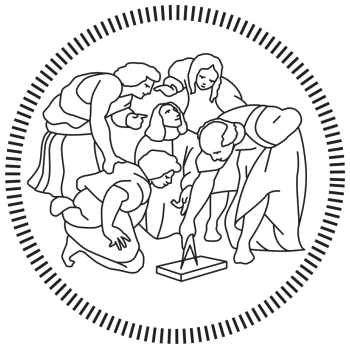
POLITECNICO
MILANO 1863

Image Analysis and Computer Vision



Anomaly Detection Based on Autoencoder and Denoising Convolutional Neural Network

Supervisor:

Prof. Vincenzo Caglioti — Politecnico di Milano

Co-Supervisor:

Dr. Giacomo Boracchi — Politecnico di Milano

Şemsi Yiğit Özgümüş

Matriculation ID: 893558

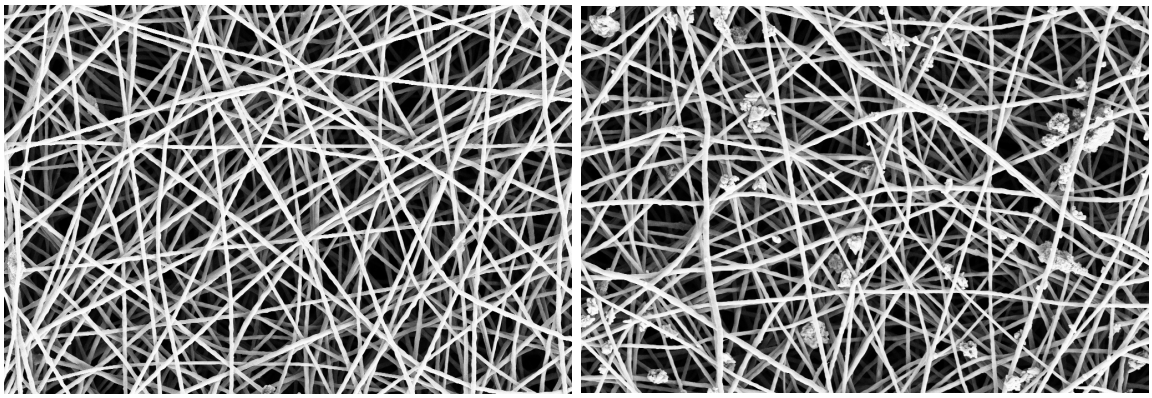
Yusuf Yiğit Pilavci

Matriculation ID: 892973

July 2019

Introduction

Anomaly or novelty detection can be defined as finding patterns in given data that has new or unexplained characteristics [6]. It is a significant topic in the computer vision studies because it has the potential to provide solutions to vast array of application domains. The underlying reason for this significance stems from the fact that depending on the context, unobserved or unexplained data can be defined in some form of anomaly and generally the anomalous data may provide insightful, sometimes critical information that can be used for the benefit of the system. For example in IT industry, network security systems uses anomaly detection methods to scan through abnormalities in the network traffic to prevent intrusion attacks. In finance, the expenditure of the clients are monitored against unusual activities to detect potential frauds. Industrial production lines can use anomaly detection systems to monitor both the quality of the manufacturing process and performance of the machinery for maintenance purposes. Developments in technology and the increased importance of data makes anomaly detection a significantly relevant issue to solve.



(a) SEM Image sample with no anomalies (b) SEM Image sample with anomalous regions

Figure 1: Samples from SEM Image Dataset

The problem that we particularly examine is automated detection of defects in nanofibrous materials which are widely used in various fields such as sensors, medicine, filtration and water treatment [9]. For such examination, we analyze Scanning Electron Microscope images(see Fig.1(a)). Defects which occur in production process are mostly visible flatten areas or beads which basically create non-uniformity (see Fig.1(b)) with respect to other parts of analyzed image and automated detection of these particular ares saves considerable time and cost.

In terms of problem definition of anomaly analysis, detection task of defects in SEM images has a perfect fit. As aforementioned, defected parts create non-uniformity w.r.t. the rest of image which we address as "anomaly" for the rest of paper. The main task is, as done in other anomaly detection tasks,to mark the defected areas in the image to discriminate those from the "anomaly-free" areas. To do so, in perspective of machine learning, we design an unsupervised classification network that aims to learn the distribution of "anomaly-free" images without any knowledge on anomalous parts in training time. In the run-time, the same network outputs "anomaly score" that is derived from the deviation of input item from trained normality distribution. Better quality of this score results in better accuracy of anomaly detection. Our proposed method is identified as a reference-free model which will be explained in following section.

Related Work

In this section, we list the previous works and state of the art related to anomaly detection in images and proceed with particular case of SEM images. Proposed methods for anomaly detection in images can be grouped in reference-based and reference-free ones [6]. Reference-based methods compares the input image with a template image which is anomaly-free by definition and result of this comparison decides in run-time whether input includes anomalous component or not [14]. However, they are not feasible for considered SEM images where filament structures follow pseudo random rather than geometric pattern [4].

