12-1-2023

A New Approach to Synthetic Image Evaluation

Majid Memari
Southern Illinois University Carbondale, memari.majid@hotmail.com

A NEW APPROACH TO SYNTHETIC IMAGE EVALUATION

by
Majid Memari

B.S., Azad University, 2010
M.B.A., Azad University, 2015
M.S., Southern Illinois University Carbondale, 2018

A Dissertation
Submitted in Partial Fulfillment of the Requirements for the
Doctor of Philosophy Degree

School of Computing
in the Graduate School
Southern Illinois University Carbondale
December 2023

DISSERTATION APPROVAL

A NEW APPROACH TO SYNTHETIC IMAGE EVALUATION

by

Majid Memari

A Dissertation Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in the field of Computer Science

Approved by:

Khaled R. Ahmed, Chair

Shahram Rahimi, Co-Chair

Dunren Che

S. Yaser Samadi

Henry Hexmoor

Graduate School
Southern Illinois University Carbondale
August 10, 2023

AN ABSTRACT OF THE DISSERTATION OF

Majid Memari, for the Doctor of Philosophy degree in Computer Science, presented on June 19, 2023, at Southern Illinois University Carbondale.

TITLE: A NEW APPROACH TO SYNTHETIC IMAGE EVALUATION

MAJOR PROFESSOR: Dr. Khaled R. Ahmed

This study is dedicated to enhancing the effectiveness of Optical Character Recognition (OCR) systems, with a special emphasis on Arabic handwritten digit recognition. The choice to focus on Arabic handwritten digits is twofold: first, there has been relatively less research conducted in this area compared to its English counterparts; second, the recognition of Arabic handwritten digits presents more challenges due to the inherent similarities between different Arabic digits. OCR systems, engineered to decipher both printed and handwritten text, often face difficulties in accurately identifying low-quality or distorted handwritten text. The quality of the input image and the complexity of the text significantly influence their performance.

However, data augmentation strategies can notably improve these systems' performance. These strategies generate new images that closely resemble the original ones, albeit with minor variations, thereby enriching the model's learning and enhancing its adaptability. The research found Conditional Variational Autoencoders (C-VAE) and Conditional Generative Adversarial Networks (C-GAN) to be particularly effective in this context. These two generative models stand out due to their superior image generation and feature extraction capabilities.

A significant contribution of the study has been the formulation of the Synthetic Image Evaluation Procedure (Algorithm 3), a systematic approach designed to evaluate and amplify the generative models' image generation abilities. This procedure facilitates the extraction of

meaningful features, computation of the Fréchet Inception Distance (LFID) score and supports hyper-parameter optimization and model modifications.

ACKNOWLEDGEMENTS

I am profoundly grateful to my mentor, Dr. Ahmed, whose unwavering guidance has been pivotal in propelling me through this significant phase of my academic pursuit. His unyielding support, resilience, and vast knowledge have played an indispensable role in helping me achieve my ambitions. I am equally indebted to Dr. Rahimi, who has worked tirelessly alongside Dr. Ahmed, providing invaluable support throughout my research journey. I would like to extend my sincere appreciation to Dr. Carver, Dr. Che, Dr. Samadi, and Dr. Hexmoor for their generous dedication of time and presence at my defense. Their encouragement and participation have made a profound impact on this accomplishment.

DEDICATION

To my older brother, Hamid, for your unwavering support, guidance, and belief in me, I dedicate this dissertation to you. Your help has been invaluable. With gratitude, Majid

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Image processing involves the digital alteration of an image once it has been digitized and transferred to a computer. This manipulation is carried out by software programs to enhance the image's usefulness, such as enabling image recognition. Image processing is a comprehensive field that includes topics like image/video processing, analysis, communications, sensing, modeling, computational imaging, electronic imaging, information forensics and security, 3D imaging, medical imaging, and machine learning applications to these areas. For several decades, image processing has been a hotbed of research and development, leading to numerous successful applications across various sectors like entertainment, digital photography, video conferencing, video monitoring and surveillance, satellite imaging, healthcare, distance learning, digital archiving, cultural heritage, and the automotive industry [22, 27, 64, 65, 98].

Optical Character Recognition (OCR) is a critical technology in numerous industries, as it enables the identification and conversion of printed or handwritten text into machine-readable formats. Arabic Handwritten Digit Recognition is a challenging task in the field of Optical Character Recognition (OCR). OCR is a technology used to convert different types of documents, such as scanned paper documents, PDF files, or images captured by a digital camera, into editable and searchable data. While OCR technology has made significant strides in recent years, the accuracy of Arabic Handwritten Digit Recognition remains a problem. Several factors contribute to this issue. Arabic numerals are cursive and context-sensitive, meaning their shape and form can change depending on their position in a word. This characteristic makes them more complex and challenging to recognize than Latin numerals. Additionally, variations in individual handwriting styles, writing tools, and paper quality can further complicate the recognition process [5, 49]. Improving the accuracy of Arabic Handwritten Digit Recognition is an active area of research. Techniques such as deep learning, convolutional neural networks, and other machine learning algorithms are being explored to enhance recognition accuracy [61]. However, despite these efforts, achieving high accuracy remains a significant challenge [13, 31]. Optical Character Recognition (OCR) is a rapidly evolving technology with significant implications across various industries. However, the accurate recognition of distorted, low-quality, or noisy text remains a challenge, especially for handwritten text. Data augmentation and real-time monitoring are vital components in addressing these challenges and improving OCR performance [16, 32, 90, 94, 104, 112].

This research aims to explore the use of data augmentation techniques and real-time monitoring for enhancing the capabilities of OCR systems. Despite considerable advancements in machine learning and computer vision techniques, OCR systems continue to struggle with accurately interpreting distorted, low-quality, or noisy text, particularly in the case of handwritten text, which exhibits significant variability [16, 23, 112].

Data augmentation is essential for enhancing OCR performance due to a variety of reasons. By artificially expanding and diversifying the training dataset through the introduction of synthetic images with various variations, transformations, and noise levels, data augmentation techniques help train more robust and accurate OCR systems capable of handling diverse input data. One crucial aspect of data augmentation is supplementing the limited training data with synthetic data generated by generative models. Real-time monitoring ensures an optimal balance between the quality and quantity of generated images, allowing the OCR model to effectively learn from the augmented data without being negatively affected by poor-quality samples [37, 95, 117]. Generative model training can be unstable, with the performance of the generator and discriminator potentially fluctuating throughout the training process. Real-time monitoring is essential for tracking these changes and making necessary adjustments to maintain stability and avoid problems such as mode collapse or vanishing gradients. By continuously keeping track of the training process, real-time monitoring can provide valuable insights into the model's performance and identify the appropriate intervention points [37, 76, 94].

Generated data can sometimes result in overfitting if the model begins to memorize the synthetic images rather than learning the underlying patterns. Real-time monitoring helps detect such issues early on and implement corrective action, such as reducing the amount of generated data or adjusting the training parameters. By ensuring that the model focuses on learning the essential features and patterns, real-time monitoring contributes to the development of a more robust and generalizable OCR system [74, 116].

Monitoring generative model performance in real-time also plays a significant role in improving efficiency by identifying when the model has achieved an acceptable level of quality and can cease the training process, thus saving time and computational resources. This optimization is particularly important when working with large-scale OCR systems or when training multiple models simultaneously, as it can lead to substantial time and cost savings [84, 124].

Evaluating the impact of generated images on OCR performance during the training process is another important aspect of real-time monitoring. This assessment helps determine the effectiveness of the augmentation strategy and whether adjustments are needed to better address specific OCR challenges, such as recognizing distorted or noisy text. By continuously measuring the impact of synthetic data on OCR performance, researchers can fine-tune their models and improve overall accuracy [18, 121].

Lastly, real-time monitoring facilitates the comparison of different generative models and hyperparameter settings based on their influence on OCR performance. This comparison allows for the selection of the most promising configurations for further refinement and optimization. Through the continuous evaluation of model performance and the identification of the most effective settings, real-time monitoring can significantly contribute to the development of more accurate and efficient OCR systems [26, 50].

The study aims to improve the performance of Optical Character Recognition (OCR) through the use of data augmentation techniques, specifically by incorporating synthetic data generated by generative models. It introduces real-time monitoring to achieve an optimal balance between the quality and quantity of generated images. Addressing instability issues encountered during generative model training, such as mode collapse or vanishing gradients, the research enhances the overall stability of the model. Measures are implemented to prevent overfitting, ensuring that the model doesn't memorize synthetic images and instead improves its generalization capability. Efficiency is enhanced by determining the appropriate stopping point for the training process based on the model's performance. The study continuously evaluates the impact of generated images on OCR performance and adjusts augmentation strategies, accordingly, ensuring the model's adaptability. A comparative analysis of different generative models and hyperparameter settings is provided, which contributes to further optimizing the OCR system.

In the following chapter, we delve into a detailed discussion on the existing works related to Arabic Handwritten Digit Recognition. We explore the methodologies and techniques employed in these studies, their contributions to the field, and the results they have achieved. Importantly, we also critically examine the limitations and shortcomings of these works, identifying areas where they fall short or fail to address specific challenges in Arabic Handwritten Digit Recognition. This analysis will provide a comprehensive understanding of the current state of the field and pave the way for our research to address these gaps and contribute to the advancement of Arabic Handwritten Digit Recognition models.

CHAPTER 2

BACKGROUND

2.1 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is a technology that aims to recognize text within digital images, such as scanned documents, photographs, or screenshots. It involves a process of converting images of text into machine-encoded text that can be easily edited, searched, or processed by computers. OCR systems have been widely used in various applications, such as digitization of historical documents, automatic license plate recognition, and text recognition in mobile devices [13, 91, 97].

The OCR process involves several steps, including image pre-processing, feature extraction, and text recognition. Image pre-processing involves enhancing the quality of the image by removing noise, correcting orientation, and resizing the image. Feature extraction is the process of identifying salient features of the image, such as edges, corners, and curves, that are relevant to the recognition task. Text recognition is the process of converting the image features into machine-encoded text using various algorithms, such as neural networks, decision trees, or rule-based systems [62].

OCR systems have been challenged by various factors that can affect the accuracy and robustness of the recognition process. One of the main challenges is the quality of the input image, which can be degraded by various factors such as lighting conditions, paper quality, and image distortion.

Another challenge is the complexity and variability of the text to be recognized, which can include different fonts, languages, and writing styles. OCR systems also face challenges in handling noise, skew, and other forms of image deformation, which can affect the recognition accuracy [77].

To address these challenges, researchers have explored various techniques and algorithms to improve the performance of OCR systems. These include feature extraction methods such as histogram of oriented gradients (HOG) [43], convolutional neural networks (CNN) [33], and deep learning-based approaches [81]. Other techniques involve using machine learning algorithms for text recognition, such as hidden Markov models (HMM) [47], support vector machines (SVM) [99], and Recurrent Neural Networks (RNN) [110]. Recently, researchers have also explored the use of image generation techniques, such as Variational Autoencoders (VAE) [73] and Generative Adversarial Networks (GAN) [55], to improve OCR system performance by generating synthetic images that are more robust to noise and distortion [95].

Overall, OCR systems play a critical role in various applications that require text recognition from images. However, they face challenges in handling image degradation, variability, and complexity. To improve OCR system performance, researchers have explored various techniques, including image pre-processing, feature extraction, text recognition algorithms, and image generation techniques.

2.1.1 Deep Learning in OCR systems

OCR systems have become an essential tool for digitizing text from scanned documents, images, and videos. However, traditional OCR systems often struggle with recognizing characters and words accurately in low-quality images, degraded texts, and noisy backgrounds. Deep learning techniques have shown significant promise in improving the performance and robustness of OCR systems. Deep learning is a subfield of machine learning that involves training artificial neural networks with multiple layers of neurons to extract and represent complex features from input data. Deep learning techniques have been successfully applied to a wide range of OCR tasks, including text detection, segmentation, recognition, and post-processing [133].

One of the main advantages of deep learning for OCR systems is its ability to automatically learn and extract high-level features from raw images, such as edges, textures, and patterns, without the need for hand-crafted features [126]. Deep learning models can also adapt and generalize to new and diverse OCR tasks and domains, making them highly versatile and scalable. Moreover, deep learning models can leverage large amounts of training data to improve their performance and accuracy, even in challenging and complex OCR scenarios.

Some of the popular deep learning models used in OCR systems include CNNs, RNNs, and their variants, such as Long Short-Term Memory (LSTM) networks and Attention mechanisms. These models have shown remarkable progress in achieving state-of-the-art results in OCR benchmarks, such as the MNIST [39], CIFAR-10 [78], and ICDAR datasets [68]. However, these models are typically designed to operate on clean and well-defined input data, and may not perform well in real-world OCR scenarios, where the input data may be noisy, blurred, or distorted.

Arabic Handwritten Digit Recognition is a challenging OCR task due to the variations in writing style, size, shape, and slant of different writers, as well as the noise and distortion in the images [6]. Arabic Handwritten Digit Recognition has many applications in fields such as office automation, check verification, postal address reading, and data entry. There are many methods that have been proposed for Arabic Handwritten Digit Recognition, using different features and classifiers. Therefore, researchers have explored the use of deep generative models, such as Variational Autoencoders (VAEs) [54] and Generative Adversarial Networks (GANs) [41], to generate synthetic images that can improve the robustness and adaptability of OCR systems. These models can generate realistic and diverse images that can simulate different OCR conditions, such as different fonts, sizes, resolutions, and backgrounds, allowing the OCR system to learn from a more comprehensive and diverse training set. In this literature review, we focus on comparing two conditional variants of VAE and GAN, namely Conditional VAE (C-VAE) [113] and Conditional GAN (C-GAN) [89], for their ability to generate high-quality synthetic images that can improve OCR system performance.

Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) are two popular generative models in deep learning. A VAE consists of two main components: an encoder that maps an input image to a latent space representation, and a decoder that maps the latent space representation back to an output image. During training, the VAE minimizes the reconstruction error between the input and output images, while regularizing the latent space distribution to follow a Gaussian distribution. This regularization encourages the VAE to generate diverse samples and enables it to perform various image generation tasks.

On the other hand, a GAN consists of a generator and a discriminator network that play a two-player minimax game. The generator takes a random noise vector as input and generates a fake image, while the discriminator takes a real or fake image and outputs a binary classification result. During training, the generator learns to generate images that can fool the discriminator, while the discriminator learns to distinguish between real and fake images. The GAN framework has been shown to generate high-quality images in a variety of domains [2].

