Anomaly Detection in Construction Sites

Martin Tsekov
Karlsruhe Institute of Technology
utzno@student.kit.edu

Patrick Deubel
Karlsruhe Institute of Technology
patrick.deubel@student.kit.edu

Lei Wan
Karlsruhe Institute of Technology
urmdu@student.kit.edu

Abstract

Anomaly detection has applications in a variety of domains: network intrusion detection [1], video surveillance [9] and medical diagnosis [16]. The main idea of anomaly detection is to model normal behavior and detect deviations from it, e.g. monitoring rarely occurring accidents on a video camera. This paper evaluates reconstruction-based methods [16] and an embedding similarity-based method [4] for anomaly detection on images from construction sites. We show that both types of methods can achieve good classification performance on a clean dataset, but applying them on a noisy dataset results in poor performance. To try to alleviate this issue, two cleaning methods have been applied to the noisy dataset.

1. Introduction

Anomaly detection is a method that aims to identify data samples which differ from normal behavior. For this purpose, a model first needs to learn what a normal sample looks like and detect deviations from that. The method has a variety of applications, including network intrusion detection, video surveillance and diagnosis in the medical domain. The methods used in this paper focus on surface inspection, specifically for detection of cracks in images taken at construction sites. Figure 1 shows exemplary normal and abnormal images from the Concrete-Cracks dataset [17].

There were two tasks to perform: anomaly classification and localization. Firstly, a given image needs to be classified as normal or abnormal, and if there is a crack it needs to be localized. To achieve these tasks, the models presented in this paper are trained only on images without cracks. Abnormal samples can be detected if their predicted anomaly score exceeds a certain threshold.

Figure 1. Normal and abnormal examples from the Concrete-Cracks dataset [17]

2. Related Work

A wide variety of methods can be applied for anomaly detection. Ruff et al. [11] group them into four categories: one-class classification, probabilistic, reconstruction-based and distance-based methods. Traditional shallow approaches include for example k-nearest neighbor and k-means.

Novel deep learning approaches like Autoencoder (AE) and Generative Adversarial Networks (GAN), as well as their variants have been rising in interest for different applications. For example, Baur et al. [2] and Zimmer et al. [16] employed autoencoders to detect anomalies in Brain MR Images. Schlegl et al. [13] used a GAN to model normal retinal OCT images.

A different approach are embedding similarity-based methods that try to model what normality looks like in each patch of an image, like PaDiM [4] or SPADE [3].

3. Methods

This section presents two reconstruction-based methods, the Variational Autoencoder (VAE) and a U-Net-based VAE, as well as the PaDiM framework. Appendix A shows the architecture of AnoVAEGAN [2].

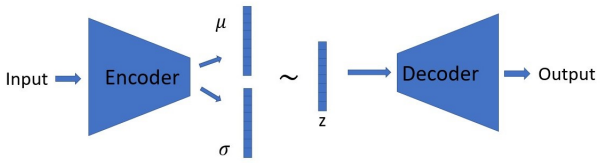

Figure 2. The architecture of the vanilla VAE

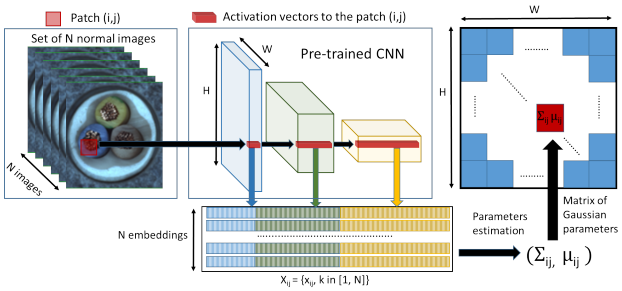

Figure 3. The architecture of the PaDiM framework [4]

3.1. Reconstruction-based methods

Reconstruction-based methods are widely used for unsupervised anomaly detection. Their goal is to train a neural network which can reconstruct normal data samples but fail to reconstruct abnormal data samples. Using the reconstruction error, which is the difference between the original sample and the reconstructed sample, a classification as normal or abnormal can be performed. In addition, by taking the reconstruction error for example in sliding windows, it is also possible to achieve anomaly localization [16].

