXEL
DATA CLASS	PRAUC	PRO	PRAUC	PRO	AUROC	AUROC	PRAUC	PRO	AUROC	AUROC
1	**99.00**	**92.00**	97.25	78.83	93.25	81.68	99.96	80.59	**99.90**	**84.55**
2	94.00	89.00	**95.50**	**92.22**	75.29	92.20	96.60	87.58	**79.36**	87.85
3	**77.00**	86.00	19.10	**90.49**	67.18	91.96	96.85	95.68	**99.65**	**95.72**
mean	**90.00**	**89.00**	70.62	87.18	78.57	88.61	97.80	87.95	**92.97**	**89.37**

Table 3: Results on BTAD of *VT-ADL* [24] and GMM and NF i11 with *DeiT*.

Third, we examined the performance effects of replacing traditional monolithic transformers with hierarchical versions using the *EsVit* and the *EfficientFormer* models described in Section 3.1. We also consider the CNN-based *ResNet-50* backbone as a baseline. As proposed in [11] and discussed in Section 3.2, we used the output of several blocks of the *ResNet* backbone and averaged the results. In case of the GMM, we trained only two models with 50 Gaussians for the last two blocks, due to hardware limitations. We trained and evaluated the methods on five selected classes of the MVTecAD dataset. The classes were chosen separately for both the GMM and the NF i11 approaches based on the performance of our previous experiment with the *DeiT* encoder. As a representative sample for each model, we included classes with high, medium and low performance according to our experiments. For GMM these are the classes *cable*, *carpet*, *grid*, *hazelnut* and *tile*. For the NF i11 we use the classes *bottle*, *carpet*, *hazelnut*, *leather* and *screw*.

4 Results and Discussion

4.1 Comparison with the VT-ADL and FastFlow models

Table 3 shows that our GMM model performs better than *VT-ADL* in two of three classes of the BTAD dataset in the localization task and one class in the detection class. The bad detection performance on class three may result from the model highlighting anomalies correctly but also producing spots with high anomaly scores in normal images (see Figure 3). This results in a high false positive rate when using the maximum of the patch anomaly scores to classify the image. Considering also the composition of the test dataset for this class can explain the gap between detection and localization performance, since it has about ten times more normal than anomalous samples. The overall localization performance of our model on the MVTecAD dataset is higher than the one of *VT-ADL*. These results show, that using a pre-trained transformer encoder in scenarios with relatively small datasets can be beneficial compared to training a transformer end-to-end.

Table 4 shows, that our implementation of the NF model could not reach the values reported for *FastFlow* in [31]. However, the use of only 80% of the training data and a lower image size may have negatively affected the performance.

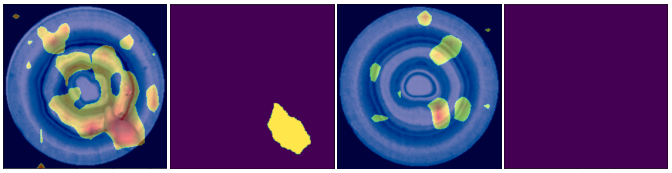

Fig. 3: The third class of the BTAD dataset, processed by the GMM with *DeiT*. Anomaly map and ground truth for a defective (left) and normal (right) sample.