Conditional Variational Autoencoders (C-VAE) and Conditional Generative Adversarial Net- works (C-GAN) are extensions of VAEs and GANs that take additional input information, such as class labels or text descriptions, to generate images with specific attributes or styles. In

C-VAE, the encoder and decoder networks are conditioned on the input information, which is fed into the latent space as an additional input. This enables the C-VAE to generate images with specific attributes, such as digit classification.

Similarly, C-GANs condition the generator and discriminator networks on the input information. The generator takes both the random noise and the input information as input, while the discriminator takes both the real or fake image and the input information as input. By conditioning the GAN on input information, C-GANs can generate images with specific attributes or styles, such as facial expressions or scene types [122].

Overall, VAEs and GANs, as well as their conditional variants, are powerful generative models that can generate diverse and high-quality images in various domains. By conditioning these models on input information, researchers can generate images with specific attributes or styles, making them valuable tools for image generation tasks, such as improving OCR systems. Present previous work in using VAEs, GANs, and their conditional variants for image generation and OCR system improvement [20].

Conditional Variational Autoencoders (C-VAE) are a type of generative model that can learn to generate new data samples with specific attributes. C-VAE consists of two main parts: an encoder that maps input data to a latent space, and a decoder that maps the latent space to the output space. In C-VAE, both the encoder and decoder are conditioned on additional information, such as class labels or attributes, to generate more specific and meaningful samples. This conditioning is achieved by incorporating the conditional information as additional inputs to the encoder and decoder networks [82].

During training, C-VAE minimizes the reconstruction loss between the input and the reconstructed output, as well as the Kullback-Leibler (KL) divergence between the learned latent distribution and a prior distribution. The KL divergence encourages the learned latent distribution to be close to a prior distribution, such as a standard normal distribution, which regularizes the latent space and enables efficient sampling. The overall objective of C-VAE is to minimize the sum of these two losses [12].

One of the key features of C-VAE is its ability to perform latent space interpolation. This means that by traversing the learned latent space, C-VAE can generate smooth and meaningful transitions between different attributes or classes. For example, by interpolating between the latent vectors of "cat" and "dog", C-VAE can generate images that smoothly transform from one animal to the other [88].

C-VAE has been shown to be effective in various image generation tasks, such as generating realistic images of faces, digits, and objects with specific attributes. In addition, C-VAE has also been applied to other domains, such as natural language processing, speech synthesis, and recommendation systems [34].

However, C-VAE has some limitations, such as difficulty in generating high-quality and diverse samples, and sensitivity to the choice of hyperparameters. In summary, C-VAE is a powerful generative model that can learn to generate specific and meaningful data samples by incorporating conditional information into the training process. Its architecture consists of an encoder and decoder that are conditioned on additional information, and it is trained by minimizing the reconstruction loss and KL divergence [36, 82].

C-VAE’s key features include latent space interpolation, which enables smooth transitions between attributes or classes, and its applicability to various domains. However, it also has limitations, such as difficulty in generating high-quality and diverse samples and sensitivity to hyperparameters [29].

2.2 Conditional Variational Autoencoders (C-VAE)

Conditional Variational Autoencoders (C-VAE) have gained popularity in recent years as a powerful tool for generating high-quality images with controllable attributes. One of the advantages of using C-VAE for image generation is its ability to capture and disentangle complex image features into a latent space that can be easily manipulated to generate new images with desired attributes. This is particularly useful in OCR system improvement as it allows for the generation of synthetic images that are similar to the real ones, but with enhanced attributes such as contrast, rotation, and skew [69].

Another advantage of C-VAE is its ability to handle incomplete or noisy data. This is particularly useful in OCR systems as images can be degraded due to various factors such as lighting conditions, image resolution, and perspective distortion. C-VAE can learn to reconstruct these images by filling in missing pixels or removing noise while preserving important features such as text and layout structure [44].

However, one of the main disadvantages of C-VAE is its tendency to produce blurry images, especially when dealing with complex or high-resolution images. This is because the reconstruction loss function used in C-VAE does not take into account the high-frequency details of the image, leading to blurriness in the generated images [25].

Another limitation of C-VAE is its dependence on the quality of the training data. C-VAE requires a large amount of high-quality training data to learn the underlying distribution of the data and generate high-quality images. In the absence of such data, the generated images may suffer from artifacts and low-quality features [45].

Lastly, the training process for C-VAE can be computationally expensive, especially when dealing with high-resolution images or large datasets. The training process requires multiple forward and backward passes through the network, which can be time-consuming and resource- intensive. This limits its scalability and practicality for large-scale OCR systems [128].

In summary, while C-VAE offers several advantages for image generation and OCR system improvement, such as its ability to capture complex features and handle incomplete data, it also has some limitations, such as its tendency to produce blurry images and its dependence on high-quality training data. Researchers and practitioners need to carefully consider these factors when deciding whether to use C-VAE for their OCR system improvement projects [25].

2.2.1 C-VAE applications in OCR systems

Conditional Variational Autoencoders (C-VAE) have shown promising results in generating realistic and diverse images for various applications, including OCR systems. C-VAE allows for the generation of conditional samples by encoding the input image and conditioning the decoding process on the desired output. C-VAE outperforms traditional VAEs in generating high-quality, diverse images that are suitable for OCR tasks. The authors also noted that C-VAE provides better reconstruction accuracy than traditional VAEs, which is important for OCR applications [38].

Despite its advantages, C-VAE also has some limitations in OCR systems. The performance of C-VAE is highly dependent on the quality of the input image. The authors noted that if the input image is noisy or has low resolution, C-VAE may fail to generate accurate OCR results. Additionally, C-VAE requires a large amount of labeled data for training, which can be a challenge for OCR systems where the availability of labeled data is often limited [46].

However, C-VAE has demonstrated its effectiveness in enhancing OCR performance within certain domains. By employing C-VAE for handwritten digit recognition, notable advancements in OCR capabilities were observed in comparison to conventional techniques.

Moreover, the authors emphasized that C-VAE exhibits greater resilience to variations in handwriting styles and noise, further solidifying its superiority over traditional methods. [3].

C-VAE exhibits a noteworthy advantage in producing superior images while maintaining control over their variability. It proves to be a valuable tool for generating images with diverse styles, fonts, and backgrounds, offering significant support to OCR systems tasked with recognizing text in various settings. Moreover, the authors highlighted its capacity to generate images with adjustable levels of distortion, a feature that proves beneficial in augmenting training data and enhancing OCR performance [96].

In summary, C-VAE has shown great potential in improving OCR performance by generating realistic and diverse images. While it has some limitations, such as its dependence on input image quality and the requirement for labeled data, it has also demonstrated strengths such as its effectiveness in specific domains, controllability of generated images, and robustness to variations. These findings suggest that C-VAE is a promising tool for improving OCR systems and can be further developed and optimized for specific OCR tasks and domains.

2.2.2 C-VAE challenges in OCR systems

C-VAE has shown promising results in improving OCR systems by generating high-quality synthetic images that can augment limited training data. However, there are still several challenges and limitations associated with using C-VAE in this context. One of the main challenges is the limited scalability of C-VAE models. As the size and complexity of the training data increase, C-VAE models may struggle to capture the underlying patterns and generate meaningful synthetic images [63].

Another challenge is the difficulty of selecting appropriate hyperparameters for C-VAE models. Tuning hyperparameters such as the number of latent dimensions and the learning rate can significantly affect the performance of C-VAE models. However, there is no universally optimal set of hyperparameters for C-VAE, and determining the best values often requires extensive trial and error [103].

A related challenge is the sensitivity of C-VAE to the quality and quantity of the input data. If the training data is noisy, incomplete, or biased, C-VAE models may learn spurious or irrelevant features that negatively affect their performance. Moreover, C-VAE models may struggle to generalize to new data that is significantly different from the training data [10].

In addition to these challenges, there are several future directions for C-VAE research in OCR systems. One direction is to investigate the potential benefits of using different variations of C-VAE, such as Wasserstein C-VAE [125] or Variational Hierarchical C-VAE [129], for image generation and OCR system improvement [35, 137]. Another direction is to explore the use of C- VAE in conjunction with other deep learning techniques, such as transfer learning or reinforcement learning, to enhance its performance and scalability.

Finally, there is a need for more research on the interpretability and explain ability of C-VAE models in OCR systems. C-VAE models can generate high-quality synthetic images, but it is often unclear how they achieve this and what underlying patterns they capture. Developing methods to interpret and visualize the learned representations of C-VAE models can improve our understanding of their behavior and facilitate their adoption in real-world applications.

2.3 Conditional Generative Adversarial Networks (C-GAN)

There are several advantages and disadvantages to using Conditional Generative Adversarial Networks (C-GAN) for image generation and Optical Character Recognition (OCR) systems [52]. One of the main advantages of C-GAN is its ability to generate high-quality images that match specific semantic conditions [115]. This makes it a powerful tool for tasks like image synthesis, data augmentation, and OCR system improvement. C-GAN has been shown to outperform other generative models, such as Variational Autoencoders (VAEs), in terms of image quality and semantic accuracy [131].

Another advantage of C-GAN is its ability to learn from unstructured data. Unlike some other deep learning models, C-GAN does not require labeled data or explicit annotations in order to learn. Instead, it can learn directly from raw image data, making it a powerful tool for tasks like image synthesis and OCR system improvement. This can save time and resources in data preparation and labeling [28].

However, one disadvantage of C-GAN is its high computational cost and training time. C- GAN requires large amounts of data and computing power to train effectively. This can limit its applicability in certain contexts where computational resources are limited. Additionally, C-GAN can be sensitive to hyperparameters, such as the learning rate and batch size, which can impact its performance and stability [123].

Another potential disadvantage of C-GAN is its susceptibility to mode collapse [80]. Mode collapse occurs when the generator produces a limited range of output images, which can limit the diversity and quality of the generated images. While this issue can be addressed through various techniques, such as minibatch discrimination and feature matching, it remains a challenge for C-GAN and other generative models [53, 83].

In conclusion, C-GAN has several advantages and disadvantages for image generation and OCR systems. Its ability to generate high-quality images that match specific conditions and learn from unstructured data make it a powerful tool for these tasks. However, its high computational cost and sensitivity to hyperparameters and mode collapse are potential limitations that must be addressed. Despite these challenges, C-GAN remains a promising avenue for improving OCR systems and generating high-quality images.

2.3.1 C-GAN applications in OCR systems

Conditional Generative Adversarial Networks (C-GAN) have demonstrated their potential in OCR systems, yielding promising outcomes. By leveraging C-GAN, the quality and diversity of synthetic images utilized for OCR training have been enhanced, consequently improving performance on real-world data. Notably, researchers successfully employed C-GAN to generate synthetic handwriting images for OCR system training, and the resulting system outperformed conventional OCR systems. These findings underscore the capability of C-GAN to significantly enhance OCR system performance [56, 115].

A prominent advantage of C-GAN in OCR systems lies in its capacity to generate images tailored to specific semantic conditions. For instance, C-GAN proves highly effective in producing images of handwritten digits or letters with precise styles, thickness, and orientation. This versatility enables OCR systems to be trained on a diverse range of image data, thereby enhancing generalization and robustness. Notably, GAN was employed to synthesize Chinese characters with distinct stroke styles and thickness, resulting in notable improvements in OCR system performance [30, 118].

Nonetheless, a potential drawback of employing C-GAN within OCR systems is its vulnerability to mode collapse, wherein the diversity and quality of generated images may be compromised, consequently affecting OCR system performance. To mitigate this concern, various approaches have been suggested, including gradient penalty regularization and feature matching. By implementing C-GAN with gradient penalty regularization, synthetic images of handwritten digits were successfully generated for OCR system training, yielding notable enhancements in performance compared to conventional OCR systems [60, 106].

Additionally, C-GAN's drawback in OCR systems lies in its considerable computational cost and extensive training time. Effectively training C-GAN necessitates substantial amounts of data and computational power, potentially restricting its suitability in certain scenarios. To tackle this challenge, several strategies have been put forth, including the utilization of pre-trained networks and transfer learning. By employing C-GAN with transfer learning, synthetic images of Chinese characters were successfully generated for OCR system training, resulting in improved performance and reduced training time [24].

In conclusion, C-GAN has shown promising results for improving OCR system performance by generating high-quality and diverse synthetic images. Its ability to generate images with specific semantic conditions is a key strength, but its susceptibility to mode collapse and high computational cost are potential limitations. However, several techniques have been proposed to address these issues, such as gradient penalty regularization and transfer learning. Future research can further explore the potential of C-GAN and other generative models for OCR system improvement.

2.3.2 C-GAN challenges in OCR systems

While Conditional Generative Adversarial Networks (C-GAN) show great promise for im- proving Optical Character Recognition (OCR) systems, there are still several challenges that must be addressed before their full potential can be realized. One challenge is the lack of large-scale datasets for training and evaluating C-GANs for OCR systems. The lack of high-quality datasets with diverse and challenging images can limit the ability of C-GANs to learn and generalize to new tasks. Developing new datasets and evaluation protocols that are tailored to OCR tasks can help address this challenge [115].

Another challenge is the selection of appropriate semantic conditions for generating images that improve OCR systems. Selecting the right semantic conditions, such as font style, size, and orientation, can have a significant impact on the performance of OCR systems. However, finding the right set of conditions can be a difficult and time-consuming process. Developing automated methods for selecting semantic conditions and evaluating their impact on OCR performance can help address this challenge [127].

A related challenge is the generalization of C-GAN models to new OCR tasks and domains. C-GANs trained on a specific OCR task or domain may not perform well on new tasks or domains. Developing methods for transferring and adapting C-GAN models across different OCR tasks and domains can help improve their generalization capabilities [102].

Another challenge is the integration of C-GANs into existing OCR systems. Integrating C- GANs into existing OCR pipelines can be a complex and challenging process that requires careful consideration of factors such as speed, accuracy, and scalability. Developing efficient and effective integration methods that balance these factors can help address this challenge [70].

In conclusion, while C-GANs show great promise for improving OCR systems, several challenges must be addressed before their full potential can be realized. These challenges include the lack of large-scale datasets, the selection of appropriate semantic conditions, the generalization to new tasks and domains, and the integration into existing OCR systems. Addressing these challenges will require the development of new methods and techniques that can improve the performance, scalability, and generalization capabilities of C-GANs for OCR tasks.

