Adaptation of Unsupervised Surface Anomaly Detection to Domain Shift using Limited Target Domain Samples

Matej Dobrevski, Danijel Skočaj

Faculty of Computer Science, University of Ljubljana, Slovenia.
e-mail: matej.dobrevski@fri.uni-lj.si

Abstract

In many realistic visual surface inspection scenarios we can expect that the distribution of images will change with time. To solve this problem, existing methods would require the acquisition of a new training set and model re-training. In this work, we present a method for adapting the CS-Flow unsupervised surface anomaly detection method to a domain shift using only a few samples from the target domain. The proposed approach does not require access to the source domain data beyond the initial training. We construct domain-shifted datasets using the VisA dataset and show that the proposed approach is more effective than using the limited amount of samples for direct training.

1 Introduction

Visual surface anomaly detection is becoming a standard tool in many industries. There exists a variety of methods that perform well on multiple standard datasets. However, once such a system is deployed its performance is likely to decrease with time because of gradual changes in illumination, background, acquisition settings, or raw materials. Once the domain shift has been detected, it is necessary to collect a new dataset and re-train the model.

Normalizing flow models are generative models that transform a complex and usually unknown distribution into a tractable distribution (like multivariate Gaussian distribution). Multiple surface anomaly detection methods [8, 2, 16] use them to model the distribution of features of normal samples, obtained from a pre-trained feature extractor, with a simple distribution. Then, anomalous objects are detected as low probability points from this distribution, using a distance metric and setting a simple threshold. However, the distribution of the features of normal samples is also affected by changes in the domain. As such, if the domain shifts (because of e.g. illumination), these methods would detect the sample as anomalous, even though the surface of the object has not actually changed.

All feature extractors encode information about the domain in addition to information about the object of interest. It is a non-trivial task to separate the domain-specific and object-specific information from the common feature space. Some researchers have observed [19, 4] that the encoding of domain-specific information is predominantly done by the early layers of deep networks. If we were to obtain a distance metric of the feature discrepancy between the source and target domains, it should be possible to update the feature extractor, so that the embedding space for the target domain is closer to the embedding space for the source domain.

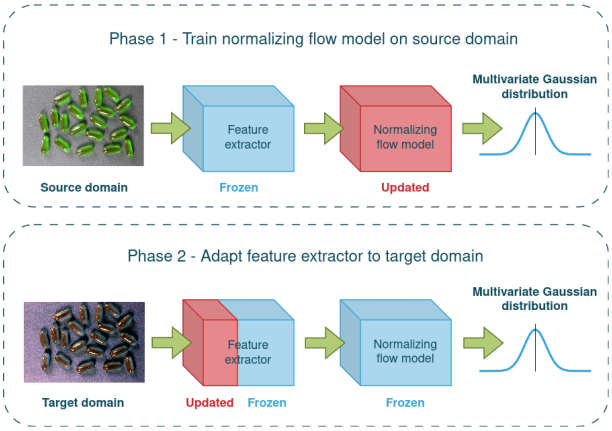

Figure 1: We propose a method for few-shot domain adaptation in surface anomaly detection scenarios. In the first phase, the normalizing flow model is trained on the source domain. In the second phase, the first layers of the normalizing flow model are adapted to the target domain.

In this work, we propose to use a flow model trained on normal samples from a source distribution as a distance metric for aligning the shift between a target and source domain, by using a few normal samples from the target domain to constrain the feature extractor to retain object-specific information. We present a method for **few-shot domain adaptation for unsupervised surface anomaly detection**. We use the well-known CS-Flow [8] method for unsupervised surface anomaly detection, which models the feature distribution of normal samples using a cross-scale normalizing flow. Once the normalizing flow has been trained on the source domain, its weights are frozen. When a domain shift occurs we use the flow model to adapt the feature extractor to the target domain. We use the recent VisA [20] dataset and create five domain-shifted versions for evaluation. We show that our method is more effective than using the limited amount of target domain data for training the anomaly detection method.