4.2 Comparison of GMMs and NF models

architecture	NF i11		NF i7		GMM		VT-ADL	FastFlow	
dataclass	AD	AL	AD	AL	AD	AL	AL	AD	AL(AUROC)
bottle	**99.84**	85.95	98.57	**93.64**	98.49	45.24	94.90	100.00	97.70
cable	**94.42**	**94.68**	81.73	84.87	61.94	78.03	77.60	100.00	98.40
capsule	**92.94**	93.42	92.82	**95.04**	82.69	93.69	67.20	100.00	99.10
carpet	96.07	95.92	87.40	87.40	**97.11**	**97.06**	77.30	100.00	99.40
grid	**98.08**	93.79	96.66	**96.11**	76.44	71.21	87.10	99.70	98.30
hazelnut	**98.46**	95.81	87.18	93.06	65.43	**96.86**	89.70	100.00	99.10
leather	**100.00**	88.10	99.97	**98.83**	98.37	98.62	72.80	100.00	99.50
metal nut	**99.71**	**92.98**	87.93	88.69	60.95	84.54	72.60	100.00	98.50
pill	**92.17**	**94.25**	81.29	90.83	69.01	91.35	70.50	99.40	99.20
screw	**86.53**	**95.96**	77.09	84.00	52.43	46.37	92.80	97.80	99.40
tile	**99.96**	91.43	96.03	**93.47**	92.50	90.38	79.60	100.00	96.30
toothbrush	90.83	93.50	88.61	95.24	**95.56**	**95.84**	90.10	94.40	98.90
transistor	**95.71**	**96.52**	88.21	93.29	76.83	92.31	79.60	99.80	97.30
wood	92.46	87.73	**98.25**	**93.86**	93.33	90.66	78.10	100.00	97.00
zipper	**96.68**	90.91	92.67	**97.31**	79.28	95.81	80.80	99.50	98.70
mean	**95.59**	**92.73**	90.29	92.38	80.02	84.53	80.71	99.37	98.45
std	**4.00**	**3.28**	6.92	4.38	15.49	17.41	8.18	1.44	0.94

Table 4: Image AUROC (AD) and PRO (AL) of the NF model with feature maps from last and seventh *DeiT* block, the GMM, *VT-ADL* and *FastFlow*. Standard deviation (std) is calculated across classes. *FastFlow* reported pixel AUROC.

Table 3 and 4 show that the overall performance of the NF model is better than the GMM. However, the GMM performs better in localization tasks on most of the surface classes. The high standard deviation between the classes on MVTecAD for the GMM is mainly caused by the classes *bottle* and *screw*, which could not be learned correctly. The NF-model has a relatively small standard deviation between the classes and thus is more robust when applied to new classes. Together with the smaller size this makes it more suitable for a use in production scenarios. The weakness of NF i11 in localization performance on some classes can be eliminated when using NF i7 instead (see Table 4). However,

the overall detection performance of NF i7 is about five percent points less. The anomaly maps show that when using the last block, the highlighted areas are more coarse grained but the model is also more certain. Using the seventh block highlights anomalies more precisely but is also more uncertain and produces small anomaly spots on anomaly-free images. Regarding the model size (see Table 1), the NF model has a clear advantage over the GMM.

4.3 Performance of the Backbones

	AUROC AD				PRO SCORE			
dataclass	DeiT	EffFormer	EsViT	ResNet	DeiT	EffFormer	EsViT	ResNet
cable	62.00	52.00	**89.00**	79.00	78.00	**81.00**	72.00	61.00
carpet	**97.00**	78.00	93.00	74.00	**97.00**	86.00	70.00	58.00
grid	66.00	**78.00**	75.00	68.00	**72.00**	71.00	61.00	54.00
hazelnut	66.00	54.00	**97.00**	73.00	**97.00**	88.00	68.00	51.00
tile	96.00	74.00	**99.00**	65.00	**90.00**	74.00	49.00	54.00
mean	77.40	67.20	**90.60**	71.80	**86.80**	80.00	64.00	55.60

Table 5: AD and AL score of different backbones for the GMM on MVTecAD.

Table 5 shows that for the GMM the overall localization performance is the best with the *DeiT* backbone. It outperforms the hierarchical backbones on all classes except for the *cable* class. In contrast, the *EsViT* model performs best in three of five classes in the detection task and has the overall best detection performance. A possible reason for the gap between localization and detection performance can be seen on the generated anomaly maps in Figure 4. The *EsViT* model does locate the anomaly correctly but also highlights large areas in the background. On the normal image there is no area highlighted at all. A possible reason for the bad performance of the *ResNet* backbone is the usage of only 50 Gaussians and two output layers as discussed in Section 3.6.