On the other hand, reference-free methods are more applicable because they can detect anomalies by using features that can consistently create varying responses for normal and anomaly regions [4]. This concept is also well-known as novelty detection [11]. In this project, we followed the methodologies under this branch.

This problem of particular dataset of SEM images previously studied in [4], [5] and [3] and these methods are based on learning a sparse dictionary that represents the normal regions and in run-time, if the input does not conform with the learned dictionary, it is detected as anomaly. Another approach that introduces convolutional neural networks is established by Napoletano et al. [9]. To extract feature for "normality" concept, they use a pretrained network called ResNet which consists of residual networks and trained by large number of images [7]. After feature extraction, for test phase, they create a dictionary by using K-means clustering to have the representation of normal regions.

Another powerful model for novelty detection problems is auto-encoders. Auto-encoders are quite useful tools to learn the dynamics of a distribution, especially defined over images and to generate examples that corresponds to a sample on learned distribution. Therefore, they are easily deployed for the motivations of novelty detection as done in [1], [8] and [10]. Particularly, in Sabokrou et al.[13], leverages two staged network that is composed by auto-encoders and convolutional

neural networks is explored for the novelty detection. By having these priors, in our work, we exploited benefits of auto-encoders [2] and denoising network [12] to create the representations that can give the understanding of normality.

Methods

Preprocessing

The proposed method aims to learn the normality concept from the anomaly-free image regions. Therefore, rather than deploying one complete image to the overall network, one needs to use patches from images to directly analyze local regions. Therefore, the dataset is needed to be decomposed to smaller patches with predetermined sizes. We empirically set patch size to 32×32 for both train(normal images) and test(images with anomaly) regarding trade-off between computational cost and precision.

For the training images, patches locations are randomly selected and for the test images, overlapped regions are extracted as different patches to create enough number of samples for more reliable analysis of the model.

Overall Architecture

Architecture of the proposed method can be seen in figure 2. It consists of an autoencoder network and a denoising network. The Specifics of each network will be explained in their own sections.

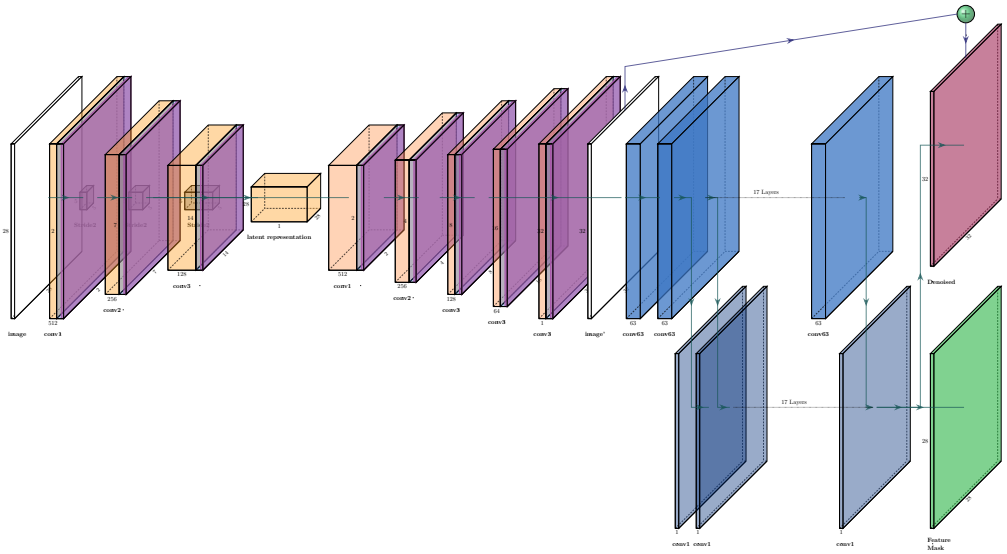


Figure 2: Proposed Anomaly Detection Architecture

Autoencoders

Autoencoder network is selected as the foundation of the proposed anomaly detection framework for two main reasons. At first, it provides a substantial performance in both learning the latent representation and creating reconstructions from those latent representations as mentioned in the related works. Other reason is that we wanted to observe the inclusion of the denoising network to the performance of the anomaly detection method so we wanted a reconstruction based approach as a basis. We implemented 3 different types of autoencoder networks to test their performance and decide which type is suitable for the denoising network addition. These are namely:

- Standard Auteoncoder
- Denoising Autoencoder
- Convolutional Variational Autoencoder.

In terms of capacity, all the autoencoder variants have the number of layers with the exception being convolutional variational autoencoder which has no batch normalization after each layer. Autoencoder architecture is the combination of an encoder and decoder network. Encoder network creates a mapping from image(X) to latent space(Z) and decoder creates the inverse mapping.

$$\phi : X \mapsto Z \quad \text{(Encoder)}$$
$$\psi : Z \mapsto X \quad \text{(Decoder)}$$

The primal training objective for the autoencoders is the reconstruction loss. Depending on the architecture of the decoder or the desired task, the source of the parameters for the objective function can change. Autoencoder learns the representation of the data by minimizing the reconstruction loss.

