HIGH SCHOOL EDITION
Journal of Student Research
Volume 11 Issue 3 (2022)

A Novel Out-of-Distribution Detector Based on Autoencoder and Binary Classifier with Auxiliary Input

Sungmin Kim[1] and Kyoung-Hyoun Kim[#]

[1]Bergen County Academies, Hackensack, NJ, USA
[#]Advisor

ABSTRACT

At industrial production levels, anomaly detection is crucial to maintaining adequate levels of safety standards and quality assurance in products. Numerous previous research focuses on detecting the anomaly samples, yet their methods cannot filter the unexpected outliers that have completely different image features. In this research, we propose a novel out-of-distribution detector based on an autoencoder and binary classifier with auxiliary input. Given the input image, the autoencoder produces the latent variable and reconstructed image. The difference image is generated and fed to the binary classifier to classify whether the input image is an outlier or a normal sample. The latent variable which contains useful feature-level information is fed to the intermediate layer of the classifier to produce the precise classification results. We also propose noise-addition augmentation to make the trained model consistently perform against various kinds of noise-containing images which are often found in real-world scenarios in industrial environments. The proposed method achieves AUC of 0.9718 and 0.5908 in the MNIST and CIFAR10 datasets, respectively. Through these experiments, we have shown that the proposed method outperforms the previous state-of-the-art methods.

Introduction

Anomaly detection is an important industrial problem that has been well-studied in the computer vision field. During the early stage of the anomaly detection study, numerous research focused on utilizing high-quality features of the input images to capture in-distribution representation [1, 2]. These methods can easily achieve human-level accuracy and are easily calibrated to different industrial fields or image domains. Yet, these methods often fail when the unexpected outliers are passed as input since their distribution is completely different from abnormal or normal samples. One possible naive approach to solve this problem is to collect the outliers in the training set and train the model to learn their distribution. However, it is practically impossible to gather a large dataset of outlier cases. For this reason, it is necessary to train the network in an unsupervised or semi-supervised approach only using normal samples without any outlier samples.

To solve this problem, we propose a novel autoencoder-based anomaly detection system. The proposed system can be trained in a semi-supervised fashion without having outlier samples. We consider outlier detection as a type of OOD (Out-Of-Distribution) detection task. In order to accomplish an OOD detection task, it is crucial to determine the in-distribution of the sample images. To find the distribution of the normal samples, we exploit the autoencoder architecture to determine the manifold for the normal samples. We also propose a noise addition data augmentation technique to make the trained model consistently perform against various kinds of noise-containing images which are often found in real-world scenarios. The proposed method achieves AUC of 0.9718 and 0.5908 in the MNIST and CIFAR10 datasets, respectively. Through the experiments, we have shown that the proposed method outperforms the previous state-of-the-art methods.

Methods

In this chapter, we explain the overview of the proposed out-of-distribution detection system. We split the system into two different architectures as we solve out-of-distribution detection as distribution-aware classification.

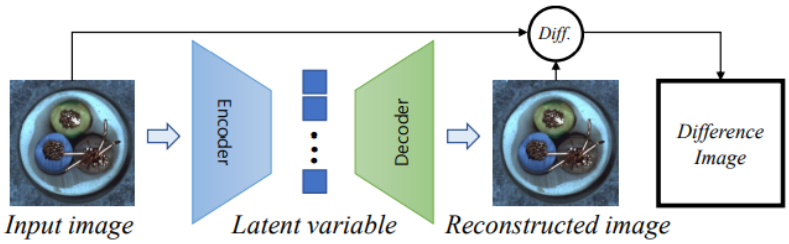

Figure 1. The flow chart of the proposed autoencoder.

Figure 1 shows the architecture of the proposed autoencoder that produces the difference image between the original input image and the reconstructed image. We simply inherit the pipeline of general autoencoders to train the proposed model. As the training process performs, the encoder becomes a feature extractor that transforms the input image into the latent space that represents the normal samples in the feature space. The decoder takes the latent variable and tries to reconstruct the original input image.