The results in Table 6 show, that *DeiT* is the backbone that results in the best detection performance for the NF model. Nevertheless, *EsVit* follows with the second best detection performance. Interestingly, while in general the NF achieved the best results, for *EsViT* the GMM resulted in a better performance. The *ResNet* backbone outperforms *DeiT* in two localization tasks but performs worse in detection tasks. Figure 5 shows the anomaly maps generated with the different backbones.

Hierarchical transformers create smaller models than CNNs, producing compact, information-rich patch embeddings, as highlighted in Table 1, where the patch embedding size significantly affects the GMM and NF sizes. Unlike *DeiT*, these models are also smaller or equal in size to *ResNet*. For high-resolution images, the adequacy of transformer patch embedding for identifying small anomalies needs evaluation. Smaller embeddings can result in coarser feature maps due to upsampling from the patch embedding level.

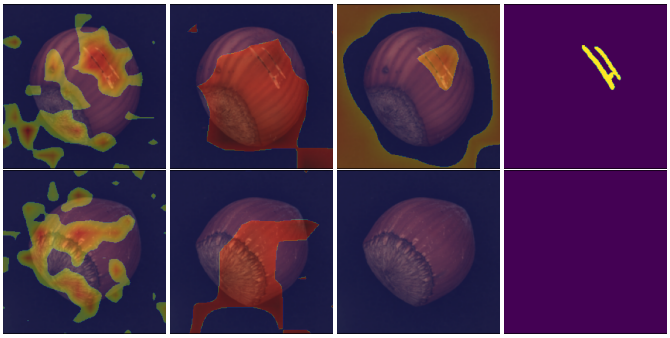

Fig. 4: Anomaly maps for the *hazelnut* class of a GMM with the backbones *DeiT*, *EfficientFormer*, *EsViT* and the corresponding ground truth.

	AUROC AD				PRO SCORE			
dataclass	DeiT	EffFormer	EsVit	ResNet	DeiT	EffFormer	EsVit	ResNet
bottle	**100.00**	98.00	93.00	97.00	86.00	79.00	66.00	**97.00**
carpet	**96.00**	78.00	87.00	89.00	**96.00**	78.00	82.00	88.00
hazelnut	**98.00**	89.00	93.00	91.00	**96.00**	92.00	73.00	90.00
leather	**100.00**	77.00	96.00	**100.00**	88.00	85.00	63.00	**97.00**
screw	**87.00**	43.00	65.00	50.00	**96.00**	60.00	57.00	84.00
mean	**96.20**	77.00	86.80	85.40	**92.40**	78.80	68.20	91.20

Table 6: AD and AL score of different backbones for the NF model on MVTecAD.

4.4 Considerations and Limitations for Practical Application

Anomaly maps can be used in various manufacturing scenarios such as explanation of a model's choice, making decisions based on the anomaly size or to double check a model's decision [6]. However, it is important to notice that the experiments in this work are conducted on benchmark datasets that have high quality, which is hard to achieve in a real-world scenario. In manufacturing, metrics should be selected considering the composition of training data and the severity of false positives and false negatives. These aspects should also be considered when deciding on the thresholding strategy for the PRO-score and for AD. Our experiments on the detection performance show, that AUROC score should be preferred when false-negatives are expensive while PRAUC is more suitable when false-positives lead to problems. To evaluate the quality of a model when facing highly imbalanced datasets, both scores should be considered. For localization tasks, the PRO-score should be preferred. The model selection is highly influenced by the available data, hardware and real-time requirements. If enough hardware is available, the superior performance of *DeiT* or a comparable monolithic transformer should be chosen. In case of strong hardware limitations and/or real-time requirements, smaller and faster hierarchical transformer are the models of choice. Yu et al. report an inference time of eight milliseconds with *DeiT* and their *FastFlow* model [31], what can be regarded as a baseline

