Project Paper

Anomaly Detection on FashionMNIST

1 Introduction

Identifying unusual patterns in a dataset is a challenging problem that can be addressed using anomaly detection. In particular, generative adversarial network (GAN) and autoencoder-based methods have been adopted in such problems.

In this project, we explore this alternative approach implemented in the 'GAN-based Anomaly Detection in Imbalance Problems' paper [0], where a modified GAN architecture and different loss functions are combined in order to enhance the efficiency for defect detection.

Especially, this model consists of an autoencoder as the generator and two different discriminators for each of normal and anomaly input from the FashionMNIST dataset. This differentiation among the images is done by selecting one class and defining it as normal input to our model and afterwards sampling images from the other classes using K-Means clustering to build our anomalous data.

The goal is then to implement a reconstruction-based approach : the idea is that the generator would reconstruct normal data with small noise while the reconstruction error for anomaly data is much higher. Adversarial training from the original GAN subsequently is introduced through the two discriminators who will decide whether the image is original or a reconstructed one.

2 Method

2.1 Network Architecture

In order to perform anomaly detection, a GAN-based generative model is proposed (Figure 1). During the training phase, we want our model to minimize the reconstruction error of the image input when normal data is fed to the generator, maximize it otherwise. The latter is in the form of an autoencoder which structure follows that of a U-Net [1]. The number of layers and kernels per layers is chosen to be identical to the architecture presented in our reference paper. In other words, we stayed loyal to the U-Net architecture while keeping track of the number and parameters and dimensions they have in the paper.

On the other hand, the discriminator's structure follows that of the PatchGan Discriminator ; it receives an image as input and outputs a value in the range of [0, 1] along with a 4x4 matrix in the same range. The first return is the probability of the image being real of fake while the second return represents the probability of the corresponding image patch being real or fake (as represented by the matrices on the right most side of the figure). The number of layers and kernels per layers is again chosen to be identical to the architecture presented in our reference paper.

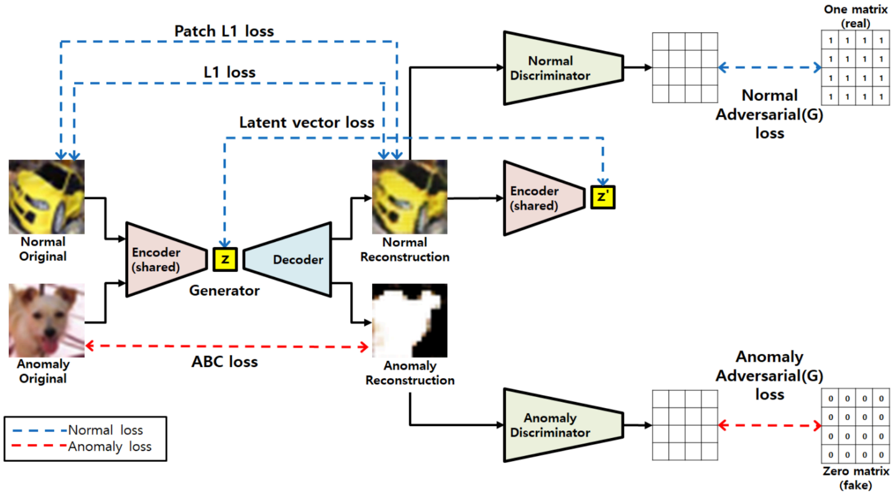


Fig. 1: Pipeline of the proposed approach for anomaly detection.

The losses used to minimize error with normal input are shown in the figure in blue and the ones used to maximize error with anomaly input in red (we will see them in detail later).

2.1 Loss functions

There is a total of eight loss functions used in our model : six for training the generator, one for the normal discriminator and one for the anomaly discriminator.

The loss function used for training the discriminator is adopted from LSGAN [2] as shown in Eq. (1).

$$\min_D V_{\text{LSGAN}}(D) = [(D(x) - b)^2] + [(D(G(x)) - a)^2] \tag{1}$$

With a and b representing the labels for fake and real data, respectively.