3.1.1 Variational Autoencoder

In Figure 2 the architecture of Variational Autoencoder [16] is shown. A VAE is composed out of an encoder and a decoder. The encoder compresses the input samples into a latent space and the decoder reconstructs original samples based on their latent representation. The model is optimized by minimizing the loss function, which is the sum of the reconstruction error and the Kullback–Leibler divergence (KL-divergence). The KL-divergence is a measurement of similarity between the latent distribution and an imposed prior distribution, which is generally a standard normal distribution. The loss function $\mathcal{L}$ is defined as

$$\mathcal{L} = -D_{KL}(q(z|x)||p(z)) + \mathbb{E}_{q(z|x)}[\log p(x|z)] \quad (1)$$

where $p(z)$ is the prior distribution of the latent space and $q(z|x)$ and $p(x|z)$ are the probability distribution of the encoder and the decoder, respectively [16].

3.1.2 U-Net-based VAE

The U-Net-based VAE uses skip connections between the encoder and decoder, similar to the U-Net architecture [10]. With these connections the information can be propagated directly from the input to the output and the spatial information in the output images can be preserved. In this paper we only add skip connections between the last two layers of the encoder and decoder, because the number of added skip connection also influences the reconstruction performance of the model. With an increasing amount of skip connections, the quality of the reconstruction improves a lot but the downside is that this behavior can also be observed for abnormal images. This results in a degrading performance for anomaly detection because the reconstruction error between normal and abnormal samples is no longer significantly different. Therefore the model can no longer distinguish between normal and abnormal.

3.2. Patch Distribution Modeling Framework

The second method that was used for the anomaly detection is called Patch Distribution Modeling Framework (PaDiM) [4]. In Figure 3 the architecture of PaDiM is shown. It consists out of a convolutional neural network (CNN), which is pre-trained on the ImageNet dataset [12].

3.2.1 Training of PaDiM

Each sample of the N training samples is used as an input for the pre-trained CNN. From these inputs, the feature embeddings from the first three layers of the CNN are extracted. The first layer outputs a feature embedding of size $H \times W \times C$, where H and W are the height and width of the embeddings and C is the number of channels in this first layer. The height and width values determine the number of patches PaDiM will use. For each one of these $H \cdot W$ patches, a multivariate gaussian distribution is modelled, which indicates what normality looks like in the respective patch [4].

For the calculation of the distributions, the N extracted feature embeddings from the three layers of the CNN are concatenated. For a patch at position $(i, j), 1 \leq i \leq W, 1 \leq j \leq H$, only the slice of the embedding that corresponds to this patch, $\mathbf{X}_{ij}$, is used for the calculation of the distribution. This can be seen in Figure 3, where the patch and its corresponding slices are highlighted in red. To determine $\mathcal{N}(\boldsymbol{\mu}_{ij}, \boldsymbol{\Sigma}_{ij})$, we calculate $\boldsymbol{\mu}_{ij}$ as the sample mean of $\mathbf{X}_{ij}$ and $\boldsymbol{\Sigma}_{ij}$ is calculated as

$$\boldsymbol{\Sigma}_{ij} = \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{x}_{ij}^k - \boldsymbol{\mu}_{ij})(\mathbf{x}_{ij}^k - \boldsymbol{\mu}_{ij})^T + \epsilon\mathbf{I} \quad (2)$$

where $\mathbf{x}_{ij}^k$ is the slice of the embedding at patch (i, j) from training image k and $\epsilon\mathbf{I}$ is a regularisation term to make $\mathbf{\Sigma}_{ij}$ a full rank matrix and invertible [4].