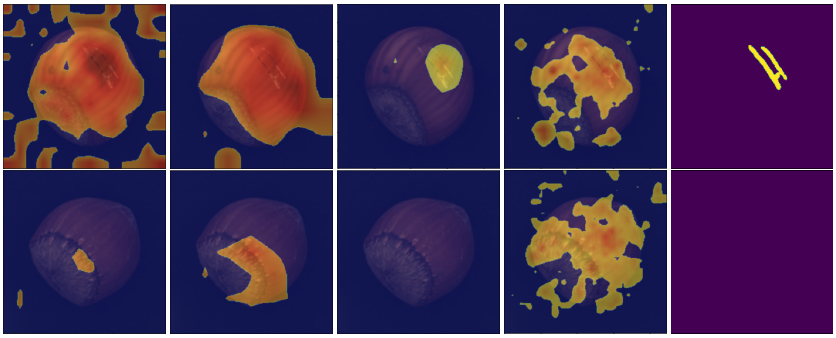

Fig. 5: Anomaly maps for the *hazelnut* class of the NF model with the backbones *DeiT*, *EfficientFormer*, *EsViT*, *ResNet* and the corresponding ground truth.

for hierarchical transformer models. When processing images with a much higher resolution, adjusting the patch-size might be required and thus a training of the backbone becomes necessary.

5 Conclusion

In our work, we provided a comprehensive overview of SoTA vision transformer models and evaluated different paradigms for visual anomaly detection in industrial visual quality control. We implemented two anomaly detection methods with four different image encoding backbones, all of them pre-trained on ImageNet1k. We trained our anomaly detection models on the datasets MVTecAD and BTAD and compared our results with two existing approaches from the literature. Finally, we discussed important aspects to consider when applying these approaches to production. We showed that using transformer models can improve the performance of anomaly detection models and reduce the overall size compared to *ResNet*. Moreover, we showed that using pre-trained transformer models can have an advantage over training from scratch. Hierarchical transformer models are worth to evaluate for further use in production scenarios with limited computational capacity. Future research could perform experiments with using the output of more than one layer of the encoder model as done with *ResNet* and proposed in [29] for their segmentation transformer.

Acknowledgments. Christoph Hönes has received funding from SAP SE. Christoph Hönes and Miriam Alber were employed by esentri AG who also provided computational resources.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ardizzone, L., Bungert, T., Draxler, F., Köthe, U., Kruse, J., Schmier, R., Sorrenson, P.: Framework for Easily Invertible Architectures (FrEIA) (2022), https://github.com/vislearn/FrEIA
2. Bae, J., Lee, J.H., Kim, S.: Image anomaly detection and localization with position and neighborhood information, https://arxiv.org/pdf/2211.12634v2.pdf
3. Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., Steger, C.: The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. International Journal of Computer Vision **129**(4), 1038–1059 (2021). https://doi.org/10.1007/s11263-020-01400-4
4. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4182–4191 (2020). https://doi.org/10.1109/CVPR42600.2020.00424
5. Bishop, C.: Mixture density networks. Workingpaper, Aston University (1994)
6. Choi, B., Jeong, J.: Viv-ano: Anomaly detection and localization combining vision transformer and variational autoencoder in the manufacturing process. Electronics **11**(15), 2306 (2022). https://doi.org/10.3390/electronics11152306
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
8. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp, https://arxiv.org/pdf/1605.08803
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale, https://arxiv.org/pdf/2010.11929
10. Fan, Y., Wen, G., Li, D., Qiu, S., Levine, M.D., Xiao, F.: Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. Computer Vision and Image Understanding **195** (2020). https://doi.org/10.1016/j.cviu.2020.102920
11. Gudovskiy, D., Ishizaka, S., Kozuka, K.: Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows, https://arxiv.org/pdf/2107.12571
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition, https://arxiv.org/pdf/1512.03385
13. Hyun, J., Kim, S., Jeon, G., Kim, S.H., Bae, K., Kang, B.J.: Reconpatch : Contrastive patch representation learning for industrial anomaly detection, http://arxiv.org/pdf/2305.16713
14. Kim, Y., Jang, H., Lee, D., Choi, H.J.: Altub: Alternating training method to update base distribution of normalizing flow for anomaly detection, https://arxiv.org/pdf/2210.14913v1.pdf
15. Lei, J., Hu, X., Wang, Y., Liu, D.: Pyramidflow: High-resolution defect contrastive localization using pyramid normalizing flow, https://arxiv.org/pdf/2303.02595v1.pdf
16. Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L., Gao, J.: Efficient self-supervised vision transformers for representation learning, https://arxiv.org/pdf/2106.09785.pdf

