Detecting Rare Diseases: Autoencoders for Detecting Anomalies in Medical Imaging

Stijn van het Reve
University of Twente
The Netherlands
s.j.vanhetreve@student.utwente.nl

ABSTRACT

This research project explores the application of autoencoders (AEs) for anomaly detection in medical images. The project investigates the performance and limitations of autoencoders in medical imaging scenarios where the quality of the dataset plays a pivotal role.

KEYWORDS

Deep learning, autoencoder, medical imaging

1 INTRODUCTION

Image anomaly detection is a field within computer vision and machine learning that focuses on identifying unusual or anomalous patterns in images. The goal is to develop algorithms that can differentiate between normal and anomalous images, where anomalies could represent defects, irregularities, or unexpected variations in the visual content. This has applications in various domains.

Anomaly detection is an unsupervised task that involves learning a standard profile based on normal data examples and subsequently recognizing samples that deviate from this norm as anomalies. One approach to address this task is by utilizing an autoencoder (AE), a type of neural network trained to reconstruct its input. Using autoencoders for identifying outliers is commonly utilized in the realms of cybersecurity, industry, finance, and healthcare. In this document, we will look at autoencoders for anomaly detection in medical images. Due to an AE's ability to train in an unsupervised way, it can be applied to medical datasets that are often unlabeled. [1]

An AE consists of an encoder, a compressed representation layer, and a decoder. (Figure 1) The encoder takes input data and transforms it into a compressed representation. This compressed representation layer is called the latent space/bottleneck layer. The decoder takes the compressed representation generated by the encoder and reconstructs the original input data. For anomaly detection, the goal is to produce an output that resembles the input data as closely as possible. The underlying assumption is that a well-trained autoencoder will capture the latent subspace of normal samples. After training, the autoencoder is expected to exhibit a low reconstruction error (commonly the Mean Squared Error or Binary Cross-Entropy) for normal samples and a high reconstruction error for anomalies. In essence, an anomaly does not represent the bottleneck layer well enough and therefore the decoder will perform worse in reconstructing the original input. [1]

AE's have been used successfully for anomaly detection. However, there are limitations. The main difficulty is choosing an effective dissimilarity metric and searching for the right degree of compression (the size of the bottleneck). Moreover, general drawbacks of using autoencoders are the limited interpretability of extracted features, overfitting, choosing the right architecture for the task, and its computational cost. Medical images have additional challenges. The datasets themselves are limited and consist of pictures without clearly defined shapes. The images hold a large amount of data and are frequently of high resolution. Anomalous cases are often very similar to normal cases and are hard to discern. [2]

1.1 Research Questions

To see how an AE performs in anomaly detection concerning medical data, I propose the following research questions:

(1) How well can AEs reconstruct medical images?
(2) What is the performance of AEs in detecting artificial anomalies in medical images?
(3) What is the performance of AEs in detecting real anomalies in medical images?
(4) What is the computational cost of reconstructing medical images with AEs?

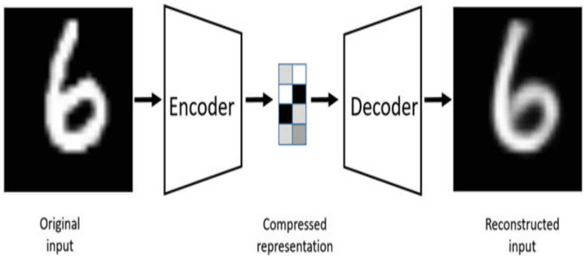

Figure 1: Autoencoder, Schematic Representation

The paper is structured as follows: Section 2 is about the cutting-edge research being done in anomaly detection using AE's, both in general and for medical datasets. Section 3 is about the methodology to conduct my experiments. The results will be presented in Section 4. Section 5 is a discussion about the results of the study and Section 6 is the conclusion where answers to the research questions are given.