Now for the generator, when the input image is normal we have four loss functions. First one is the L1 distance between the original image and the generated one as defined in Eq. (2).

$$\mathcal{L}_{\text{recon}} = \|x - G(x)\|_1 \tag{2}$$

Second loss function is the patch loss. Normal and generated images are separated into M patches and the average of the n biggest reconstructions errors among the patches is selected as shown in Eq. (3).

$$\mathcal{L}_{\text{patch}} = f_{avg}(n)(\left\|x_{patch(i)} - G(x_{patch(i)})\right\|_1), i = 1, 2, ..., m \tag{3}$$

Eq. (4) represent the latent vector loss, which is the difference between latent vectors of the generator for the normal image and of the cascaded encoder for the reconstructed image.

$$\mathcal{L}_{\text{enc}} = \|G_E(x) - G_E(G(x))\|_1 \tag{4}$$

Now, we introduce the adversarial loss in Eq. (5) in its general form, that is used to update the generator.

$$\min_G V_{\text{LSGAN}}(G) = [(D(G(x)) - y)^2] \tag{5}$$

y is equal to 1 when a reconstruction image (fake) is into the discriminator. However, since anomaly reconstructed images should be generated differently from real ones then y is equal to 0 when we want to classify anomaly.

Finally, ABC loss as shown in Eq. (8) is used to maximize the L1 reconstruction error for anomaly data.

$$\mathcal{L}_{\text{ABC}} = -\log(1 - e^{-\mathcal{L}_\theta(x_i)}) \tag{8}$$

Total loss function is a weighted combination of all these losses. We used the same weights as the reference paper as they returned the most optimal results. At the end of each training epoch, we fit a simple Logistic Regression classifier in order to predict if an image is normal or anomalous based on the reconstruction error between the original image and the reconstructed one.

3 Experiments

3.1 Results

The main metric to evaluate performance in our case is the Area Under the Receiver-Operating Characteristics (AUROC) [3], which tells us how efficient a model is. Especially, an AUROC of 0.5 suggests that a model has a predictive ability no better than random guessing and more than 0.9 is considered outstanding. We get an AUROC score approximately equal to 0.99155, very close to the scores in our reference paper, depending on the experiment. This result illustrates our model's great performance at distinguishing between positive and negative classes. This is even more confirmed with the following confusion matrix (Fig. 2).

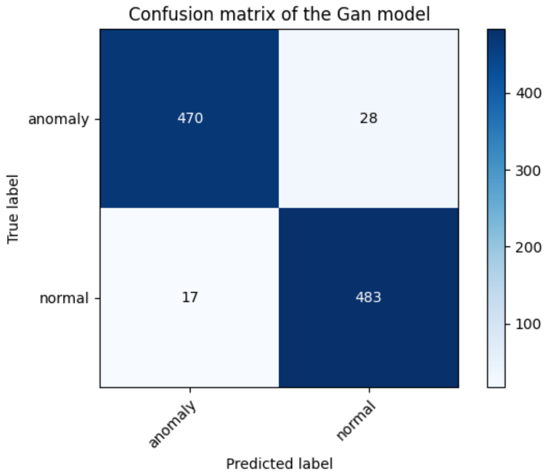
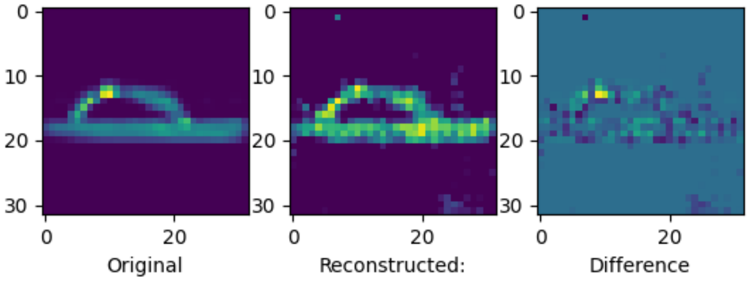


Fig. 2: Confusion matrix of the GAN model

After training our model for 50 epochs, we get the following results, as shown on Fig. 3, the original image, the reconstructed image and the reconstruction error for normal and anomaly data.

Normal :



Anomaly :

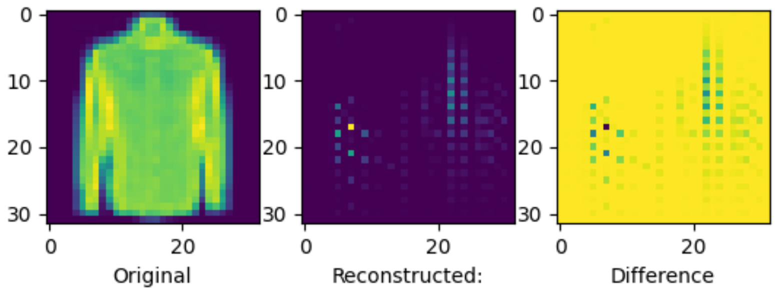


Fig. 3: Visualization of experimental results

We can see how the normal image has a nearly perfect reconstruction and how the anomaly image has a very noisy reconstruction.

