Exploring Autoencoders for Dimensionality Reduction and Anomaly Detection

Ellie Kienast, Emmanuel Lyngberg
Department of Mathematics
Georgia Institute of Technology
December 6, 2023

***Abstract*— This research paper presents the architecture of Autoencoders and its application in dimension reduction and anomaly detection. Autoencoders are a form of neural network that are composed of three layers: an encoder, a latent layer, and a decoder. This structure allows it to be used in both dimension reduction and anomaly detection, of which we will explore through two simulated data sets as well as two medical imaging data sets sourced from Kaggle. Through this exploration we detail some of the strengths and weaknesses of Autoencoders as well as important points to mention for future work with this architecture.**

***Index Terms*— Autoencoders, Neural Networks, Dimension Reduction, Anomaly Detection**

I. INTRODUCTION

The Autoencoder architecture was first explicitly introduced (but referenced vaguely or without direct understanding a few years prior in other work) by Mark A. Kramer as a form of non-linear principal component analysis. In his paper he describes the basic architecture as a feedforward network, now called the encoder, followed by a "bottleneck layer", now called the latent layer, and then completed by an output layer, now called the decoder, that reproduces the network inputs. The goal was to provide a general purpose feature extraction algorithm to produce features that retain the highest amount of information without the need for the scientists to fully understand the nature of the relationships between features in the dataset. However, the popularity of Autoencoders explicitly for dimensionality reduction did not take hold until over a decade later thanks in part to work by G.E. Hinton and R. R. Salakhutdinov.

This report first explains in depth the architecture itself, and its ability to handle non-linear relationships unlike PCA. With that established, the second section explores that non-linear ability in comparison to PCA with two simulated datasets: The Swiss roll and S-curve. The third section then discusses the natural extension of Autoencoders in anomaly detection. There are two datasets employed for anomaly detection. The first dataset consists of chest x-ray scans exhibiting healthy lungs and lungs suffering from Pneumonia. The second dataset consists of brain MRI scans that depict healthy brains, brains with pituitary gland tumors, brains with glioma tumors, and brains with meningioma tumors. Within each section, a brief discussion and description of the data used and the reasoning behind its choice is given prior to discussing implementation, algorithms, and results.

II. AUTOENCODER ARCHITECTURE

As was introduced previously, the autoencoder is a specific feedforward neural network that exhibits an hourglass shape in that it first compresses input into a latent space, from which it decompresses into a reconstruction that matches the input dimensions. If the latent space were to retain the same dimensionality as the input, we would no longer be capturing the spirit of an autoencoder as there would be no real compression being used. Below is a visualization of the autoencoder architecture and how it takes an input (in this case the image of a digit), compresses it, and then tries to reconstruct the digit.

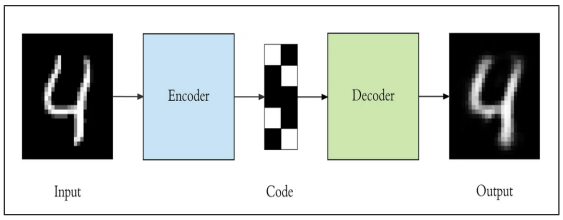

Fig. 1. Visualization of the autoencoder architecture

In gauging the effectiveness of the autoencoder's ability to generate a valuable latent layer, we use mean-squared error loss when comparing the reconstruction to the original input. One could use a sister loss function such as the L1 loss, but each one can be chosen based on the data and end goal in mind. When building an autoencoder, the main hyperparameters to consider are the size of the latent dimension, the number of layers between input and the compressed form, and the activation functions chosen. It is important to note here that it is the non-linear activation functions that are a hallmark of neural networks that give autoencoders the ability to learn non-linear relationships within the data and allow it to reconstruct non-linear data considerably more accurately than PCA. Additionally, it is also important to mention here that there exist many variations of the autoencoder (one is actually called a variational autoencoder) that are designed to tackle more specific sub-problems in the realm of dimension reduction or reconstruction. Here we focus on the general architecture and tuning process.