Table 1 summarizes the state of the art of image generation techniques in improving OCR performance. These methods offer various advantages, such as handling noise and distortion or generating high-quality, diverse images. However, they also have limitations, such as training instability, blurry image generation, or resource requirements.

Model	Description	Advantages	Limitations
Variational Autoencoders (VAEs)	A generative model that learns a latent representation of the input data, generating new samples.	Effective at modeling complex data distributions Capable of handling noise and distortion	Can produce blurry images. Less control over generated image features
Generative Adversarial Networks (GANs)	A framework where a generator and discriminator compete to improve the generated image quality.	Generate high-quality, sharp images. Effective at generating diverse images	Training can be unstable. Can suffer from mode collapse (limited variety in samples)
Cycle-Consistent Adversarial Networks (CycleGANs)	An unsupervised image-to-image translation technique that leverages cycle consistency.	Enables unsupervised learning for image translation. Can generate visually consistent images	May struggle with complex transformations. Limited to paired domain mappings
StyleGAN (and its variants)	A GAN architecture that allows control over the generated image's style, content, and stochastic features.	Enables fine-grained control over image generation. Generates high-quality images	\Requires large datasets and computational resources. Potential overemphasis on style
BigGAN	A GAN architecture that scales up in capacity and size, resulting in high-quality images.	Generates high-resolution, high-quality images. Capable of generating diverse images	Requires large datasets and computational resources. Can suffer from mode collapse

Table 1: Generative Models

2.4 Comparative Analysis of C-VAE and C-GAN

Conditional Variational Autoencoders (C-VAE) and Conditional Generative Adversarial Net- works (C-GAN) are two popular deep learning architectures used for image generation and im- proving Optical Character Recognition (OCR) systems. Here is a comparison of the two methods based on several aspects: C-GAN is generally considered to produce higher-quality images than C-VAE due to its ability to generate sharp and detailed images. C-VAE tends to produce images that are smoother and more blurred [14].

C-VAE is generally faster and more computationally efficient than C-GAN due to its simpler architecture and training process. C-VAE also requires fewer samples to achieve good results than C-GAN. However, C-GAN can be trained on larger and more complex datasets due to its ability to learn from unstructured data [48] .

C-VAE is generally more stable and easier to train than C-GAN due to its deterministic nature and use of an explicit loss function. C-GAN is more prone to mode collapse and other stability issues, which can make it more difficult to train. However, recent advancements in C-GAN training, such as Wasserstein GANs, have improved stability and convergence [87].

Both C-VAE and C-GAN are sensitive to hyperparameters, such as learning rate, batch size, and regularization. However, C-GAN is generally more sensitive to hyperparameters due to its more complex architecture and training process. Finding the optimal set of hyperparameters can be a time-consuming and challenging process for both [132].

Both C-VAE and C-GAN can be suitable for different OCR tasks and domains depending on the specific requirements and constraints. C-VAE is generally more suitable for tasks that require smooth and blurred images, while C-GAN is more suitable for tasks that require sharp and detailed images. Additionally, C-GAN can be more suitable for tasks that require learning from unstructured data, while C-VAE can be more suitable for tasks that require faster and more efficient training [105].

Other factors to consider when comparing C-VAE and C-GAN include their interpretability, flexibility, and scalability. C-VAE is generally more interpretable than C-GAN, as it explicitly models the underlying latent space. C-GAN is more flexible than C-VAE, as it can learn from unstructured data and produce images that match specific conditions. C-GAN can also be more scalable than C-VAE, as it can be trained on larger and more complex datasets [85].

In conclusion, both C-VAE and C-GAN have their strengths and weaknesses when it comes to image generation and OCR system improvement. Choosing between the two methods depends on the specific requirements and constraints of the task at hand. While C-GAN tends to produce higher-quality images and is more suitable for learning from unstructured data, C-VAE is generally faster, more stable, and more interpretable.

2.5 Evaluation Metrics for Generative Models

Evaluation metrics play a crucial role in assessing the performance of generative models, allowing us to measure the quality, diversity, and fidelity of the generated samples. These metrics are designed to provide quantitative insights into how well the models capture the characteristics of the real data distribution.

Evaluation metrics for generative models can be broadly categorized into two groups, shown below in the figure 1:

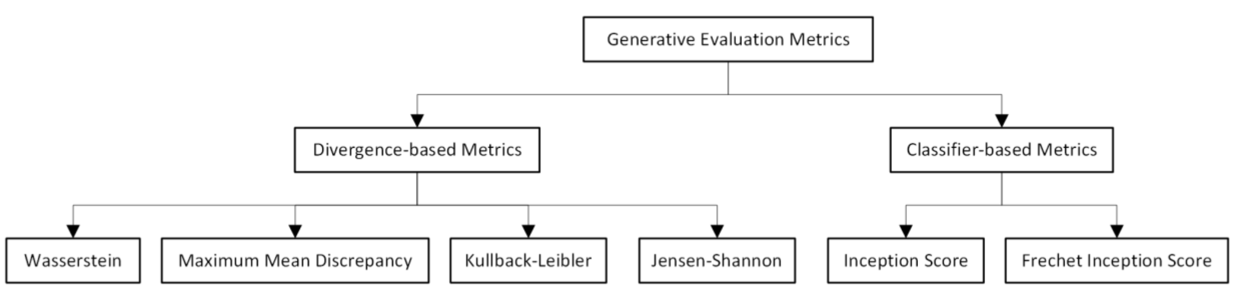

Figure 1: Evaluation Metrics

2.5.1 Classifier-based Metrics

These metrics utilize pre-trained classifiers to extract features from both real and generated samples, enabling a comparative analysis. Two commonly used metrics in this category are the Inception Score (IS) and the Fréchet Inception Distance (FID).

Inception Score (IS): Measures the quality and diversity of generated samples by evaluating both the conditional and marginal distributions. It utilizes a pre-trained Inception model to extract features and provides an aggregate score that combines these two aspects [15].

The IS measures these two aspects as follows:

Diversity: Good models should produce a variety of different images, not just variations of the same image. This is quantified by the entropy of the marginal distribution over labels. A higher entropy means more diversity.

Quality: Good models should generate images that look like the training data. This is quantified by the conditional distribution of the labels given the generated images. If a model is good, the conditional distribution should have low entropy for each image (i.e., the model is confident about the label of the image).

Let's denote:

- $p\,(y|x)$ as the conditional distribution of the label y given the image x, produced by the Inception model.
- $p\,(y)$ as the marginal distribution over labels, obtained by integrating over all images x in the generated data: $p\,(y) = \int p\,(y|x)dx$.

The Inception Score is then defined as:

$$IS = exp(\ E\_x[\ KL(p(y|x)\ ||\ p(y))\ ]\ )$$

where:

- E_x denotes the expectation over all generated images x.
- KL is the Kullback-Leibler divergence, which measures how one probability distribution is different from a second, reference probability distribution.

Fréchet Inception Distance (FID): Quantifies the distance between the feature representations of real and generated samples, leveraging a pre-trained Inception model. The FID score provides a measure of similarity between the distributions and is particularly useful for assessing the overall fidelity and diversity of generated samples. The Fréchet distance is a metric for measuring the similarity between two curves. Mathematically, for two continuous curves $C1 : [0,1] \rightarrow M$ and $C2 : [0,1] \rightarrow M$ in a metric space (M, d), it's defined as:

$$F(C1, C2) = inf\ max\ d(C1(\alpha(t)), C2(\beta(t))), for\ all\ t \in [0,1]$$

The Fréchet Inception Distance (FID) score extends this concept to compare the distributions of real and generated images in the feature space of a pre-trained Inception model. The steps to calculate the FID score include preprocessing the images, passing them through the Inception model, and calculating the mean and covariance of the activations for both real and generated images. The FID score is then computed as:

$$FID = d^2 + trace$$

where:

- $d^2 = \left|\left|\mu_{real} - \mu_{gen}\right|\right|^2$ is the squared Euclidean distance between the means.
- $trace = Tr(\Sigma_{real} + \Sigma_{gen} - 2 * \left(\Sigma_{real} * \Sigma_{gen}\right)^{\frac{1}{2}})$ is the trace of the sum of covariance matrices and their square root. A lower FID score indicates the generated images are of higher quality and more similar to the real ones.

Although we have provided an overview of its calculation here, a more comprehensive discussion of the FID score, particularly its fine-tuning to better suit specific datasets, will be elaborated in the Methodology chapter. This forthcoming section will provide detailed insights into how we adapt and apply the FID score in the context of our specific research and data. By optimizing this measure for our unique dataset, we aim to achieve a more precise evaluation of the performance of our generative models. Stay tuned for this in-depth exploration in the upcoming chapters.

2.5.2 Divergence-based Metrics

These metrics estimate divergences or distances between the distributions of real and generated samples, providing a more principled measurement of fidelity and diversity. Some commonly used metrics in this category include Kullback-Leibler divergence (KL), Jensen-Shannon divergence (JS), Wasserstein distance (W), and Maximum Mean Discrepancy (MMD).

Kullback-Leibler Divergence (KL): Measures the divergence between the distributions of real and generated samples. It quantifies the difference in information content between the two distributions [40]. The KL Divergence between two probability distributions P and Q is defined as:

$$KL(P \mid\mid Q) = \sum P(x) * log\left(\frac{P(x)}{Q(x)}\right) for\ all$$

It measures the difference in the information content between the two distributions, P and Q.

Jensen-Shannon Divergence (JS): Calculates the divergence between two probability distributions, typically the real and generated data distributions. It captures the similarity and difference between the two distributions [86].

The JS Divergence is another method to measure the similarity between two probability distributions, P and Q. It's defined as:

$$JS(P \,||\, Q) \;=\; 1/2 \,*\, KL(P \,||\, M) \;+\; 1/2 \,*\, KL(Q \,||\, M)$$

where M is the average of P and Q, defined as:

$$M \;=\; 1/2 \,*\, (P \;+\; Q)$$

Wasserstein Distance (W): Also known as Earth Mover's Distance, it measures the distance between the real and generated data distributions by computing the minimum cost of transforming one distribution into another [130].

The Wasserstein distance between two probability distributions P and Q is defined as the solution of the following optimization problem:

$$W(P,Q) \;=\; inf \sum |x_i \;-\; y_i| \;*\; P(T=i) \; for\ all\ i$$

where the infimum is taken over all joint distributions of (X,Y) with marginal distributions P and Q.

Maximum Mean Discrepancy (MMD): Measures the distance between the means of the real and generated data distributions in a reproducing kernel Hilbert space. It provides a principled way to measure the similarity between the two distributions [21].

The MMD between two distributions P and Q in a reproducing kernel Hilbert space H with a kernel k is defined as:

$$MMD(P,Q) \;=\; sup \; \left|\left|E_P[k(X,.)] - \; E_Q[k(Y,.)]\right|\right|_H$$

Evaluation Metric	Methodology	Pros	Cons
Inception Score (IS)	Uses a pre-trained Inception model to extract features from real and generated samples, compares the conditional and marginal distributions of generated data.	Measures both the realism and diversity of the generated samples.	Sensitive to the choice of classifier, potential bias towards certain types of data, does not account for issues like class-conditional generation or memorization.
Fréchet Inception Distance (FID)	Measures the distance between real and generated data distributions in the feature space of an Inception network.	Provides a holistic view of the quality and diversity of generated samples, better correlation with human judgement than IS.	Assumes high-level and complex features in images, may not be effective for low-quality images.
Kullback-Leibler Divergence (KL)	Estimates the divergence between real and generated data distributions.	Provides a principled measurement of the fidelity and diversity of the generated samples.	Computationally expensive, requires access to the true data distribution or its samples, difficult to interpret or compare across different models or datasets.
Jensen-Shannon Divergence (JS)	Calculates the divergence between two probability distributions, typically between the real and generated data distributions.	Symmetric measure, provides a principled measurement of the similarity between real and generated samples.	Computationally expensive, requires access to the true data distribution or its samples, difficulty to interpret or compare across different models or datasets.
Wasserstein Distance (W)	Also known as the Earth Mover's Distance, measures the distance between the real and generated data distributions.	Provides robust and principled ways to measure the fidelity and diversity of the generated samples.	Computationally expensive, requires a lot of resources and time, could be difficult to interpret.
Maximum Mean Discrepancy (MMD)	Measures the distance between means of the real and generated data distributions in a reproducing kernel Hilbert space.	Does not require density estimation, provides a principled way of measuring the similarity between two distributions.	Difficult to choose an appropriate kernel, requires access to the true data distribution or its samples, can be computationally intensive.

Table 2: Evaluation Metrics for Generative Models

where sup denotes the supremum, E_P and E_Q denote the expectations under the distributions P and Q, X and Y are random variables with distributions P and Q respectively, and $||.||_H$ denotes the norm in the Hilbert space H.

Divergence-based metrics offer a more rigorous and theoretically grounded approach but may be computationally expensive and require access to the true data distribution or its samples. When selecting evaluation metrics for generative models, researchers must consider factors such as the specific characteristics of the data, research goals, computational efficiency, and interpretability. Understanding the strengths and limitations of different metrics is essential to choose the most appropriate evaluation approach for a given scenario. Table 2 provides a comparative analysis for various evaluation metrics for generative models:

As our research focuses specifically on real-time optical character recognition (OCR), it be- comes increasingly important to find an evaluation metric that not only accurately measures the performance of generative models but also does so quickly. Optical character recognition is a time-sensitive application where rapid generation and evaluation of synthetic images are crucial for efficient operation. Handwritten Optical Character Recognition (OCR) is a challenging task due to the high degree of variation in handwriting. Handwriting varies significantly from person to person, and even the same person's handwriting can change over time or under different conditions. It can also be affected by factors such as the writing instrument used, the writing surface, and the speed at which the person is writing. This makes it more difficult for OCR systems to correctly identify characters, as there is no single 'template' that can be used to match each character.

Image generation, particularly through Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), has the potential to aid in this task. Generative models can be trained to generate a wide range of examples for each character, which can help an OCR system to learn the diversity of appearances each character can have. This can lead to a more robust OCR system that is better able to handle the variability in handwriting.