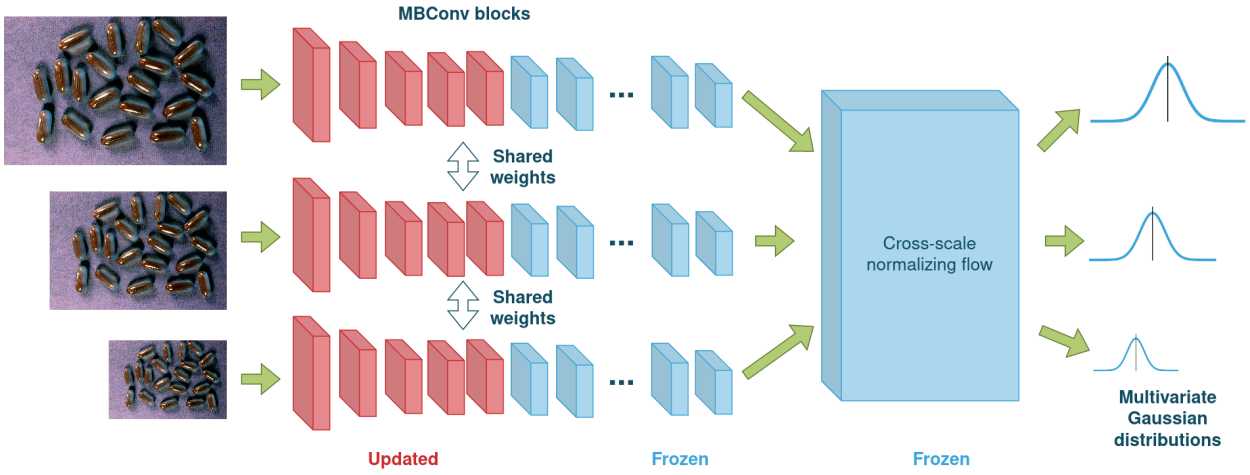

Figure 2: During the phase of adaptation to the target domain, only the initial blocks of the EfficinetNet-B5 feature extractor are updated. The network is updated using the same images at three scales.

2 Related work

Unsupervised surface anomaly detection aims to detect anomalies on the surface of objects while only training on normal samples. The two most prominent groups of methods that solve this problem are image reconstruction-based methods, and feature statistics-based methods. Image reconstruction-based approaches like AnoGAN [9] and RIAD [18], learn to reconstruct the normal samples. If the reconstruction of the original image "fails" then the image contains anomalies. Feature statistics-based methods like GaussianAD [6], PaDiM [1], PatchCore [7], CS-Flow [8], FastFlow [16] use a (usually pre-trained) feature extraction network and model the distribution of features of normal samples. At test time, some type of distance measure is used to classify an image as normal or anomalous. Recently, a third group of methods [17], which create synthetic anomalies and train a discriminator has emerged.

Domain adaptation seeks to learn on a source domain and perform well on a target domain. Most often, the target domain data is limited in some way. There are numerous problem settings: the target data can be labeled but there is not enough of them for building a good model, it can be partially labeled or unlabeled. There can also be only a few samples from the target domain. The most popular approaches either learn a transformation from one domain to the other or learn representations that are invariant to the domain.

Test-Time Adaptation is an interesting new subfield in domain adaptation, that addresses the problem of domain shift during test time. The methods in this field, like TENT [13], TTT [10], TTT++ [4] and TTTFlow [5], assume that during training we have access only to the source domain, and during testing only to the target domain data. This is challenging because it is impossible to directly measure any discrepancies. TTTFlow uses a normalizing flow to model the feature distribution for the source domain and is the inspiration for the method proposed in this paper.

3 Method

We are using a pre-trained feature extractor $f_{FE}(x) = y$ that maps an image $x \in X$ to its feature representation $y \in Y$ and a normalizing flow model that learns a bijective mapping from the unknown distribution p_Y of the feature space Y to a latent space Z with a multivariate Gaussian distribution p_Z.

In the **first phase** of training, the feature extractor's weights are kept frozen and the normalizing flow model $z = f_{NF}(y)$ is trained on the source domain using normal samples from the source domain:

$$p_Y(y) = p_Z(z)\left|\det \frac{\partial z}{\partial y}\right|. \tag{1}$$

For convenience, the equivalent objective being optimized is the log-likelihood:

$$\mathcal{L}(y) = -\log p_Y(y) = \frac{\|z\|_2^2}{2} - \log\left|\det \frac{\partial z}{\partial y}\right|. \tag{2}$$

This is identical to the training procedure for anomaly detection on the source domain. At test time, we can classify the whole image as anomalous by using a threshold θ:

$$\text{Annomalous}(x) = \begin{cases} 1 & \text{if} \quad p_Z(z) < \theta \\ 0 & \text{if} \quad p_Z(z) \geq \theta. \end{cases} \tag{3}$$

Since the flow model is fully convolutional, we can localize an anomalous region in the image x by looking at the score for each (i, j) position of the feature map y by aggregating the z values along the channel dimension.

