Anomaly Detection

Johnny Antoun, Richard Fremgen, Jaskaran Singh

November 29, 2022

Abstract

Anomaly detection is one of the most heavily used data mining techniques in academic and industry, that has garnered an extensive amount of research interest over the years. Anomaly detection can be achieved numerous ways, as one of the most common ways researchers detect anomalous images or data is through training a neural network. In this paper, we discuss a series of reconstruction autoencoders that were trained and testing to perform anomalous image detection and localization. Competing models architectures were evaluated using both pixel-level and image-level performance metrics.

1 Introduction

The ability for humans to detect irregularities in images is a relatively trivial task that only requires a small amount of cognition and familiarity with the object of interest. Training a digital network to perform a similar task is a challenging, but worthwhile task because depending on the subject and with enough training data, a neural network can detect abnormalities at or better than of human and offer one major advantage: *scalability*. Anomaly detection is a technique that utilizes machine learning to identify abnormal behavior and features in a data set when compared to an established pattern. Such a process commonly consists of training a neural network on a data set consisting of normal examples, in order to learn what distinct characteristics of the data constitute *normality*. This trained network can then be used on to detect anomalous events and observations that significantly deviate from the standard behavior observed in the data. As such, the objective of this project is to train a series of reconstruction autoencoders in order to perform anomaly detection and localization on the CIFAR-10 data set, in order to evaluate an optimal model that can differentiate between normal and anomalous images.

1.1 Motivation and importance of the problem

The rise of the big data era has been influential over the research and development of machine learning methods that can scale and handle such vast amounts of

data collected. One of the primary methods used by researchers to detect outliers and irregularities is anomaly detection, which is used to solve problems such as detecting fraud in financial transactions, to identifying medical abnormalities in X-ray scans, and to even monitoring sensor readings on airplanes. Such a process provides immense value and power when implemented because they are able to quickly detect deviations from the norm, which is some case can be a critical event one wishes or avoid, or can represent potential opportunities.

2 Related works

Due to the popularity and frequent use of the CIFAR-10 data set for machine learning classification problems, there is a multitude of related works where researchers constructed similar autoencoders to detect anomalous images. Reiss, Cohen, Bergamn and Hoshen developed a system known as PANDA that utilized the CIFAR-10 data set as a one-class classification (OCC) setting using the Deep SVDD method with elastic regularization and was able to achieve an AUROC of 98.9. However, the most accurate model to date was developed by Mirzaei and colleagues in their paper *Fake It Till You Make It: Near-Distribution Novelty Detection by Score-Based Generative Models*, which deployed a near novelty detection network an achieved an AUROC of 99.1

3 Details of the project

The CIFAR-10 dataset consists of 60,000 32x32 images of ten different classes; 50,000 of which are training images and the remaining 10,000 are test images. This data set is commonly used in the machine learning community for training and testing autoencoders, as for purposes of this project, class 0 (airplane) was chosen as the *normal* data class. A 80-20 training-validation set split was used on the class 0 images, meaning that our training data consisted for 4,000 airplane images, and the validation set consisted of 1,000 airplane images. The training and validation sets were filtered to only contain class 0, or airplane images, since the objective of training the autoencoder was to be able to properly reconstruct these types of *normal* images. The test set, however, consisted of all ten classes, or 10,000 images, because the purpose of testing our network was to verify that our network could differentiate between normal images (class 0) and abnormalities (classes 1-9).

3.1 Contribution of each member of the team

The three team members of this project were supervised by the Nick Konz, ECE-685 teaching assistant. Antoun, Fremgen, and Singh took a systematic, collaborative approach for this project. All team team members were involved with every stage of the training and testing of the anomaly detection neural network, in addition to the writing this paper. Existing autoencoder code from

the ECE-685 discussion sections and homework provided motivation for the code used in this report

4 Experimental results

4.1 Model 1

The first model we fit is a convolutional autoencoder. Both the encoder and decoder consist of 3 two-dimensional convolution layers (stride = 2 and padding = 1). All convolutional layers are followed by a ReLU activation function except for the last layer where we opted to use a Sigmoid activation function. In terms of optimizer, we decide to use Adam optimizer to minimize MSE loss. The structure of the convolutional autoencoder is included Figure 1.

