Detecting One-Pixel Attacks Using Variational Autoencoders

Janne Alatalo, Tuomo Sipola, and Tero Kokkonen(✉)

Institute of Information Technology,
JAMK University of Applied Sciences,
Jyväskylä, Finland
{janne.alatalo, tuomo.sipola, tero.kokkonen}@jamk.fi

Abstract. In the field of medical imaging, artificial intelligence solutions are used for diagnosis, prediction and treatment processes. Such solutions are vulnerable to cyberattacks, especially adversarial attacks targeted at machine learning algorithms. One-pixel attack is an adversarial method against image classification algorithms based on neural networks. In this study, we show that a variational autoencoder can be used to detect such attacks in the context of medical imaging. We use adversarial one-pixel images generated from the TUPAC16 dataset and apply the variational autoencoder as a filter before letting the images pass to the classifier. The results indicate that the variational autoencoder model efficiently detects one-pixel attacks.

Keywords: Variational Autoencoders, Anomaly Detection, Artificial intelligence, Deep Learning, Machine Learning, One-Pixel Attack, Cyber Defence, Cyber Security, Medical Imaging

1 Introduction

Artificial Intelligence (AI) has a strong role in diagnostic and therapeutic medical imaging [29], including neural network Deep Learning (DL) and Machine Learning, that have grown rapidly and been proved to be valuable in healthcare applications [18,22]. In addition to prediction and diagnosis capabilities, AI applications can be used for upgrading and improving the efficiency of healthcare services, which reduces healthcare costs and shortens patient waiting times [26]. Cyber threats against healthcare are real, as increasing number and types of threats targeting healthcare are expected to become more frequent [20]. Because of the increased usage of AI applications in healthcare, potential and effective threats against AI applications have also been recognized. There are proven attack vectors, some of which are crafted to deceive classification models [25,3].

One-pixel attack is a method for fooling a neural network classifier by changing just one pixel in the input image [28,24,16]. These attacks can also be further developed to make them more difficult for a human to identify [27,15]. Understanding the nature of successful one-pixel attacks is useful for defending against them [2]. One-pixel attack is an example of an adversarial attack, which use a

The original article appeared as: Janne Alatalo, Tuomo Sipola and Tero Kokkonen. "Detecting One-Pixel Attacks Using Variational Autoencoders." In: Information Systems and Technologies (WorldCIST 2022). Ed. by Álvaro Rocha, Hojjat Adeli, Gintautas Dzemyda and Fernando Moreira. Vol. 468. Lecture Notes in Networks and Systems. Cham, Switzerland: Springer, 2022, pp. 611–623. DOI: 10.1007/978-3-031-04826-5_60

modified image to fool a neural network classifier, also in the medical imaging domain [11,3].

In order to defend against an attack, there must be current knowledge about what is happening in the environment. This knowledge of the surrounding environment is called Situational Awareness (SA). When considering well-known definition of SA by Endsley [10] or classical decision-making models (OODA-loop [23] and Gartner's four stages of an adaptive security architecture [19]), it can be seen that sensor information has an important role in achieving proper SA as well as in making correct decisions based on sensor detections. In our study, a similar model is used against one-pixel attacks. The implemented Machine Learning model detects modified pixels from image information.

Furthermore, detection of adversarial attacks has been studied in general [35] but also in relation to medical images [17] and one-pixel attacks [32]. Particularly for pre-trained object detectors, Chiang et al. introduce Adversarial Pixel Masking (APM), which is a data preprocessing MaskNet outputting a mask. The masked input image will be fed to the object detector [8]. Nguyen-Son et al. introduce a one-pixel attack, detection, and defense framework (OPA2D) including three functionalities: Attack Improvement (OPA2D-ATK), Attack Detection (OPA2D-DET) and Attack Defense (OPA2D-DEF), where attack detection is based on classification between the original and adversarial images attacked by one-pixel attacks [21]. Autoencoders have been used for robust anomaly detection in images [5] and adversarial sample detection [30]. Variational autoencoders are also useful for adversarial example detection [6].

The goal of this paper is to detect the one-pixel attacks against digital pathology images. We used a variational autoencoder to create a detector to provide sensor information about the input images indicating whether the images are trying to use the attack or not. Firstly, we introduce the variational autoencoder, anomaly score to rank the images, and evaluation metrics used to measure the success of the approach. Then, the experiment setup is described. Lastly, results are presented with examples.