The authors of PaDiM stated that not all channels of each embedding layer have to be used as this is computationally and memory intensive. Instead, randomly selecting only a subset of all channels across the three layers has proven to be nearly as good as using all of them, while decreasing the complexity for training and testing time [4].

3.2.2 Inference

During inference, the test images are used as an input for the pre-trained CNN and the feature embeddings are extracted. Then a score map for each image is created. This is done, by calculating the Mahalanobis distance [8] per patch. Let $\mathbf{x}_{ij}$ be the embeddings for patch (i, j) of the test images and $\mathcal{N}(\boldsymbol{\mu}_{ij}, \mathbf{\Sigma}_{ij})$ is the learned distribution of PaDiM at that patch. Then we compute the distance $M(\mathbf{x}_{ij})$ as

$$M(\mathbf{x}_{ij}) = \sqrt{(\mathbf{x}_{ij} - \boldsymbol{\mu}_{ij})^T \mathbf{\Sigma}_{ij}^{-1} (\mathbf{x}_{ij} - \boldsymbol{\mu}_{ij})}[4]. \quad (3)$$

A large distance $M(\mathbf{x}_{ij})$ means, that the corresponding patch is likely abnormal. Therefore $\mathbf{M} = (M(\mathbf{x}_{ij}))_{1 \leq i \leq W, 1 \leq j \leq H}$ forms the anomaly map for an input image. To classify the whole image, we take the maximum of $\mathbf{M}$ and if it exceeds a certain threshold, then the image is considered an anomaly [4]. Choosing the threshold is considered a hyperparameter.

4. Evaluation

This section presents the evaluation of the reconstruction-based methods and the PaDiM framework on two datasets: Concrete-Cracks [17] and the SDNet2018 [7]. Issues which were encountered with the SDNet2018 dataset are elaborated and results from attempted solutions are also provided.

4.1. Datasets

The Concrete-Cracks dataset has a single category, while SDNet2018 has three: decks, pavements and walls. The images in the datasets are labeled as abnormal or normal, i.e. images with or without cracks, but bounding boxes or segmentations of the cracks are not available. The dataset splits are as follows: 65% for training, 15% for validation and 20% for testing. The datasets were originally introduced for classification, but were repurposed for anomaly detection.

4.2. Evaluation method

All of the presented methods are trained only on normal samples. We trained the VAE with a latent space dimensionality of 128. The encoder and decoder have 5 layers each, starting with 32 kernels at layer 1 and doubling in each layer. Our models were trained for 30 epochs using the Adam optimizer [6] with a learning rate of 0.005 and batch size of 32.

PaDiM models require only a single iteration through the normal samples. The embedding dimensions are chosen at random in each run and remain the same for the whole experiment. Two hyperparameters were evaluated. As backbones, ResNet18 [5], Wide-ResNet50-2 [15] and EfficientNet-B5 [14] were used. The number of embedding dimensions was set to 100 or 200 per patch, as using the full embedding vector can lead to memory issues.

For evaluation, the receiver operating characteristic (ROC) curve is calculated from both the normal and abnormal test set, and the area under the ROC curve (AUROC) is used to compare our models. The intersection over union (IOU) metric is normally better suited, but requires ground truth segmentations, which are not available in the two datasets.

The ROC curve is generated based on the anomaly scores of the images, i.e. the mean pixelwise difference in case of the VAE models, and the anomaly score maps in the PaDiM case. For the VAE, the threshold which maximizes the difference between true and false positives is selected, and for PaDiM, the threshold that maximizes the F1 score is used. Samples with anomaly scores higher than the generated threshold are identified as anomalies. For PaDiM the threshold is used for creating the binary mask and will have an impact on the segmentation result as well.

Model	**AUROC**
Vanilla VAE	0.93
U-Net-based VAE	0.97
ResNet18-100/-200	0.96 / 0.98
WideResNet50-2-100/-200	0.97 / 0.97
EfficientNet-B5-100/-200	0.97 / 0.98

Table 1. AUROC for reconstruction-based methods and PaDiM on Concrete-Cracks. Reconstruction-based methods are the VAE and U-Net-based VAE. For PaDiM, three backbones were used and two different amounts of embeddings per patch, 100 and 200.