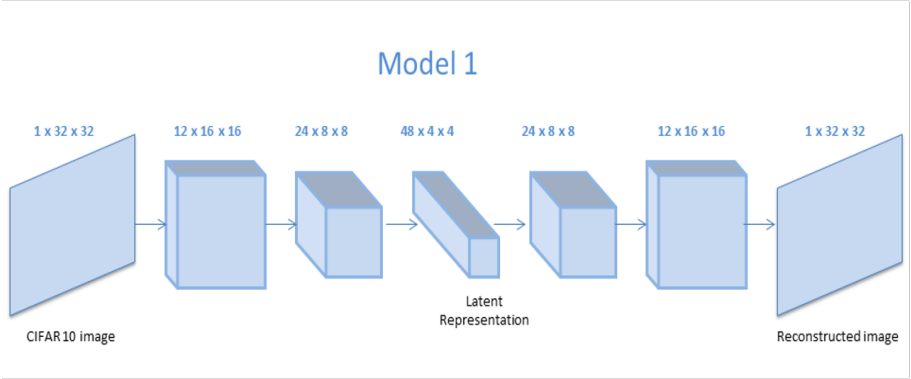

Figure 1: Model 1 Architecture

We first start by training and validating the autoencoder with different options for learning rate and weight decay over 10 epochs and calculating the average training loss and validation loss per epoch. The learning rates we try are 0.1, 0.01 and 0.001 while the weight decay values we use are 0.01, 0.001 and 0.0001. We decided to use the model with the lowest average validation error, which was the model with a learning rate of 0.01 and a weight decay of 0.0001 (average training loss 2.38 and average validation loss 0.45). In Figure 2, we include the validation and training error curves for this model over all 10 epochs. We use this model this for both anomaly detection at the level of images and pixels as follows.

Image level: For this approach we find the average mean squared error between an input image and a reconstructed image. We either classify an image as anomalous or airplane based on the value of some threshold "c" we set. If the average mean squared error between the input image and reconstructed image is greater than or equal to "c", we predict that the image is anomalous. Otherwise, we predict airplane (average mean squared error between input image and reconstructed image less than c). Based on the various values of c we obtain

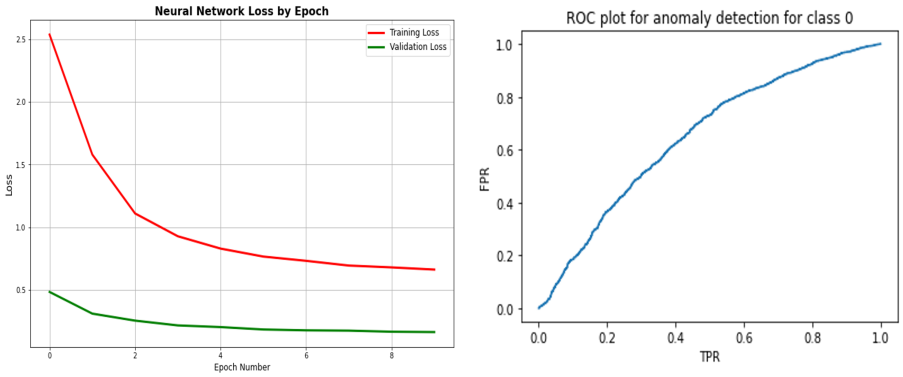

Figure 2: Model 1 Overall Loss and ROC Curve

values for true positive rate and false positive rate. The higher we set c, the less false positives we get and the more true positives we obtain and vice versa. We can plot these different true positive and false positive rates using a Receiver Operating Characteristic (ROC) and compute an area under the curve as a measure of the accuracy of the classifier. Below we include the ROC curve for model 1 using different values of "c" and obtain an area under the curve of 0.35 (generally we would like to see values closer to 1).

Pixel level: For this approach we decide to create an absolute difference image which essentially the absolute value of the difference between pixels. The difference image we obtain consists of three channels (RGB) as the images we start off with are three channels. We also create an absolute difference image with one channel where we are simply averaging across the three channels. The middle two pictures consist of three channels where the right-most images consist on of channel.

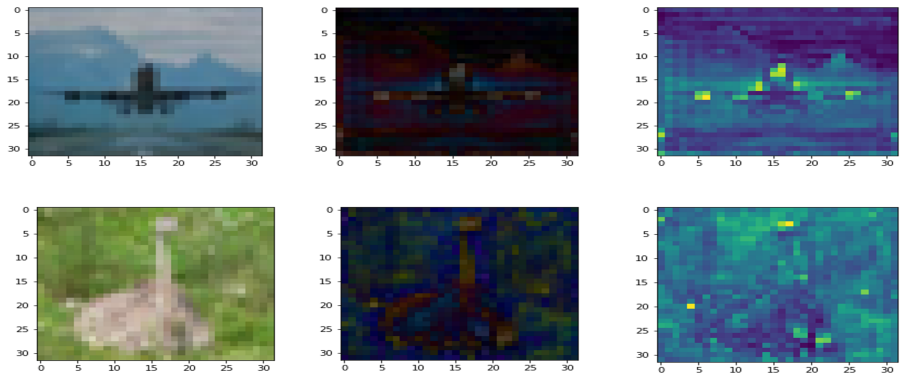

Figure 3: Model 1 Pixel Anomaly

4.2 Model 2

We use a similar structure for model II except we add one layer to each of the encoder and decoder. We use a similar loss function and optimizer, but we run for 25 epochs rather than 10. The structure of the convolutional autoencoder is included in Figure 4.