In the **second phase**, we adapt the feature extractor f_{FE} to the target domain, so that when a sample $x_{shifted} \in X_{shifted}$ from a shifted domain arrives, the feature extractor will map it to the same feature space Y. To do this, the weights of the trained normalizing flow model f_{NF} are frozen, as well as most of the weights of

the later stages of the feature extractor f_{FE}. Only the first five stages (or blocks) of f_{FE} are being updated, by using the same loss as in (3) calculated from a few normal samples from $X_{shifted}$ and backpropagating the gradients through f_{NF} and f_{FE}, as illustrated in Figure 1.

The approach is demonstrated using the EfficientNet-B5 [12] model pre-trained on Imagenet-1K as a feature extractor. The normalizing flow model is a cross-scale flow as described in [8]. Because the cross-scale flow uses features extracted from three different resolutions of the input image, a single target domain sample generates three training samples for the adaptation of the feature extractor. The stages that are being updated are the first five MBConv [11] blocks.

4 Experiments

4.1 Dataset

We evaluate the method using the recent VisA [20] dataset. Since there are no available datasets that represent the domain shifts we are interested in, we create five domain-shifted versions of the dataset: VisA-brg, VisA-coljit, VisA-gray, VisA-ljs, and VisA-ld. The *VisA-brg* dataset was obtained by simply permuting the channels of the original images as RGB→BRG. The *VisA-coljit* dataset was generated using the color jittering technique for image augmentation; in our case, the same image transformation was applied to all images. All images were first transformed into HSV representation, all channels were multiplied with a constant factor, and then the images were reverted to the RGB representation. *VisA-gray* is simply a grayscale version of the original. *VisA-ljs* and *VisA-ld* were generated using the WCT2 [14] method for style-transfer, where two arbitrarily chosen images (images of Ljubljana sunrise and Luka Dončič) were used as a style image. A number of images from these datasets, showing the influence of the applied transformations, are shown in Figure 3. For the evaluation, all images are resized to 512×512 pixels. The results were generated using 5-fold cross-validation.

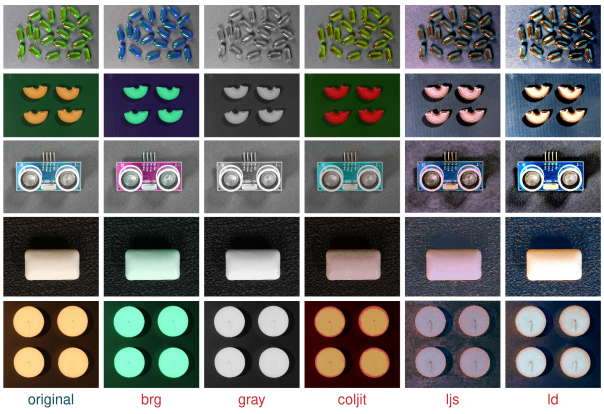

Figure 3: Examples from the domain-shifted dataset for five object categories. In the leftmost column, we see the objects as they look in the original dataset. The other columns depict the domain-shifted images.

4.2 Baselines

As the best-base baseline, we train the CS-Flow models on the training sets of each of the shifted domains and evaluate them on the test sets in the corresponding domains. The average AUROC score across the five domains is 0.91, represented as the blue line in Figure 4. As the worst-case baseline, we train the CS-Flow model on the training set of the original domain and evaluate it on test sets of the shifted domains. As represented by the red line in Figure 4, this results in a significant drop in performance and an average score of 0.64.

As a more realistic baseline, we train the model on randomly chosen 1, 5, or 10 images from each target domain and evaluated on the corresponding test set. From the results, we can see that even using a single image from the target domain is better than using the complete source domain. It is also somewhat surprising that using 10 images did not bring significant improvements in the performance.

As another realistic baseline, we added the few samples from the target set to the complete source set and trained a model on the joined set. We can see that in this case, whether we used only 1 or we used 5 samples made a large difference. There is a further increase in performance if we use 10 images. The results from these baselines are represented by the orange and green bars in Figure 4, respectively.

Figure 4: The AUROC metric, using the CS-Flow method, as evaluated on the (shifted) target domain. The blue line is the best-case scenario, training on the complete training set of the shifted datasets. The red line is the worst-case scenario when we train only on the original domain. The orange bars are the results from training on a few samples from the shifted domains, the green bars are the results from training on the complete original dataset plus a few samples from the shifted domains and the dark blue bar is the score for the proposed adaptation method, first trained on the complete original domain and adapted on a few samples from the shifted domains.