Standard autoencoder learns to encode the given data and optimized to reconstruct it from the latent representation. Denoised autoencoder adds a noise to the input data as a factor of distortion to make the reconstruction and latent representation learning more robust. Different from the first two models, CVAE learns the first and second moment of the probability distribution of the latent representation. It uses reparameterization to sample the latent representation and obtain reconstruction. The objective functions to train these models are provided below.

$$\mathcal{L}_{SAE} = \|x - (\psi \circ \phi)(x)\|^2, \quad x \sim X \in \mathbb{R}^d \tag{1}$$

$$\mathcal{L}_{DAE} = \|x - (\psi \circ \phi)(\hat{x})\|^2, \quad \hat{x} = x + \epsilon : x \sim X \in \mathbb{R}^d, \quad \epsilon \sim \mathcal{N}(\mu, \sigma^2) \tag{2}$$

$$\mathcal{L}_{CVAE} = -\mathbb{E}_{z \sim q_\phi(z|x_i)} \left[\log p_\psi \left(x_i|z\right)\right] + \mathbb{KL}\left(q_\phi \left(z|x_i\right) \|p(z)\right) \tag{3}$$

Equation 1 and 2 represents the ℓ_2 norm of the difference between the reconstructed sample and the input. Equation for CVAE consists of two terms. First

one represents the reconstruction loss and the second term is a regularizer for the training. KL means Kullback-Leibler divergence between the encoder's conditional distribution and latent space distribution. This divergence measures how much information is lost when using decoder to reconstruct from the encoded image.

The anomaly score for detection of anomalies of the autoencoder architectures is computed using the second norm of the reconstructed query samples. For each model, anomaly score equation is provided below. Performance metrics for autoencoder architecture alone are provided in table 1.

$$\mathcal{A}_{SAE}(x),\ \mathcal{A}_{DAE}(x),\ \mathcal{A}_{CVAE}(x) = \|x - (\psi \circ \phi)(x)\|^2 \tag{4}$$

To improve the performance of the anomaly detection, we added denoiser network to the anomaly detection pipeline. Next section will explain how it is integrated to the architecture.

Denoising Network

In latter part of our network, we adapt CNN-denoising network proposed in [12]. The original purpose of this network is the removal of the noise from the input image and it is trained in the supervised way in which input is image with added Gaussian or Poisson noise and it is compared with the image without noise. Previous results in [12] already reports the purposes the different layers and properties of output. Regarding these and their experimental results, one can conclude that in denoising network, original output of denoising operation results in smoother representation of the image w.r.t. noisy image. They also report that throughout the certain layers' outputs (bottom part of denoising network in Fig. 2), converge to determine statistics on noise which will be referred as the mask for the rest of report. However, empirically, it is not only the captured information and, from the reported qualitative results of different layers' output, one can easily notice that some significant (edge-like) features can be also captured through this process. Having this prior, we extracted additional output out of the original denoising network.

In overall, output of our network may leverage either of two outputs of this network which corresponds to different enhanced representations. In particular, first one is the original output of the network which is the result of addition of mask on input image for the denoising and it is expected to be smooth. On the other hand, the second one is only the mask and we also observe that in our problem, it is more likely that it corresponds to significant features of previously generated representations. In Fig. 2, as aforementioned, one can find in second part of overall architecture (denoising network), purple and green outputs are the original (smooth) output and mask(feature representation), respectively.

Training of this network is done sequentially, meaning that in the first part of the training autoencoder network is trained and then autoencoder is used with its fixed weights to provide reconstructed samples to denoising network in the second

stage of the training. This increases the total training time slightly but since the computation graph is also divided into separate training stages the difference is negligible.

The loss function to training of denoising network naturally differs from original one as the eq. 5 states:

$$\mathcal{L}_{Den} = \|x - Den((\psi \circ \phi)(x))\|^2 \tag{5}$$

Notice that in eq. 5, we compare input of overall network (not the input of denoising network) with the output that we obtain. Additionally, during the experiments, we observe slow process for learning due to small gradient values therefore, similarly done in Autoencoder training, we add Gaussian noise to input of denoising network.

In the test phase, two different anomaly scores which based on two different output of the image are computed as following:

$$x_{rec} = (\psi \circ \phi)(x), \quad x \sim X \in \mathcal{R}^d \tag{6}$$

$$Den(x_{rec}) \mapsto x_{den}, \ f_{mask} \tag{7}$$

$$\mathcal{A}_{Smooth} = \|x - x_{den}\|^2 \tag{8}$$

$$\mathcal{A}_{Feature} = \|x - f_{mask}\|^2 \tag{9}$$

Higher anomaly scores indicate higher probability for occurrence of an anomaly in the tested patch. Since in both cases, the overall network is supposed to learn the distribution over normal samples, outputs will generate high values in anomaly cases. Thanks to the refinement of denoising network, we present a quite reliable and robust anomaly score to distinguish normal and anomalous patches.

