Exploring Enhanced Architectures for Convolutional Autoencoders

Mert Erkol
Computer Science Department

Asu Guvenli
Physics Department

Zahra Koulaeizadeh
Computer Science Department

***Abstract*—This report presents an exploration of enhanced architectures for convolutional autoencoders. The methodology involved replacing the Rectified Linear Unit (ReLU) activation function with LeakyReLU, modifying the network architecture to include skip connections and an additional convolutional layer, and investigating the effects of different loss functions including L1, Binary Cross-Entropy (BCE), and a combination of both. The experiments aimed to improve the performance of the autoencoder in terms of reconstruction accuracy and feature extraction capabilities. The findings provide insights into the potential enhancements achievable through alternative architectural choices and loss functions in convolutional autoencoders.**

***Index Terms*—Convolutional Autoencoders, skip connections, residual blocks**

I. Introduction

Convolutional Autoencoders (CAEs) have emerged as powerful deep learning models for unsupervised feature learning and data representation. They are particularly well-suited for processing high-dimensional data, such as images, due to their ability to capture spatial hierarchies and local dependencies. CAEs consist of two main components: an encoder and a decoder. The encoder performs a series of convolutional and pooling operations, gradually reducing the spatial dimensions of the input data while extracting meaningful features. The decoder, on the other hand, employs transpose convolutions to reconstruct the original input from the learned features. By training the CAE to minimize the reconstruction error, it is possible to obtain latent representations that effectively capture the salient characteristics of the input data.

Skip connections, also known as residual connections, have proven to be beneficial in various neural network architectures. They involve the direct connection of earlier layers to later layers, bypassing intermediate layers. This architectural design allows for the flow of information from the lower-resolution feature maps to higher-resolution feature maps, thereby facilitating gradient propagation and alleviating the vanishing gradient problem. Skip connections have been successfully employed in deep residual networks (ResNets) and U-Net architectures, among others, to improve the overall performance of the models. In the context of CAEs, skip connections can potentially enhance the reconstruction quality and the ability to capture fine details by preserving and leveraging low-level spatial information.

In this project, we investigate the application of skip connections in Convolutional Autoencoders. We explore different architectural configurations by integrating skip connections at various stages of the CAE. By doing so, we aim to evaluate the impact of skip connections on the reconstruction performance and feature learning capabilities of the CAE. To assess the effectiveness of the proposed enhanced architectures, we conduct extensive experiments on three distinct datasets: MNIST Dataset, CIFAR-10 Dataset, and CINIC-10 Dataset. Through our evaluation, we aim to provide insights into the benefits and limitations of incorporating skip connections into Convolutional Autoencoders and shed light on their potential for improving the quality of learned representations and reconstruction accuracy.

II. Related Work

Convolutional Autoencoders (CAEs) have gained significant attention in the field of image processing and computer vision due to their ability to learn compact representations of input images through unsupervised learning. To provide a visual understanding of the working principle of CAEs, Figure 1 illustrates a schematic diagram showcasing the typical architecture of a Convolutional Autoencoder, depicting the encoding and decoding processes involved in the reconstruction of input images.

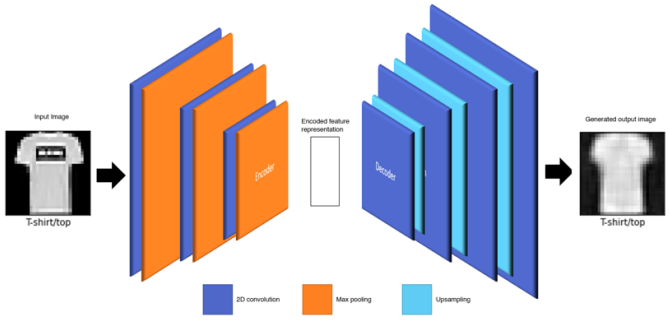

Fig. 1. Convolutional AutoEncoders Structure

The utilization of CAEs has shown promising results in various applications, including image denoising, dimensionality reduction, and anomaly detection. However, to further improve the performance of CAEs, researchers have explored enhanced architectures and techniques, such as skip connections, which allow for the integration of information from different layers in the network.

"Image Resolution Enhancement Using Convolutional Autoencoders with Skip Connections" [3] proposes an approach

that utilizes CAEs with skip connections to improve image resolution, restoration, and denoising. By incorporating skip connections from the encoder's initial layers to the decoder's final layers, the authors achieve impressive performance in reconstructing images.

Despite the advancements in CAE architectures and related techniques, there are still gaps and limitations that exist in the current research. One limitation is the lack of comprehensive evaluations and comparisons of different CAE architectures and techniques. Many studies focus on specific applications or datasets, making it challenging to generalize the findings. Moreover, the interpretability of CAEs remains a challenge, as understanding the learned representations and their corresponding features can be difficult. This hinders the broader adoption of CAEs in practical applications.

The goal of this project is to address these gaps and limitations in existing research. We aim to explore enhanced architectures for CAEs by incorporating skip connections and other novel techniques. Through a comprehensive evaluation and comparison of different architectures, we seek to identify the most effective approaches for improving the performance of CAEs. Additionally, we will investigate methods to enhance the interpretability of CAEs, providing insights into the learned representations and features. By addressing these research gaps, this project aims to contribute to the advancement of CAEs and their applications in image-processing tasks.

III. Methodology

We initially constructed a basic convolutional autoencoder and conducted experiments with different parameters, including the number of convolutional and fully-connected layers in both the encoder and decoder parts, kernel sizes, strides, padding, and the number of channels or nodes in each layer. To maintain simplicity, we opted for a symmetric architecture. Following hyperparameter tuning, the architecture depicted in Figure 2 was selected.

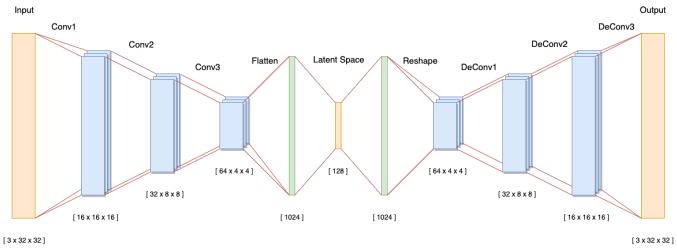

Fig. 2. Base Convolutional AutoEncoder architecture. Tensor sizes are given for CIFAR-10 AND CINIC-10 datasets.

We aimed to improve the performance of convolutional autoencoders by implementing various modifications. In this section, we describe the key changes we made to the original architecture and the experiments we conducted.

A. 1. Activation Function Replacement:

To enhance the expressiveness of the autoencoder, we replaced the Rectified Linear Unit (ReLU) activation function with LeakyReLU. LeakyReLU allows for a small negative slope for negative input values, which can help alleviate the vanishing gradient problem and promote more robust feature extraction.

B. 2. Improved Network Architecture:

We further improved the autoencoder's architecture by making two significant changes. First, we removed one of the fully connected layers from the original design. Instead, we introduced an additional convolutional layer, resulting in a network structure consisting of four convolutional layers followed by one fully connected layer (4 CNN + 1 FC). This modification aimed to increase the network's capacity for capturing more intricate features in the encoded representation.

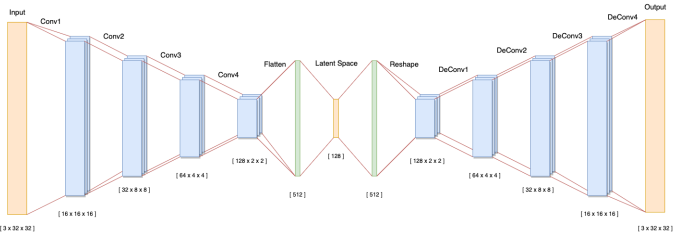

Fig. 3. Improved Convolutional AutoEncoder architecture. Tensor sizes are given for CIFAR-10 AND CINIC-10 datasets.

Secondly, we incorporated skip connections within the encoder. Skip connections establish direct connections between earlier layers and later layers of the network, allowing information from lower-resolution feature maps to bypass the bottleneck structure. By enabling the flow of both high-level and low-level features, skip connections can enhance the network's ability to reconstruct the input data faithfully.

C. 3. Loss Function Variation:

We explored the impact of different loss functions on the training and reconstruction performance of the autoencoder. Specifically, we experimented with four types of loss functions: L1 loss, MSE, Binary Cross-Entropy (BCE) loss, and a combination of both. L1 loss measures the absolute difference between the input and the reconstructed output, while BCE loss computes the binary cross-entropy between the input and output pixel values. By combining these two loss functions, we aimed to leverage their complementary properties and potentially achieve a more comprehensive reconstruction.

Through our experiments, we evaluated the performance of the modified autoencoder architecture by comparing the results obtained with each variation. The evaluation metrics included reconstruction accuracy, feature extraction capabilities, and training convergence. By systematically examining these changes and their effects, we aimed to uncover insights into the potential enhancements that can be achieved by exploring alternative architectural choices and loss functions in convolutional autoencoders.

IV. Evaluations

We utilized PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) as evaluation metrics to assess the performance of our enhanced architectures for convolutional autoencoders.

A. Peak Signal-to-Noise Ratio (PSNR)

PSNR [7] is a widely used metric for measuring the quality of reconstructed or compressed images. It quantifies the difference between an original image and a reconstructed image by considering the peak value of the signal and the amount of noise present.

The mathematical representation of PSNR is as follows :

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \tag{1}$$

in which MAX represents the maximum possible pixel value of the image and MSE stands for Mean Squared Error, which is calculated by averaging the squared differences between corresponding pixels in the original and reconstructed images.

The higher the PSNR value, the lower the perceived distortion between the original and reconstructed images. PSNR is often expressed in decibels (dB).

B. Structural Similarity Index (SSIM)

The Structural Similarity Index Measure (SSIM) [6] is a method used to predict the perceived quality of digital images and videos, including those in digital television and cinema. It measures the similarity between two images by considering their structural information and incorporating important perceptual phenomena like luminance masking and contrast masking.

SSIM differs from other techniques such as MSE or PSNR, which estimate absolute errors. Instead, SSIM focuses on the perceived change in structural information caused by image degradation. It recognizes that pixels in an image have strong inter-dependencies, especially when they are spatially close, providing valuable information about the objects' structure in the visual scene.

SSIM compares two images using a formula:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2}$$

Where :

- μ_x and μ_y: They represent the average (mean) values of x and y, respectively. These values are calculated by summing all the pixel values in the signal or image and dividing by the total number of pixels.
- σ_x^2 and σ_y^2: These are the variances of x and y, respectively. They measure the statistical dispersion or spread of the pixel values in the signal or image.
- σ_{xy}: It represents the covariance between x and y. The covariance measures the degree to which the pixel values of x and y vary together.
- C_1 and C_2: These are small constants added to the formula to avoid instability when the means and variances are close to zero. They are typically set to small positive values.

V. Datasets

We conducted experiments on three distinct datasets: MNIST, CIFAR-10, and CINIC-10.

A. MNIST

The MNIST dataset [5] is a commonly employed benchmark in the domain of image classification. It comprises 60,000 grayscale images of handwritten digits, ranging from 0 to 9, each with a resolution of 28x28 pixels. The dataset is split into 50,000 training images and 10,000 test images. Our objective was to assess the effectiveness of the enhanced CAE architectures in accurately reconstructing and classifying these digit images, leveraging the well-established MNIST dataset.

B. CIFAR-10

For a more comprehensive evaluation, we utilized the CIFAR-10 dataset [8], another popular benchmark for image classification tasks. CIFAR-10 comprises 50,000 color images categorized into ten distinct classes, with each image having dimensions of 32x32 pixels. The dataset is partitioned into 40,000 training images and 10,000 test images. By conducting experiments on CIFAR-10, we aimed to evaluate the performance of the enhanced CAE architectures in handling more complex and diverse datasets and to assess their capabilities in accurately reconstructing and classifying the color images present in this dataset.

C. CINIC-10

In addition to MNIST and CIFAR-10, we also incorporated the CINIC-10 dataset [4] into our experiments. CINIC-10 is a composite dataset that combines images from CIFAR-10 and the ImageNet dataset. It consists of 270,000 training images and 15,000 test images distributed across ten different classes. By including CINIC-10 in our experiments, we aimed to evaluate the generalization capabilities of the enhanced CAE architectures and their performance on a more realistic and challenging dataset. This allowed us to assess how well the architectures could handle diverse visual content and classify images from a broader range of categories.

VI. Experiments

A. Optimizers

We experimented with seven different optimizers, tuning the learning rate for each optimizer while keeping the remaining parameters constant for the purpose of comparison. Similarly to the rest of this work, we evaluated the performance using the PSNR and SSIM metrics. The results can be observed in Figure 3. We can say Adam and Adamax outperformed the others.

B. Loss Functions

To observe the effect of the loss functions in the performance of our network we tried 4 loss functions:

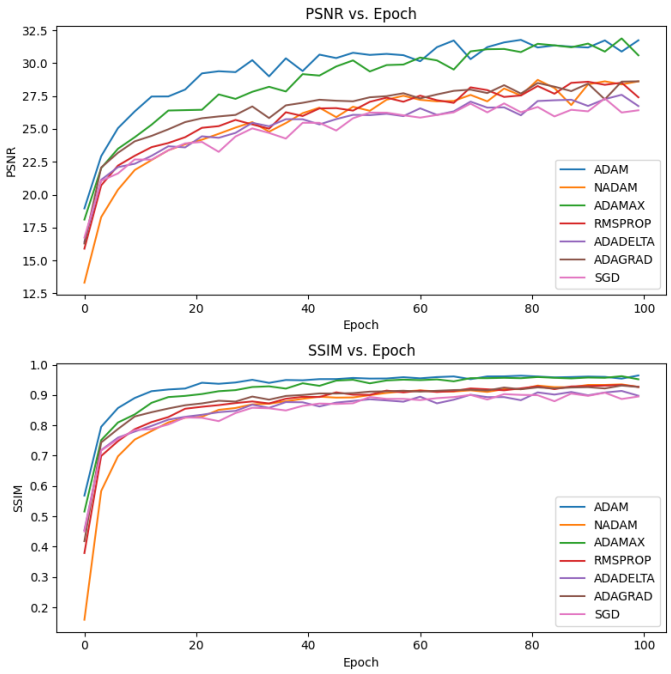

Fig. 4. Comparison of optimizers

1) L1 Loss (Mean Absolute Error):: L1 loss [1], also known as Mean Absolute Error (MAE), measures the average absolute difference between the input and the reconstructed output. It is a common loss function used for regression tasks and can be formulated as follows:

$$L1Loss = \frac{1}{N}\sum |X - X'| \tag{3}$$

2) Mean Squared Error (MSE): Mean Squared Error (MSE) is a widely used loss function that measures the average squared difference between the input and the reconstructed output. It is commonly used for regression tasks and provides a measure of the overall reconstruction error. The formula for MSE is as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(X_i - X_i')^2 \tag{4}$$

where N represents the total number of elements in the input data.
X_i is the i-th element of the input data.
X_i' is the i-th element of the reconstructed output.

3) Binary Cross-Entropy Loss:: Binary Cross-Entropy (BCE) [2] loss is commonly used for binary classification tasks, but it can also be applied to autoencoder reconstruction tasks by treating each pixel value as a binary prediction. BCE loss measures the dissimilarity between the input and the reconstructed output by computing the binary cross-entropy between their pixel values. The formula for BCE loss is as follows:

$$BCELoss = -\frac{1}{N}\sum[X\log(X') + (1-X)\log(1-X')] \tag{5}$$

Where:

- N represents the total number of elements in the input data.
- X is the input data (binary values).
- X' is the reconstructed output (predicted probabilities).

C. Combination of L1 and MSE Loss:

To leverage the complementary properties of L1 and MSE loss, we explored a combination of these two loss functions. The combined loss function can be defined as follows:

$$CombinedLoss = \alpha \cdot \text{L1 Loss} + \beta \cdot \text{MSE Loss} \tag{6}$$

in which α and β are weighting factors that determine the relative importance of each loss component.

The results of our experiments is shown in results section.

D. Skip-Connections

Initially, we explored the integration of skip connections within the encoder and decoder stages of the architecture. Figure 5 illustrates the different skip connection combinations that were evaluated. This picture represents connections applied on Base-CAE Network's encoder part. Since the tensor size in each of these layers was different, reshaping of the residual was necessary to enable skip connections. We experimented with two different approaches for reshaping the residuals: bilinear interpolation and zero padding, and a convolutional reshaper.

In the first approach, bilinear interpolation and zero padding were employed to reshape the residual tensors. Bilinear interpolation was used to upsample or downsample the residuals to match the dimensions of the target layer. Additionally, zero padding was applied to adjust the spatial dimensions of the residuals, ensuring compatibility with the target layers.

In the second approach, we employed a convolutional reshaper to reshape the residuals. A convolutional layer with a suitable kernel size and stride was utilized to adjust the tensor dimensions and align them with the target layers.

But none of the methods improve the PSNR or SSIM and we did not include them in the code or report to not confuse anyone. There is Residual Block class in the code that is unused. We think that we did not see the real affect of a residual block inside the encoder since our network is not deep enough.

VII. Results

In this study, we investigated the impact of skip connections on the performance of an encoder-decoder architecture. Specifically, we explored the effects of skip connections within the encoder and decoder, as well as connections across the bottleneck layer.

Initially, we incorporated skip connections within the encoder and decoder, expecting them to enhance information flow and improve performance. However, our experiments revealed that these internal skip connections did not yield a

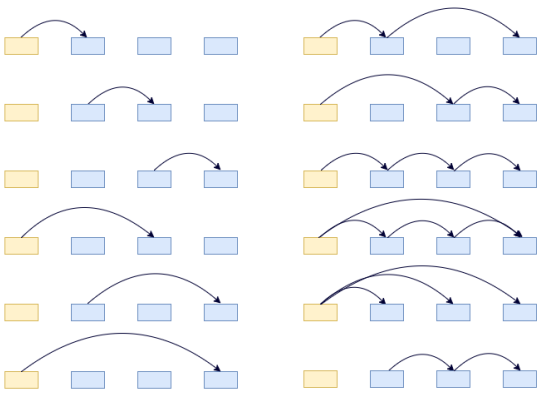

Fig. 5. Various combinations of skip-connections have been implemented on the encoder part.

significant increase in the overall performance of the model. Despite the additional connections, the improvement in loss, PSNR, and SSIM were negligible compared to the baseline architecture without skip connections. A comparison of these metrics is shown below for the tree dataset for 3 networks, namely Base-CAE, Improved-CAE and Improved-CAE with skip-connections.

Model	Loss	PSNR	SSIM
BaseCAE	0.040	24.10	0.95
ImprovedCAE-no-skip	0.025	26.26	0.96
ImprovedCAE-with-skip	0.007	31.50	0.97

TABLE I
RESULTS ON MNIST DATASET WITH MSE LOSS

Model	Loss	PSNR	SSIM
BaseCAE	0.008	20.75	0.64
ImprovedCAE-no-skip	0.004	23.92	0.81
ImprovedCAE-with-skip	0.009	30.41	0.96

TABLE II
RESULTS ON CIFAR-10 DATASET WITH MSE LOSS

Model	Loss	PSNR	SSIM
BaseCAE	0.01	20.55	0.59
ImprovedCAE-no-skip	0.005	24.20	0.80
ImprovedCAE-with-skip	0.009	31.24	0.97

TABLE III
RESULTS ON CINIC-10 DATASET WITH MSE LOSS

Subsequently, we explored the possibility of introducing skip connections across the bottleneck layer of the encoder-decoder architecture. These connections were designed to establish direct links between the encoding and decoding stages, bypassing the bottleneck layer. The intention was to enable a more seamless information flow and potentially improve the model's ability to reconstruct input data.

Surprisingly, our experiments demonstrated that skip connections across the bottleneck layer yielded remarkable results. The model incorporating these connections achieved a significant increase in performance compared to the baseline and the

Model	PSNR	SSIM
ImprovedCAE-MSE	24.20	0.8020
ImprovedCAE-L1	23.61	0.8028
ImprovedCAE-BCE	23.54	0.8120
ImprovedCAE-L1-MSE(0.1 - 1)	23.80	0.8168
ImprovedCAE-L1-MSE(0.5 - 1)	23.68	0.8123

TABLE IV
LOSS SELECTION TUNING

models with skip connections within the encoder and decoder. We saw significant improvements in loss, PSNR, and SSIM values indicating the effectiveness of these cross-bottleneck connections in enhancing the model's overall capability to reconstruct the input data.

The output images of the tree dataset for 3 networks, namely Base-CAE, Improved-CAE and Improved-CAE with skip, are given in Figures 6-14. The important parameters used were a batch size of 256, a learning rate of 0.001, Adam optimization, and a latent size of 256.

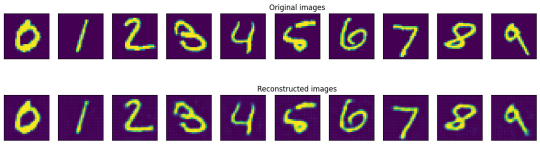

Fig. 6. Original and Reconstructed images for Base-CAE Network for MNIST Dataset.

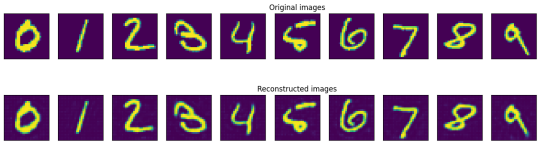

Fig. 7. Original and Reconstructed images for Improved-CAE Network for MNIST Dataset.

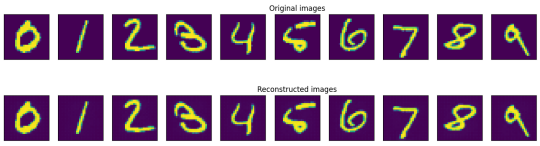

Fig. 8. Original and Reconstructed images for Improved-CAE Network with skip-connections for MNIST Dataset.

These findings underscore the importance of carefully considering the placement and nature of skip connections within an encoder-decoder architecture. While the internal skip connections did not result in notable performance gains, the skip connections across the bottleneck layer proved to be highly effective. It is worth noting that these connections could

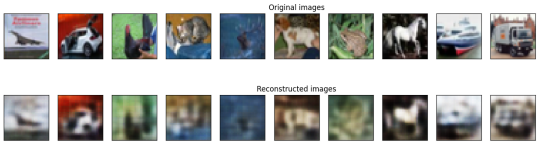

Fig. 9. Original and Reconstructed images for Base-CAE Network for CIFAR-10 Dataset.

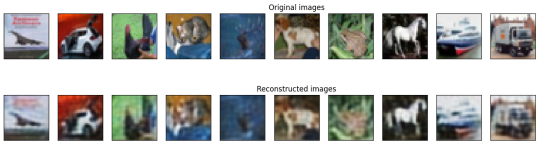

Fig. 10. Original and Reconstructed images for Improved-CAE Network for CIFAR-10 Dataset.

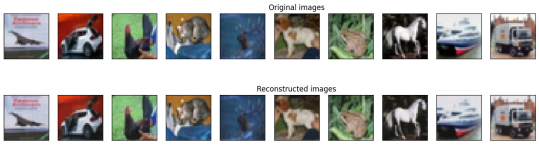

Fig. 11. Original and Reconstructed images for Improved-CAE Network with skip-connections for CIFAR-10 Dataset.

Fig. 12. Original and Reconstructed images for Base-CAE Network for CINIC-10 Dataset.

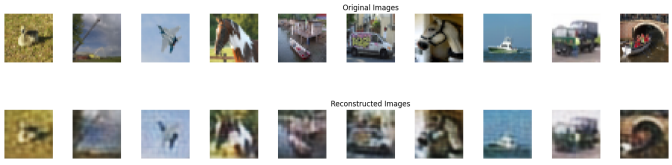

Fig. 13. Original and Reconstructed images for Improved-CAE Network for CINIC-10 Dataset.

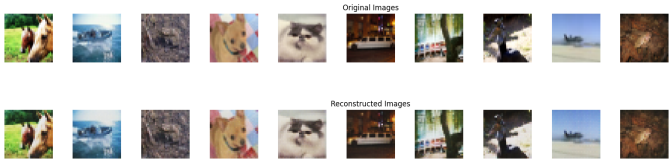

Fig. 14. Original and Reconstructed images for Improved-CAE Network with skip-connections for CINIC-10 Dataset.

be perceived as a means of circumventing the latent space, potentially giving the impression of "cheating" or bypassing certain encoding constraints.

The success of skip connections across the bottleneck layer has significant implications for various applications of encoder-decoder architectures. Although these connections may raise concerns about potentially undermining the intended encoding process, they have demonstrated the potential to significantly improve performance in tasks such as image reconstruction, semantic segmentation, and natural language processing, where encoder-decoder models are commonly employed.

In conclusion, our experiments revealed that skip connections within the encoder and decoder did not yield a significant performance increase. However, skip connections across the bottleneck layer proved to be highly effective, leading to substantial improvements in performance. Despite the concerns regarding the potential circumvention of the latent space, these findings highlight the importance of strategically placing skip connections within an encoder-decoder architecture to achieve optimal results.

ACKNOWLEDGMENT

We would like to extend our heartfelt thanks to Dr. Ethem Alpaydin for his exceptional instruction during the Introduction to Machine Learning course. His guidance and teachings have been instrumental in the successful completion of this project.

REFERENCES

[1] https://insideaiml.com/blog/LossFunctions-in-Deep-Learning-1025.
[2] https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a.
[3] Hemant Bhojwani, Vishwam Bhavsar, Ruchi Gajjar, and Manish Patel. Image resolution enhancement using convolutional autoencoders with skip connections. In *2021 2nd International Conference on Range Technology (ICORT)*, pages 1–5. IEEE, 2021.
[4] L. Darlow, E.J. Crowley, A. Antoniou, S. Ravi, R. Maior, C. Donahue, M. Camplani, H. Yousefzadeh, and A. Vedaldi. Cinic-10: An open dataset for benchmarking convolutional neural networks. https://github.com/BayesWatch/cinic-10, 2018.
[5] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
[6] imatest. SSIM. https://www.imatest.com/docs/ssim/.
[7] National Instruments. PSNR. https://www.ni.com/en-tr/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/peak-signal-to-noise-ratio-as-an-image-quality-metric.html.
[8] Alex Krizhevsky and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). Technical report, University of Toronto, 2009.