3.2 Comparison with other baselines

Along with our model, we built two other baselines : a random one and a non-trivial one. The latter consists of an autoencoder that is trained with only normal samples. Based on the final train loss, we set a threshold that will determine, in the validation step, whether the input is normal or anomaly, i.e. if the reconstruction error is smaller than the threshold then the image is classified as normal, anomaly otherwise.

We computed the AUROC score and plotted the ROC curve for all three models, as shown on Fig. 4.

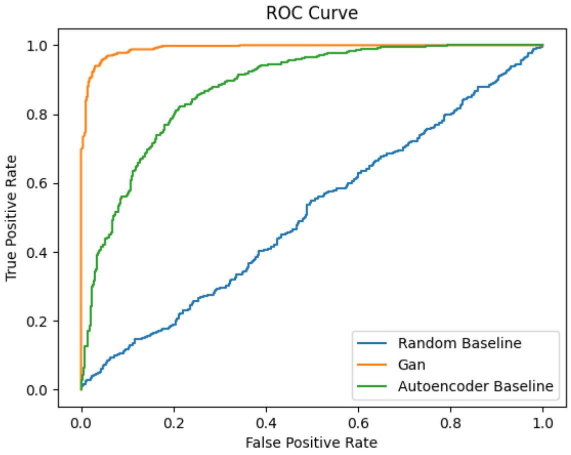


Fig. 4: ROC curve of the Random Baseline, GAN model and Autoencoder Baseline

We also decided to look at these baseline's confusion matrices, as presented in Fig. 5 and Fig. 6.

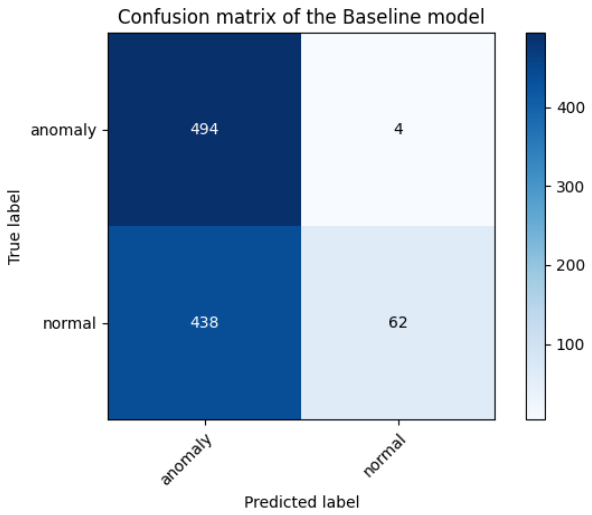


Fig. 5: Confusion matrix of the Autoencoder Baseline

This baseline mostly fails at predicting the normal class, as it is returning the anomaly label for most cases. One might suggest that this is due to the threshold value which is not optimal for differentiating the two classes.

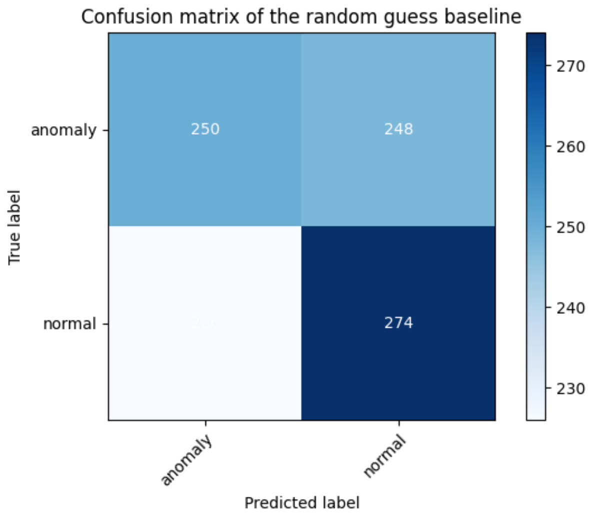


Fig. 6: Confusion matrix of the Random Baseline

[0] : http://intlab.skuniv.ac.kr/paper/GAN-based_Anomaly_Detection_in_Imbalance_Problems.pdf

[1] : Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

[2] : LSGAN : Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2017)

[3] : https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html