Experiments and Results

This section will present conducted experiment results and provide an analysis regarding the inclusion of the denoiser network to the anomaly detection task.

Performance Metrics

For the interpretation of the experiment results, we used 4 performance metrics. These are precision, recall, F1 score and area under receiver operating characteristic curve (AUROC). **Recall** is the ability of a model to find all the relevant cases within a dataset. In our case detection of all anomalies in a test would give us a recall of 1.0. **Precision** on the other hand is the ability of a classification model to identify **only** the relevant data points. While recall expresses the ability to find all relevant

instances in a dataset, precision expresses the proportion of the data points our model says was relevant actually were relevant. The calculation of the both metrics is given below

$$\textbf{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{10}$$

$$\textbf{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{11}$$

Precision and recall comprises a trade off situation. If the model favors the precision, the recall decreases because eliminating the false positives inadvertently increases the false negative rate and vice versa. To give equal importance to both metrics, **F1 score** is used. The F1 score is the harmonic mean of precision and recall taking both metrics into account in the following equation:

$$\mathrm{F}_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{12}$$

The last metric we use to interpret the model performance is area under receiver operating characteristic curve, **AUROC** for short. ROC curve visualizes the trade of relationship between the false positive and true positive rate. True positive rate is actually recall. False positive rate is the probability of false detection for the system. The calculations for these rates are provided below:

$$\textbf{True Positive Rate} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{13}$$

$$\textbf{False Positive Rate} = \frac{\text{false positives}}{\text{true negatives} + \text{false positives}} \tag{14}$$

the AUROC value can be obtained by calculating the area under the ROC curve which has a range between 0 and 1 with a higher number indicating better classification performance.

Experiment Settings

The experiments are all part of the same training cycle. All the models trained on the same GPU with the same batch size and the epochs to preserve consistency. All the configurations related to the results presented in this report can be found in the project repository[1] Highest F1 score is considered as the primal factor to compare the other metrics. Additional metric graphs (Figure 5, 3) and histograms(Figure 4) can be found in the appendix. Since the training method of the networks is one after the other, all the different anomaly score computations are done on the same experiment. Hence we divided each of our experiments into three stages.

[1] https://github.com/yigitozgumus/IACV_Project

Experiments

In the first stage, we trained the autoencoder network to have an initial observation about its performance on the dataset. Table 1 shows the score for all three models with their reconstruction scores.

Table 1: Anomaly Detection scores of the implemented models alone

Autoencoder Reconstruction				
Models	**Metrics**			
	AUROC	Precision	Recall	F1 Score
Autoencoder	0.63934	0.15418	0.24354	0.18882
Denoising Autoencoder	0.61357	0.12040	0.28527	0.16933
C. Variational Autoencoder	**0.70764**	**0.19773**	**0.31233**	**0.24216**

Convolutional variational autoencoder performed better than other models in terms of overall performance. Unlike AE and DAE, CVAE learns the mean and the variance of the distribution of the trainig dataset as opposed the latent representation itself, so learned latent representation is sampled from the mean and variance to obtain the reconstruction. AE and DAE learns to reconstruct the given image better than CVAE in this case but while they become better at reconstructing, they inadvertantly learn to reconstruct some parts of the anomalous regions in some test examples. The placement and the contrast of the anomalous regions are also a contributing factors to this performance loss.

Table 2: Anomaly Detection scores with the addition of Denoiser Network as reconstruction smoother

Reconstruction Smoothing with Denoiser Network				
Models	**Metrics**			
	AUROC	Precision	Recall	F1 Score
Autoencoder	**0.78648**	**0.26277**	**0.41507**	**0.32181**
Denoising Autoencoder	0.75003	0.22560	0.35636	0.27629
C. Variational Autoencoder	0.71852	0.21012	0.33190	0.25733

In the table 2, the results from the reconstruction smoothing is presented. Here the reconstructed sample is fed through to the denoising network and the denoised output of this network is used to compute the anomaly score. Denoising network normalizes the reconstruction constrast by applying the layered convolutions. The contrast differences in the denoised reconstructions are much more similar to the original samples. All models shows an improved performance and both their recall and precision scores are improved.

Table 3: Anomaly Detection scores with the addition of Denoiser Network as feature extraction method

Feature Mask from Denoiser Network				
Models	**Metrics**			
	AUROC	Precision	Recall	F1 Score
Autoencoder	0.76437	0.55964	0.35361	0.43338
Denoising Autoencoder	**0.80022**	**0.60295**	**0.38098**	**0.46692**
C. Variational Autoencoder	0.68515	0.18650	0.29460	0.22841

In our third experiment we focused on the secondary output of the denoising network. [12] creates a 1 convolution from each main 63 convolution layer and adds to the image to apply denoising. After the training, these added de-noising filters become a feature mask of the image and extracted from network as a secondary output. Table 3 shows the performance of the proposed method with this feature being used to compute anomaly score as in equation 9. CVAE model didn't improve its overall performance because the learned feature mask is obtained from the reconstruction and since it is sampling based and not directly obtained from the input image, using the feature mask didn't improve neither precision nor recall. Standard AE and especially DAE obtained a significant performance increase with the feature mask obtained from the denoising network.