Given that autoencoders are unsupervised, the training process requires more care and finesse than what one might traditionally encounter when training a supervised neural network. When discussing anomaly detection there will be a larger discussion around this fact, but we can introduce this idea by noting that in many cases it can be quite difficult to understand the latent space itself when dealing with higher dimensional data, due to the fact that we cannot visualize the latent dimension well past three dimensions. Thus, since we cannot directly know ahead of time what the autoencoder will consider valuable information, or force it to focus on specific information in a supervised fashion, we can only tune the activation functions and depth of the layers based on our understanding of the linearity (or lack thereof) in the data as well as the inherent complexity in the relationships we choose to explore. Whilst this seems like a disadvantage at first, it can actually be quite valuable and is used quite often as an initial preprocessing technique to help scientists that are unsure about new data they have obtained to better understand the nature of the data through the encoding and decoding process, and the subsequent reconstruction losses obtained from that process.

III. DIMENSIONALITY REDUCTION IN COMPARISON WITH PCA

A. Dimension Reduction Capability

In order to examine the dimension reduction capabilities of autoencoders, we utilized two simulated datasets: the Swiss Roll and the S-Curve.

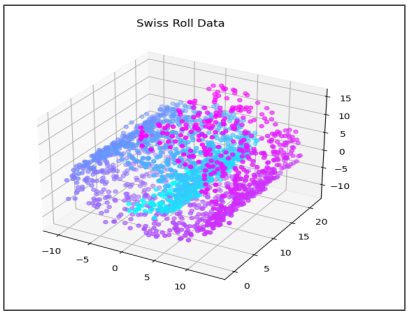

Fig. 2. Swiss Roll dataset generated with Scikit-learn.

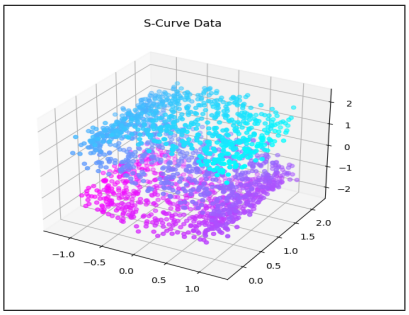

Fig. 3. S-Curve dataset generated with Scikit-learn.

Initially for each dataset, we encoded the data into a two-dimensional latent representation.

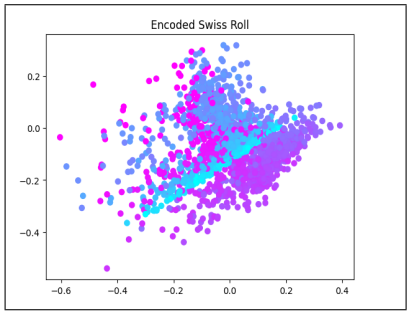

Fig. 4. Two-dimensional latent representation of Swiss Roll dataset.

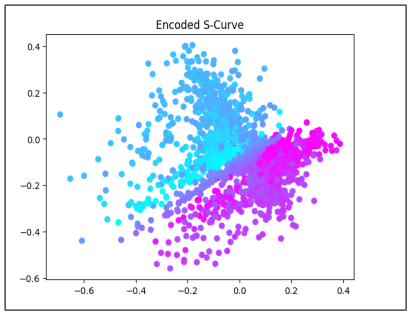

Fig. 5. Two-dimensional latent representation of S-Curve dataset.

However, to measure the dimension reduction ability of autoencoders, it is necessary to examine the latent representations of our datasets for a variety of latent dimension values. In particular, we quantified this ability by computing the associated mean-squared error for each latent dimension. The mean-squared error is calculated by comparing the decoded version of our latent representation to the original input. This error indicates how well the autoencoder processed and retained important features and underlying structure when performing dimension reduction. We see the resulting errors for the training and test sets for both the Swiss Roll and S-Curve data below.