However, the evaluation metrics for generative models, which are often used to assess the quality of generated images, have limitations when it comes to recognizing low-quality images. These metrics, such as the Inception Score (IS), Fréchet Inception Distance (FID), Kullback- Leibler Divergence (KL), and others, are designed to measure the overall quality and diversity of the generated images, rather than focusing on specific details that might be crucial for OCR [4, 58]. For example, a model might generate images that score highly on these metrics, indicating that they have high quality and diversity, but if the images are not high-resolution or clear enough, the OCR system may still struggle to correctly identify characters.

Furthermore, these metrics can be computationally intensive, which makes it harder to use them in real-time applications or when dealing with large volumes of data. Therefore, while image generation can potentially improve the performance of handwritten OCR systems, there is still a need for improved evaluation metrics that can accurately assess the quality of generated images in the context of OCR, particularly when dealing with low-quality images or real-time applications.

In this chapter, we conducted a thorough examination of the literature concerning Generative Models designed to enhance Optical Character Recognition (OCR) performance, as well as the assessment metrics employed for these models. In the following chapter, we will delve into the specific Generative Models we deployed to augment the Arabic Handwritten Digit Dataset in OCR. Additionally, we will introduce our innovative method for accurately calculating the Fréchet Inception Distance (FID) score, providing a robust measure for the quality of the images generated.

CHAPTER 3

METHODOLOGY

The purpose of this methodology chapter is to provide a detailed and comprehensive explanation of the research design, data collection, and analysis methods employed in this study. These methodological components are essential for understanding the process and rationale behind the investigation into the effectiveness of Conditional Generative Adversarial Networks (C-GANs) and Conditional Variational Autoencoders (C-VAEs) for enhancing Optical Character Recognition (OCR) performance in Arabic handwritten digit recognition.

In this study, we aim to compare C-GANs and C-VAEs for image generation, introduce a new approach to correctly calculate the FID score for monitoring the quality of generated images. We also propose another new approach to examine the performance of the improved OCR systems using Saliency Maps. The research design, data collection, and analysis methods will guide the investigation and ensure that the results can provide valuable insights into the use of generative models for OCR performance enhancement.

3.1 Research Design

The overall research design of this study is a comparative, quantitative, and experimental approach. This design was chosen because it allows for a systematic comparison of C-GANs and C-VAEs in generating synthetic images and enhancing OCR performance. By employing an experimental approach, we can measure and analyze the impact of different generative models on OCR systems and identify the most effective strategies for improving accuracy and efficiency. The choice of using GANs and VAEs for data augmentation in OCR systems is based on their proven effectiveness in quickly generating synthetic images and their potential for improving the performance of machine learning models. Both C-GANs and C-VAEs have been successfully applied to a wide range of computer vision tasks, including image synthesis, style transfer, and image inpainting, making them relevant and promising candidates for enhancing OCR performance [9].

3.2 Dataset

The dataset used for this study is the Arabic Handwritten Digits Dataset (AHDD), which was created by El-Sawy et al. [108] The dataset contains 60,000 grayscale images of handwritten Arabic digits, with each digit having 6,000 samples. The images are 28x28 pixels in size and represent Arabic digits ranging from 0 to 9 which is shown below in figure 2. This dataset is characterized by its variability in writing styles, as well as the presence of noise and distortions, which makes it a suitable choice for investigating the effectiveness of GANs and VAEs in improving OCR systems. The dataset was divided into three subsets: training, validation, and test sets.

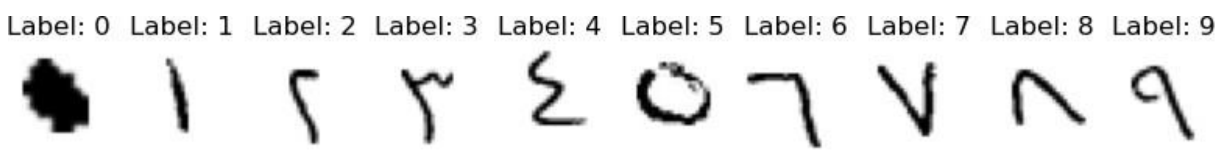

Figure 2: Arabic Handwritten Digits Dataset

Following the standard practice in machine learning research, 70% of the data was allocated to the training set, 15% to the validation set, and 15% to the test set [35]. This division ensures that the models are trained on a sufficiently large dataset while providing separate subsets for model selection and evaluation.

3.3 Image Generation

The process of generating synthetic images using C-GAN and C-VAE involved several steps, including pre-processing and hyperparameter tuning. The pre-processing steps included data normalization, which involved scaling the pixel values to a range between 0 and 1, and data augmentation through random transformations, such as rotation and scaling. For both C-GAN and C-VAE, various architectures and hyperparameter settings were explored to optimize the quality of the generated images. Hyperparameters included learning rates, batch sizes, and the number of training epochs [42].

3.3.1 C-GAN

The algorithm of the Conditional Generative Adversarial Network (C-GAN) [89] , as shown in figure 3, is an extension of the GANs architecture, which was initially proposed by Goodfellow et al. [55]. GANs consist of two neural networks: a generator that creates synthetic data samples and a discriminator that distinguishes between real and synthetic samples. These two networks are trained simultaneously in a min-max game, where the generator tries to create samples that can fool the discriminator, while the discriminator attempts to accurately classify the samples as real or synthetic.

C-GANs incorporate additional conditional information (e.g., class labels) into both the generator and discriminator networks, allowing the model to generate samples based on

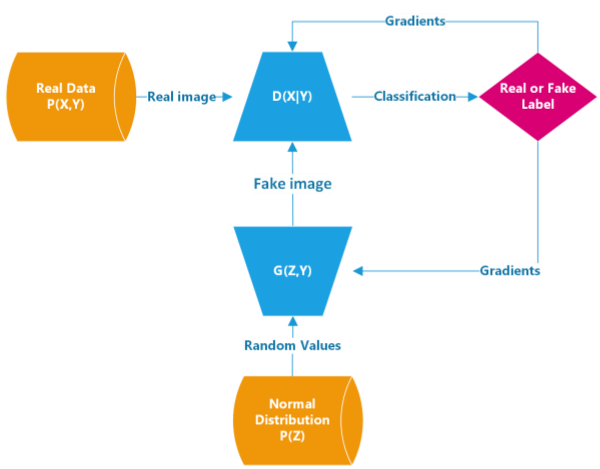

Figure 3: C-GAN Architecture

specific conditions. The conditional information is typically concatenated with the input noise vector for the generator and with the data samples for the discriminator. This modification enables the generation of more targeted and diverse samples, making C-GANs particularly useful for various applications, such as data augmentation and image synthesis [51]. The Conditional

Generative Adversarial Network (C-GAN) is described in Algorithm 1. The architecture of C-GAN is illustrated in Figure 3.

For C-GANs, the loss function is based on binary cross-entropy, which is the original GAN loss function proposed by Goodfellow et al. [55]. The C-GAN loss function aims to optimize the generator (G) and the discriminator (D) through a min-max game. The generator tries to create samples that can fool the discriminator, while the discriminator attempts to accurately classify the samples as real or synthetic. Conditional information, such as class labels, is incorporated into both the generator and discriminator networks. The C-GAN loss function can be expressed as:

$$\begin{aligned} \mathrm{L}(G, D) = \quad & \mathbb{E}_{x,y\sim p_{\mathrm{data}}(x,y)}[\log D(x,y)] \\ & + \mathbb{E}_{z\sim p_z(z), y\sim p_{\mathrm{data}}(y)}\big[\log\big(1 - D(G(z,y),y)\big)\big] \end{aligned} \tag{1}$$

The architecture of this Conditional Generative Adversarial Network (CGAN) consists of two main components: a generator and a discriminator.

Generator: The generator is responsible for generating new synthetic data. It takes a random noise vector and a class label as input, which are then passed through a series of dense and transposed convolution layers. The output of the generator is a 28 × 28 grayscale image. The generated image is conditioned on the input class label through an embedding layer, a dense layer, and a reshaping operation, after which the label-informed tensor is concatenated with the generator input.

Discriminator: The discriminator’s task is to differentiate between real and fake (generated) data. It receives an image and outputs two values: one indicating whether the image is real or fake, and the other indicating the class of the image. The architecture of the discriminator includes convolution layers, batch normalization, LeakyReLU activation functions, dropout layers, and finally, two dense layers for the two outputs.

Algorithm 1 Conditional GAN

Initialize G and D with random weights
Set hyperparameters: α, T, B
for $t = 1$ to T **do**
    for mini-batch x, y **do**
        Sample noise z
        Generate fake samples $\hat{x} = G(z, y)$
        Update the discr.: compute $\mathcal{L}_D$ and update θ_D
        Sample new noise z
        Generate new fakes $\hat{x} = G(z, y)$
        Update the gen.: compute $\mathcal{L}_G$ and update θ_G
    end for
end for

C-GAN Parameters:

Latent Dimension: The size of the random noise vector that is fed into the generator as an input. This random noise vector, often called a latent vector, is a crucial aspect of Generative Adversarial Networks (GANs). It provides the random seed or the initial point of randomness that the generator will use to produce an output. In this work, we used a latent dimension equal to 100. The size (i.e., the number of elements in the vector) is a hyperparameter and can be tuned for different results.

Batch Size: Refers to the number of training examples utilized in one iteration. In this experiment, we used a batch size of 16. In the context of training deep learning models, a 'batch' refers to the subset of the dataset that is used for a single update to the model weights during training. Using a batch size of 16 means that the weights will be updated after 16 examples have been processed. Choosing the right batch size impacts learning as it influences the accuracy of the model's gradient estimate, the stability of the learning process, and the training time.

Epochs: A measure of the number of times all of the training vectors are used once to update the weights in the model. We trained the model for 10 epochs. Running the training process for 10 epochs means going through this entire process 10 times. More epochs could lead to better learning up to a point, after which the model might just be overfitting.

Optimizer: The optimizer is the algorithm used to change the attributes of the neural network such as weights and learning rate in order to reduce the losses. Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure [8] to update network weights iteratively based on training data. The learning rate (0.001) controls how much to change the model in response to the estimated error each time the model weights are updated, and epsilon (1×10^{-8}) is a very small number to prevent any division by zero in the implementation.

Loss Functions: Loss functions are used to measure how well the model is doing at learning the mapping function. Binary Cross-Entropy loss [59] is used for binary classification problems. It is suitable when models output probabilities for the two classes. Sparse Categorical Cross-Entropy loss is a form of categorical cross entropy that is very useful when dealing with large categorical output classes. It is used when the labels and predictions are in the form of integers rather than one-hot encodings.

In the training loop, the discriminator and the generator are trained in an alternating manner. First, the discriminator is trained on a batch of real samples and a batch of fake samples (generated by the generator). Then, the generator is trained using the combined GAN model, where the discriminator's weights are frozen.

The discriminator tries to correctly classify real and fake images, while the generator tries to generate images that the discriminator cannot distinguish from real images. Over time, both the generator and discriminator improve their capabilities, creating a kind of arms race, leading to the generation of more realistic images.

3.3.2 C-VAE

A Conditional Variational Autoencoder (C-VAE) as shown in figure 4 is a variant of the VAE architecture introduced by Kingma et al. [73]. VAEs are generative models that learn to encode data samples into a lower-dimensional latent space and then decode the latent representations back into the original data space. VAEs consist of two main components: an encoder network that learns the approximate posterior distribution of the latent variables given the data, and a decoder network that learns the likelihood of the data given the latent variables [71].

C-VAEs integrate conditional information (e.g., class labels) into both the encoder and decoder networks, enabling the generation of data samples conditioned on specific variables. The conditional information is typically concatenated with the input data for the encoder and with the latent variables for the decoder. This modification allows the model to generate more diverse and context- specific samples, which can be beneficial for tasks such as image synthesis, data augmentation, and semi-supervised learning.

Both C-GANs and C-VAEs have be implemented using popular machine learning frameworks like TensorFlow [101] . This facilitates efficient model development, training, and evaluation, enabling researchers to thoroughly investigate the performance of both generative models in various contexts, including OCR enhancement. The architecture of the C-VAE is illustrated in Figure 4.

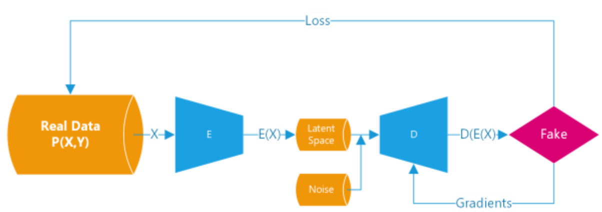

Figure 4: C-VAE Architecture

For C-VAEs, the loss function is a combination of the reconstruction loss and the KL (Kullback- Leibler) divergence [12]. The reconstruction loss measures the difference between the input data and the generated data after encoding and decoding, ensuring that the C-VAE can accurately reconstruct the input samples. The KL divergence measures the difference between the approximate posterior distribution learned by the encoder and the prior distribution of the latent variables, encouraging the C-VAE to learn a smooth and structured latent space. The C-VAE loss function can be expressed as:

$$L_{\text{C-VAE}}(x,y) = \begin{array}{l} \mathbb{E}_{z\sim q(z|x,y)}[\log p(x|z,y)] \\ -D_{\text{KL}}(q(z|x,y)||p(z|y)) \end{array} \tag{2}$$

Both loss functions play a critical role in the training process and determine the effectiveness of the C-GAN and C-VAE models in generating synthetic images for data augmentation and improving the performance of OCR systems.

The optimization algorithm used for both generative models and the OCR model was the Adam optimizer, which has been shown to be effective in training deep neural networks [138]. Training parameters, such as the learning rate, batch size, and the number of training epochs, were selected based on a of literature recommendations and empirical testing [19].

CVAE is structured with an encoder, a decoder, and a reparameterization step in the middle. The encoder converts the input data into a latent representation, while the decoder reconstructs the data back from the latent space.

Encoder: The Encoder is defined as a class. It’s a neural network that transforms input data into two parameters in a latent space, which are means and log_vars (logarithm of variances). These parameters are used to sample a latent vector, z. If the CVAE is conditional (i.e., "conditional" is set to True), the encoder will consider the labels of the input data (encoded as one-hot vectors) along with the input data itself.

Decoder: The Decoder is also defined as a class. It's another neural network that transforms a given latent vector back to the original data. If the CVAE is conditional, the decoder will consider the labels of the input data (encoded as one-hot vectors) along with the latent vector. The VAE model is defined to have the encoder and decoder as its components. The forward method of the VAE is where the encoder's output is sampled (using the reparameterization trick to allow backpropagation) and passed through the decoder [66].