Conclusion

In this project, we implemented an anomaly detection model using autoencoder network and explored the addition of denoising network and exploited for the superior performance. In our experiments, out of all three different autoencoder networks, convolutional variational autoencoder, in its own, achieves the best performance in terms of previously explained metrics. We speculate that such performance improvement stems from the imperfections in its reconstructions which are derived by decoding of a sampled latent representation. While this provides with an advantage in the reconstruction based anomaly detection, CVAE does not improve the performance of the merged model with denoising network, in fact, it performs worse with the feature mask based anomaly detection. Reconstructions acquired from the autoencoder and denoising autoencoder network however improved after the additional denoising operation. Denoised reconstructions provided better anomaly scores in all the performance metrics.

As aforementioned, in the overall architecture that we propose, we manage to extract and analyze different outputs of denoising network which result in reconstruction smoothing and feature mask. Based on these two representation, two different anomaly scores become available by computing the residual of them with query image and our experimental results clearly indicate the latter one is able to

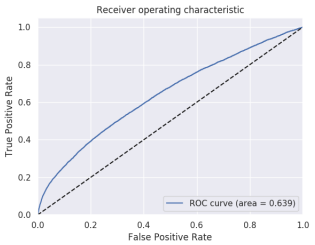
instantiate a combination that outperforms all the others (see Table 3). In particular, the comparison between Table 2 and 3 shows that feature mask anomaly score with the combination of autoencoder network had a small decrease in its capacity for recall while obtained double of its previous precision score. On the other hand, with the combination of denoising autoencoder network improved both the precision and recall considerably, by giving the best performance among all the models.

With a closer look on the layers of denoising network, we observe that within 10 layers, created feature mask is very similar to the output of 20 layers and they behave quite similar in terms of the performance, also. In other words, the features of the SEM images might be efficiently extracted in very early layers of the network and this observation of ours with SEM dataset is surprisingly consistent with the observations of Remez et al. [12] related to outputs of intermediate layers.

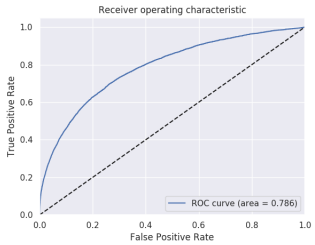
In overall, we present a state-of-art model for anomaly detection which is an arduous task to formulate, implement and solve for any specific dataset. In our case, our proposed models learn a "normality" concept from SEM image patches without seeing any anomalous image. Regarding the recent works related to this task, natural extension on this work can be replacement of former part with generative adversarial networks (GAN) which are quite powerful and robust for learning distributions for images and generating samples out of these distributions. Another possible future work is to analyze and improve the quality of feature masks derived from denoising network. Regarding performance boost with addition of it, one direction to explore is to search for architectural changes that is more related to motivations of anomaly detection.

Appendix

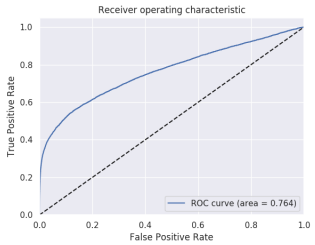
ROC Curves and AUROC Values of the Experiments