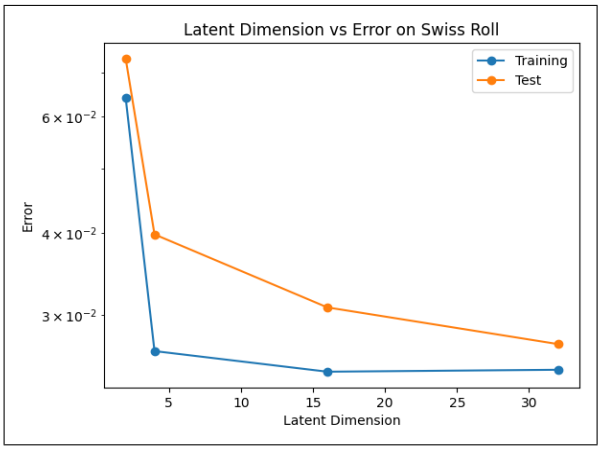

Fig. 6. Graph depicting the mean-squared error as a function of latent dimension for the Swiss Roll dataset.

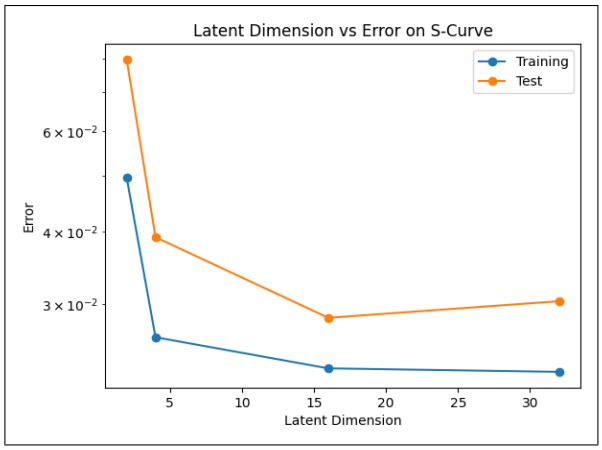

Fig. 7. Graph depicting the mean-squared error as a function of latent dimension for the S-Curve dataset.

It is clear that as the latent dimension increases, the mean-squared error across both the training and test sets decreases. This is an expected result, as a greater latent dimension means more features are retained in the compressed form from which the decoded output is reconstructed. Thus, we have a more accurate reconstruction.

B. Generating New Samples

Using the encoded latent representations of both simulated datasets, we can generate new, decoded representations of our data.

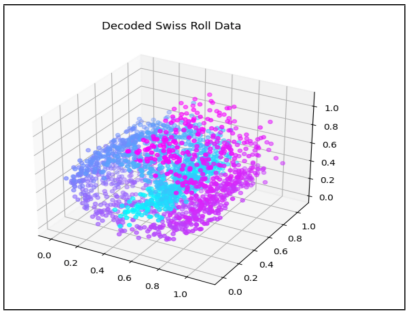

Fig. 8. Decoded output from two-dimensional Swiss Roll latent representation.

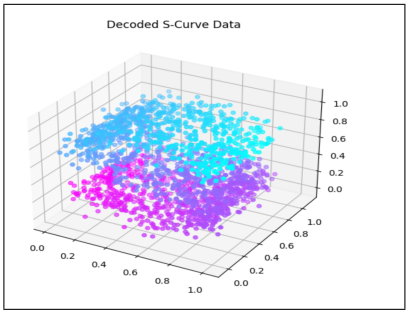

Fig. 9. Decoded output from two-dimensional S-Curve latent representation.

Examining Fig. 8. and Fig. 9., we find that the decoded outputs generated by our autoencoder hold a strong resemblance to the original input. The effectiveness of our autoencoder's reconstruction ability is also highlighted by the decreasing mean-squared errors presented in Fig. 7. Though we only present the decoded representations from two-dimensions here, we observed improved accuracy in decoded outputs generated from higher-dimensional latent representations.