C-VAE Parameters:

Number of Epochs: The C-VAE model is configured with 10 epochs. The term 'epoch' refers to a single iteration where the whole dataset is exposed to the model, encompassing both forward and backward passes. Therefore, in this scenario, the entire dataset will make ten complete journeys through the model.

Batch Size: The batch size has been set to 16. This configuration determines that during the training process, the model will receive sixteen samples from the dataset at every step, again covering both forward and backward passes. While smaller batch sizes may speed up the training process, it could also destabilize it, resulting in more fluctuations in loss. Striking a balance is crucial [114].

Learning Rate: A learning rate of 0.001 is established. The learning rate dictates the extent of alteration in the model parameters in response to the estimated error each time the model's weights are updated. Setting it too high may cause the model to overshoot the optimal solution, while a too low rate may either slow down the training or cause it to stagnate. The value of 0.001, used here, is a common choice for many training scenarios. It is also important to note that the Adam optimizer is being utilized, with a learning rate of 0.001 and epsilon set at 1×10^{-8} [114].

Encoder and Decoder Layer Sizes: The architecture of the model is defined by the encoder and decoder layer sizes. The encoder's input layer consists of 784 neurons (28x28, reflecting the size of the input image), followed by a layer with 512 neurons. While this particular architecture is fairly simple with just two layers, more could be added for models that need to handle more complexity. The decoder part of the VAE is also delineated by layer sizes. Its input layer, linked to the latent space, has 512 neurons, followed by a layer with 784 neurons. This reflects a mirrored structure to the encoder but in the reverse sequence [134].

Latent Space Dimensionality: Lastly, the dimensionality of the latent space, otherwise known as the latent size, is set to 10. This parameter determines the size of the compressed representation of the input data. The selection of the latent size represents a trade-off; while a larger latent space might capture more complex representations, it also increases computational complexity and potentially heightens the risk of overfitting.

The loss function used for training the CVAE is the sum of the reconstruction loss (Binary Cross Entropy) and the KL divergence, which is a measure of how one probability distribution diverges from a second, expected probability distribution. The Adam optimizer is used to minimize this loss.

3.3.3 Optimization

The Adam (Adaptive Moment Estimation) optimizer is a popular optimization algorithm for training deep neural networks, introduced by Kingma et al [72] . It is an extension of the stochastic gradient descent (SGD) algorithm [8] that adapts the learning rate for each parameter individually, based on the first and second moments of the gradients. This adaptive learning rate approach allows the optimizer to converge faster and achieve better performance compared to standard SGD. The Adam optimizer computes the first moment (mean) and the second moment (uncentered variance) of the gradients using exponential moving averages. It then corrects the bias in these moment estimates and updates the parameters using the corrected moments. The update rule for each parameter can be expressed as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t,$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2,$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \hat{v}t = \frac{v_t}{1-\beta_2^t} \theta t + 1 = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{3}$$

Here, g_t denotes the gradient at time step t, β_1 and β_2 are the exponential decay rates for the first and second moment estimates, m_t and v_t represent the first and second moment estimates, $\hat{m}_t$ and $v\hat{}_t$ are the bias-corrected moment estimates, α is the learning rate, ϵ is a small constant to avoid division by zero, and θ_t is the parameter at time step t. The choice of the Adam optimizer for both the generative models and the OCR model in this study was based on its proven effectiveness in training deep neural networks. Training parameters, such as learning rate, batch size, and the number of training epochs, were selected based on a combination of literature recommendations and empirical testing [107].

3.4 Evaluation

Fréchet distance, named after Maurice Fréchet, a French mathematician, is a measure of similarity between curves [7]. It captures the intuition that two curves are close if you can travel along both curves at the same time while keeping the leash (a line that connects the points on two curves) short. More formally, the Fréchet distance is defined as the minimum leash length that is sufficient for both ends of the leash to traverse their respective curves from start to end. One can imagine walking a dog on a leash: you can move at different speeds, pause, and even reverse, but you can't teleport or leave the path.

In mathematics, the Fréchet distance between two curves in a metric space is a measure of the extent to which the curves are close to each other in their overall shapes rather than merely at individual points. This is particularly useful in many scientific fields, including computational geometry, computer graphics, and GIS systems, where comparing the similarity of different trajectories, paths, or time series data is important [57].

Mathematically, the Fréchet distance can be defined as follows:

$Let\ C1 : [0,1] -> M\ and\ C2 : [0,1] -> M$ be two continuous curves in a metric space (M, d), where d denotes the distance in M.

We are looking for reparameterizations $\alpha, \beta : [0,1] -> [0,1]$, which are continuous and non-decreasing with $\alpha(0) = \beta(0) = 0\ and\ \alpha(1) = \beta(1) = 1.$

The Fréchet distance is then defined as the infimum of all constants $\varepsilon \geq 0$ such that there exist reparameterizations α and β with the property:

$$sup\ t \in [0,1]\ d(C1(\alpha(t)), C2(\beta(t))) \leq \varepsilon \tag{4}$$

The expression 4 represents the maximum length of the leash needed for a given pair of reparameterizations α and β.

Therefore, the Fréchet distance is the minimum maximum leash length over all pairs of reparameterizations. In other words, it represents the shortest possible leash that would allow someone traversing the first curve and a dog traversing the second curve to stay connected by the leash for all possible ways of moving along the curves. In the case of discrete curves (also known as polygonal curves), the computation of the Fréchet distance is significantly simplified because the number of points on the curve is finite. We can imagine each curve as a sequence of points, and we're interested in comparing these sequences in a way that respects the order of the points.

3.4.1 Fréchet Inception Distance (FID) Score

The Fréchet Inception Distance (FID) score is a metric used for evaluating the quality of images generated by generative models. It calculates the distance between the real and generated images' feature representations, where these representations are calculated using an Inception model [120].

The use of the Inception v3 model is crucial in the FID score calculation because it is capable of extracting high-level features from the images, allowing a meaningful comparison between the real and generated images. It's also important to note that the Inception v3 model is used as a fixed feature extractor and is not trained or fine-tuned during the FID score calculation.

Inception v3 is a convolutional neural network (CNN) developed by Google for image recognition tasks. This model is an enhancement of its predecessors, Inception v1 (also known as GoogLeNet) and Inception v2 [119].

The defining feature of the Inception v3 architecture is the use of "modules," or small clusters of layers, repeatedly deployed within the network. These "Inception modules" are designed to discern patterns at varying scales within the image.

One such module utilizes factorized 7×7 convolutions to reduce computational complexity while preserving the capability to identify high-level features from the input. This is done by breaking down the 7×7 convolution into a sequence of 1×7 and 7×1 convolutions, reducing both the number of parameters and computational cost.

The inception architecture is further extended with the integration of pool projection, another module that includes a pooling operation running parallel to convolution operations. This enhancement increases the model's capacity to handle diverse inputs and combats overfitting.

The architecture begins with a standard convolutional and max pooling layer, succeeded by multiple inception modules. Auxiliary classifiers are included in the network, which helps propagate back gradient information into the deep network, mitigating the vanishing gradients issue.

Generative models can leverage the Inception v3 model to extract features from various layers, enabling the computation of the Fréchet Inception Distance (FID) score. Inception v3, a pre-trained model on the ImageNet dataset, serves as a foundation for feature extraction. The FID score is employed to assess the quality of images generated by these generative models.

Rather than using the default "pool3" layer, corresponding to the 2048-dimensional final average pooling features, other layers can be used:

- 64-dimensional features: These correspond to features extracted post the first max pooling layer. Such features will represent very basic image features such as edges and colors.
- 192-dimensional features: These correspond to features extracted post the second max pooling layer. These features will capture more complex patterns compared to the first layer but will still be relatively low-level.
- 768-dimensional features: These correspond to features extracted before the auxiliary classifier (pre-aux classifier features). These features represent a higher level of abstraction and are closer to the final classification layer.

- 2048-dimensional features: These correspond to features extracted from the final average pooling layer (this is the default in the FID score). These high-level features offer a good balance between low-level image features and high-level semantic content.

Figure 5: ImageNet Features learned by Inception v3[100]

Figure 5 illustrates the learned features from different level layers of Inception V3, which was trained on the ImageNet dataset. The depicted features span from basic characteristics found in the lower convolutional layers (left) to increasingly abstract features observed in the higher convolutional layers (right).

The model culminates with an average pooling layer and a SoftMax layer providing the final classification results.

Pre-training a neural network on a large-scale dataset like ImageNet allows the model to learn useful feature representations from the data. When applied to a specific task on a new dataset, it can be fine-tuned, meaning the weights learned during pre-training are minutely adjusted to align with the new data. This approach usually results in superior performance than training a model from scratch, especially when the new dataset is relatively small, as the pre-training assists in preventing overfitting. Thanks to its diverse and high volume of classes, the ImageNet dataset allows models like Inception v3 to learn a broad spectrum of feature representations.

The process of calculating the Fréchet Inception Distance (FID) score starts by preprocessing the real and generated images to align with the input requirements of the Inception model. Typically, this preprocessing involves resizing the images to the input size expected by the model and normalizing the pixel values.

Next, a pre-trained Inception model is loaded. This model is often trained on a large dataset like ImageNet, providing it with a broad understanding of various image features. Once the model is prepared, the real images are passed through the Inception model. The activations from a selected layer are then obtained, capturing the high-level features of the real images.

Similarly, the generated images are also passed through the same Inception model. Activations from the same selected layer are extracted, providing a representation of the high-level features present in the generated images.

The next step involves statistical analysis of these extracted activations. The mean of the activations from the real images, denoted as μ_{real}, and the mean of the activations from the generated images, denoted as μ_{gen}, are calculated. Subsequently, the covariance matrix of the activations from the real images, denoted as Σ_{real}, and the covariance matrix of the activations from the generated images, denoted as Σ_{gen}, are computed.

Further, the squared Euclidean distance between the means of the real and generated activations is computed. This is represented as:

$$d^2 = \left|\mu_{real} - \mu_{gen}\right|^2$$

This measurement offers a quantifiable assessment of the difference between the features of the real and generated images.

Following this, the trace of the sum of the covariance matrices and their square root is calculated.

This is expressed as:

$$race = \left(\Sigma_{real} + \Sigma_{gen} - 2 * \left(\Sigma_{real} * \Sigma_{gen}\right)^{\frac{1}{2}}\right)$$

This calculation provides an additional measure of the difference in the dispersion of the real and generated image features.

Finally, the FID score is calculated using the following formula:

$$FID = d^2 + trace$$

This score reflects both the difference in the mean and dispersion of the real and generated image features, providing a comprehensive measure of the similarity between the real and generated images. This score is used as a standard measure to evaluate the performance of generative models. The FID score measures the similarity between the feature distributions of the real and generated images. It takes into account both the differences in means (representing content) and the differences in covariances (representing variety) between the two distributions.

The smaller the FID score, the closer the generated image distribution is to the real image distribution, indicating higher quality and similarity. Conversely, a larger FID score indicates greater dissimilarity between the distributions, implying lower quality and less resemblance to the real images.

It is important to note that FID is generally used with Inception activations from a specific layer of the Inception model, typically the activations from the last layer. The activations represent high-level features extracted from the images, which are then used to compute the mean and covariance. This approach enables a meaningful comparison between the real and generated image distributions.

3.4.2 FID Score Limits

While the Fréchet Inception Distance (FID) score is widely used and provides a holistic view of the quality and diversity of generated images, it has its limitations. The traditional FID score calculation relies on high-level features extracted from the Inception model, which may not be suitable for low-fidelity or low-quality images [67, 75, 136].

In our investigation, we discovered that the FID score can be significantly improved by using lower-dimensional Inception activations specifically tailored to low-quality images. By focusing on these lower-dimensional features, such as shapes and edges, which are more relevant and prominent in low-quality images, we can obtain a more accurate measure of image similarity that aligns better with human perception.

However, this realization raises further questions that need to be addressed. We need to investigate the complexity of visual features present in our dataset, in this case, Arabic handwritten digit images, and determine the complexity of features in the pre-trained Inception model that should be used to calculate the FID score. This investigation will enable us to fine-tune the FID computation and enhance the evaluation of generative models in the context of real-time optical character recognition tasks.

By exploring these questions and finding the optimal approach, our research aims to establish a more accurate and efficient method for evaluating generative models when generating images with different feature complexity, specifically focusing on the task of recognizing handwritten character images. The goal is to develop an evaluation method that aligns well with human perception and improves the performance of generative models in real-time optical character recognition applications.

3.4.3 Low-dimensional Fréchet Inception Distance Score

We have introduced the Low-dimensional Fréchet Inception Distance (LFID), which is a designed metric that evaluates the discrepancy between the distributions of real and generated images. By focusing on lower-level image features vital for character recognition, LFID allows for a detailed assessment of synthetic image quality in relation to its impact on Optical Character Recognition (OCR) systems. This metric is crucial during training, enabling model alterations to yield images with optimal feature quality, thus improving OCR performance. Furthermore, LFID facilitates early termination of the learning process when models reach desired performance, helping to conserve resources required for model training. We have adapted the LFID to suit character recognition by lowering feature levels, thus reducing computational load when calculating the Fréchet distance. This modification enhances LFID's suitability for real-time data augmentation, despite its low dimensionality. Even with reduced dimensions, LFID has been shown to offer accurate image quality evaluations in comparison to the standard Fréchet Inception Distance (FID). By maintaining a robust correlation with human perceptual judgments, LFID guarantees reliable evaluations of generated images. The LFID allows for the quick evaluation of generated images without sacrificing accuracy, facilitating real-time monitoring and adjustments during training. This immediate feedback helps in spotting and addressing issues with the synthetic images, leading to changes in the model architecture or training parameters. Consequently, this enhances the quality of the generated images, improving OCR performance. In essence, LFID is an efficient and accurate alternative to traditional FID for monitoring image generation tasks, including OCR. It minimizes the dimensionality of Inception features, speeding up the evaluation of generated image quality without losing accuracy. Hence, it's an ideal choice for real-time monitoring and adjustments during training. To assess the performance of C-GAN and C-VAE in producing synthetic images and enhancing OCR performance, we utilized a combination of innovative and conventional metrics. The LFID is a crucial part of this evaluation, effectively assessing the quality of images generated by the models during training. LFID's design is computationally efficient while preserving the ability to precisely evaluate image quality. The study also implemented ablation experiments to assess the significance of different components of the generative models and the OCR systems. This method, which involves the systematic removal or modification of specific components or hyperparameters, enabled the researchers to understand their influence on overall model performance. This broad assessment approach confirmed the findings and shed light on the aspects contributing to the improved OCR performance through generative-based data augmentation.