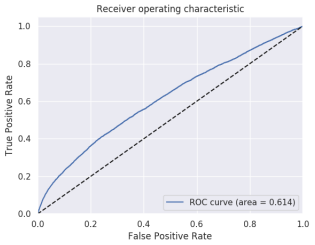
(a) Autoencoder Reconstruction



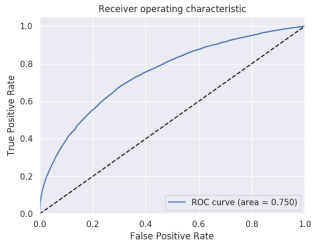
(b) Autoencoder + Denoiser Smoothing



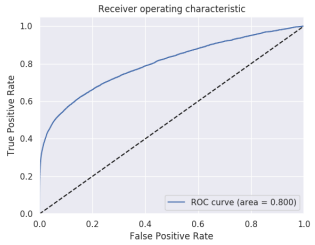
(c) Autoencoder + Denoiser Feature Mask



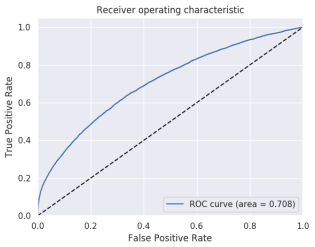
(d) Autoencoder Reconstruction



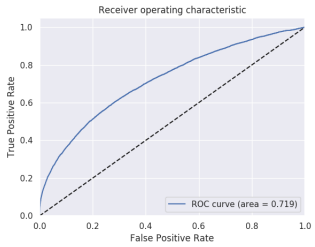
(e) Autoencoder + Denoiser Smoothing



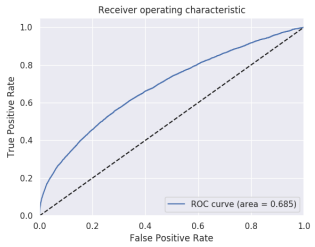
(f) Autoencoder + Denoiser Feature Mask



(g) Autoencoder Reconstruction



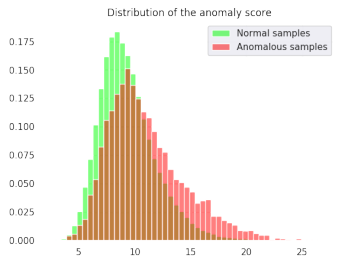
(h) Autoencoder + Denoiser Smoothing



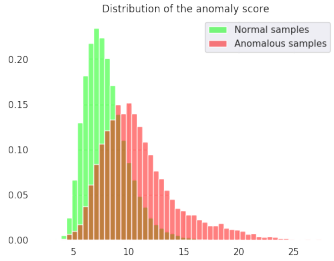
(i) Autoencoder + Denoiser Feature Mask

Figure 3: ROC tables for all the experimented models.the ROC curve shows the TPR/FPR Rate of the model. First row is the normal Autoencoder, second row is the Denoising Autoencoder and the third is the Convolutional Variational Autoencoder

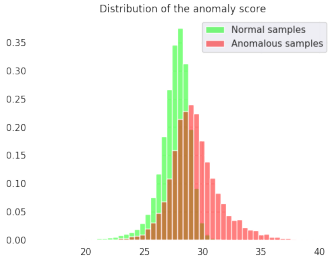
Seperation of the Distribution According to the Anomaly Scores



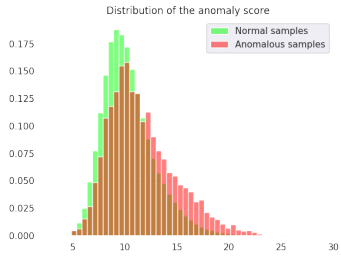
(a) Autoencoder Reconstruction



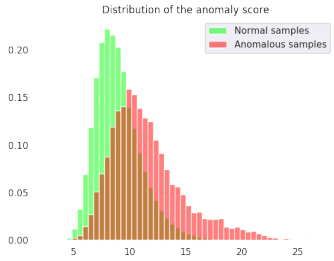
(b) Autoencoder + Denoiser Smoothing



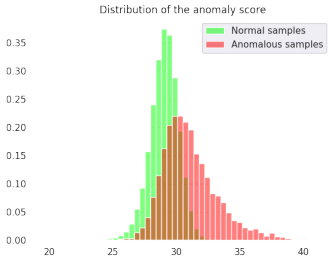
(c) Autoencoder + Denoiser Feature Mask



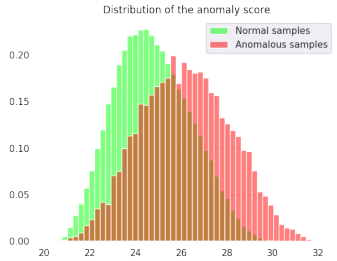
(d) Autoencoder Reconstruction



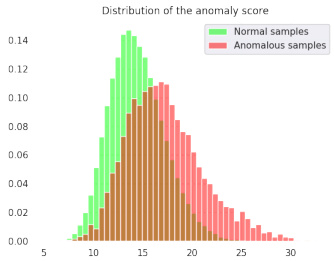
(e) Autoencoder + Denoiser Smoothing



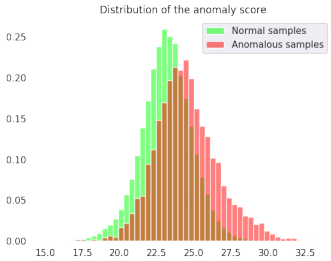
(f) Autoencoder + Denoiser Feature Mask



(g) Autoencoder Reconstruction



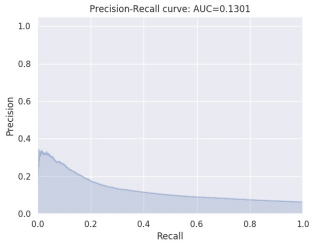
(h) Autoencoder + Denoiser Smoothing



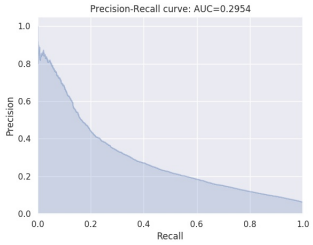
(i) Autoencoder + Denoiser Feature Mask

Figure 4: Histogram tables for the experimented models. Histograms shows the separation of anomalous and normal samples in the test set.First row is the normal Autoencoder, second row is the Denoising Autoencoder and the third is the Convolutional Variational Autoencoder