40th Twente Student Conference on IT, Feb. 2nd, 2023, Enschede, The Netherlands

2 STATE OF THE ART / RELATED WORK

Ever since the introduction of AE's, constant research has taken place in this field of study, resulting in valuable improvements. [12][13] While deep learning using AE's has made significant strides in identifying image anomalies, it remains challenging to apply these methods to complex images, such as those within the medical domain. Given the dynamic nature of this field, researchers persist in exploring new architectures and techniques to enhance the performance of autoencoders, but despite a large number of anomaly detection methods that appeared in recent years, only several papers included medical images in their experiments. As there is no general approach to utilizing autoencoders in the medical field, researchers have tried to suggest new baselines for anomaly detection for different medical imaging scenarios resulting in complex models. [2][3] Other research has shown that while deep learning-based autoencoders could have great potential in the medical field, the way of determining performance using the reconstruction error is outdated. New ways to regulate anomaly scores have been introduced. [4] While there is evidence of advanced autoencoders achieving high accuracy for outlier detection in medical datasets, there are complications for these models. Like any other expert system, the proposed models are highly dependent on the training data. As a result, when unseen data is fed as an input, the system calls it an anomaly, which is a problem for AE's in general. [5] This work examines if relatively simple AE's can be used to address the anomaly detection challenge for medical datasets of manageable complexity. To examine the limitations and potential in a transparent and systematic matter in the given time frame, it is practical to start with working AE code that is readily available.

3 METHODOLOGIES

In this section, we outline the framework that guides the study, encompassing the critical aspects of the research process. First, the experiment environment and datasets are described, then the way of conducting the experiments is explained and the evaluation metrics are given.

3.1 Environment

Google Colab will be the main application used to implement code in Python Notebooks, it is favored for data science due to its free access to GPUs, eliminating the need for local setup. It comes with pre-installed libraries like TensorFlow and Pandas. It also integrates with Google Drive for easy online work. Colab supports data visualization libraries for creating informative visualizations and with an active community and many educational resources, it is a good choice for this research project. [8]

3.2 Datasets

In order to effectively apply the autoencoder technology, the datasets that are used must be of high quality. Medical data, and therefore medical datasets are more difficult to obtain and use for machine learning purposes. Legal constraints and lack of medical professionals to create and label the data inputs make medical datasets more sparse. Nevertheless, there are useful and trusted datasets available

.

3.2.1 MNIST. A well-known and publicly available dataset is the MNIST (Modified National Institute of Standards and Technology) dataset. This labeled dataset is a collection of 28×28×1 pixel grayscale images of handwritten digits (0-9) and is often used as a starting point to develop and test computer vision algorithms. It has 60.000 training images and 10.000 testing images. While the simplicity of the dataset makes it widely used and acknowledged, getting a high accuracy on MNIST doesn't necessarily translate to success on more challenging tasks. [6]

3.2.2 PneumoniaMNIST. The PneumoniaMNIST is based on a prior dataset of 5,856 pediatric chest X-Ray images. The dataset holds normal lung scans and lung scans where pneumonia is the diagnosis. The source images are gray-scale, and their sizes are (384−2,916)×(127−2,713). The images are center-cropped with a window size of the length of the short edge and are resized to 28×28×1 [7]. Due to the large amount of information in X-Ray images, the randomness/entropy is high [11]. This dataset is a good representation of a complex medical dataset.

3.2.3 OCTMNIST. The OCTMNIST is based on a prior dataset of 109,309 valid optical coherence tomography (OCT) images for retinal diseases. The dataset is comprised of 4 diagnosis categories, of which 3 are malignant. The source images are gray-scale, and their sizes are (384−1,536)×(277−512). The images are center-cropped with a window size of the length of the short edge and are resized to 28×28×1[7]. Although the randomness in the images is less than that of the Pneumonia dataset, the OCT dataset visually resembles the MNIST dataset more and is interesting to investigate.

3.3 Procedure

The way of carrying out the experiments will be the same for each dataset, starting with the MNIST dataset to establish a baseline of trusted results and then switching focus to the medical datasets. With each of the 4 experiments, we try to focus on 1 of the 4 research questions. Due to the small size of the images, having a simple but easy-to-adjust autoencoder architecture will be productive. It is crucial that the architecture of the autoencoder is the same while tackling the different datasets.

3.3.1 Preprocessing. All of the datasets contain images with a resolution of (28x28x1). Since we are trying to establish the difference between the baseline dataset and the medical datasets, it is beneficial to work with images that are of the same size since the input shape of the autoencoder should match the image resolution.