2 Method

2.1 Variational autoencoder

Let $\boldsymbol{x} \in \mathbb{R}^{N \times M}$ be a sample from the data, where N is the number of samples and M is the dimension of the data. The variational posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with parameters ϕ from the encoder function provides the latent variable $\boldsymbol{z}$ and the probabilistic decoder part with parameters θ is $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. During the training, we want to find the function parameters ϕ and θ that can produce the most faithful reconstructions of the images. Theoretically, the cost function

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p_\theta(\boldsymbol{z})) \tag{1}$$

is maximized during the training of the variational autoencoder. There is a reconstruction term and a Kullback–Leibler divergence D_{KL} between the two probability distributions. The reconstruction term handles the log-likelihood of getting the input image $\boldsymbol{x}$ given the sampled image $\boldsymbol{z}$ that comes from the distribution of the encoder. The divergence term compares the distribution related to the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and the prior $p_\theta(\boldsymbol{z})$. The closer to each other they are, the smaller the term gets. For a more comprehensive explanation of the theory behind variational autoencoders, please see Kingma and Welling [14] or Goodfellow et al. [13].

Variational autoencoder works very similarly to the normal autoencoder architecture, except that the output of the encoder module is a probability distribution. It is this difference that makes it possible to sample the training data in a more fine-grained manner. The power of anomaly detection with autoencoders originates from the model architecture. The model consist of two parts, encoder and decoder. The encoder module encodes the network input to some lower dimensional latent space which the decoder network tries to restore as similar as possible to the original input. Consequently, the network is forced to learn a compressed representation of the input data. Training the network with a dataset containing samples from the same distribution causes the network to learn features that are common between the samples. The model uses these common features to compress the data samples to the latent space with very little information loss. However, when the model gets a sample that is not from the same distribution as the training dataset, the sample might not compress very well to the latent space, and the reconstruction of the sample differs significantly from the original input sample. By comparing the input sample to the decoder output, we can compute a score for how well the sample was reconstructed. This score correlates very well with how similar the input sample was to the training dataset. In other words, how many of the features in the sample the autoencoder network has seen in the training dataset. We can use the reconstruction error score as an anomaly score that we use to classify if the sample is an anomaly or not [13].

2.2 Model architecture

Figure 1 shows the model architecture used in this study. The model consists of two sub models, encoder and decoder. The encoder sub model takes in 64 × 64 RGB color images that are encoded to $\boldsymbol{\mu}, \ln \boldsymbol{\sigma^2} \in \mathbb{R}^H$ pairs, where H is the dimensionality of the latent space, and the $\boldsymbol{\mu}$ and $\ln \boldsymbol{\sigma}^2$ are the mean and log-variance of the normal distribution that the input image is encoded to. Latent vector $\boldsymbol{z}$ is sampled from the encoder output distribution $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma^2})$, and that latent vector is fed to the decoder sub model. The output of the decoder sub model is the same shape as the encoder input $64 \times 64 \times 3$. The $\boldsymbol{z}$ sampling from the latent space distribution is done using the well-known *reparameterization trick* [13,14] that allows the gradient to flow through random node in the model graph.

The reconstruction error is computed by taking the mean squared error (MSE) between the original input image and the decoder output. This reconstruction error is added to the Kullback-Leibler divergence error (KL error) be-