3.4.4 Synthetic Image Evaluation Procedure

This section on outlines the process for evaluating image quality generated by generative models, primarily using the Low-dimensional Fréchet Inception Distance (LFID) score. We'll cover feature extraction with the Inception V3 model, visualizing learned features, selecting relevant complexity levels, and computing FID score. LFID measures the similarity between real and generated images' feature distributions, guiding model adjustments and hyper-parameter tuning. Early stopping will be discussed as a strategy to prevent overfitting during model training. This section aims to provide a concise understanding of the evaluation process for low-dimensional synthetic images.

Feature Extraction: This initial phase of the evaluation process involves extracting meaningful features from images using a large, pre-trained model. The model is the Inception V3 architecture, renowned for its proficiency in handling a wide variety of image-related tasks.

Pretrained Model: We employ the Inception V3 model, which has been pre-trained on the ImageNet dataset [79].

Mathematically, the Inception V3 model works by convolving the image with learned kernels, applying batch normalization, and then a rectified linear unit (ReLU) non-linearity, following this general formula:

$$X\_new = ReLU(BN(Conv(X\_old, W) + b))$$

Here, Conv is the convolution operation with kernel W, BN stands for batch normalization, ReLU is the rectified linear unit non-linearity, and b is the bias. This operation is performed several times throughout the model to extract and transform features from the image.

Further in the model, these features are flattened and passed through a fully connected layer to generate a fixed-size vector for each image. If we denote the flatten operation as Flatten, and the fully connected layer operation as FC, this can be expressed as:

$$Z = FC\,(Flatten(\,X_{last}))$$

Here, X_{last} is the last feature map generated by the convolution operations. The vector Z then represents the extracted features from the image.

The feature extraction process can be viewed as a function f that takes an image I and returns a feature vector Z:

$$Z = f\,(I)$$

This function f is what we refer to as the Inception V3 model pre-trained on the ImageNet dataset. The vectors Z are what we use in the subsequent stages of the evaluation process.

Feature Visualization

After the feature extraction, the next phase involves visualizing the extracted features. This helps in understanding what kind of image properties the model has learned to identify and highlight.

Visualizing Learned Features (ImageNet): We then move on to visualize the features that the Inception V3 model, pre-trained on ImageNet, has learned. This can provide crucial insights into the core patterns and structures the model perceives as significant in an image.

Visualizing Learned Features (Fine-Tuned Model): This involves visualizing the features learned by the Inception V3 model, pre-trained on ImageNet, and subsequently fine-tuned on real images. Fine-tuning often leads to a model better suited to the specific task at hand, in this case, feature extraction. This step allows us to observe how this adaptation process has influenced the model's feature recognition abilities.

Selecting Relevant Complexity Level: Lastly, we identify and select the complexity level of the ImageNet features learned by Inception V3 that are most relevant to our real images, Arabic Handwritten Digit Dataset (AHDD). This provides a focal point to ensure the features maintain the essential characteristics found in the actual images, thereby improving the quality of the synthetic image evaluation. We do this by computing the variance of each feature across the dataset and ranking the features based on this variance. Mathematically, this can be expressed as:

$$\sigma\_i\hat{}2 \ = \ var(Z[:,i]), for\ i\ in\ \{1,\dots,n\}$$

Where Z [:, i] represents the i-th feature across all images in the dataset. The var function computes the variance, and σ^2 is the variance of the i-th feature. n in this context represents the total number of features in the feature vector extracted from the images by the Inception V3 model, with i varying from 1 to n. These features represent the distinct characteristics learned by the model from the images. Inception V3 often is used to extract a 2048-dimensional feature vector from an image when using the final layer before the classification layer, implying that n would typically be 2048 in such a case, but it may vary depending on the specific implementation and layer chosen. Features with higher variance are typically considered more relevant since they capture more variability and, therefore, more information about the dataset Calculating LFID

Real Images: Calculate the mean and covariance of the distribution of the lower-level features extracted from the real images. These statistical measures capture the central tendency and dispersion of the feature distribution.

Generated Images: Similarly, compute the mean and covariance of the distribution of the lower-level features extracted from the generated images.

LFID: Use these statistical measures (mean vectors and covariance matrices) from both the real and generated images to compute the Fréchet distance. This distance quantitatively measures the dissimilarity between the two distributions, essentially capturing how far apart the real and generated images are in the feature space.

$$d^2 = \left|\mu_{real} - \mu_{gen}\right|^2$$

$$Tr = \left(\Sigma_{real} + \Sigma_{gen} - 2 * \left(\Sigma_{real} * \Sigma_{gen}\right)^{\frac{1}{2}}\right)$$

$$LFID = d^2 + Tr$$

Image Quality Evaluation: Use the LFID as a measure of the quality of the generated images. A lower LFID indicates that the generated images are more similar to the real images in terms of their lower-level feature distributions, suggesting higher quality.

Model Adjustment: Adjustment: The need for model adjustment is determined based on the LFID score. From our experiments on the Arabic Handwritten Digit Dataset (AHDD), we have found that an LFID score less than 20 indicates significant similarity between the generated and real images. Therefore, if the LFID score exceeds this threshold, T = 20, it suggests that the generated images are not sufficiently similar to the real images, warranting adjustments. This can be formally expressed as:

$$If\ LFID > T, then\ adjust\ model\ parameters$$

Hyper-parameter tuning: The adjustments can encompass changes to the model's architecture, modifications to the training parameters, or alterations in the method of image generation. This is typically conducted through an optimization process, such as grid search or Bayesian optimization, which aims to minimize the LFID score.

We search for the optimal values of the parameters used to train the C-GAN and C-VAE. The parameters of interest might include:

- The architecture of the CNN, including the number of layers (n_layers), size of the filters (filter_size), the size of the stride (stride_size), batch size (batch_size), and optimizer (optimizer_type).
- The latent dimension size (latent_dim).
- The batch size (batch_size).

Formally, this can be represented as a minimization problem:

$$minimize\ LFID(n\_layers, filter\_size, stride\_size, batch\_size, optimizer\_type, latent\_dim)$$

Monitoring LFID: After each adjustment, we monitor the LFID to assess the impact on the image generation quality. Formally, we could express this as:

$$LFID\_new\ =\ compute\_LFID(C - GAN\ or\ C - VAE\ with\ new\ parameters)$$

$$If\ LFID\_new\ <\ LFID\_old, then\ keep\ the\ new\ parameters$$

This iterative process of adjustment and monitoring continues until we reach a point where the LFID does not significantly decrease with further tuning, or we hit a preset limit on the number of iterations or time.

3.4.5 Early Stopping

The early stopping process is an integral part of model training, designed to halt training when the model's performance starts to plateau or degrade. This process is driven by monitoring the LFID metric during the training of the generative models [135].

The LFID score offers a precise assessment of the synthetic image quality, especially in terms of their potential impact on OCR performance. Thus, it is used as a performance criterion to determine when the models have attained optimal performance. Mathematically, the early stopping condition can be defined as follows:

$$If\ |\ LFID\_(i)\ -\ LFID\_(i-1)\ |\ <\ \varepsilon, for\ i\ =\ current\ epoch, then\ stop\ training$$

Here, ε is a small threshold value that determines the sensitivity of the early stopping procedure. If the absolute change in the LFID score between two consecutive epochs ($i-1$ and i) is less than ε, the training is stopped.

This strategy is especially beneficial in two main ways: First, it ensures efficient use of computational resources by avoiding needless training once the model performance has stabilized, as shown in Figure 6. Second, it helps prevent overfitting, a common pitfall where the model performs exceptionally well on training data but poorly on unseen data.

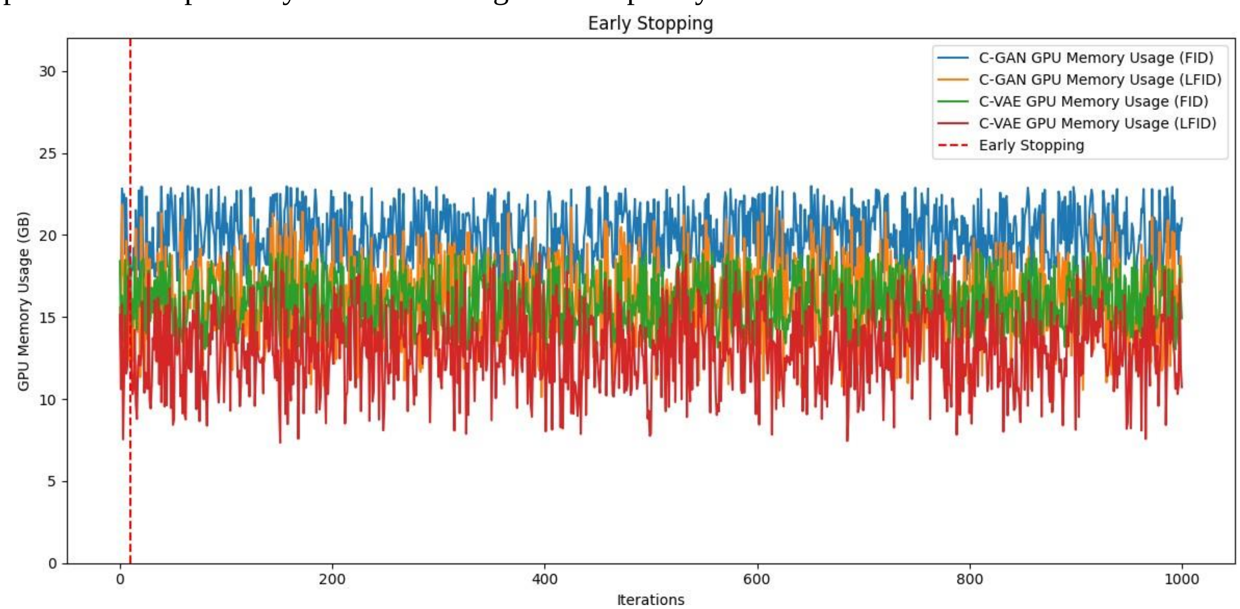

Figure 6: Early Stopping

This mechanism implies a balance between training the model to improve its performance (as measured by the LFID score) and preventing the model from overfitting to the training data, thereby generalizing poorly to new data. The early stopping technique, therefore, plays a critical role in ensuring that the trained model is both efficient and effective.

Please note that the parameter ε and the number of epochs before triggering early stopping (patience) should be chosen carefully, as too small a value might stop training prematurely, while too large a value might result in overfitting.

Algorithm 3 Synthetic Image Evaluation Procedure

procedure EVALUATION
    Phase 1: Feature Extraction
    $model \leftarrow$ load_InceptionV3_pretrained_model()
    $X\_n \leftarrow$ ReLU(BN(Conv(X_o, W) + b))
    $Z \leftarrow$ FC(Flatten(X_l))
    Phase 2: Feature Visualization
    Visualize_features(Z)
    Phase 3: Selecting Relevant Complexity Level
    for $i = 1$ to n **do**
        $\sigma_i^2 \leftarrow$ var($Z[:, i]$)
    end for
    Phase 4: Calculating the LFID
    Compute $\mu\_r, \mu\_g, \Sigma\_r, \Sigma\_g$
    $d^2 \leftarrow ||\mu\_r - \mu\_g||^2$
    $Tr \leftarrow \Sigma\_r + \Sigma\_g - 2 * sqrt(\Sigma\_r * \Sigma\_g)$
    $LFID \leftarrow d^2 + Tr$
    Phase 5: Image Quality Evaluation
    If $LFID > T$ then adjust model parameters
    Phase 6: Hyper-parameter tuning
    Set $params = (n\_l, f\_s, s\_s, b\_s, o\_t, l\_d)$
    Minimize $LFID(params)$
    $LFID\_n \leftarrow$ compute_LFID(C-GAN or C-VAE with new parameters)
    If $LFID\_n < LFID\_o$ then keep the new parameters
    Phase 7: Early Stopping
    If $|LFID_(i) - LFID_(i-1)| < \varepsilon$ then stop training
end procedure

Finally, the early stopping technique can be used in conjunction with other regularization techniques and the aforementioned model adjustment process, providing a comprehensive approach to controlling the complexity and improving the performance of the trained models.

In the context of our Synthetic Image Evaluation Procedure algorithm, we use several abbreviations for brevity and to accommodate space constraints within the algorithm block. The following explains what each abbreviation stands for:

- n_l: Stands for the number of layers in the neural network.

- f_s: Represents the filter size used in the convolutional layers of the neural network.
- s_s: Denotes the stride size, which is the number of pixels shifts over the input matrix.
- b_s: Signifies the batch size used in the training process of the neural network.
- o_t: Corresponds to the type of optimizer used during the training process.
- l_d: Represents the latent dimension in the context of Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs).
- X_n: Stands for the new input tensor after applying the ReLU, BN and Conv operations.
- X_o: Denotes the original input tensor before any operation.
- X_l: Stands for the last layer's output.
- μ_r: Stands for the mean of the real data.
- μ_g: Signifies the mean of the generated data.
- Σ_r: Represents the covariance matrix of the real data.
- Σ_g: Denotes the covariance matrix of the generated data.
- $LFID\_n$: Corresponds to the new calculated LFID metric after parameter adjustment.
- $LFID\_o$: Represents the old LFID metric before parameter adjustment.

We believe that these abbreviations maintain the clarity of the algorithm while efficiently utilizing the available space.

The Synthetic Image Evaluation Procedure (Algorithm 3) is a multi-phase algorithm for assessing the quality of images generated by generative models. It involves feature extraction using Inception V3, visualization of learned features, selection of relevant complexity levels, and computing the Low-dimensional Fréchet Inception Distance (LFID) score. The LFID score serves as a performance measure to optimize generative models through hyper-parameter tuning and adjustments. Early stopping prevents overfitting. This systematic process improves generative models for low-dimensional synthetic image generation.

3.5 Saliency Maps

Convolutional Neural Networks (CNNs) have become a go-to solution for OCR due to their adeptness at handling image data. These networks use a series of filters to extract relevant features from images and then utilize these features to recognize and detect objects, including text characters. However, despite their efficacy, it's often difficult to understand what specific features a CNN uses to detect or classify an object, which might be a problem in critical applications were knowing why a certain prediction was made is as important as the prediction itself. This lack of interpretability or transparency in decision-making is commonly referred to as the 'black box' problem [17].