3.3.2 Model architecture. The starting model architecture is one that works well in detecting anomalous images after training on the MNIST dataset. Having a high accuracy using MNIST allows for comparison with medical datasets.

Figure 2 is a visual representation of the AE architecture. It is clear that by definition of an AE, the output shape should be the same as the input shape (28x28x1). The network uses convolutional, maxpooling and upsampling layers. Convolutional layers are a fundamental building block of convolutional neural networks (CNN's), which are widely used in computer vision. The primary use of convolutional layers is to detect patterns and features in the input data. It has been shown that using a convolutional network is a good choice for medical imaging [9].

Max pooling is a down-sampling operation commonly used in convolutional neural networks to reduce the spatial dimensions of the input feature maps. It helps retain the most important information while discarding less relevant details. We use max pooling layers in the encoding part of the network.

Upsampling is the counterpart to the max pooling operation. The purpose is to reconstruct the input data from the learned compact representation. The upsampling layers can be found in the decoding part of the network.

The model is trained in 25 epochs with a batch size of 60. Adam optimizer is used. More details about the network can be found at the end of this document. (Figure 6)

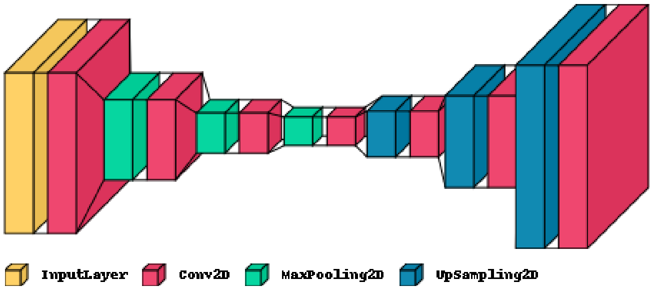

Figure 2: Autoencoder Architecture, Schematic Representation

3.3.3 Experiment 1: Reconstruction Error. To determine the success of the autoencoder reconstructing the input images we can use the commonly used Binary Cross-entropy (BCE). The BCE loss is a measure of the difference between the input data and the reconstructed output. As our input values are in range [0,1], it is appropriate to use. The mathematical expression for BCE loss in the context of autoencoders is as follows:

$$\text{BCE Loss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)) \quad (1)$$

Where:

N is the number of training examples.

y_i represents the true binary value (0 or 1) for the i-th example.

$\hat{y}_i$ represents the predicted probability that y_i equals 1 for the i-th example.

3.3.4 Experiment 2: Artificial anomaly detection. After training the model on the training data, we can test it on images that we know to be anomalies. For example, we can test 500 anomalies and see how many of them the model classifies correctly. To classify, we have to create a threshold for whether we consider the input image to be an anomaly. The threshold is based on the mean and standard deviation of the reconstruction error for training images. Having the threshold be 2 standard deviations below and above the mean gives a trustworthy metric to base our anomaly decision on.

For detecting artificial anomalies we distinguish 2 different cases.

1: The anomaly is an image that is not part of the dataset. We use the fashionMNIST dataset for this purpose [10].

2: The anomaly is an image of the dataset, corrupted by Gaussian noise. (Same amount of noise for every anomalous image for every dataset. Scale = 0.2)

3.3.5 Experiment 3: Real anomaly detection. To detect real anomalies in the data, we train the autoencoder on normal images. After this training, the AE should have learned relevant features in the normal data and the reconstruction error for anomalous data should be higher. The MNIST dataset holds no real anomalies, so testing for this won't yield a result.

3.3.6 Experiment 4: Computational cost. The computational cost of a neural network depends on various factors, most of them being part of the training process (model architecture, amount of training data, etc.) By using the same architecture and training process for different datasets we can determine whether medical datasets have different computational costs by comparing the time it takes to train the model.

3.4 Evaluation

Experiment 1 will measure the reconstruction error, we will use the Binary Cross-entropy for this.

Experiments 2-3 will use the percentage of anomalies the model correctly classifies as anomalies given the standard deviation metric. 500 anomalies are tested.

Experiment 4 measures the amount of training input images divided by how long it takes to train the model in seconds.