4.3. Evaluation on the Concrete-Cracks dataset

The performance of reconstruction-based models, as well as the PaDiM models with different backbones and number of embedding dimensions are shown in Table 1. It can be seen that all methods achieved good performance with AUROC close to 1.00, meaning they can easily distinguish between the two classes.

4.4. Evaluation on SDNet2018 Dataset

In Table 2 the mean AUROC from 5 experiments for the reconstruction-based methods and PaDiM with dif-

Model	Mean AUROC
Vanilla VAE	0.51
U-Net-based VAE	0.52
ResNet18-100/-200	0.63 / 0.64
WideResNet50-2-100/-200	0.63 / 0.63
EfficientNet-B5-100/-200	0.67 / 0.67

Table 2. Mean AUROC over 5 experiments for reconstruction-based methods and PaDiM on SDNet2018. Reconstruction-based methods are the VAE and U-Net-based VAE. For PaDiM, three backbones were used and two different amounts of embeddings per patch, 100 and 200.

ferent backbones and number of embeddings is shown. Reconstruction-based methods achieved low performance with AUROC close to 0.50, meaning they can hardly distinguish between normal and abnormal samples. In comparison to that, the PaDiM models showed an improvement with a mean AUROC of up to 0.67.

Figure 4 show the reconstruction and the pixelwise difference from the U-Net-based VAE for a normal and abnormal sample from the SDNet2018 dataset. Firstly, it can be seen that the small hole in the normal sample is not reconstructed and appears in the pixelwise difference. This results in an increased anomaly score although the hole is considered a normality. Secondly, the crack in the abnormal image could not be reconstructed, which is beneficial to our approach, but since the cracks in the SDNet2018 dataset can be very small they do not contribute much to the anomaly score of a sample. Therefore, it is hard to find an appropriate threshold to distinguish the two classes. A similar behavior can also be observed with PaDiM which considers the small holes as anomalies as well. A false positive and a true positive example can be found in Appendix B.

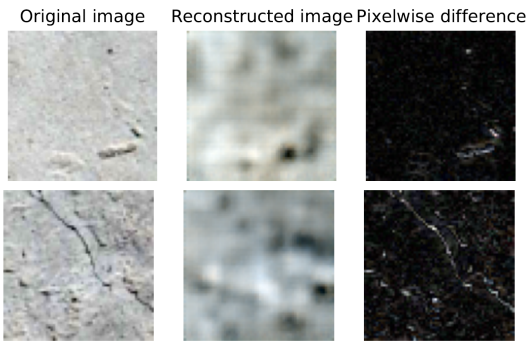

Figure 4. Normal sample (top) and abnormal sample (bottom) from SDNet2018 reconstructed by a U-Net-based VAE, as well as the pixelwise difference between the original and the reconstructed image.

PaDiM-ResNet18-100	# on Vanilla SDNet	# on cleaned SDNet
SDNet2018	0.63	-
SDNet2018 cleaned	0.64	0.67

Table 3. Mean AUROC results over 5 experiments for PaDiM-ResNet18-100 on the vanilla SDNet2018 and VAE cleaned versions

4.5. SDNet2018 challenges

Both types of evaluated methods experienced difficulties with the SDNet2018 dataset. It appears to be noisy and unclean, and has three categories of which the pavement category differs the most. Our models falsely recognize certain objects as anomalies. Moreover, many images contain hardly visible cracks. Therefore, the models cannot find a proper threshold which separates the two classes. This threshold choice also has an impact on the resulting segmentations in PaDiM.