4.3 Proposed approach

We trained a CS-Flow model on the complete source domain, and adapted the feature extractor as per our method, using 1, 5, or 10 images from the target domain. From the results in Figure 4, we can see that performing adap-

tation in this manner brings significant improvement in the scores, especially in the case where we are performing one-shot adaptation. As the number of available samples from the target domain increases, the improvement in performance that we get from performing adaptation is decreased. This is to be expected, as having a large number of samples from the target domain, decreases the importance of samples from the source domain. If this number is large enough, the samples from the source domain will start causing more issues during training, than helping. The detailed per-object results are presented in Figure 5.

4.4 Comparison with related work

Finally, since there are no methods that we can directly compare the proposed method to, we compare the results with related unsupervised surface anomaly detection methods in the case of one-shot learning. All methods are trained on one image from the original domain, and evaluated on the complete test set, only the results for CS-Flow are generated both with and without domain adaptation. From the results in Table 1 we can see that the basic CS-Flow has worse performance than SPADE or PatchCore. CS-Flow with the proposed adaptation (which unlike the rest of the methods has access to the source domain) outperforms these methods. Our proposed method achieved comparable results to the recently proposed zero/few-shot anomaly detection method WinCLIP [3]. Note that we used images with resolution 512×512, while WinCLIP used the full resolution images.

Table 1: One-shot learning, for unsupervised surface anomaly detection.

Method	Image-level AUROC %
SPADE [15]	79.5
PaDiM [1]	62.8
PatchCore [7]	79.9
CS-Flow [8]	74.1
WinCLIP [3]	83.8
CS-Flow with DA	83.7

5 Conclusion

In this work, we presented an approach for few-shot unsupervised domain adaptation for unsupervised anomaly detection. The approach is suitable for adapting anomaly detection methods that use a normalizing flow to model the distribution of features of normal samples. We created a dataset for evaluating the method, generated from the VisA dataset and simulating domain-shift in the acquisition of industrial images. We showed that it is more effective than simply using the few samples that are available for training an anomaly detection method.

While the method shows significant improvement when compared to simply training with a few samples, further improvements are needed for the method to achieve scores that are near the performance on the original domain. This might be possible to achieve by using an additional normalizing flow model, by using perturbations of the few target domain samples, in order to artificially increase the number of domain-specific samples.