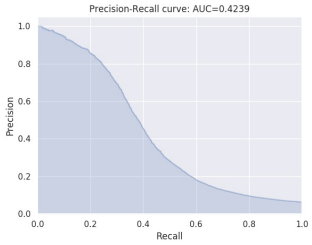
PRC Curve changes of the Models regarding the experiments



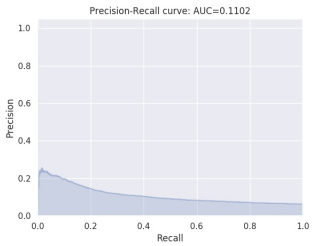
(a) Autoencoder Reconstruction



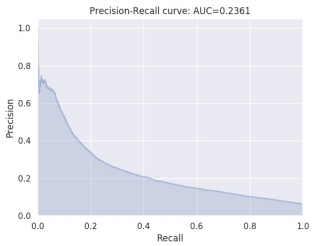
(b) Autoencoder + Denoiser Smoothing



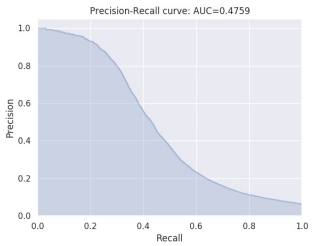
(c) Autoencoder + Denoiser Feature Mask



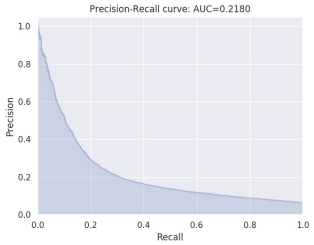
(d) Autoencoder Reconstruction



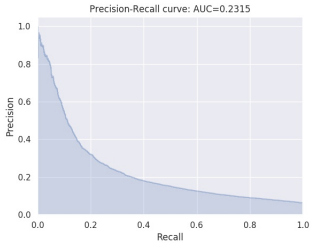
(e) Autoencoder + Denoiser Smoothing



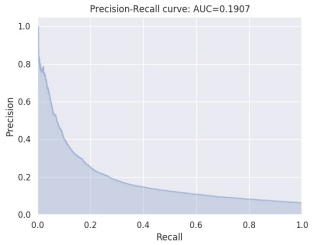
(f) Autoencoder + Denoiser Feature Mask



(g) Autoencoder Reconstruction



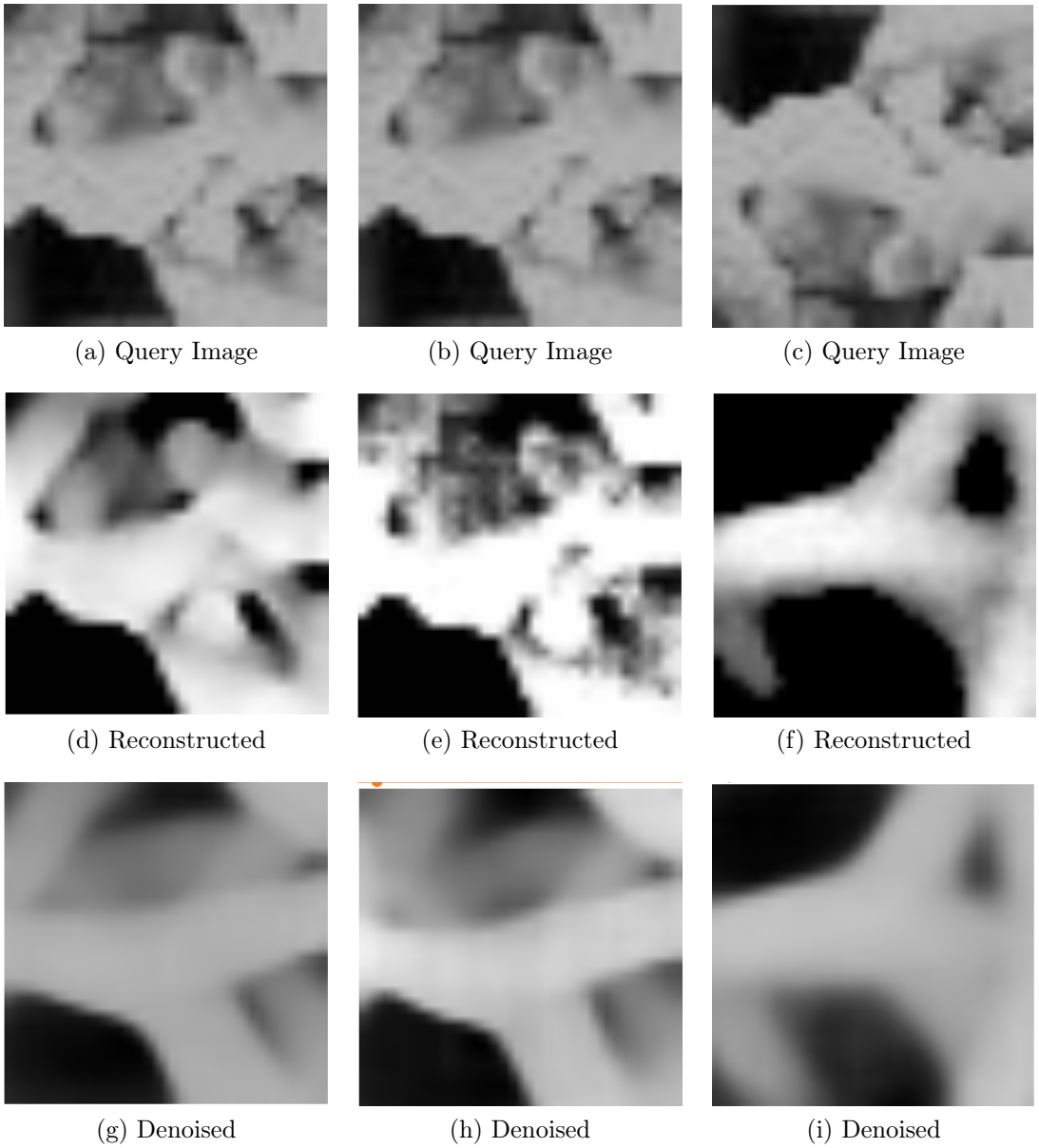
(h) Autoencoder + Denoiser Smoothing



(i) Autoencoder + Denoiser Feature Mask

Figure 5: PRC curves for the experimented models. PRC curve shows the trade off between the precision and recall. First row is the normal Autoencoder, second row is the Denoising Autoencoder and the third is the Convolutional Variational Autoencoder

Image samples from different stages of the Pipeline



(a) Query Image (b) Query Image (c) Query Image

(d) Reconstructed (e) Reconstructed (f) Reconstructed

(g) Denoised (h) Denoised (i) Denoised

Figure 6: Image samples from different stages of the anomaly detection pipeline. First column is from standard autoencdoer, second column is denoising autoencoder and third column is convolutional variational autoencoder

(j) Mask Obtained (k) Mask Obtained (l) Mask Obtained

(m) Residual Image (n) Residual Image (o) Residual Image

Figure 6: Image samples from different stages of the anomaly detection pipeline. First column is from standard autoencdoer, second column is denoising autoencoder and third column is convolutional variational autoencoder

Bibliography

[1] Jinwon An and Sungzoon Cho. "Variational autoencoder based anomaly detection using reconstruction probability". In: *Special Lecture on IE* 2 (2015), pp. 1–18.