17. Li, H., Wu, J., Chen, H., Wang, M., Shen, C.: Efficient anomaly detection with budget annotation using semi-supervised residual transformer, http://arxiv.org/pdf/2306.03492
18. Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., Ren, J.: Efficientformer: Vision transformers at mobilenet speed. arXiv preprint arXiv:2206.01191 (2022)
19. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows, https://arxiv.org/pdf/2103.14030
20. Mathian, E., Liu, H., Fernandez-Cuesta, L., Samaras, D., Foll, M., Chen, L.: Haloae: An halonet based local transformer auto-encoder for anomaly detection and localization, https://arxiv.org/pdf/2208.03486.pdf
21. Mishra, P., Verk, R., Fornasier, D., Piciarelli, C., Foresti, G.L.: VT-ADL: A vision transformer network for image anomaly detection and localization. In: 30th IEEE/IES International Symposium on Industrial Electronics (ISIE) (June 2021)
22. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection, https://arxiv.org/pdf/2106.08265v2.pdf
23. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019), https://arxiv.org/pdf/1801.04381
24. Tao, X., Gong, X., Zhang, X., Yan, S., Adak, C.: Deep learning for unsupervised anomaly localization in industrial images: A survey. IEEE Transactions on Instrumentation and Measurement **71**, 1–21 (2022). https://doi.org/10.1109/TIM.2022.3196436
25. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention, https://arxiv.org/pdf/2012.12877
26. Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., Shlens, J.: Scaling local self-attention for parameter efficient visual backbones, https://arxiv.org/pdf/2103.12731
27. Wang, T., Chen, Y., Qiao, M., Snoussi, H.: A fast and robust convolutional neural network-based defect detection model in product quality control. The International Journal of Advanced Manufacturing Technology **94**(9-12), 3465–3471 (2018). https://doi.org/10.1007/s00170-017-0882-0
28. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pvt v2: Improved baselines with pyramid vision transformer. Computational Visual Media **8**(3), 415–424 (2022). https://doi.org/10.1007/s41095-022-0274-8
29. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers, http://arxiv.org/pdf/2105.15203.pdf
30. You, Z., Yang, K., Luo, W., Cui, L., Le Xinyi, Zheng, Y.: Adtr: Anomaly detection transformer with feature reconstruction, https://arxiv.org/pdf/2209.01816.pdf
31. Yu, J., Zheng, Y., Wang, X., Li, W., Wu, Y., Zhao, R., Wu, L.: Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows, https://arxiv.org/pdf/2111.07677
32. Zhang, K., Wang, B., Kuo, C.C.J.: Pedenet: Image anomaly localization via patch embedding and density estimation. Pattern Recognition Letters **153**, 144–150 (2022). https://doi.org/10.1016/j.patrec.2021.11.030

33. Zhang, Z., Zhang, H., Zhao, L., Chen, T., Arik, S.O., Pfister, T.: Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding, https://arxiv.org/pdf/2105.12723.pdf
34. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=BJJLHbb0-