C. Comparison with PCA

We begin by briefly discussing the inherent differences between dimension reduction via PCA and autoencoders. Firstly, PCA is a linear technique that seeks to find the principal components, which represent linear combinations of the original features. Autoencoders, on the other hand, have the ability to learn non-linear transformations. Furthermore, PCA focuses on capturing the global structure and variance in the data while autoencoders have the potential to capture local structure as well as non-linear data patterns. Due to these differences, we observe that the two-dimensional latent representations of our simulated datasets look much different when compressed using PCA. We can clearly observe these differences by comparing the figures below with the encoded version presented in Fig. 4 and Fig. 5.

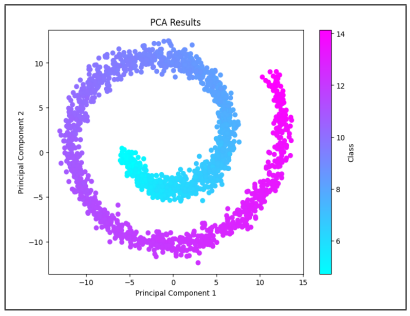

Fig. 10. Two-dimensional latent representation of Swiss Roll dataset using PCA.

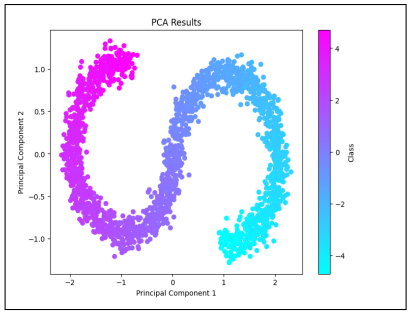

Fig. 11. Two-dimensional latent representation of S-Curve dataset using PCA.

We now look to compare PCA and autoencoder dimension reduction ability. For our purposes, we quantified dimension reduction performance by evaluating the reconstruction error associated with mapping our latent representations generated by PCA and autoencoder back to the data's original dimension. Without loss of generality,

we focus on results for our Swiss Roll dataset. Due to nature of PCA, we can only examine reconstruction error across one, two and three dimensions. In Fig. 9. below, we plot the reconstruction error as a function of latent dimension for both PCA and autoencoder.

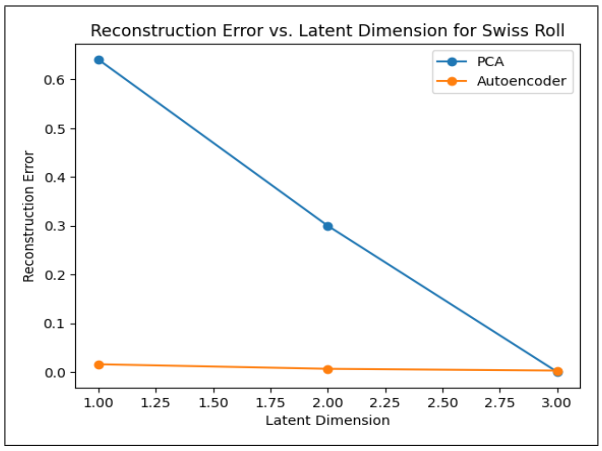

Fig. 12. Graph displaying reconstruction error for PCA and autoencoder for various latent dimensions.

We find that the reconstruction error for PCA is consistently higher than that of our autoencoder for each latent dimension. As previously mentioned, PCA performs best when the underlying structure of the data is linear in nature. Autoencoders, alternatively, have a greater ability in capturing non-linear patterns. Hence, in the case of the Swiss Roll dataset, which has an apparent non-linear structure, our autoencoder outperforms PCA in compressing the data across all latent dimensions.

IV. ANOMALY DETECTION WITH BRAIN TUMORS AND PNEUMONIA

One of the beautiful qualities inherent in the architecture of autoencoders is the fact that it can be used to reconstruct data from an encoded, compressed layer. In this way, it is able to help us better understand the most important features and relationships in complex data which can help in making models more efficient and more accurate. However, this unique architecture also allows us to solve interesting problems in a similar spirit to Generative Adversarial Networks (GANs) because of its reconstruction or generative abilities. The rationale here is that by allowing the autoencoder to organically determine the most relevant features in a dataset and train to reconstruct based on those features, we can feed the network a particular class of data such that it becomes very adept at reconstructing that particular kind of data. Then, if it were to encounter new data that contained some form of anomaly, or deviation in feature characteristics, it would suddenly stumble and create a reconstruction with a significant increase in reconstruction loss. It is then this disparity in reconstruction loss that is used to determine whether or not the data given was anomalous. The added benefit is that if we carefully construct the autoencoder, it can even uncover relationships in complex data that give away the anomalies in ways we could not foresee. As an example, if a particular cancer attacked liver cells, one would naturally suspect that an autoencoder trained on data used to determine the existence of the cancer would struggle because of the deviation in the number of healthy liver cells. However, it could also be that before there is a significant decrease in liver cells, there is a particular hormonal marker that is weakened or heightened due to the presence of the cancer that would not be immediately obvious to an oncologist studying the disease. These kinds of insights further the attractiveness of the autoencoder.

With the concept of anomaly detection using autoencoders introduced, we now move onto the task selected in testing the anomaly detection capabilities of a convolutional autoencoder with regard to brain MRI scans and chest x-rays. The architecture used was relatively simple and straightforward for the ease of reproducibility as well as the fact that not much computational power was available to train with, considering convolutional networks for image processing can be quite costly. The architecture is outlined below:

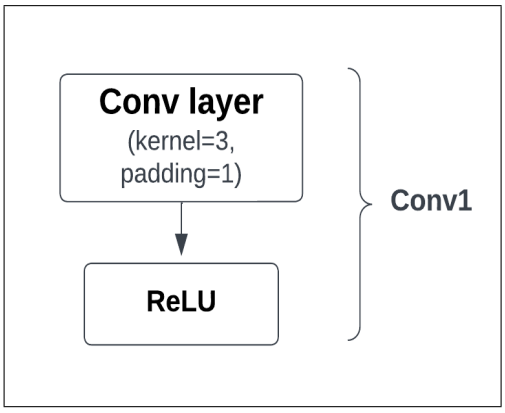

Fig. 13. The first type of convolutional layer followed by ReLU activation, denoted conv1

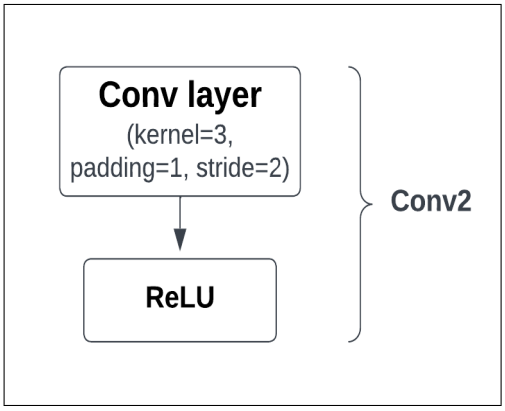

Fig. 14. The second type of convolutional layer followed by ReLU activation, denoted conv2

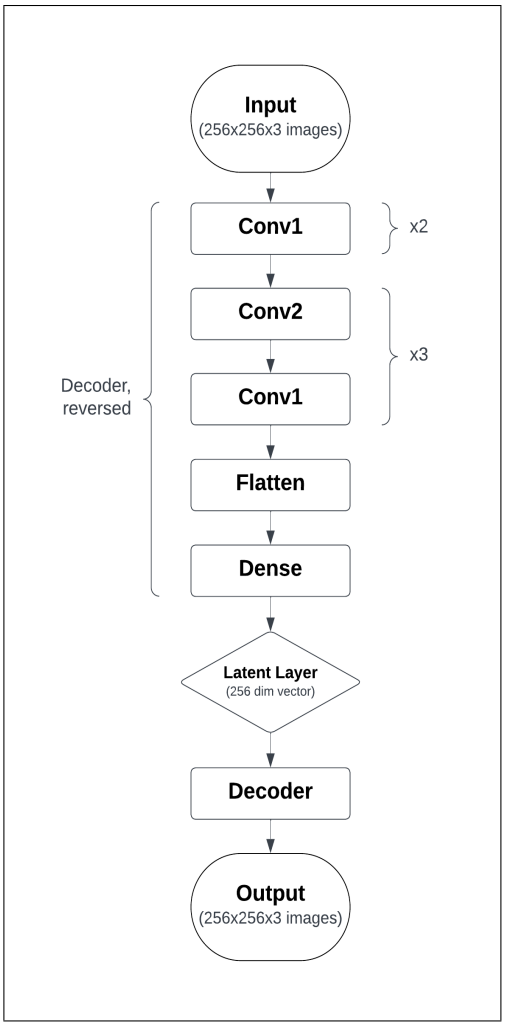

Fig. 15. Visualization of the convolutional autoencoder architecture

Figures 13 and 14 are used to simplify the full architecture diagram in Figure 15 by combining the two convolutional layers we used along with their activation functions. Altogether there are 5 convolutional layers, each with a non-linear ReLU activation layer between them. To then reach the final latent layer representation, we flatten the output of the final convolution and compress it to a vector of length 256. This was chosen to make it easier to understand how powerful this process can be, as the latent representation has dimensions 256x1, whilst the original input had dimensions 256x256x3. We then train two convolutional autoencoders (both with this same architecture) on a dataset of only healthy chest x-rays and one with only healthy brain scans, with the goal in mind of creating models that can reconstruct healthy scans with high accuracy. Then, we test the first model on a dataset containing both healthy chest x-rays and chest x-rays of patients with Pneumonia. Similarly, we test the second model on a dataset containing both healthy brain scans and brain scans exhibiting glioma, meningioma, and pituitary tumors. Note that both datasets were not perfectly balanced (between 60-40 and 70-30 unhealthy to healthy) due to the datasets containing significantly more unhealthy scans. This is not surprising as a hospital or clinic is more likely to store scans exhibiting Pneumonia or brain tumors for educational purposes, whereas they regularly see healthy scans that are less likely to be saved unless a patient would need to see them later for a particular reason.

The following plots show the training loss per epoch for both autoencoders training on chest x-rays and brain MRI scans individually.

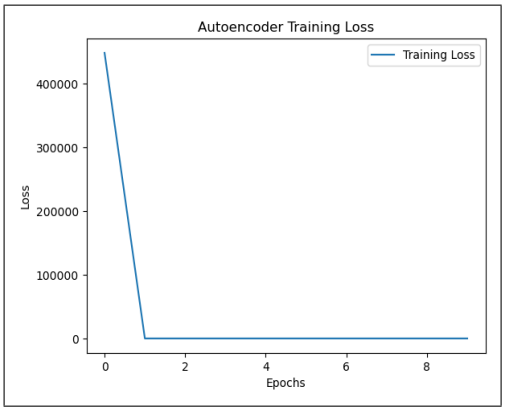

Fig. 16. Loss per epoch for training on healthy brain MRI scans

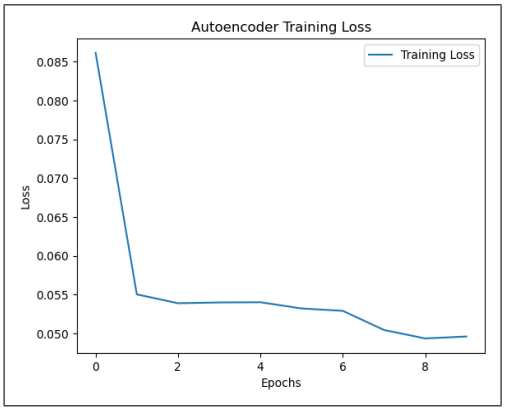

Fig. 17. Loss per epoch for training on healthy chest x-ray scans

As we can see, the model improves quite quickly in both cases, finishing the training process with very low training loss in both cases: 0.05 for the chest x-rays and under 0.01 for the brain MRI scans. However, this is only a start, as the important part is disparity in reconstruction losses obtained when we test on a dataset containing healthy and unhealthy scans. Below are the results for both the brain MRI scans and the chest x-rays. Note that we label all types of brain tumors as anomalous and did not differentiate between tumors in separate testing runs.

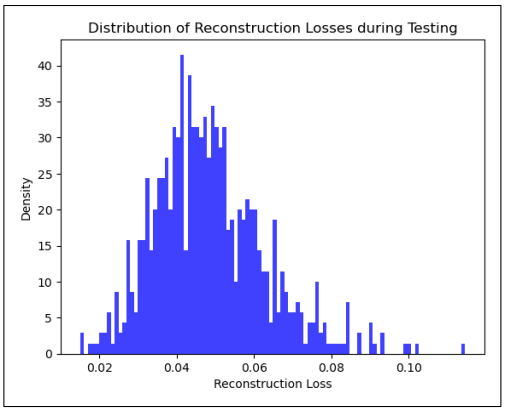

Fig. 18. The distribution of reconstruction losses per image for 700 chest x-ray images

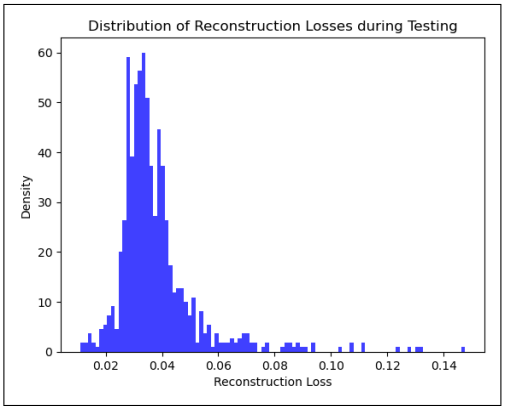

Fig. 19. The distribution of reconstruction losses per image for 800 brain MRI scan images

From these results, we can see that the model did indeed train well on the healthy images as it is reconstructing them well with mean MSE error below 0.05 in both cases. However, the model is also performing very similarly for anomalous MRI scans and chest x-rays. In tuning the model, a separation in mean MSE loss between anomalous and non-anomalous scans was achievable, however the variance was too great to be able to set a consistent threshold for consistently high accuracy. In the case of brain MRI scans, we achieved 65 percent accuracy on the validation set, and 55 percent accuracy for chest x-rays. To remedy this, we attempted to pre-process the images by heightening contrast in the images in an effort to enhance anomalous color deviations in anomalous images, however this proved to have little effect. These results highlight the unpredictability and difficulty to tune autoencoders which is partially due to the complex hidden nature of the latent space but also the fact that it is an unsupervised learning method, and therefore will pick up on information it finds most useful, regardless of whether it suits our specific goal.

V. CONCLUSIONS

Autoencoders have powerful dimension reduction capabilities due to their complexity. They are able to capture non-linear data structures effectively in data compression, as shown through our examinations with the Swiss Roll and S-Curve datasets. Furthermore, we were able to clearly establish that data compression performance via our autoencoder has a positive correlation with latent dimension. Using our autoencoder, we were able to generate new samples from the latent representations of our data in the form of a decoded output. Our autoencoder proved to do this effectively across multiple latent dimensions. In comparing PCA and autoencoders for dimension reduction, we found that our autoencoder displayed superior peformance compared to PCA. This is likely the result due to non-linearity in our simulated datasets. We should consider that for datasets with inherently linear structure, dimension reduction via PCA has the potential to perform equally.

As we saw in the section on anomaly detection, the power of autoencoders for image reconstruction cannot be overstated. However, due to the complex and hidden nature of the latent layer in higher dimensions, tuning the model to focus on the differentiating features between normal and anomalous images can be a difficult task. Other than network depth and latent dimension there are no other model hyperparameters to tune, requiring further preprocessing even with convolutional layers. This further cements the necessity for a deep understanding of the anomalous characteristics that the autoencoder can decipher, and therefore how one might go about preprocessing the images to highlight those key characteristics. In the case of medical images, the high image resolution, variety of image angles, and inherent noise present in a brain MRI or chest x-ray make it especially challenging. Additionally, if we wish to support physicians with this kind of technology, we require a greater level of robustness and accuracy than we might elsewhere given that decisions made from these results can alter lives. Altogether, it is clear that there is great potential for this general architecture and work continues to be done on altering the architecture to better suite specific needs, however with this complexity comes an up hill challenge to achieve consistent and reproducible results.

DIVISION OF WORK

Proposal: Both group members completed half of the written proposal that was submitted.

Proposal Presentation: Ellie created the presentation slide layout and both group members filled out the slides according to their respective research on the proposal.

Project Code: Ellie was responsible for the dimension reduction and PCA comparison portion and Emmanuel was responsible for the anomaly detection portion.

Final Presentation: Ellie created the presentation slide layout and both group members filled out the slide according to the work they did for the project coding/analysis

Final Project Report: Emmanuel created the report, wrote the introduction, architecture, and anomaly de-

tection sections. Ellie wrote the sections for dimension reduction, comparison with PCA, and generation. Both group members edited and formatted the final report, and added respective references.

REFERENCES

[1] Bandyopadhyay, H. (2023, July 11). Autoencoders in Deep Learning: Tutorial and Use Cases [2023]. V7. [online]. Available: https://www.v7labs.com/blog/autoencoders-guide#h2. [Accessed 3 Dec. 2023].

[2] Dertat, A. (2017, Oct 3). Applied Deep Learning - Part 3: Autoencoders. [Online] Available: https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798. [Accessed 3 Dec. 2023].

[3] Herath, Jerome D. (2020, Dec 27). Anomaly detection with deep autoencoders. Available: https://www.dinalherath.com/2020/autoencoder/. [Accessed 3 Dec. 2023].

[4] Kramer, Mark A. (1991, Feb). Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. [Online]. Available: https://people.engr.tamu.edu/rgutier/web_courses/cpsc636_s10/kramer1991nonlinearPCA.pdf. [Accessed 3 Dec. 2023].

[5] Mooney, P. (2017). Chest X-Ray Images (Pneumonia) [Dataset]. Available: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia.

[6] Nickparvar, M. (2021). Brain Tumor MRI dataset [Dataset]. Available: https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset.

[7] Hinton, G. E. and Salakhutdinov, R. R.(2006, July 28). Reducing the Dimensionality of Data with Neural Networks. [Online]. Available: https://www.cs.toronto.edu/~hinton/absps/science.pdf. [Accessed 3 Dec. 2023].

[8] Shannon, Ali. (2019). Autoencoders. [Online]. Available: https://github.com/techshot25/Autoencoders/blob/master/README.md. [Accessed 14 Oct. 2023].

[9] Vineyadula. (2023, Jan 23). Swiss Roll Reduction with LLE in Scikit Learn. [Online]. Available: https://www.geeksforgeeks.org/swiss-roll-reduction-with-lle-in-scikit-learn/. [Accessed 14 Oct. 2023].

[10] Weng, L. (2018, Aug 12). From Autoencoder to Beta-VAE. [Online]. Available: https://lilianweng.github.io/posts/2018-08-12-vae/. [Accessed 3 Dec. 2023].