[2] Pierre Baldi. "Autoencoders, unsupervised learning, and deep architectures". In: *Proceedings of ICML workshop on unsupervised and transfer learning*. 2012, pp. 37–49.

[3] Giacomo Boracchi, Diego Carrera, and Brendt Wohlberg. "Novelty detection in images by sparse representations". In: *2014 IEEE Symposium on Intelligent Embedded Systems (IES)*. IEEE. 2014, pp. 47–54.

[4] Diego Carrera et al. "Defect detection in SEM images of nanofibrous materials". In: *IEEE Transactions on Industrial Informatics* 13.2 (2016), pp. 551–561.

[5] Diego Carrera et al. "Scale-invariant anomaly detection with multiscale group-sparse models". In: *Proceedings of IEEE International Conference on Image Processing (ICIP)*. Phoenix, AZ, USA, Sept. 2016, pp. 3892–3896. DOI: 10.1109/ICIP.2016.7533089.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.

[7] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[8] Valentin Leveau and Alexis Joly. "Adversarial autoencoders for novelty detection". In: (2017).

[9] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. "Anomaly detection in nanofibrous materials by cnn-based self-similarity". In: *Sensors* 18.1 (2018), p. 209.

[10] Stanislav Pidhorskyi et al. "Generative Probabilistic Novelty Detection with Adversarial Autoencoders". In: *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 6823–6834. URL: http://dl.acm.org/citation.cfm?id=3327757.3327787.

[11] Marco AF Pimentel et al. "A review of novelty detection". In: *Signal Processing* 99 (2014), pp. 215–249.

[12] Tal Remez, Or Litany, and Raja Giryes. "Class-Aware Fully Convolutional Gaussian and Poisson Denoising". In: *IEEE Transactions on Image Processing* 27 (2018), pp. 5707–5722.

[13] Mohammad Sabokrou et al. "Adversarially Learned One-Class Classifier for Novelty Detection". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 3379–3388.

[14] Maria Zontak and Israel Cohen. "Defect detection in patterned wafers using anisotropic kernels". In: *Machine Vision and Applications* 21.2 (2010), pp. 129–141.