A Generated anomaly maps from different model configurations

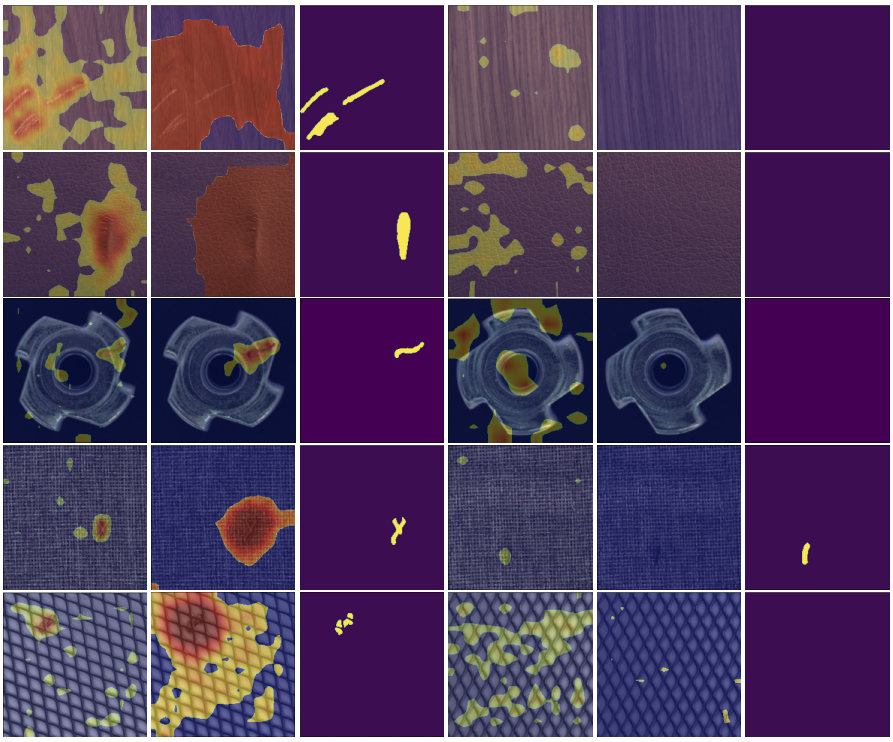

Fig. 6: Anomaly maps for the classes *wood*, *leather*, *metal nut*, *carpet* and *grid*, from the first row to the last, from the MVTecAD dataset with the NF model and the *DeiT* encoder. In each row, the first anomaly map is created with NF i7, the second with NF i11 and the third image shows the corresponding ground truth.

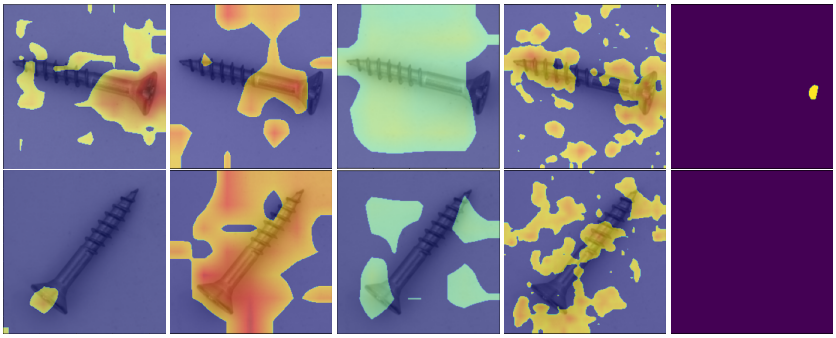

Fig. 7: Anomaly maps for the *screw* class of the NF model. The used backbones are from left to right: *DeiT*, *EfficientFormer*, *EsViT*, *ResNet* and the corresponding ground truth. It can be noted, that *DeiT* was the only backbone with which the model was able to capture the small anomaly.

B Class distributions of the used datasets

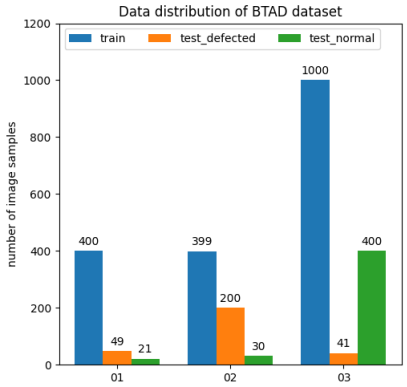

Fig. 8: Distribution of the different classes of the BTAD dataset.