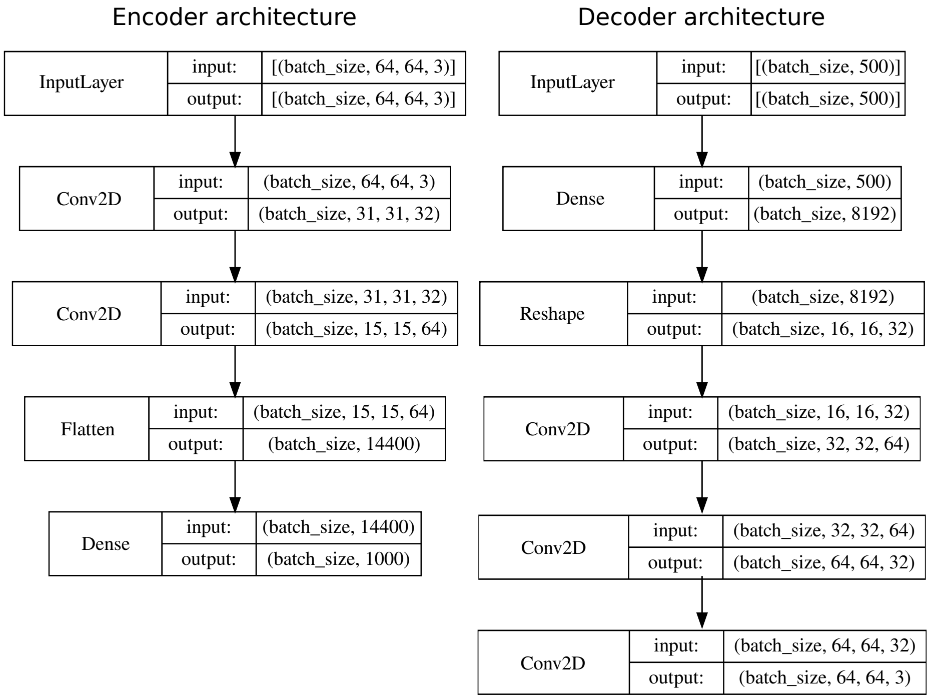
Encoder architecture
InputLayer
input: [(batch_size, 64, 64, 3)]
output: [(batch_size, 64, 64, 3)]
Conv2D
input: (batch_size, 64, 64, 3)
output: (batch_size, 31, 31, 32)
Conv2D
input: (batch_size, 31, 31, 32)
output: (batch_size, 15, 15, 64)
Flatten
input: (batch_size, 15, 15, 64)
output: (batch_size, 14400)
Dense
input: (batch_size, 14400)
output: (batch_size, 1000)
Decoder architecture
InputLayer
input: [(batch_size, 500)]
output: [(batch_size, 500)]
Dense
input: (batch_size, 500)
output: (batch_size, 8192)
Reshape
input: (batch_size, 8192)
output: (batch_size, 16, 16, 32)
Conv2D
input: (batch_size, 16, 16, 32)
output: (batch_size, 32, 32, 64)
Conv2D
input: (batch_size, 32, 32, 64)
output: (batch_size, 64, 64, 32)
Conv2D
input: (batch_size, 64, 64, 32)
output: (batch_size, 64, 64, 3)

Fig. 1: Variational encoder model architecture

tween the sample's latent space distribution and normal distribution with mean 0 and variance 1. The final loss function for the model is presented in Equation 2. In the equation, $\mathcal{S}$ is the set of samples for which the loss is calculated for. Function $e(\boldsymbol{x})$ is the sample's latent normal distribution computed from the decoder output mean and log-variance pairs. Function $d_{\boldsymbol{z}\sim e(\boldsymbol{x})}(\boldsymbol{z})$ is the reconstruction of the image. Here the encoder $e(\boldsymbol{x})$ corresponds to the theoretical $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and the decoder $d_{\boldsymbol{z}\sim e(\boldsymbol{x})}(\boldsymbol{z})$ corresponds to the $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ in Equation 1. Similarly, D_{KL} is the Kullback-Leibler divergence.

$$L(\mathcal{S}) = -\frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{x}\in\mathcal{S}} \mathrm{MSE}(\boldsymbol{x}, d_{\boldsymbol{z}\sim e(\boldsymbol{x})}(\boldsymbol{z})) + \lambda \cdot D_{\mathrm{KL}}(e(\boldsymbol{x}) \parallel \mathcal{N}(\boldsymbol{0},\boldsymbol{1})), \qquad (2)$$

The used loss function in Equation 2 is practically the computable version of the theoretical loss function presented in Equation 1, with the addition of the KL error scaling with λ. Scaling the KL error is a commonly used technique to balance the reconstruction and KL error terms for the best model performance [4].

2.3 Anomaly score

A metric is needed to compare the difference between the images and the reconstructions that the variational autoencoder creates. As adversarial images can be prominently different, we chose a distance metric that emphasizes the extreme distances. Minkowski distance is a generatization of the Euclidean distance. The distance between the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as follows:

$$D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} , \tag{3}$$

where x_i and y_i are the ith element of the corresponding vectors and p is the order of the Minkowski distance [34]. Please note that we use the symbol p to signify this (and not the decoder) in the results section. We interpret a large distance between the image and its reconstruction to indicate that the image does not come from the training data distribution. Finally, a decision threshold for the maximum allowed distance can be set for the detector.

2.4 Evaluation metrics

We use operating characteristic (ROC) curve to evaluate the performance of the detector classifier. The curve is plotted with the true positive rate (TPR) as a function of the false positive rate (FPR). The closer the curve is to the upper left corner of the plot the better the performance is for the classifier. Because the best possible result is a line with its only points in the vertices of the unit square, area under curve (AUC) is a common way of measuring the performance of a classifier [33]. In our case, we measure the performance of the detector, which classifies incoming images as suspicious or normal.

In addition to the TPR and FPR metrics, it is interesting to know how the detector model works to the test dataset after choosing the optimal threshold based on the validation dataset. In this study our proposed method screens the input images for anomalies before the images go to the final classifier. This is a similar solution to the one what Tong et al. propose in [30]. In their paper they note that not all undetected adversarial samples are successful attacks, and for that reason they propose *undetected rate* metric that takes into account how many of the undetected samples were actually successful attacks. This is an important metric because in our previous study [2] we concluded that for one-pixel attack to be effective, the change in the pixel color must be very large. Small color changes in the pixel were not often enough to flip the predicted output class. Based on this, we can conclude that for the one-pixel attack detector to be adequately effective, it only needs to detect very large pixel changes. For that reason, in addition to studying how the detector detects the adversarial images, we also studied how many of the non-detected adversarial images actually were successful attacks.

3 Experiment setup

3.1 Data and target classifier

We used the TUPAC16 dataset, which contains digital pathology images related to breast cancer [31]. The dataset consists of 64 by 64 sized PNG images. The

IBM MAX Breast Cancer Mitosis Detector served as the target classifier. It gives a numeric score that indicates whether the image contains mitoses possibly related to cancer or not [1,9]. We have earlier created several one-pixel attacks against the target classifier [16]. We want to detect those attacks in this research.

3.2 Training and deployment

There are two stages in the experiment. Firstly, the classifier and detector need to be trained using the same data source. In this experiment, we use the already trained target classifier. Training the detector to create a probabilistic model of typical non-attack images is the main focus of this experiment. Secondly, the classifier and the detector are deployed and given images as input. This way we can evaluate how the detector fares against adversarial images. A schematic presentation of the experiment setup is given in Figure 2.

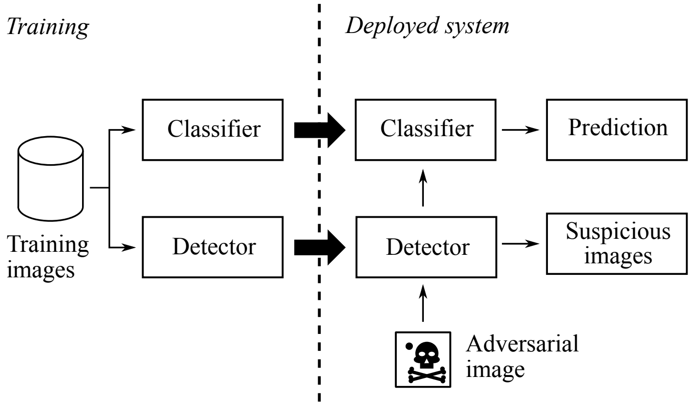

Fig. 2: Schematic presentation of the experiment. The left side shows the training of the classifier and the detector using the same data source. The right side shows the deployed system, where the detector filters incoming images before they reach the classifier.

During the deployment, the first step that happens to an incoming image is to be inspected by the variational autoencoder detector. The detector encodes the image and immediately decodes it as a reconstruction of the image. Then the difference between the images is calculated, which is the Minkowski distance in our case. If the image and the reconstruction are too different, it is taken aside as a potential threat. Otherwise, the target classifier may take the image as an input, and make predictions based on it.

3.3 Training parameters

The model architecture was implemented using Tensorflow deep learning framework. The latent space dimensionality H was chosen to be 500 through experimentation. The KL loss scaling factor λ was set to 0.001. The model was trained using Adam optimizer with learning rate of 0.001 and batch size of 100.

4 Results

4.1 Minkowski distance exponent

After training the detector model with the normal data in the training dataset, we evaluated the performance of different p values for the anomaly score Equation 3. The evaluation was done by plotting the ROC curves for validation dataset with the different p values and calculating the AUC value for the plots. As seen from the Figure 3, by increasing the p value we get better separation for the adversarial and normal image anomaly scores, which can be seen from the ROC plots as a tighter bend for the line and increased area under curve for the higher p values.