Given an input image I, the proposed encoder, E, compresses the high-dimensional features to produce the latent variable z. This latent variable is then processed through the decoder D to output the reconstructed image, I, which has the same dimensional qualities as the original input. Finally, the difference in the pixel data of the original image and the reconstructed image, I_{Diff}, is found by taking the absolute value of the difference as follows: $|I - \hat{I}|$. Here, we define the encoder, E, and the decoder, D, by E: I → z and D: z → I, respectively.

The underlying assumption here is that since the training dataset only contains the normal samples the encoder learns to encompass the distribution of the normal sample. Thus, given outlier samples, the encoder fails to transform the image into in-distribution and the decoder also fails to reconstruct the image at the test time. This significantly affects the generated difference image, and the following binary classifier can easily detect if the input sample is an outlier or not.

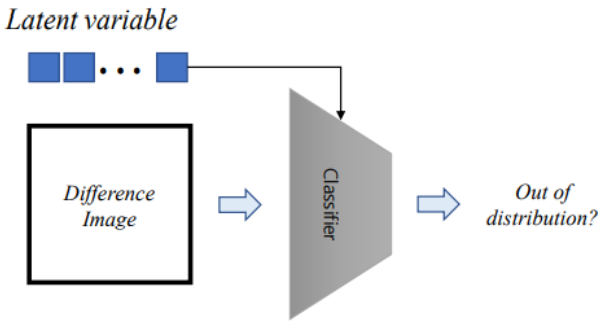

Figure 2. The flow chart of the proposed out-of-distribution detector

Figure 2 shows the flowchart of the proposed binary out-of-distribution classifier. The classifier takes the difference image generated in the first step and predicts if the input image is an outlier or not. This scheme is enough to yield comparable detection performance. However, we have found that a simple auxiliary input can boost the detection performance even more while only making computational costs slightly higher.

We feed the latent variable that is extracted from the encoder in the first step to the intermediate layer of the classifier. The latent variable contains very useful feature-level distribution-related information. Thus, this simple architecture modification can significantly increase the accuracy of the trained model. The increase in computational cost is trivial compared to its performance boosts.

Implementation Details

A loss function yields the characteristic of the trained network, which is determined by the training process. As we use the autoencoder architecture, it is necessary that the reconstructed image is virtually identical to the original input image. The loss function we used to train the proposed network is represented in equation 1.

Equation (1): Reconstruction loss function:

$$L_1 = \left| I - \hat{I} \right|_1$$

Here, I and $\hat{I}$ denote the input image and reconstructed image, respectively. Next, to train the binary out-of-distribution classifier, we simply use the cross-entropy loss function that is widely used for many classification tasks.

Equation (2): Cross-entropy loss function:

$$L_2 = CE(y, \hat{y})$$

In the above cross-entropy loss function, y and $\hat{y}$ denote the prediction of the classifier and their corresponding ground-truth. To train the proposed method, the Adam [3] optimizer was used with beta1 set to 0.9 and beta2 to 0.99. The mini-batch size was set to 32 samples with the network being trained for 100 epoch. Furthermore, the learning rate was initialized to 0.0001 and decreased by a factor of 5 every 40 to 80 epochs using the MultiLRStep method implemented in PyTorch [4].

To implement the overall architecture, we exploit resnet [5] widely being used in many computer vision fields. Specifically, we choose resnet18 to implement the proposed binary classifier and both the encoder and the decoder in the autoencoder scheme. For the data augmentation, we propose a novel noise addition augmentation technique to allow the trained model to produce robust results against the low-quality input samples that commonly occurs in the real-world scenarios.

Results and Discussion

In this chapter, we conduct comparison experiments to show the superiority of the proposed method. We choose MNIST [6] and CIFAR10 [7] dataset which have completely different color and shape characteristics. For the comparison methods, we select PixCNN [8], DSEBM [9], and VAE [10] which show comparable performance in out-of-distribution detection tasks. For a fair comparison, we train the comparison methods with the same training settings and measure the AUC scores which are often used to measure the performance of the out-of-distribution detectors.

Comparison with the state-of-the-art methods

Table 1. Comparison result for MNIST(in-dist.) and CIFAR10(out-dist.)

MNIST (in-dist.) CIFAR10 (out-dist.)	AUC
PixCNN [8]	0.6141
DSEBM [9]	0.9554
VAE [10]	0.9643
Ours	0.9718

Table 2. Comparison result for CIFAR10(in-dist.) and MNIST(out-dist.)

CIFAR10 (in-dist.) MNIST (out-dist.)	AUC
PixCNN [8]	0.5450
DSEBM [9]	0.5725
VAE [10]	0.5725
Ours	0.5908

Tables 1 and 2 show the comparison results. In the first comparison experiment, we set the MNIST dataset as in-distribution samples and CIFAR10 as out-distribution samples. The proposed method achieves an AUC of 0.9718 and surpasses the previous state-of-the-art methods. The first comparison model PixCNN [8] achieves an AUC of 0.6141 and results in the worst performance within the comparison group. DSEBM [9] and VAE [10] achieve AUC of 0.9554 and 0.9643 which are 0.0089 and 0.0075 lower than the AUC of the proposed method.

We can also observe a similar comparison aspect for the second experiment as shown in table 2. In the second experiment, the dataset configuration is set opposite to that of the first. The proposed method outperforms the previous state-of-the-art methods with a noticeable performance gap. The state-of-the-art methods show relatively poor results compared to the proposed method. We attribute this to the proposed autoencoder and the use of latent variables as an auxiliary input. This modification significantly improves the accuracy of the trained model by implicitly providing distribution-related features. The effectiveness of each proposed idea is examined in the ablation study.

Ablation Study

Table 3. Ablation study results

CIFAR10 (in-dist.) MNIST (out-dist.)	AUC
w/o auxiliary input	0.5624
w/o data augmentation	0.5873
Full model	0.5908

We conduct ablation studies to review and examine how each proposed method affects the trained model. We train two ablation models without having auxiliary input which is a latent variable and the proposed noise addition augmentation technique. Table 3 shows the overall ablation study results.

The first ablation model is trained without having auxiliary latent variable input in the out-of-detection classifier. The performance gap between the full model and the ablation model which achieves the AUC of

0.5624 is 0.0284. This result clearly shows that the proposed auxiliary helps the trained out-of-distribution classifier to produce more accurate results. The trained encoder that outputs the auxiliary latent variable transforms the in-distribution and out-distribution samples in different manifold spaces. Thus, the latent variable contains very useful feature-level information to detect the out-of-distribution samples.

In the second ablation study, we examine the effectiveness of the proposed data augmentation technique. The second ablation model achieves an AUC of 0.5873 while the full model yields an AUC of 0.5908. We contribute this performance boost to the characteristics of the test dataset. The samples in the test dataset often contain blur-like noise which normally degrades the accuracy of the trained model. By applying the proposed data augmentation technique, we can provide a wide range of data samples during the training time so that the trained model can produce robust results against such noisy samples.

Conclusion

In this research, we proposed a two-step out-of-distribution detector using an autoencoder and binary classifier. In the first step, the proposed autoencoder is trained with reconstruction loss and produces the difference image and latent variable as the output. The proposed binary out-of-distribution classifier takes the difference image as the main input and the latent variable as an auxiliary input. Through extensive experiments, we have shown that the proposed method surpasses all previous state-of-the-art methods with a noticeable performance gap. We also conducted ablation studies to verify how each proposed idea affects the final accuracy. In conclusion, the proposed method achieved an AUC of 0.5908 and 0.9718 in the CIFAR10 and MNIST datasets, respectively.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

[1] Gudovskiy, Denis, Shun Ishizaka, and Kazuki Kozuka. "Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022.
[2] Rudolph, Marco, et al. "Fully Convolutional Cross-Scale-Flows for Image-based Defect Detection." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022.
[3] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).
[4] Paszke, Adam, et al. "Automatic differentiation in pytorch." (2017).
[5] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
[6] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, *29*(6), 141–142.
[7] Krizhevsky, Alex, Vinod Nair, and Geoffrey Hinton. "Cifar-10 (canadian institute for advanced research)." *URL http://www. cs. toronto. edu/kriz/cifar. html* 5.4 (2010): 1.
[8] Van den Oord, Aaron, et al. "Conditional image generation with pixelcnn decoders." *Advances in neural information processing systems* 29 (2016).
[9] Zhai, Shuangfei, et al. "Deep structured energy based models for anomaly detection." *International conference on machine learning*. PMLR, 2016.

[10] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114* (2013).