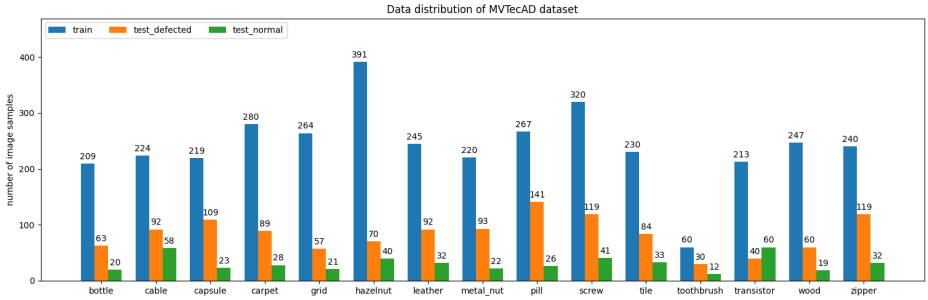

Fig. 9: Distribution of the different classes of the MVTecAD dataset.

C Ablation study and production application

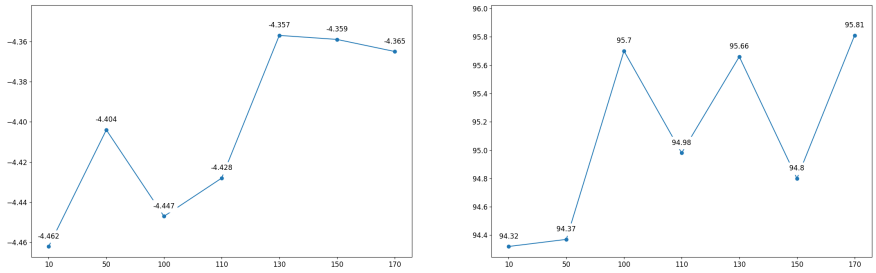

Fig. 10: Plots of loss (left) and PRO score (right) for different numbers of Gaussian's on the data class *hazelnut* from MVTecAD. The graphics show that there is no direct relation between PRO score and likelihood loss.

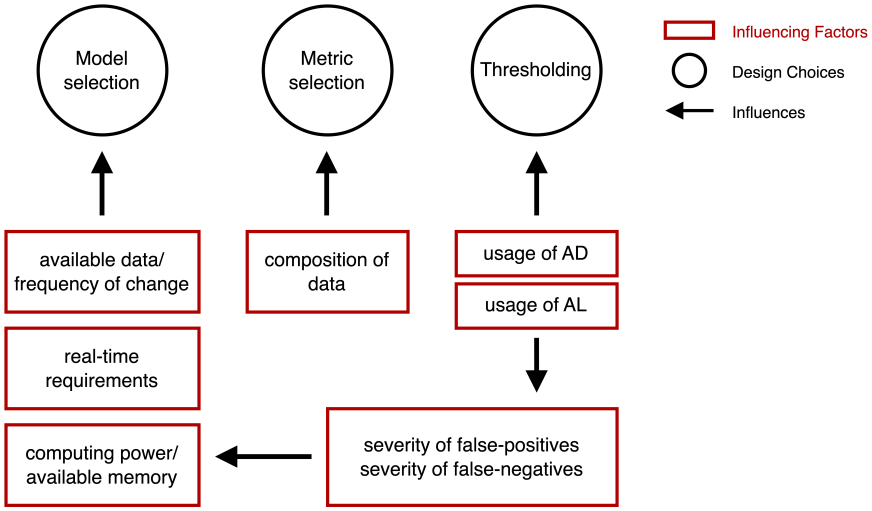

Fig. 11: High level overview on decisions to be made when applying our approaches to production scenarios.

This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: tbd. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms