Hybrid Methods

Luisa Toro Villegas
Universidad EAFIT
Medellin, Colombia
ltorov@eafit.edu.co

Olga Lucía Quintero Montoya
Universidad EAFIT
Medellin, Colombia
oquinte1@eafit.edu.co

***Abstract*—This paper encompasses the practical exploration of hybrid supervised learning models, including Autoencoders, Convolutional Neural Networks (CNNs), and Generative Adversarial Networks (GANs), aiming to gain hands-on experience using synthetic datasets before tackling real-world challenges. The tasks involve extraction of characteristics in a dataset through Autoencoders, digit image classification using LeNet-5 CNNs on the MNIST dataset, and creating Generative and Discriminative networks for MNIST while also experimenting with style transfer GANs on recorded videos. The aim is to explore areas such as feature extraction, image classification, and generative modeling while fostering a deeper understanding of machine learning techniques and their practical applications in various domains.**

***Index Terms*—Autoencoders, Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), Style Generative Adversarial Networks**

I. Introduction

This paper embarks into the world of hybrid supervised learning models, where Autoencoders, Convolutional Neural Networks (CNNs), and Generative Adversarial Networks (GANs) take center stage. Our primary objective is to gain hands-on experience by delving into synthetic datasets before venturing into the realm of real-world challenges.

One of the fundamental tasks we undertake involves the utilization of Autoencoders to extract essential characteristics from a dataset. Autoencoders, known for their remarkable ability to capture latent features within data, serve as the cornerstone for feature extraction: a pivotal step in understanding the intrinsic nature of the information.

As we dive deeper into the world of hybrid supervised learning models, our focus shifts to Convolutional Neural Networks, specifically LeNet-5, as we undertake the challenge of digit image classification. Using the MNIST dataset, a benchmark in the field of computer vision, we aim to harness the power of CNNs to distinguish and classify handwritten digits. Then, we use a dataset built on our handwritten digits and we evaluate the model's ability to classify them. This is meant to test the efficacy of a model trained on old data when faced with new handwriting styles—a reflection of the evolving ways in which the art of penmanship has transformed over time.

Furthermore, our exploration extends to the fascinating domain of Generative Adversarial Networks (GANs). Here, we endeavor to create both Generative and Discriminative networks tailored to the MNIST dataset. Additionally, we dive into the captivating world of style transfer GANs, applying them to recorded videos to investigate their potential for altering visual aesthetics. By exploring generative modeling and style transfer, we aim to uncover the creative and transformative capacities of machine learning, highlighting its wide-ranging applicability across various domains.

II. Methodology

Hybrid supervised learning models combine the strengths of both supervised and unsupervised learning. Supervised learning models are trained on labeled data, while unsupervised learning models are trained on unlabeled data. Hybrid models leverage the advantages of both approaches to achieve better performance on a variety of tasks.

A. The models

1) Autoencoders: Autoencoders, a type of unsupervised neural network, have a remarkable capacity for feature extraction. These neural networks are designed with a specific objective: to reconstruct their input data as faithfully as possible. Yet, this seemingly straightforward task allows them to capture the most crucial features embedded within the data.

During the training process, autoencoders learn to encode input data into a condensed representation known as the "encoder" or "bottleneck." This compact space focuses on its main features while filtering out noise and irrelevant information. The true magic of autoencoders emerges when they are then tasked with decoding this representation back into the original data, fine-tuning their internal weights and biases to achieve optimal reconstruction accuracy. This process not only results in faithful reconstructions but also equips the autoencoder with the capability to extract the most essential features, making them indispensable tools in data analysis, image processing, anomaly detection, and natural language processing, among other fields.

For this exercise, we will experiment different model arquitectures, varying the number of neurons in the hidden layer. The activation functions chosen were *relu* for the hidden layer and *sigmoid* using a *keras* model.

2) Convolutional Neural Networks: Convolutional Neural Networks (CNNs) belong to the group of supervised neural networks, and they shine when it comes to sorting out images into categories. These networks specialize in the art of spotting spatial features within images, and this knack proves incredibly handy for tasks like image classification. By teasing out these spatial details, CNNs become adept at classifying images into various groups based on the patterns and structures they identify.

The beauty of CNNs lies in their ability to discern important visual elements within images, making them a go-to choice for tasks where understanding the visual content is key. Whether it's recognizing objects in photos or distinguishing between different types of animals, CNNs bring a valuable toolset to the world of image analysis and classification.

3) Generative Adversarial Networks: Generative Adversarial Networks (GANs) represent a category of unsupervised neural networks employed in the creation of novel data. GANs operate through a unique dynamic involving two separate networks: a generator and a discriminator. The generator network's primary task is to produce fresh data, while the discriminator network is trained to differentiate between genuine and generated data.

The intriguing interplay between the generator and discriminator within GANs forms the foundation of their creative capabilities. While the generator strives to craft data that closely resembles real-world examples, the discriminator continuously sharpens its ability to tell the difference between authentic and artificially generated content. This delicate balance ultimately leads to the development of remarkably authentic synthetic data, making GANs an invaluable asset across various domains, from image and text generation to data augmentation and beyond.

To evaluate the discriminator, I used the dataset of our own digits. I selected subsets of the dataset to ensure efficient processing. Then, I utilized the "evaluate" function to compute the probability that my handwritten digits resembled real ones, as determined by the GAN's discriminator. This method allowed me to quantitatively assess the quality of my custom dataset, providing valuable insights into how closely my digits resembled real-world examples and guiding any necessary refinements.

4) Style Generative Adversarial Networks: Style Generative Adversarial Networks, or StyleGANs for short, represent a remarkable leap forward in the realm of generative modeling. Unlike traditional GANs that focus on generating data, StyleGANs take things a step further by allowing users to control the artistic style and content of the generated images. This capability opens up a world of creative possibilities, enabling artists and designers to craft visually stunning and highly customizable digital artwork.

At the heart of StyleGAN's innovation lies its ability to disentangle the content and style of an image, providing a level of artistic control that was previously elusive in the realm of generative models. Users can tweak various aspects of the generated images, from the color palette and texture to the overall composition, allowing for the creation of truly unique and expressive visuals. Whether it's generating hyper-realistic portraits, surreal landscapes, or abstract art, StyleGANs empower individuals to explore the boundaries of creativity and digital artistry, redefining the possibilities of AI-assisted design and content generation.

B. The data

1) Autoencoder data: I selected the data given by the professor which is for academic purposes. This dataset is characterized by its three-dimensional structure and comprises 1251 entries. In order to ensure the dataset's compatibility with various analytical techniques and to create a standardized baseline for further investigations, we applied a normalization process. This process involved scaling the data points to fit within the confines of a hypercube spanning the range $[0, 1] \times [0, 1] \times [0, 1]$. This normalization step not only aids in reducing the influence of varying scales among the dataset's features but also facilitates comparisons and analysis by bringing all data points into a common and uniform range. Fig. 1 shows the data.

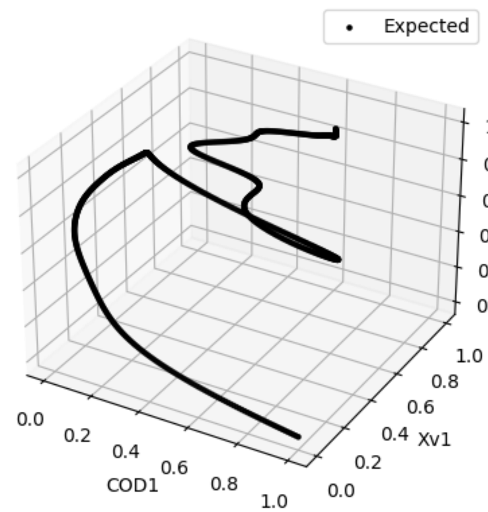


Fig. 1: Dataset for Autoencoder

2) MNIST: The MNIST dataset is a widely recognized and essential benchmark in the field of computer vision, particularly for models like LeNet-5. Comprising a collection of 28x28 grayscale images of handwritten digits (0 to 9), MNIST serves as a foundational dataset for digit image classification tasks. Its simplicity and accessibility make it a perfect starting point for evaluating and training machine learning models. MNIST's diverse set of handwritten characters, each labeled with its corresponding digit, facilitates the development and testing of algorithms, allowing researchers and practitioners to measure the performance and accuracy of models like LeNet-5 in distinguishing and classifying these handwritten symbols.

3) Our digits: Our class put together a dataset of handwritten digits that showcases the kind of handwriting often seen among young people, which can sometimes be a bit messy. These images are 28x28 greyscale, and each one is labeled with its corresponding number. This dataset can be useful for testing machine learning models in situations where handwriting isn't always perfect.

TABLE I: Experimental Design Matrix

Hidden Neurons	Training Loss	Testing Loss	Validation Loss
1	0.028	0.026	0.029
2	0.004	0.004	0.004
4	7.068e-06	7.594e-06	7.175e-06
5	0.025	0.023	0.025
6	0.002	0.003	0.003
10	2.455e-07	4.111e-07	2.094e-07

III. Results

A. Autoencoder

In the 3D plots shown in Fig. 2 and Fig. 3, we're comparing what we expected with what our model predicted across a dataset for the best case scenario of architecture: ten hidden neurons. Each curve represents this comparison, helping us see how well our model's predictions match the actual data. By visualizing this in 3D space, we get a clearer picture of where the model excels and where it might need improvement.

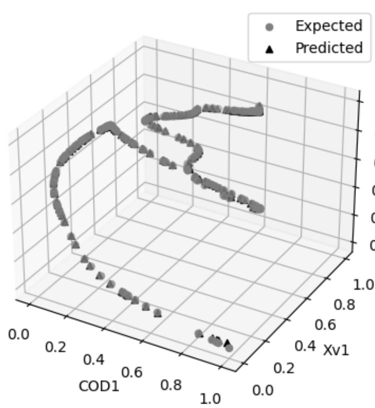


Fig. 2: Expected vs Predicted for test dataset

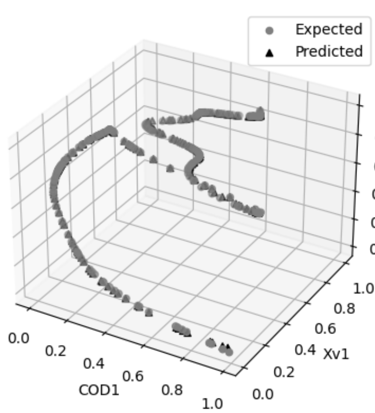
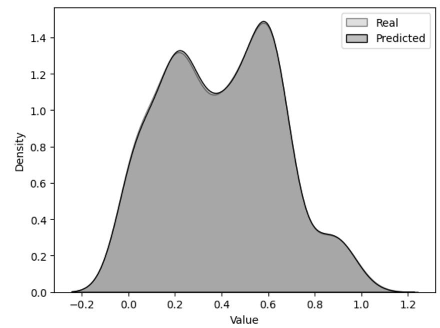
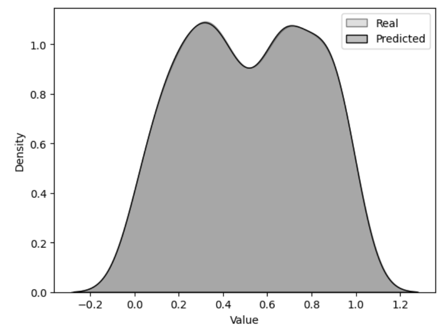


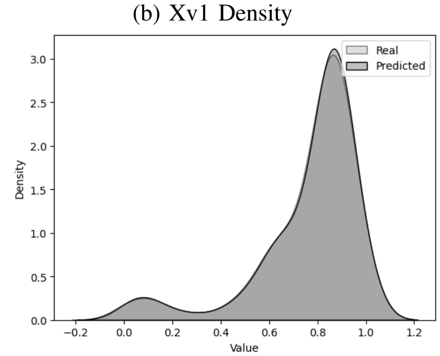
Fig. 3: Expected vs Predicted for validation dataset



(a) COD1 Density



(b) Xv1 Density



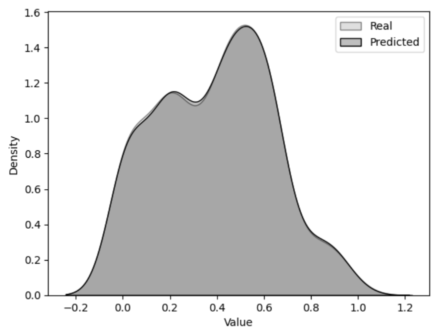
(c) t Density

Fig. 4: Expected vs Predicted densities for test dataset

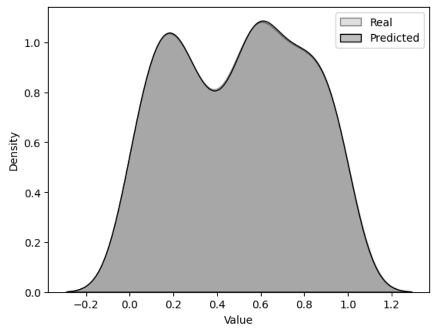
In Fig. 4 and Fig. 5, we showcase three density plots for each dataset that provide a comprehensive comparison between the expected and predicted density distributions. The former shows the densities for the test dataset, and the latter for the validation dataset. Each of these density figures offers a unique perspective on the accuracy of our model's predictions. By plotting the expected and predicted density curves side by side, we gain valuable insights into the model's ability to capture the underlying patterns and variations within the data.

B. Convolutional Neural Networks

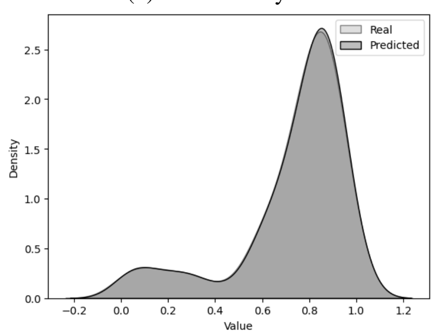
Fig. 6 and Fig. 7 display nine different handwritten numbers from the MNIST dataset, the first used for training and the second for testing. Each image shows a clear example of a single digit, giving us a quick look at the dataset's variety of writing styles. The label beneath is the predicted class given by the LeNet 5 model.



(a) COD1 Density



(b) Xv1 Density



(c) t Density

Fig. 5: Expected vs Predicted densities for validation dataset

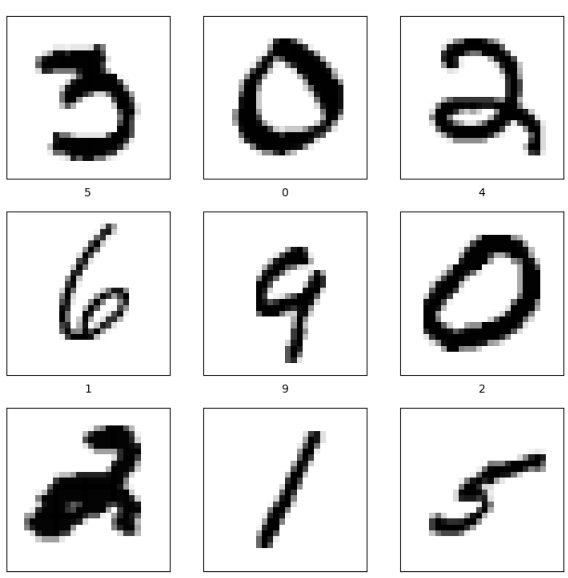


Fig. 6: Predicted train digits

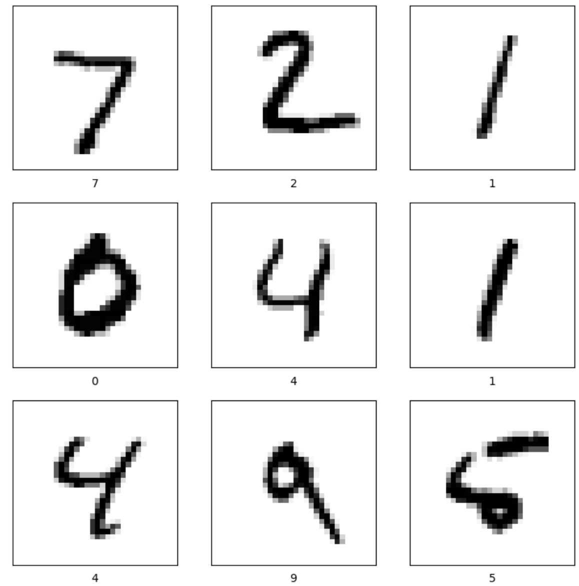


Fig. 7: Predicted test digits

Fig. 8 display nine different handwritten numbers from the dataset we created, used for validating the trained model. The label beneath is the predicted class given by the LeNet 5 model.

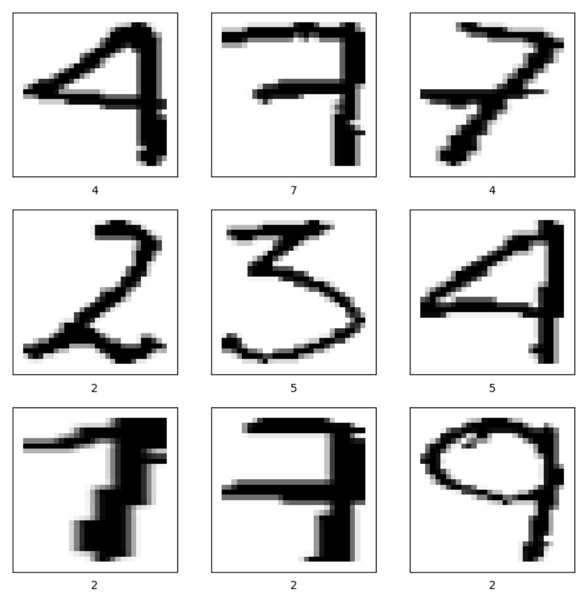


Fig. 8: Predicted validation digits

Then, we employ a performance comparison chart to assess the model's effectiveness after training. The horizontal axis (x) on the chart signifies the number of training epochs, providing a timeline of the model's progression over time. On the vertical axis (y), we can observe the model's performance, which is quantified through various metrics: accuracy for Fig. 9 and precision for Fig. 10.

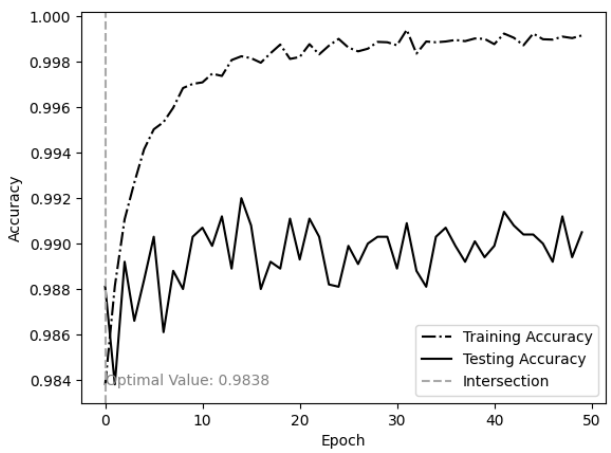


Fig. 9: Accuracy

The curve displayed on the chart represents the evolution of these metrics across epochs, offering insights into how the model's performance changes during training. Notably, for enhanced clarity, a vertical line is incorporated, intersecting the curves at the point where training and validation performance values align most optimally. This marker helps pinpoint the epoch where the model exhibits its peak performance.

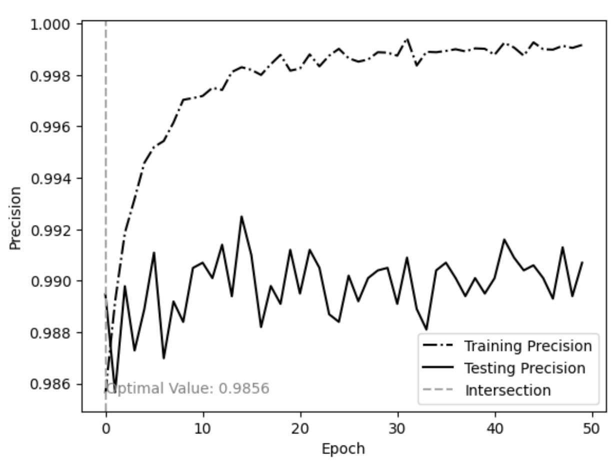


Fig. 10: Precision

In Fig. 11 and Fig. 12, we present a visual representation of a confusion matrix, a valuable tool frequently employed in the realm of machine learning for assessing classification model performance. The confusion matrix is structured as a grid, with rows and columns that correspond to the actual and predicted class labels, respectively. Each cell in the matrix contains a numerical value representing the count of instances belonging to a specific class.

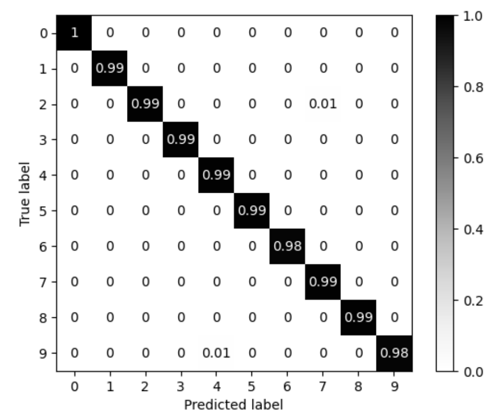


Fig. 11: Confusion Matrix for Test Dataset

The diagonal cells of the confusion matrix capture the instances that the model correctly classified, demonstrating its effectiveness in distinguishing between different classes. On the other hand, the off-diagonal cells display instances that were misclassified, highlighting areas where the model may need further improvement. This graphical depiction offers a concise and insightful overview of the classifier's performance, aiding in the evaluation and optimization of its accuracy and precision for various classification tasks.

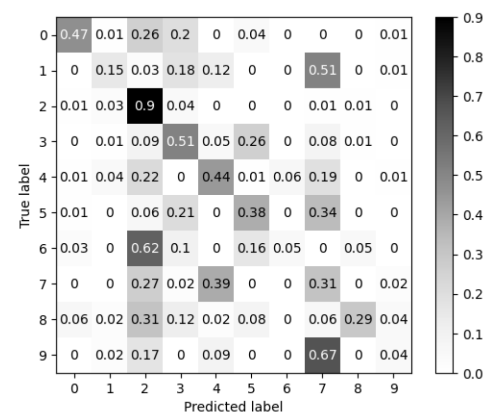


Fig. 12: Confusion Matrix for Validation Dataset

C. Generative Adversarial Networks

In Fig. 13, we showcase six digits that were generated using a Generative Adversarial Network (GAN). Each digit represents an example of the synthetic data produced by the GAN, demonstrating its ability to create realistic-looking handwritten numbers. These digits exhibit diverse styles and patterns, highlighting the versatility and creativity of the GAN in generating new and visually appealing content. The figure provides a glimpse into the potential of GANs for creating artificial data that closely resembles real-world examples, showcasing their utility in various domains, from character recognition to data augmentation.

Fig. 13: Digits Generated by GAN model

In Fig. 14, we present a visual representation of the training process for our Generative Adversarial Network (GAN). The chart illustrates the evolution of loss values over the course of the GAN's training. Specifically, it displays both the generator and discriminator losses, providing insights into how the two networks interact and learn from each other during the training iterations. Monitoring these loss curves is crucial in understanding the dynamics of GAN training, as they reflect the adversarial relationship between the generator's quest to produce realistic data and the discriminator's role in distinguishing real from generated data.

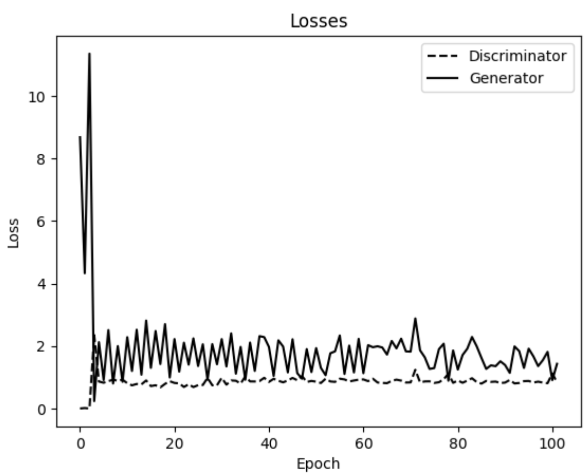


Fig. 14: Losses for each epoch of GAN training

In Fig. 15 we present the result of an evaluation process conducted on subsampled datasets of our digits. Leveraging the discriminator from our Generative Adversarial Network (GAN) model, we sought to determine the authenticity of these digits by assessing the probability that they are indeed real. Each plot showcases the changing probabilities across different subsets of the dataset, providing a visual representation of how the discriminator perceives the authenticity of the custom digits. These plots are instrumental in gauging the GAN's discriminative capabilities when applied to our unique data, shedding light on its ability to distinguish between real and synthetic samples and offering valuable insights into the model's performance and reliability for this specific task.

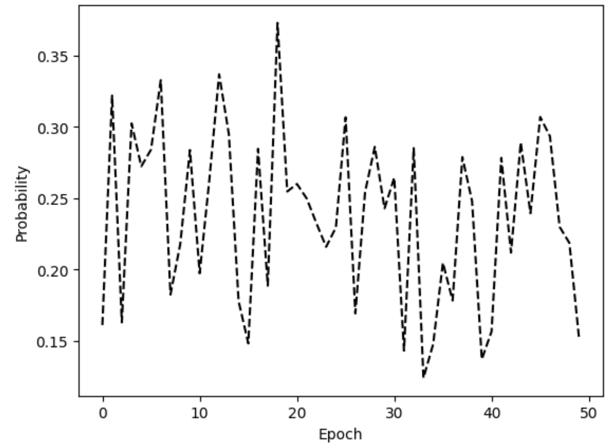


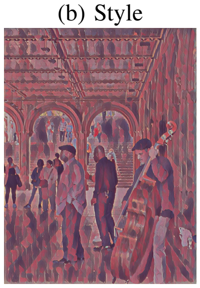
Fig. 15: Probability of being real per sampling of our numbers

D. Style Generative Adversarial Network

In Fig. 16 we present a three-image sequence that vividly encapsulates our style transfer experiment. The first image showcases the original frame extracted from the video, serving as the foundation upon which our style transformation is based. The second image represents the specific style we aimed to imbue in our new video, providing a clear visual reference for the desired aesthetic. The third image is where the StyleGAN depicts the frame styled. This transformation integrates the desired style, texture, and artistic elements from the reference image into the video frame. While the color matching may exhibit some nuances, the overall outcome demonstrates the promising potential of GAN-based style transfer techniques in achieving visually captivating and stylistically coherent video transformations.

(a) Original frame

(b) Style



(c) Styled frame

Fig. 16: Original vs generated frame

IV. Discussion and Conclusion

A. Autoencoder

When attempting to employ a compression autoencoder, characterized by an architecture with just one or two neurons in the hidden layer, the outcomes fell short of expectations, even after a lengthy training period spanning 200 epochs. These less-than-ideal results suggest that this particular architecture, coupled with the chosen activation functions, might not be well-suited for effective data compression. However, a different approach was remarkably successful when utilizing an expansion autoencoder, featuring a more extensive hidden layer with four or more neurons. This architecture demonstrated exceptional performance by accurately replicating the underlying data distribution, as vividly illustrated in Fig. 4 and Fig. 5. Additionally, Table I provides a comprehensive overview of the comparative analysis, revealing the nuances in training, testing, and validation losses across different experimental scenarios, with the most promising results observed in the case of 10 hidden neurons.

B. Convolutional Neural Networks

Our analysis revealed some notable differences between the MNIST dataset and our custom testing dataset, suggesting changes in how digits are written over time. Additionally, our dataset's zoomed-in format introduced specific challenges that may have influenced model errors by potentially altering the spatial correlations between pixels, which may contribute to model errors.

Throughout training, our LeNet5 model exhibited swift learning, achieving high accuracy and precision early in the process. However, when tested with a different dataset, the model's ability to predict was greatly affected. The confusion matrix highlighted that while the model performed well with the MNIST data it was trained on, it faced greater difficulties when confronted with our custom dataset.

In light of these observations, it's evident that modern handwritten digit recognition demands training LeNet5 models with contemporary data. The evolving nature of writing styles underscores the need for models to remain adaptable, ensuring their effectiveness in recognizing and classifying the diverse array of handwritten digits encountered today.

C. Generative Adversarial Network

Our training of a Generative Adversarial Network (GAN) on the MNIST dataset yielded visually impressive results, as illustrated in Figure 13, with the generated digits closely resembling real ones. However, a notable observation was the consistent nature of the loss function throughout the training epochs, as depicted in Figure 14. Specifically, the generator's loss exhibited a substantial decrease during the initial epochs, while the discriminator's loss demonstrated an initial increase—an encouraging sign of effective training. Nevertheless, these losses subsequently stabilized within a narrow range between 1 and 2.

While the visually appealing outcomes serve as a positive indicator of the model's capability to produce realistic data, the persistent static loss suggests potential challenges in capturing the intricacies of the underlying data distribution. Consequently, further exploration and optimization of the GAN's training dynamics may be necessary to enhance its adaptability in generating data with greater dynamism and diversity.

D. Style Generative Adversarial Network

In our experiment employing a Style Generative Adversarial Network (GAN) to transfer the style of a specific image onto a video, we achieved a noteworthy replication of the desired style in the video frames. The output exhibited a striking similarity in terms of stylistic elements, textures, and overall aesthetics when compared to the reference image. However, a notable challenge was encountered in accurately replicating the colors from the image onto the video frames. Despite the successful transfer of the style's essence, color mapping posed a significant hurdle. The colors in the video did not align precisely with those of the image, resulting in variations that affected the overall color composition and vibrancy. This points to the need for further refinement and fine-tuning in the color mapping process within the GAN model to achieve a more faithful reproduction of colors, thus enhancing the overall fidelity of the style transfer process.