Saliency maps, introduced to tackle this problem, are a class of techniques that highlight the important regions in an image that contribute to the model's final decision, thereby providing a graphical representation of how an AI system interprets visual data. Saliency maps can help us understand where the model is 'looking' when it makes a decision, i.e., which regions of the image it perceives as salient or relevant. They indicate what the CNN model considers critical in an image when identifying an object or a piece of text [1].

Image saliency maps can be particularly effective in OCR systems built on CNNs, as they allow for the examination of the decision-making process of the model. By observing the areas highlighted by the saliency map, researchers and developers can interpret what visual features the model considers crucial for text detection and recognition. This can lead to better understanding of how the model functions, which in turn can be used to improve model performance and reliability, particularly in complex environments where OCR is often put to the test [109].

The use of image saliency maps has the potential to move us a step closer to opening the ’black box’ of AI, specifically in the realm of OCR, by offering a visual interpretation of what features are deemed significant by a model. This insight can be instrumental in enhancing not only model transparency but also in designing more efficient and reliable OCR systems in the future [92].

In the forthcoming sections of this dissertation, specifically in the "Experiments and Results" segment, we will provide a detailed account of how we utilize saliency maps as an interpretative tool to understand our experimental outcomes. This will encompass the methodology we employed to generate the saliency maps, their practical application in the interpretation of our model’s decisions, and the insights we gleaned about the salient features contributing to the model’s performance. This process will allow us to provide a comprehensive understanding of our CNN-based OCR model’s behavior and reasoning. By visually showcasing what parts of the image our model deemed significant during the detection and recognition process, we will demystify the ’black box’ problem and offer greater transparency into the workings of our model, substantiating its performance and reliability.

In conclusion, the methodology chapter has outlined the systematic approach used to evaluate and enhance the image generation capabilities of Conditional Variational Autoencoders (C-VAE) and Conditional Generative Adversarial Networks (C-GAN). With a focus on improving Optical Character Recognition (OCR) systems, our experiments aimed to explore their effectiveness in the domain of Arabic handwritten digit recognition. The Synthetic Image Evaluation Procedure, as described in the preceding section, served as the foundation for conducting our experiments. This comprehensive algorithm allowed us to extract meaningful features, compute the Low-dimensional Fréchet Inception Distance (LFID) score, and optimize the generative models through hyper- parameter tuning and adjustments. In the next section, "Experiments and Results," we will present and discuss the outcomes of these experiments. The results will shed light on the performance and potential of C-VAE and C-GAN for generating high-quality synthetic images, enabling us to gain valuable insights for enhancing OCR systems.

CHAPTER 4

EXPERIMENTS AND RESULTS

This section is dedicated to describing the series of experiments conducted to analyze and compare the effectiveness of Conditional Variational Autoencoders (C-VAE) and Conditional Generative Adversarial Networks (C-GAN) for image generation in the context of improving Optical Character Recognition (OCR) systems. The emphasis of our experiment was on Arabic handwritten digit recognition [108]. For our experimental approach, we used a pre-established dataset of Arabic handwritten digits. The dataset, exhibiting a wide range of handwriting styles and various degrees of distortion, presents a robust ground for testing the adaptability and efficiency of the OCR systems.

In our initial experiment, we explored the capabilities of both C-VAE and C-GAN in generating diverse synthetic images that bear a resemblance to the original ones in the dataset. The generated images were then introduced into the OCR system to assess its recognition accuracy.

The evaluation of synthetic image quality presents a critical challenge in this domain. To address this, we employed our proposed Synthetic Image Evaluation Procedure (Algorithm 3)

. This evaluation procedure efficiently gauges the quality of the synthetic images produced by the generative models during the training phase. Crucially, it aids in identifying the point of optimal performance, allowing for timely termination of training—a vital aspect considering the usually protracted learning process of generative models.

Furthermore, we adopted Saliency Maps to scrutinize the performance of the upgraded OCR systems. These maps provide insights into which parts of the images the OCR system finds most informative, hence highlighting areas where the system's attention can be optimized.

The experimental results demonstrate that our proposed method of using most realistic synthetic data for enhancing OCR performance offers several advantages. However, it also poses some challenges. Understanding these strengths and limitations can guide future research efforts in this direction and help to devise more precise and efficient OCR systems.

A key highlight of our experiments was the improved performance of the OCR system in recognizing Arabic handwritten digits. Both C-VAE and C-GAN contributed to this enhancement, though they exhibited unique strengths and weaknesses that we discuss in detail. Notably, our proposed evaluation procedure offered faster, and more accurate results compared to traditional FID scores, particularly in the context of OCR applications.

As part of this research, we have strived to fortify the performance of Optical Character Recognition (OCR) systems by leveraging data augmentation techniques, primarily through synthetic data generated by generative models. By continuously monitoring and adjusting the balance between the quality and quantity of the images generated, we have aimed to tackle the instability issues that often arise during the training of generative models. The ultimate goal has been to improve the generalization capability of the models, enhance efficiency, and contribute to the optimization of OCR performance.

4.1 C-GAN

In this section we discuss the experiments and results for the C-GAN model:

Discriminator: The input shape is set to (28, 28, 1) for monochromatic images. The class output is designated to have ten classes, signifying the ten Arabic numerals. The Discriminator uses a *Random Normal* distribution for weight initialization, with a mean of 0.0 and a standard deviation of 0.02. It is constructed with four convolutional layers, with filters set at 32, 64, 128, and 256 for each layer respectively. The kernel size is set to (3, 3) with a stride of (2, 2) and

’same’ padding to keep the output size consistent. *Batch Normalization* is applied after the 2nd, 3rd, and 4th layers to stabilize and accelerate the learning process. The model utilizes the ’binary crossentropy’ loss function for real/fake classification and ’sparse categorical crossentropy’ for class label classification. The *Adam* optimizer, with a learning rate of 0.001 and epsilon of $1e-08$, is used to minimize the loss function.

Generator: The Generator in the C-GAN model takes a 100-dimensional latent space vector as input. The *Conv2DTranspose* function is used for upsampling with strides of (2, 2). The ’ReLU’ activation function is used after Dense layers and the first *Conv2DTranspose* layer, while ’tanh’ is used after the final layer to constrain the output values within a suitable range $(-1, 1)$.

For the complete GAN model, similar to the Discriminator, ’binary crossentropy’ and ’sparse categorical crossentropy’ are used as loss functions for real/fake and class label outputs respectively, and the *Adam* optimizer is again used with the same parameters. This setup ensures the network’s learning process is adequately controlled and guided towards generating realistic and distinct images.

Training Challenges

The C-GAN model faced significant challenges in generating authentic-looking Arabic Hand- written Digit images, as evidenced in Figure 7. This issue stems from the multifaceted nature of training Generative Adversarial Networks (GANs), which often necessitates strategic modifications to the model’s architecture.

The process requires a delicate balance between the generator’s creativity in creating realistic images and the discriminator’s ability to discern real from fake. Achieving this balance can be a non- trivial task, and slight deviations can lead to unsatisfactory results. The training challenges were surmounted through iterative experimentation and adjustment of various architectural elements, laying the groundwork for further analysis.

Training Dynamics

Training GANs can be likened to a two-player minimax game, involving the simultaneous training of two neural networks: the generator, responsible for creating images, and the discriminator, charged with differentiating between real and generated images.

The delicate equilibrium between these two networks must be maintained, and disruptions in this balance can lead to instability in training. High values of loss functions, particularly in the initial phases of training, underscore this instability, as depicted in Figure 7 Achieving convergence in this challenging scenario necessitates careful monitoring and adaptation of training strategies.

Image Quality and Sharpness

One of the redeeming features of the C-GAN model is its capability to generate images with pronounced sharpness. Through the adversarial training process, the generator learns to continually refine its output, leading to images with well-defined edges and contours. Figure 7 showcases this ability, illustrating the model's success in producing sharp and visually appealing images despite the noted training difficulties.

OCR Performance

An intriguing observation was that the enhanced sharpness did not lead to improved performance in OCR systems. Various metrics, such as accuracy, precision, recall, and F1 score, as revealed in Figure 14, highlighted this inconsistency. A closer inspection reveals potential explanations:

Overlooking Crucial Features: The C-GAN might have sacrificed essential features needed for accurate digit recognition in its pursuit of visual appeal. The complexity of representing handwritten digits may have led to this oversight, emphasizing the need for a balanced approach. Mode Collapse: A phenomenon common in GANs, mode collapse, may have occurred. In this situation, the generator starts producing limited or identical images, failing to capture the diversity in the actual data. This lack of variety could hinder the OCR's ability to generalize, explaining the subpar performance with the C-GAN-augmented dataset shown in Figure 7.

Learning Time

An additional point of interest is the discrepancy in learning time between C-GAN and C-VAE, with the former taking 10 times longer, as seen in Figure 9. This difference emphasizes the complexities and challenges associated with GAN architecture, reflecting the intricate balance needed for successful GAN training.

Focus On Unique Features

The Saliency Maps in Figure 12 highlight that the C-GAN model does not concentrate on unique features within each digit. This lack of focus on critical attributes might contribute to the discrepancies in OCR performance, indicating a potential area for model refinement and further exploration.

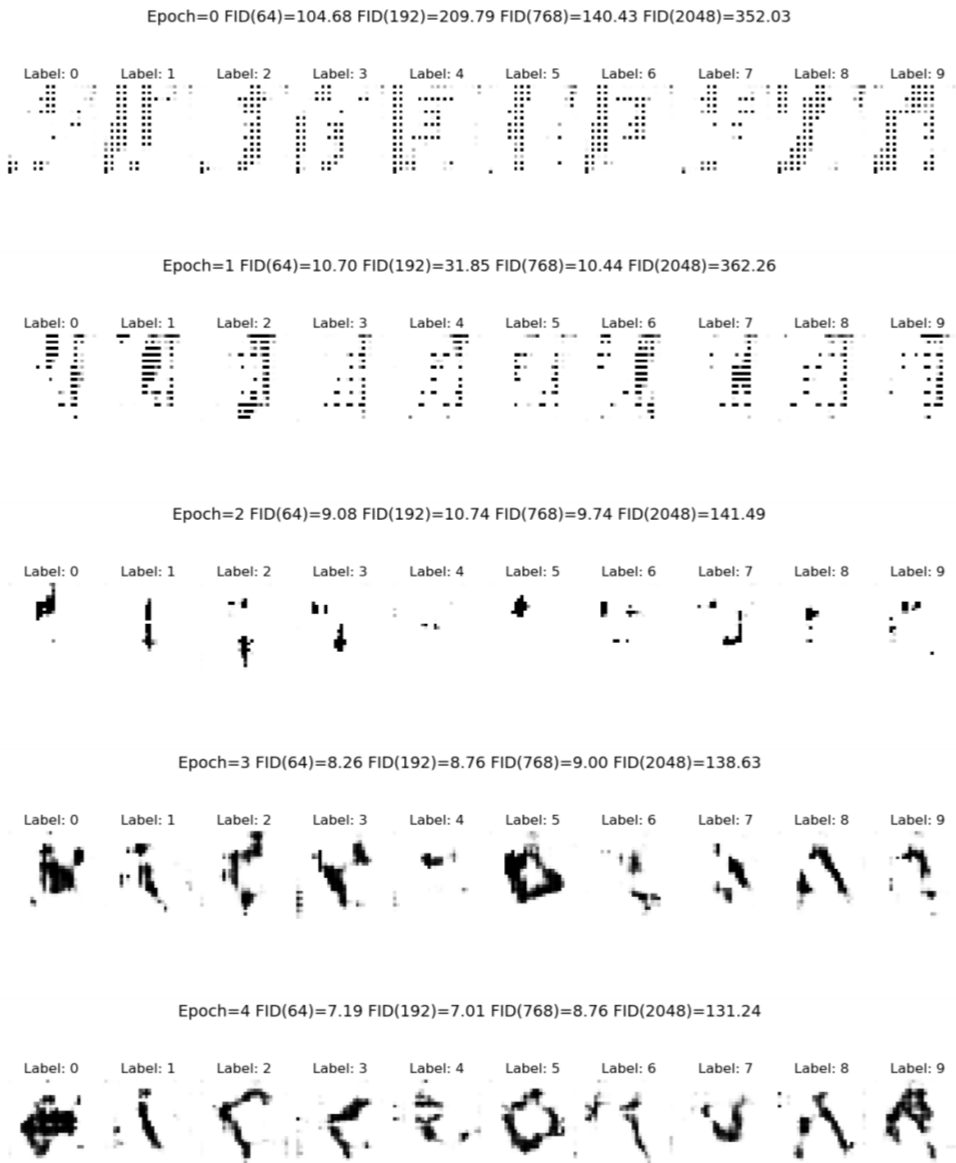

Figure 7: Generated Images by the C-GAN Model (first half)