4 RESULTS

Results of the autoencoder's performance on the datasets are available in Table 1. Figures 3-4 show samples of the PneumoniaMNIST dataset, both original and reconstructed. Figure 5 shows an example of the reconstruction error of 100 normal samples and 100 anomalous samples after the network is trained.

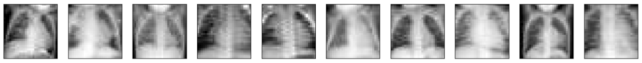

Figure 3: Original Input Images

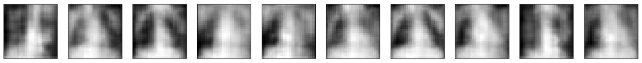

Figure 4: Reconstructed Output Images

5 DISCUSSION

This research paper evaluated the performance of a simple autoencoder structure on detecting anomalies in medical imaging. It strived to demonstrate whether the autoencoder can tackle medical data.

Table 1: Performance Statistics

	Recon. Error (BCE)	An. fashionMNIST	An. Gaussian noise	An. Real	Computational Cost
MNIST	0.3408	96.2	87.3	n/a	11000/324.35 =33.91
PneumoniaMNIST	0.7381	86.2	82.5	26.0	5232/145.52 = 35.95
OCTMNIST	0.5146	69.8	85.9	19.0	11000/336.17 = 32.72

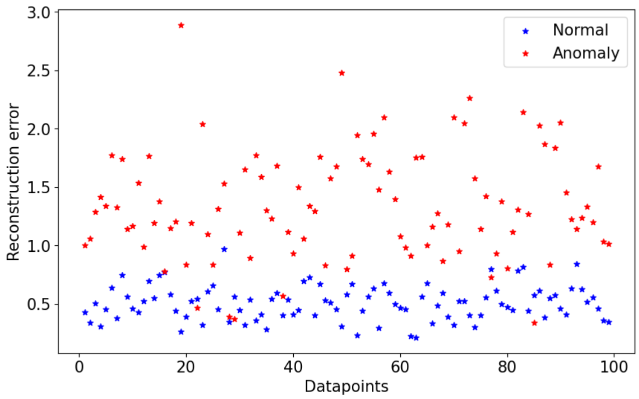

Figure 5: Reconstruction error for 100 normal and 100 anomalous samples

Although the desired result of having a high accuracy in detecting real anomalies in medical imaging was not achieved, there are still useful conclusions to be made. In this section, some general observations are highlighted and explained. Then the quality of the dataset and model are interpreted in terms of model performance and future work is suggested.

5.1 General observations

When looking at the results, the model scored well on the MNIST dataset. We can see that the reconstruction error of the PneumoniaMNIST is very high, likely because the images don't contain clearly defined shapes. The OCTMNIST resembles the MNIST better, the reconstruction error is lower.

Interestingly, training on the OCTMNIST and then feeding it fashionMNIST anomalies yielded only a score of 69.8%. A reason for this could be that the fashionMNIST contains images that look like the retinal scans or the OCTMNIST images not always being correctly centered.

Adding the same artificial noise to the images resulted in an MNIST anomaly set where the 0-9 handwritten numbers were still recognizable. The corrupted medical data, especially the PneumoniaMNIST set, resembled random data to the bare eye. It was however possible to find a noise 'sweet spot', where it was possible to see that the anomalies were X-Ray images of lungs/retinal scans while still achieving an accuracy of >75%. The computational cost for the medical datasets didn't differ from the MNIST dataset.

5.2 Dataset quality

During this research, only well-known and trusted datasets were used. Having a small image size was practical for doing experiments and it matched with our simple AE architecture. The MNIST dataset was regarded as 'normal' data due to its relevance, this is a point of discussion.

Medical data is difficult data, often containing complex structures, textures, and anatomical variations. The high variability in normal anatomy results in more diverse patterns, making it challenging for AE's to accurately capture and reconstruct all variations. Because of this reality, the reconstruction error is high, this in term means that finding anomalies is inconvenient, as the anomaly reconstruction error should be even higher for detection. Medical datasets are dissimilar, the relevant features are different for every dataset. In addition, medical anomalies are often subtle, and labeling the data is a demanding task, even for medical professionals.

5.3 Model quality

The model used in the experiments reached high accuracy on the MNIST dataset. Putting this model to the test in medical imaging proved that autoencoder performance is very dependent on the task at hand and the data available.

In cutting-edge research, the advanced AE models used are custom-designed to successfully carry out the anomaly detection challenge for a particular dataset and anomaly. Addressing this challenge involves careful architecture selection, regularization, appropriate data preprocessing, and collaboration with domain experts.

It is clear that to use this technology efficiently in the medical realm, knowledge about multiple expert fields is required.

5.4 Future work

There are avenues for future work that can be explored to address the limitations identified in this study. Future work could involve exploring a more diverse range of medical datasets. Some datasets might resemble the MNIST dataset more closely, resulting in better performance.

The way of tackling artificial anomalies created by Gaussian noise could be improved. The generation of synthetic anomalies that mimic the characteristics of real anomalies could help in fine-tuning the model architecture.

6 CONCLUSION

In conclusion, this study delved into how well AEs perform in reconstructing and detecting anomalies in medical images. The results indicate that AEs struggle with accurately reconstructing complex medical images due to their high resemblance to random data, making it challenging for AEs to pick up on relevant features.

When it comes to spotting artificial anomalies in medical images, AEs do well when trained on authentic medical data. However, finding the right amount of artificial noise is crucial for optimal performance, especially as medical data requires a higher degree of noise.

In terms of detecting real anomalies in medical images, the study's experiments didn't yield impressive results. Despite this, the literature proves that it is possible to reach high accuracy with custom-made models.

On the computational side, the cost of reconstructing medical images with AEs depends on factors like image size, model design, and training methods. The nature of the data itself doesn't play a significant role in this aspect.

REFERENCES

[1] Bank, D., Koenigstein, N., Giryes, R. (2023). Autoencoders. In: Rokach, L., Maimon, O., Shmueli, E. (eds) Machine Learning for Data Science Handbook. Springer, Cham. https://doi.org/10.1007/978-3-031-24628-9$_1$6

[2] N. Shvetsova, B. Bakker, I. Fedulova, H. Schulz and D. V. Dylov, "Anomaly Detection in Medical Imaging With Deep Perceptual Autoencoders," in IEEE Access, vol. 9, pp. 118571-118583, 2021, doi: 10.1109/ACCESS.2021.3107163.

[3] Anomaly Detection with Deep Perceptual Autoencoders. https://github.com/ninatu/anomaly_detection/

[4] David Zimmerer, Simon Kohl, Jens Petersen, Fabian Isensee, & Klaus Maier-Hein. (2020). Context-encoding Variational Autoencoder for Unsupervised Anomaly Detection – Short Paper.

[5] Siddalingappa, R., & Kanagaraj, S. (2021). Anomaly detection on medical images using Autoencoder and Convolutional Neural Network. *International Journal of Advanced Computer Science and Applications*, *12*(7). https://doi.org/10.14569/ijacsa.2021.0120717

[6] Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, *29*(6), 141–142. https://doi.org/10.1109/msp.2012.2211477

[7] Yang, J., Shi, R., Wei, D. *et al.* MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Sci Data* 10, 41 (2023). https://doi.org/10.1038/s41597-022-01721-8

[8] Bisong, E. (2019). Google Colaboratory. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA. https://doi-org.ezproxy2.utwente.nl/10.1007/978-1-4842-4470-8_7

[9] Sarvamangala, D.R., Kulkarni, R.V. Convolutional neural networks in medical image understanding: a survey. *Evol. Intel.* **15**, 1–22 (2022). https://doi-org.ezproxy2.utwente.nl/10.1007/s12065-020-00540-3

[10] Xiao, Han, e.a. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv:1708.07747, arXiv, 15 september 2017. *arXiv.org*, http://arxiv.org/abs/1708.07747.

[11] Wu, Y., Zhou, Y., Saveriades, G., Agaian, S., Noonan, J. P., & Natarajan, P. (2013). Local Shannon entropy measure with statistical tests for image randomness. *Information Sciences*, *222*, 323–342. https://doi.org/10.1016/j.ins.2012.07.049

[12] David E. Rumelhart; James L. McClelland, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations* , MIT Press, 1987, pp.318-362.

[13] Bank, D., Koenigstein, N., Giryes, R. (2023). Autoencoders. In: Rokach, L., Maimon, O., Shmueli, E. (eds) Machine Learning for Data Science Handbook. Springer, Cham. https://doi-org.ezproxy2.utwente.nl/10.1007/978-3-031-24628-9_16

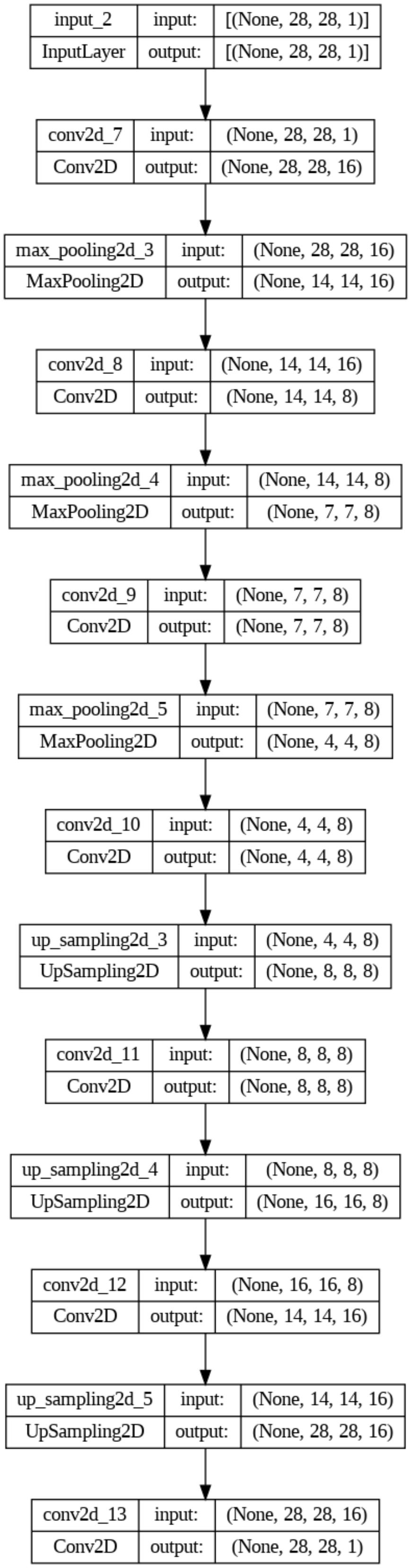
input_2 | input: | [(None, 28, 28, 1)]
InputLayer | output: | [(None, 28, 28, 1)]
conv2d_7 | input: | (None, 28, 28, 1)
Conv2D | output: | (None, 28, 28, 16)
max_pooling2d_3 | input: | (None, 28, 28, 16)
MaxPooling2D | output: | (None, 14, 14, 16)
conv2d_8 | input: | (None, 14, 14, 16)
Conv2D | output: | (None, 14, 14, 8)
max_pooling2d_4 | input: | (None, 14, 14, 8)
MaxPooling2D | output: | (None, 7, 7, 8)
conv2d_9 | input: | (None, 7, 7, 8)
Conv2D | output: | (None, 7, 7, 8)
max_pooling2d_5 | input: | (None, 7, 7, 8)
MaxPooling2D | output: | (None, 4, 4, 8)
conv2d_10 | input: | (None, 4, 4, 8)
Conv2D | output: | (None, 4, 4, 8)
up_sampling2d_3 | input: | (None, 4, 4, 8)
UpSampling2D | output: | (None, 8, 8, 8)
conv2d_11 | input: | (None, 8, 8, 8)
Conv2D | output: | (None, 8, 8, 8)
up_sampling2d_4 | input: | (None, 8, 8, 8)
UpSampling2D | output: | (None, 16, 16, 8)
conv2d_12 | input: | (None, 16, 16, 8)
Conv2D | output: | (None, 14, 14, 16)
up_sampling2d_5 | input: | (None, 14, 14, 16)
UpSampling2D | output: | (None, 28, 28, 16)
conv2d_13 | input: | (None, 28, 28, 16)
Conv2D | output: | (None, 28, 28, 1)

Figure 6: Autoencoder Model Architecture