4.6. Attempted solutions

We attempted to automatically clean the SDNet2018 dataset using a U-Net-based VAE, which was pre-trained on Concrete-Cracks. Each image from the SDNet2018 is assigned an anomaly score, which is the prediction of the VAE. The normal images that lie within the highest 10% of the scores are deleted, as many of these are considered abnormal by the VAE. The abnormal images that lie within the lowest 10% of the scores are also cleaned out, as many of them contain hardly visible cracks. The performance comparison between models trained on the vanilla SDNet2018 and the cleaned version, are shown in Table 3. After the cleaning, the mean AUROC over five experiments for PaDiM-ResNet18-100 is improved from 0.63 to 0.67.

We also tried to clean the dataset using a thresholding method but that had no improvement in the performance of PaDiM. The method is explained in Appendix C.

5. Conclusion

In this paper we applied reconstruction-based methods and an embedding similarity-based metho for anomaly detection in construction sites. Both methods achieved good performance on the Concrete-Cracks dataset, while the SDNet2018 dataset turned out to be more difficult. Reconstruction-based methods were not able to achieve good results on it. The PaDiM models showed some improvements and the performance could be further increased by using different backbones, increasing the number of embedding dimensions or through dataset cleaning.

As a future work, manual cleaning of the SDNet2018 dataset could be attempted, which we think could greatly improve the performance of both types of methods. In addi-

tion, a better choice of the threshold could be used in PaDiM to receive better segmentations. These segmentations could then also be applied in a weakly supervised manner, for example by adding a small amount of anomaly samples to the training and using their segmentations as ground truths.

References

[1] Isra Al-Turaiki and Najwa Altwaijry. A convolutional neural network for improved anomaly-based network intrusion detection. *Big Data*, 9(3):233–252, 2021.

[2] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. *CoRR*, abs/1804.04488, 2018.

[3] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *CoRR*, abs/2005.02357, 2020.

[4] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. *CoRR*, abs/2011.08785, 2020.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[7] Marc Maguire, Sattar Dorafshan, and Robert J. Thomas. Sdnet2018: A concrete crack image dataset for machine learning applications. utah state university., 2018.

[8] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *National Institute of Science of India*, 1936.

[9] Nasaruddin Nasaruddin, Kahlil Muchtar, Afdhal Afdhal, and Alvin Prayuda Juniarta Dwiyantoro. Deep anomaly detection through visual attention in surveillance videos. *Journal of Big Data*, 7(1):1–17, 2020.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[11] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021.

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[13] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *CoRR*, abs/1703.05921, 2017.

[14] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019.

[15] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press, 2016.

[16] David Zimmerer, Fabian Isensee, Jens Petersen, Simon Kohl, and Klaus H. Maier-Hein. Unsupervised anomaly localization using variational auto-encoders. *CoRR*, abs/1907.02796, 2019.

[17] Çağlar Fırat Özgenel. Concrete crack images for classification, 2019.

A. AnoVAEGAN

AnoVAEGAN is a combination of a VAE and a GAN. Both the VAE and the GAN have individual drawbacks. The reconstructed images of Autoencoders are often blurry and the training of a GAN is generally unstable, although the quality of the reconstructions are high. To avoid these problems both concepts can be combined into AnoVAEGAN [2]. The AnoVAEGAN framework consists of an encoder, a decoder and a discriminator. The VAE is used to reconstruct input samples and the discriminator is used to discriminate its input as either real or reconstructed. The framework is optimized by using two loss functions. The loss function for the VAE, $\mathcal{L}_{VAE}$, and the loss function for the discriminator, $\mathcal{L}_{Dis}$, are defined as

$$\begin{aligned}\mathcal{L}_{VAE} &= \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{prior} + \lambda_3 \mathcal{L}_{adv} \\ &= \lambda_1 \|\mathbf{x} - \widehat{\mathbf{x}}\|_1 + \lambda_2 D_{KL}(\mathbf{z}||\mathcal{N}(0, I)) \\ &\quad - \lambda_3 \log(\mathrm{Dis}(\mathrm{Dec}(\mathrm{Enc}(\mathbf{x}))))\end{aligned} \tag{4}$$

$$\mathcal{L}_{Dis} = -\log(\mathrm{Dis}(\mathbf{x})) - \log(1 - (\mathrm{Dis}(\mathrm{Dec}(\mathrm{Enc}(\mathbf{z}))))) \tag{5}$$