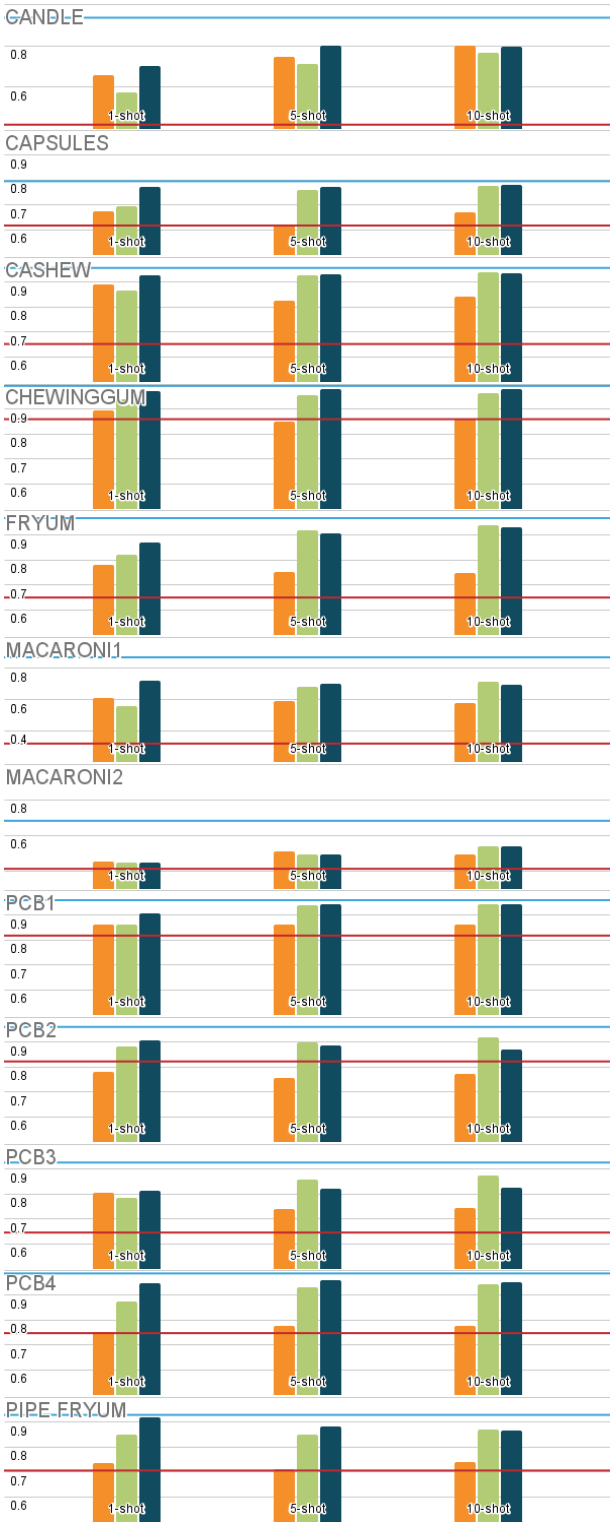

Figure 5: Results for each object from the VisA dataset.

Acknowledgements

This work was in part supported by the ARIS research project L2-3169 (MV4.0) and research programme P2-0214.

References

[1] Defard, T., Setkov, A., Loesch, A., Audigier, R.: Padim: A patch distribution modeling framework for anomaly detection and localization. In: ICPR International Workshops and Challenges. pp. 475–489. Springer International Publishing, Cham (2021)

[2] Gudovskiy, D., Ishizaka, S., Kozuka, K.: Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1819–1828. IEEE Computer Society, Los Alamitos, CA, USA (jan 2022)

[3] Jeong, J., Zou, Y., Kim, T., Zhang, D., Ravichandran, A., Dabeer, O.: Winclip: Zero-/few-shot anomaly classification and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19606–19616 (June 2023)

[4] Liu, Y., Kothari, P., van Delft, B., Bellot-Gurlet, B., Mordan, T., Alahi, A.: Ttt++: When does self-supervised test-time training fail or thrive? In: Advances in Neural Information Processing Systems. vol. 34, pp. 21808–21820. Curran Associates, Inc. (2021)

[5] Osowiechi, D., Hakim, G.A.V., Noori, M., Cheraghalikhani, M., Ayed, I., Desrosiers, C.: Tttflow: Unsupervised test-time training with normalizing flow. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2125–2126. IEEE Computer Society, Los Alamitos, CA, USA (jan 2023)

[6] Rippel, O., Mertens, P., König, E., Merhof, D.: Gaussian anomaly detection by modeling the distribution of normal data in pretrained deep features. IEEE Transactions on Instrumentation and Measurement (2021)

[7] Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14298–14308 (2022)

[8] Rudolph, M., Wehrbein, T., Rosenhahn, B., Wandt, B.: Fully convolutional cross-scale-flows for image-based defect detection. In: IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022. pp. 1829–1838. IEEE (2022)

[9] Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Information Processing in Medical Imaging - 25th International Conference, IPMI 2017, Boone, NC, USA, June 25-30, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10265, pp. 146–157. Springer (2017)

[10] Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A.A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. vol. 119, pp. 9229–9248. PMLR (2020)

[11] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2815–2823. IEEE Computer Society, Los Alamitos, CA, USA (jun 2019)

[12] Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019. Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114. PMLR (09–15 Jun 2019)

[13] Wang, D., Shelhamer, E., Liu, S., Olshausen, B.A., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021)

[14] Yoo, J., Uh, Y., Chun, S., Kang, B., Ha, J.: Photorealistic style transfer via wavelet transforms. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9035–9044. IEEE Computer Society, Los Alamitos, CA, USA (nov 2019)

[15] Yoon, J., Sohn, K., Li, C.L., Arik, S.O., Pfister, T.: SPADE: Semi-supervised anomaly detection under distribution mismatch. Transactions on Machine Learning Research (2023), featured Certification

[16] Yu, J., Zheng, Y., Wang, X., Li, W., Wu, Y., Zhao, R., Wu, L.: Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. CoRR **abs/2111.07677** (2021)

[17] Zavrtanik, V., Kristan, M., Skočaj, D.: Draem - a discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8330–8339 (October 2021)

[18] Zavrtanik, V., Kristan, M., Skočaj, D.: Reconstruction by inpainting for visual anomaly detection. Pattern Recognition **112**, 107706 (2021)

[19] Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain generalization with mixstyle. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria (2021)

[20] Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: Computer Vision – ECCV 2022. pp. 392–408. Springer Nature Switzerland, Cham (2022)