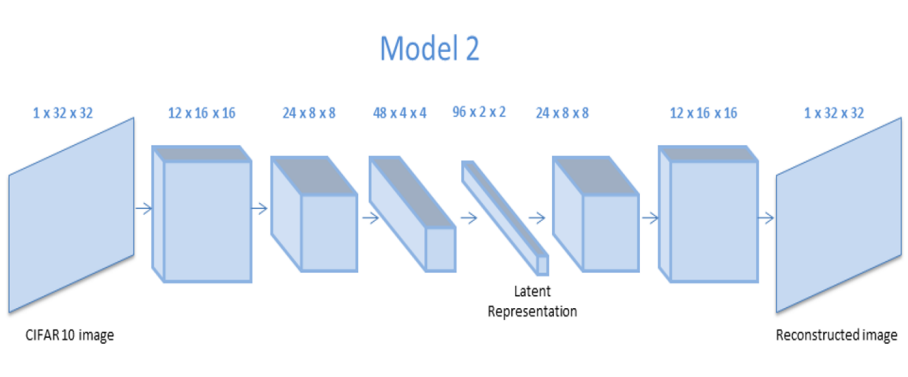

Figure 4: Model 2 Architecture

We try similar learning rates and weigh decay values as we did for model I and we decide to use the model with the lowest average validation error – model with learning rate of 0.001 and weight decay of 0.0001 (average training loss 1.28 and average validation loss 0.28). Below we include validation and training error curves for this model over all 25 epochs.

Image level: we obtain a worse result with area under the curve of 0.312 as seen in Figure 5 below.

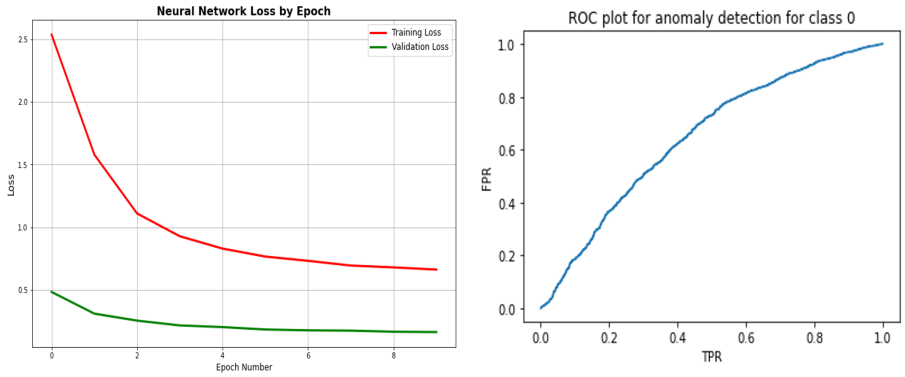

Figure 5: Model 2 Overall Loss and ROC Curve

Pixel level: We include a similar set of images as we did for model 1 absolute difference over 3 channels and absolute difference averaged over 1 channel.

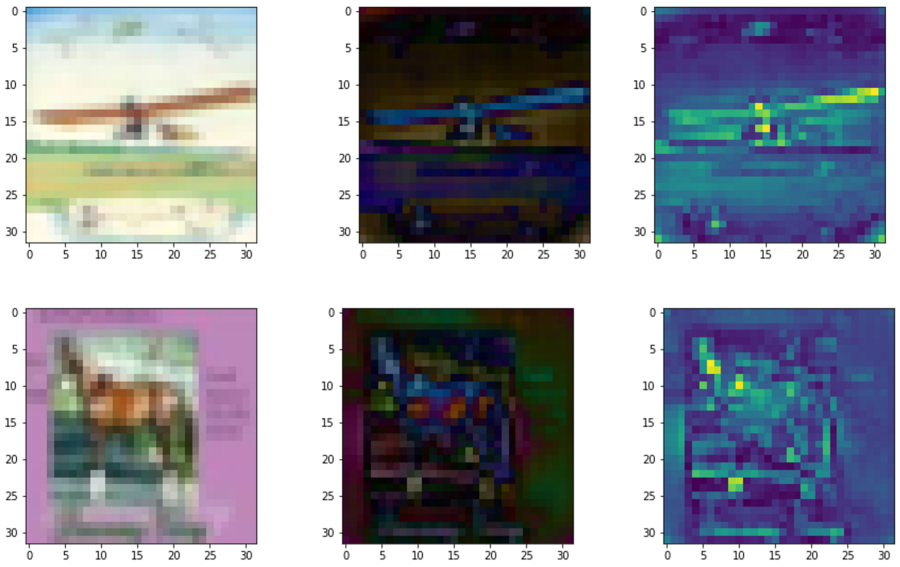

Figure 6: Model 2 Pixel Anomaly

5 Concluding remarks

In conclusion, the results from model I were not satisfactory for image level classification. We made some changes in the form of increasing layers in the auto-encoder and training for more epochs and still did not obtain the results we were looking for. Potential avenues we would like to consider for future include various forms of data augmentation such as rotating images, changing color and adding noise. Data augmentation could be useful as our training and validation sets are small in size given we are not using images of anything other than airplanes. Additionally, we could consider using different auto-encoder structure such as using fully connected layers rather than convolutional layers and comparing results.

6 References

1. *Fake It Till You Make It: Near-Distribution Novelty Detection by Score-Based Generative Models*
2. *PANDA: Adapting Pretrained Features for Anomaly Detection and Segmentation*