[2]. The VAE is trained using a weighted sum of the reconstruction error, the KL-divergence and the adversarial error. The meaning of the first two terms is similar to its usage in the vanilla VAE. The third term forces the decoder to generate images that are likely to deceive the discriminator. Meanwhile the discriminator is trained to distinguish between real and reconstructed samples.

B. PaDiM on SDNet2018

In Figure 5, a true positive and a false positive example is shown that was created using PaDiM with an EfficientNet-

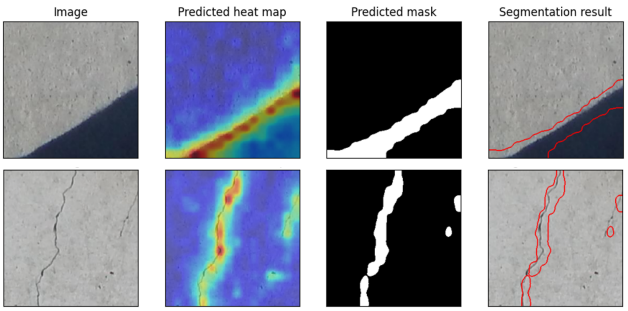

Figure 5. False positive (top) and true positive (bottom) from SDNet2018 classified by PaDiM-EfficientNet-B5-200

PaDiM-ResNet18-100	**# on Vanilla SDNet**	**# on cleaned SDNet**
SDNet2018	0.63	-
SDNet2018 cleaned stddev	0.64	0.64
SDNet2018 cleaned percentile	0.63	0.64

Table 4. Mean AUROC results for PaDiM with a ResNet18 as the backbone on the vanilla SDNet2018 and the threshold cleaned versions

B5 backbone and 200 as the number of embeddings per patch. The threshold that is generated is not as strict as in the Concrete-Cracks case and achieves better overall anomaly localization. However, similar to the U-Net-based VAE, the PaDiM model falsely assigns high anomaly scores to certain objects such as small holes and boundaries between differently colored surfaces. The true positive image shows that in some cases anomalies can be visualized well. However, the issue with hardly visible cracks persists and some of them might not be detected well even on the heatmap.

C. SDNet2018 Threshold Cleaning

As PaDiM tries to model normality using the normal training images, we suspect that having a very clean dataset is crucial to the model performance. Since the SDNet2018 does not provide that, we try to automatically clean it by using a thresholding method which is done per category of the SDNet2018 individually. For each image in the current category the pixel-wise mean is calculated and saved to a list. Then a lower and an upper threshold is calculated and used to delete images from the list that are below the lower threshold and above the upper threshold. We evaluated two choices of choosing the thresholds. The first choice is taking the mean μ of the list of pixel-wise means as well as the standard deviation σ and calculate the thresholds as $\mu \pm \sigma$. The second choice is selecting the lower and upper thresholds so that the lowest 5% and highest 5% of the images are deleted.

Figure 6. Deleted normal images from SDNet2018 using the cleaning method with the percentile threshold

In Table 4 the AUROC scores are presented for the vanilla SDNet2018 and the cleaned versions using PaDiM and a ResNet18 as a backbone. The results show that using a thresholding method to clean the dataset is not working to the extent that it is beneficial to PaDiM or they show that PaDiM in general cannot perform well on this dataset.

Some of the deleted normal images are visualized in Figure 6. Clearly, some of these images should not be included in the training, for example the ones that contain the black foil, as they cannot be used to learn the normality of concrete surfaces. One downside of the automatic cleaning is, that normal images which would be beneficial to training like the ones in the top row, get deleted as well. Therefore a manual cleaning approach would be better suited.