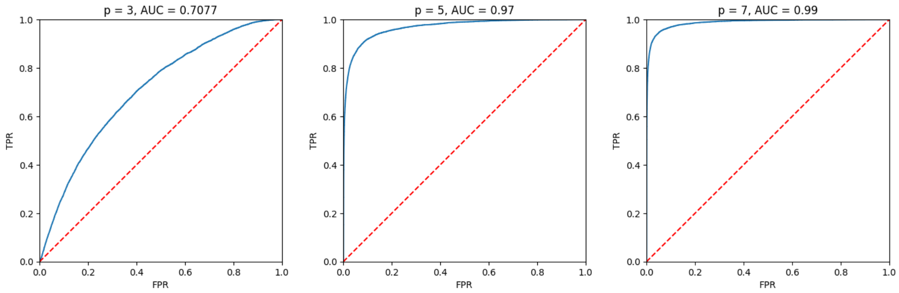

Fig. 3: ROC curves for different anomaly score p values: left $p = 3$, center $p = 5$, right $p = 7$.

Based on the experimentation on the validation dataset, we ended up choosing $p = 7$ for the anomaly score formula. The value gave a very good separation for the adversarial and normal image anomaly scores, and higher p values did not increase the model performance in any considerable amount.

4.2 Anomaly threshold

The anomaly threshold was chosen based on the anomaly scores of the validation dataset. There are different strategies that can be used when choosing the threshold. The different thresholds cause different trade-offs for the model performance. If the threshold is set too low, the model will flag many legitimate

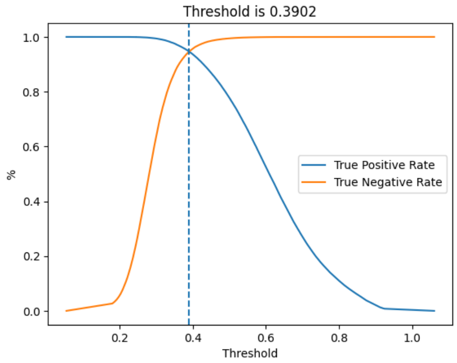

Fig. 4: The threshold value was set to maximize true positive rate and true negative rate values. The dashed vertical line indicates the optimal threshold value.

images as anomalous, reducing the detector usefulness. On the other hand, if the threshold is set too high, some of the adversarial images might not be caught by the system. In this study, we chose to set the threshold so that, the true positive rate and true negative rate, both are maximized with the threshold. Figure 4 visualizes how the threshold was selected.

4.3 Evaluation metrics results

The chosen threshold was then applied to the test dataset by classifying the images as clean or adversarial based on the anomaly score threshold. The results were evaluated as described in the section 2.4. The results of the classification are shown in the Table 1.

Table 1: Adversarial image detection rates. The number of successful attack images that were not detected by the variational autoencoder detector is the most interesting statistics.

Total number of adversarial images 8484			
Detected 7993		**Not detected 491**	
Attack failure 7671	Attack success 322	Attack failure 491	**Attack success 0**

As seen from the results, the detector network works very well for one-pixel attack detection. The detector model catches all the images that would have successfully flipped the prediction result. All the images that were not detected, did not have any effect on the final prediction. When we inspect the individual image reconstructions, we can analyze better how the detector works. Images in Figure 5 show the input images for the detector network on the left, and the reconstructed images are on the right. As seen from the images containing the one-pixel attack modification, the network never adds the pixel modification to the reconstructed image. Otherwise, the image is very accurately reconstructed with only very small details missing. Because we are using the Minkowski distance as our anomaly score, the large error in the reconstruction of the one attack pixel makes the anomaly score high enough for the detector to detect the image as an anomaly. This happens because the large p value in the Minkowski distance formula amplifies large errors in the reconstruction image.

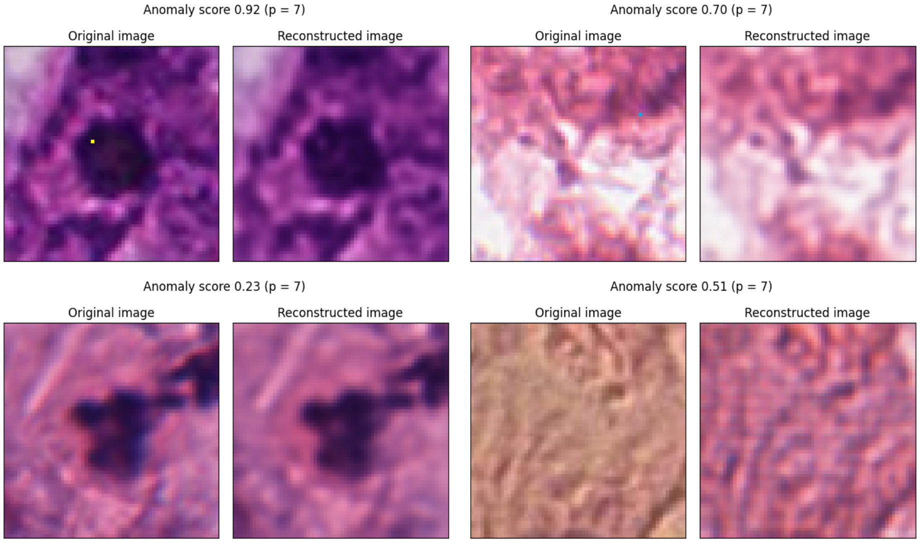

Fig. 5: Examples of image reconstruction by the model. The images on the first row include one-pixel modifications that the model has detected correctly by predicting high anomaly scores. On the second row, there are two unmodified images, of which the left one is predicted correctly not to be adversarial, but the right one is a false positive alert given by our model.

4.4 Tests with images outside the training set

As a sanity check, we also tested the detector network with totally different images that are not anyway related to the training dataset. This was done because the detector network uses very large latent space dimensionality, which can cause the autoencoder network to become too generic, because the latent space is not restrictive enough. We wanted the network to still detect images that are obviously from a totally different distribution. In Figure 6 we tested reconstructing cat images. As seen from the figure, the reconstruction is not very good for these images. In the cat's face image reconstruction, the main features of the face, such as nose, mouth and eyes, are reconstructed very well, but the image colors are way off. The same effect can be seen in the blood sample image; the features are reconstructed well, but the colors are off. From these images we can make the conclusion that the training dataset includes plenty of different shapes that the autoencoder network learns to encode to the latent space, but the dataset colors are very uniform. Although the detector is relaying on the color reconstruction errors to detect the sample images, we can conclude that the network can also detect other types of anomalities instead of only one-pixel attacks.

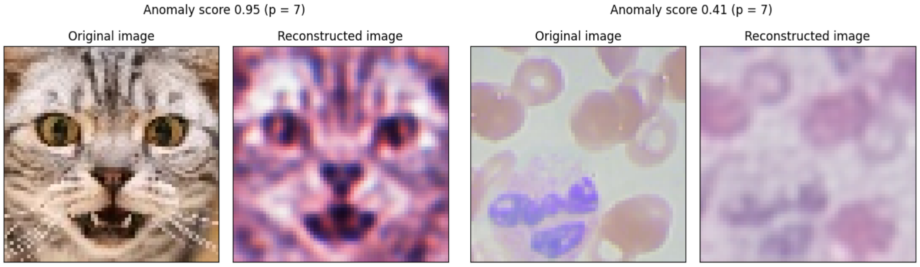

Fig. 6: Examples of image reconstruction for images from two totally different datasets. On the left, there is a reconstruction of an image of a cat's face [12]. On the right, there is a reconstruction of a blood sample image [7].

5 Conclusion

In the modern digitalized world, attacks against AI applications used in medical imaging are a real threat. One-pixel attack is an effective adversarial method, which modifies only one pixel of an image, fooling automated diagnosis. Any tampering with medical data can lead to wrong treatment or cause delays and, in the worst-case scenario, consequences can be catastrophic when considering, e.g., cancer diagnosis by imaging. Therefore, it is extremely important to detect and defend against such attacks. In this study, we have shown that autoencoders can be useful as sensors detecting one-pixel attacks against medical imaging. The probabilistic presentations learned by variational autoencoders are well suited for this task because one-pixel attacks are usually anomalies in the image.

The independently trained autoencoder successfully detected the attacks. The results indicate that in the end, regardless of the machine learning method, the anomaly score is a crucial component of the detection chain. The higher order Minkowski distances separate the images with one-pixel attacks well because a single element in the vector contributes all the information about the attack. Moreover, the threshold used in detection is always a compromise. Our evaluation metrics show that the methodology detects all the images that would have fooled the classifier. Finally, tests with images outside the training set illustrate how the detector can differentiate other types of anomalies.

We have demonstrated this capability for one type of attack, but it should be possible to detect other malicious perturbations using the same methodology. In the future, tests with other datasets will provide information about the generalizability of the detector. Other attack types should also be tested to test how a variational autoencoder can learn them. It remains to be seen how well the combination of robust classification models and additional sensor defenses protects the automated diagnosis.

Acknowledgments. This research was partially funded by *Cyber Security Network of Competence Centres for Europe (CyberSec4Europe)* project of the Horizon 2020 SU-ICT-03-2018 program. The authors would like to thank Ms. Tuula Kotikoski for proofreading the manuscript.

References

1. IBM code model asset exchange: Breast cancer mitosis detector. https://github.com/IBM/MAX-Breast-Cancer-Mitosis-Detector (2019)
2. Alatalo, J., Korpihalkola, J., Sipola, T., Kokkonen, T.: Chromatic and spatial analysis of one-pixel attacks against an image classifier (2021). arXiv:2105.13771 [cs.CV]
3. Apostolidis, K.D., Papakostas, G.A.: A survey on adversarial deep learning robustness in medical image analysis. Electronics **10**(17) (2021). DOI 10.3390/electronics10172132
4. Asperti, A., Trentin, M.: Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. IEEE Access **8**, 199,440–199,448 (2020). DOI 10.1109/ACCESS.2020.3034828
5. Beggel, L., Pfeiffer, M., Bischl, B.: Robust anomaly detection in images using adversarial autoencoders (2019)
6. Cai, F., Li, J., Koutsoukos, X.: Detecting adversarial examples in learning-enabled cyber-physical systems using variational autoencoder for regression. In: 2020 IEEE Security and Privacy Workshops (SPW), pp. 208–214 (2020). DOI 10.1109/SPW50608.2020.00050
7. Cheng, S.: BCCD dataset. https://github.com/Shenggan/BCCD_Dataset (2018)
8. Chiang, P.H., Chan, C.S., Wu, S.H.: Adversarial pixel masking: A defense against physical attacks for pre-trained object detectors. In: Proceedings of the 29th ACM International Conference on Multimedia, MM '21, p. 1856–1865. Association for Computing Machinery, New York, NY, USA (2021). DOI 10.1145/3474085.3475338
9. Dusenberry, M., Hu, F.: Deep learning for breast cancer mitosis detection (2018)
10. Endsley, M.: Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors **37**(1), 32–64 (1995). DOI 10.1518/001872095779049543
11. Finlayson, S.G., Bowers, J.D., Ito, J., Zittrain, J.L., Beam, A.L., Kohane, I.S.: Adversarial attacks on medical machine learning. Science **363**(6433), 1287–1289 (2019)

12. Ghosh, S.: Cats faces 64x64 (for generative models). https://www.kaggle.com/spandan2/cats-faces-64x64-for-generative-models (2018)
13. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2014)
15. Korpihalkola, J., Sipola, T., Kokkonen, T.: Color-optimized one-pixel attack against digital pathology images. In: S. Balandin, Y. Koucheryavy, T. Tyutina (eds.) 2021 29th Conference of Open Innovations Association (FRUCT), vol. 29, pp. 206–213. IEEE (2021). DOI 10.23919/FRUCT52173.2021.9435562
16. Korpihalkola, J., Sipola, T., Puuska, S., Kokkonen, T.: One-pixel attack deceives computer-assisted diagnosis of cancer. In: Proceedings of the 4th International Conference on Signal Processing and Machine Learning (SPML 2021), August 18–20, 2021, Beijing, China. ACM, New York, USA (2021). DOI 10.1145/3483207.3483224
17. Li, X., Zhu, D.: Robust detection of adversarial attacks on medical images. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), pp. 1154–1158 (2020). DOI 10.1109/ISBI45749.2020.9098628
18. Mazlan, A.U., Sahabudin, N.A.b., Remli, M.A., Ismail, N.S.N., Mohamad, M.S., Warif, N.B.A.: Supervised and unsupervised machine learning for cancer classification: Recent development. In: 2021 IEEE International Conference on Automatic Control Intelligent Systems (I2CACIS), pp. 392–395 (2021). DOI 10.1109/I2CACIS52118.2021.9495888
19. van der Meulen, R.: Build Adaptive Security Architecture Into Your Organization. https://www.gartner.com/smarterwithgartner/build-adaptive-security-architecture-into-your-organization/ (2017). Accessed: 3 April 2020
20. Nayyar, S.: Why healthcare could face unprecedented cyber threats in 2021. https://www.forbes.com/sites/forbestechcouncil/2021/03/17/why-healthcare-could-face-unprecedented-cyber-threats-in-2021/ (2021)
21. Nguyen-Son, H.Q., Thao, T.P., Hidano, S., Bracamonte, V., Kiyomoto, S., Yamaguchi, R.S.: Opa2d: One-pixel attack, detection, and defense in deep neural networks. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–10 (2021). DOI 10.1109/IJCNN52387.2021.9534332
22. Rafi, T.H., Shubair, R.M., Farhan, F., Hoque, M.Z., Quayyum, F.M.: Recent advances in computer-aided medical diagnosis using machine learning algorithms with optimization techniques. IEEE Access **9**, 137,847–137,868 (2021). DOI 10.1109/ACCESS.2021.3108892
23. Rogova, G.L., Ilin, R.: Reasoning and decision making under uncertainty and risk for situation management. In: 2019 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA), pp. 34–42 (2019). DOI 10.1109/COGSIMA.2019.8724330
24. Sipola, T., Kokkonen, T.: One-pixel attacks against medical imaging: A conceptual framework. In: Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, A. Ramalho Correia (eds.) Trends and Applications in Information Systems and Technologies. WorldCIST 2021, *Advances in Intelligent Systems and Computing*, vol. 1365, pp. 197–203. Springer, Cham (2021). DOI 10.1007/978-3-030-72657-7_19
25. Sipola, T., Puuska, S., Kokkonen, T.: Model fooling attacks against medical imaging: A short survey. Information & Security: An International Journal **46**(2), 215–224 (2020). DOI 10.11610/isij.4615

26. Strachna, O., Asan, O.: Systems thinking approach to an artificial intelligence reality within healthcare: From hype to value. In: 2021 IEEE International Symposium on Systems Engineering (ISSE), pp. 1–8 (2021). DOI 10.1109/ISSE51541.2021.9582546
27. Su, J., Vargas, D.V., Sakurai, K.: Attacking convolutional neural network using differential evolution. IPSJ Transactions on Computer Vision and Applications **11**(1), 1–16 (2019)
28. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation **23**(5), 828–841 (2019). DOI 10.1109/TEVC.2019.2890858
29. Tang, X.: The role of artificial intelligence in medical imaging research. BJR open **2**(1), 20190,031–20190,031 (2019). DOI 10.1259/bjro.20190031. URL https://pubmed.ncbi.nlm.nih.gov/33178962
30. Tong, L., Wang, L., Li, S., Pengfei, Z., Xiaoming, J., TongWei, Y., WeiDong, Y.: Adversarial sample detection framework based on autoencoder. In: 2020 International Conference on Big Data Artificial Intelligence Software Engineering (ICBASE), pp. 241–245 (2020). DOI 10.1109/ICBASE51474.2020.00058
31. Veta, M., Heng, Y.J., Stathonikos, N., Bejnordi, B.E., Beca, F., Wollmann, T., Rohr, K., Shah, M.A., Wang, D., Rousson, M., Hedlund, M., Tellez, D., Ciompi, F., Zerhouni, E., Lanyi, D., Viana, M., Kovalev, V., Liauchuk, V., Phoulady, H.A., Qaiser, T., Graham, S., Rajpoot, N., Sjöblom, E., Molin, J., Paeng, K., Hwang, S., Park, S., Jia, Z., Chang, E.I.C., Xu, Y., Beck, A.H., van Diest, P.J., Pluim, J.P.: Predicting breast tumor proliferation from whole-slide images: The TUPAC16 challenge. Medical Image Analysis **54**, 111–121 (2019). DOI 10.1016/j.media.2019.02.012
32. Wang, P., Cai, Z., Kim, D., Li, W.: Detection mechanisms of one-pixel attack. Wireless Communications and Mobile Computing **2021**, 8891,204 (2021). DOI 10.1155/2021/8891204
33. Wlodarczak, P.: Machine Learning and its Applications. CRC Press, Boca Raton, London, New York (2019)
34. Xu, G., Zong, Y., Yang, Z.: Applied Data Mining. CRC Press, Boca Raton, London, New York (2013)
35. Xu, H., Ma, Y., Liu, H.C., Deb, D., Liu, H., Tang, J.L., Jain, A.K.: Adversarial attacks and defenses in images, graphs and text: A review. International Journal of Automation and Computing **17**(2), 151–178 (2020). DOI 10.1007/s11633-019-1211-x