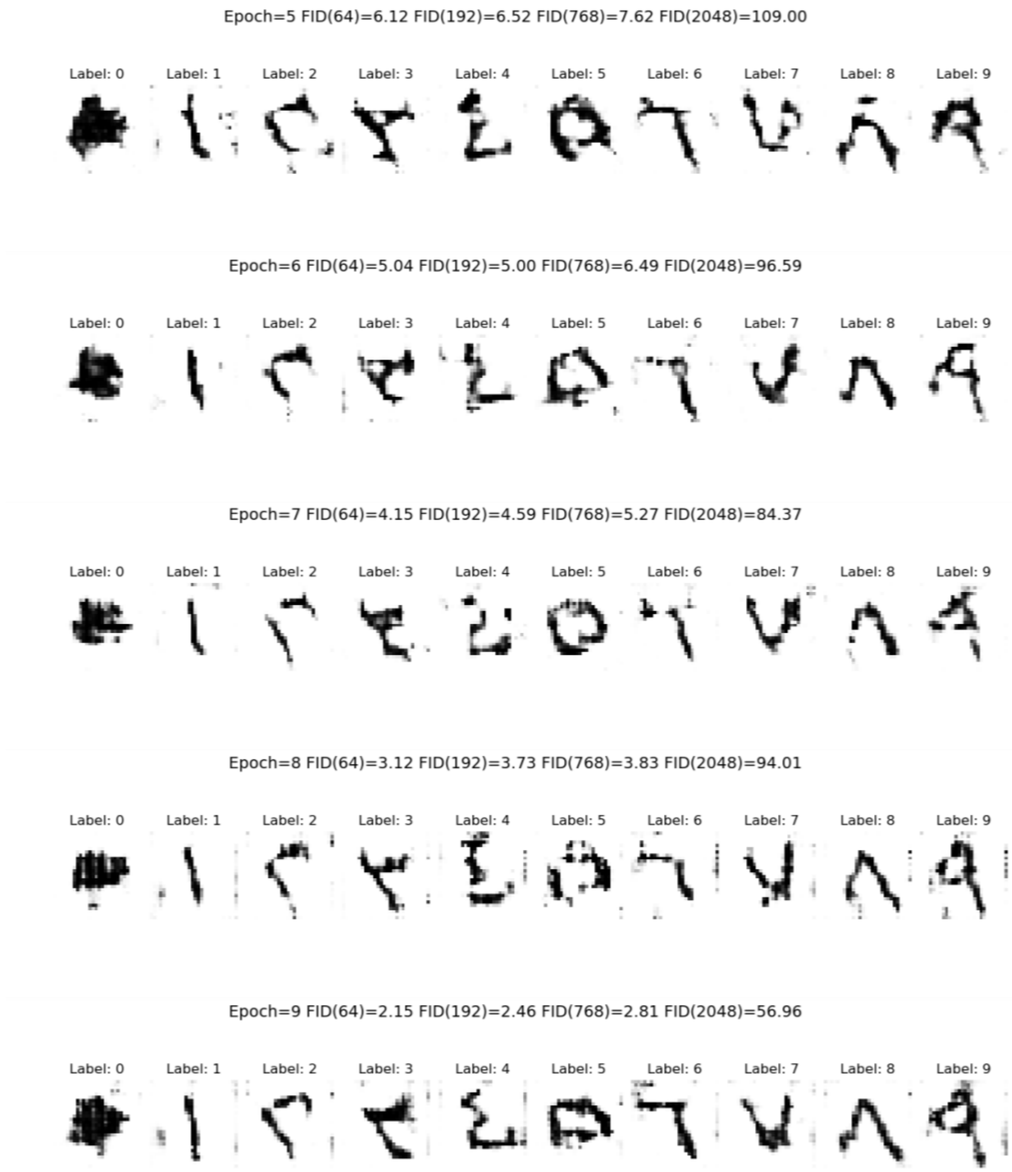

Figure 7: Generated Images by the C-GAN Model (second half)

4.2 C-VAE

In this section we discuss the experiments and results for the C-VAE model:

The Conditional Variational Autoencoder (C-VAE) implemented in this experiment uses several parameters to guide the learning process.

The training process for the C-VAE model is configured to run for 10 epochs. An epoch represents a full pass of the entire dataset through the C-VAE model. The batch size is set to 8, indicating that the model processes eight samples at a time during the training process.

Learning rate, a critical hyperparameter in any optimization algorithm, is set to 0.001. It dictates the size of the steps the model takes towards the minimum of the loss function. A smaller learning rate means the model will learn slowly, but it might result in better performance as it allows the model to fine-tune the weights.

The encoder and decoder architectures of the CVAE model are defined by their layer sizes. The encoder layer sizes are set to [784, 512], which means that the encoder network first maps the input to a 784-dimensional vector, then further down to a 512-dimensional vector. The decoder layer sizes are set to [512, 784], indicating that the decoder network maps the latent representation from a 512-dimensional vector up to a 784-dimensional vector.

The latent size, set to 10, refers to the dimensionality of the space into which the encoder compresses the input data, and from which the decoder generates the output. The loss function for this CVAE model is a combination of Binary Cross Entropy (BCE) and Kullback-Leibler Divergence (KLD). BCE measures the error between the model's output and the actual data, while KLD measures how much the learned latent distribution deviates from a predefined prior distribution.

Adam optimizer is used with a learning rate of 0.001 and epsilon of 1×10^{-8} to guide the learning process and ensure optimal convergence of the model's weights. The 'classes' parameter is set to 10, as there are ten classes of Arabic numerals for the model to learn and generate.

Efficiency In Learning and Image Generation

The C-VAE model demonstrated an impressive ability to mimic the Arabic Handwritten Digit dataset swiftly. As seen in Figure 8, the model began to produce convincing images resembling real handwritten digits after just a single epoch of training. This rapid learning (shown in Figure 10) progression illustrates the efficiency of C-VAEs in understanding the specific underlying data distribution for this task.

Blurriness In Generated Images

Despite its efficiency, the C-VAE model was not without flaws. A notable shortcoming was the blur in the generated images, a common issue with C-VAEs, as shown in Figure 8. The origin of this blurriness is traced to the C-VAE's architecture, which promotes a smooth, continuous latent space through the incorporation of a regularization term in the loss function. While this approach ensures continuity, it leads to averaged output over the distribution, resulting in the observed blur.

Impact On OCR Performance

Surprisingly, the blurriness in the generated images did not impede the OCR performance. As the metrics in Figure 14 demonstrate, the C-VAE-generated images contributed positively to OCR performance. This success can be attributed to the VAE model's ability to retain the essential features necessary for accurate digit classification. Even with the blur, the key attributes distinguishing each digit remained intact, aiding the OCR model in successful recognition.

Focus On Unique Digit Features:

The Saliency Maps presented in Figure 13 further affirmed the C-VAE model's focus on unique features within each digit. By concentrating on these characteristics, the model ensured that the crucial elements for digit classification were preserved.

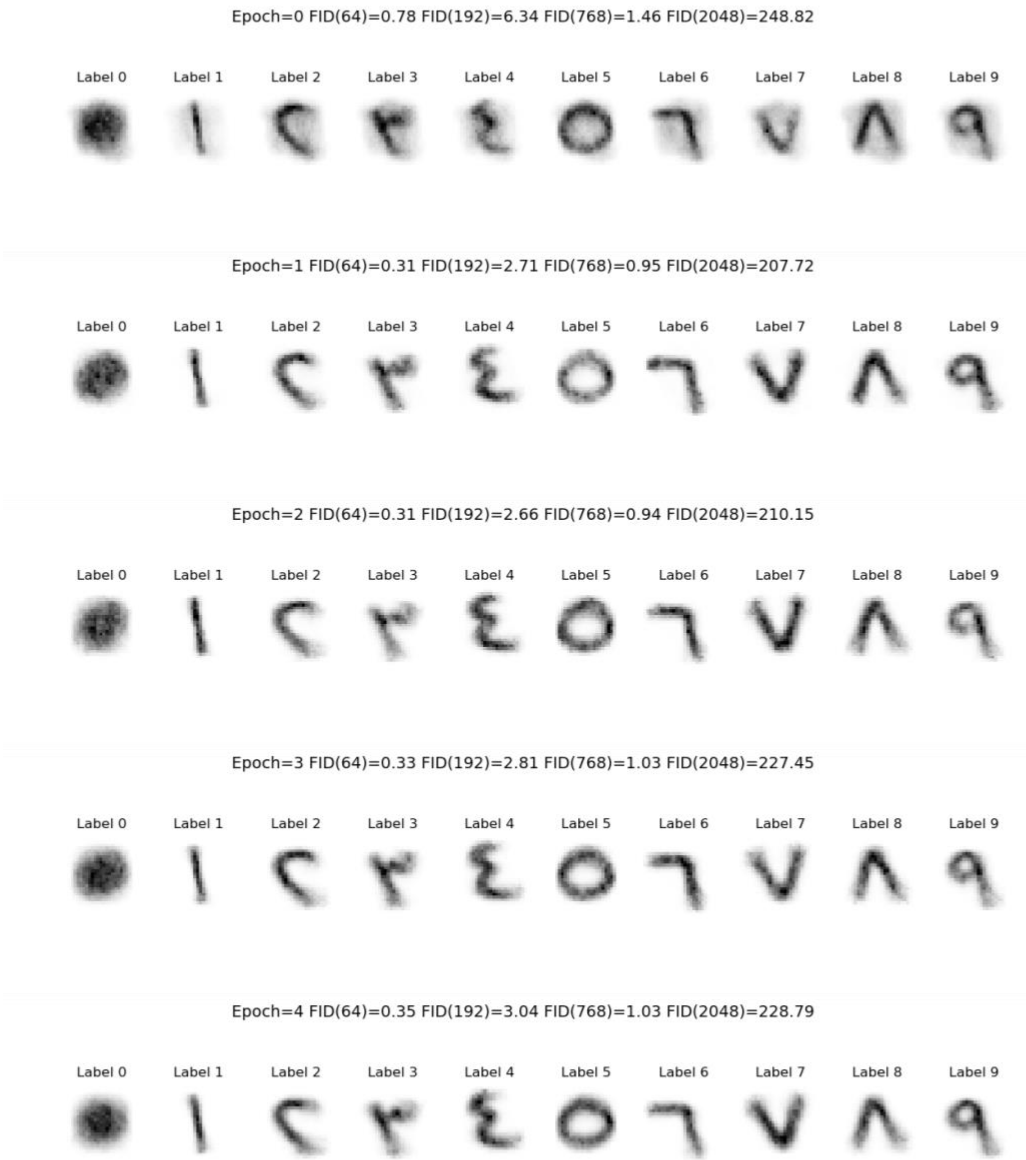

Figure 8: Generated Images by the C-VAE Model (first half)

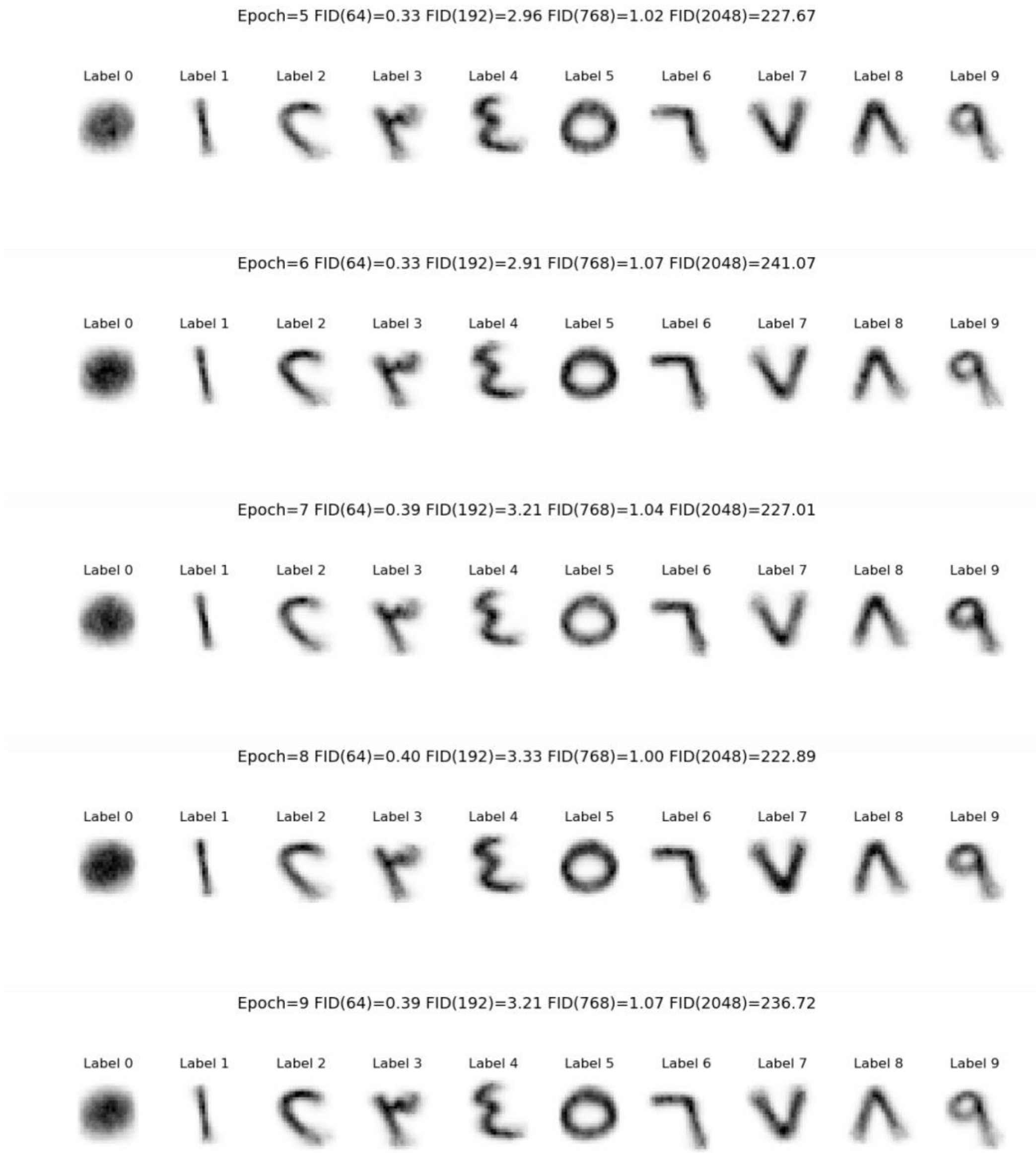

Figure 8: Generated Images by the C-VAE Model (second half)

4.2 Training Procedure

The training procedure for both C-GAN and C-VAE models consists of several crucial steps, designed to effectively train the models and optimize their performance in generating synthetic images for OCR tasks. These steps include:

1. Mini-batch Preparation: The Arabic Handwritten Digits training dataset is preprocessed, and mini-batches of 64 real images and their corresponding class labels are created.

2. Noise and Label Sampling: For each mini-batch, random noise vectors are sampled for the C-GAN model, and latent vectors are sampled from the encoder's output distribution for the C-VAE model. These vectors are combined with class labels.
3. Generator/Encoder Training: The C-GAN generator is trained to generate synthetic images from noise vectors and class labels, while the C-VAE encoder is trained to encode input images into a latent space representation while considering class labels.
4. Discriminator/Decoder Training: The C-GAN discriminator is trained to distinguish between real and generated images while considering class labels, and the VAE decoder is trained to reconstruct input images from sampled latent vectors and class labels.
5. Latent Space Regularization: An essential component of C-VAE training is the regularization of the latent space using the KL divergence loss. This loss encourages the model to learn a smooth and meaningful latent space representation.
6. Training Iterations: The training process involves updating the generator/encoder and discriminator/decoder weights to improve their performance progressively. Limiting the number of training iterations helps balance performance and computational efficiency, preventing overfitting.

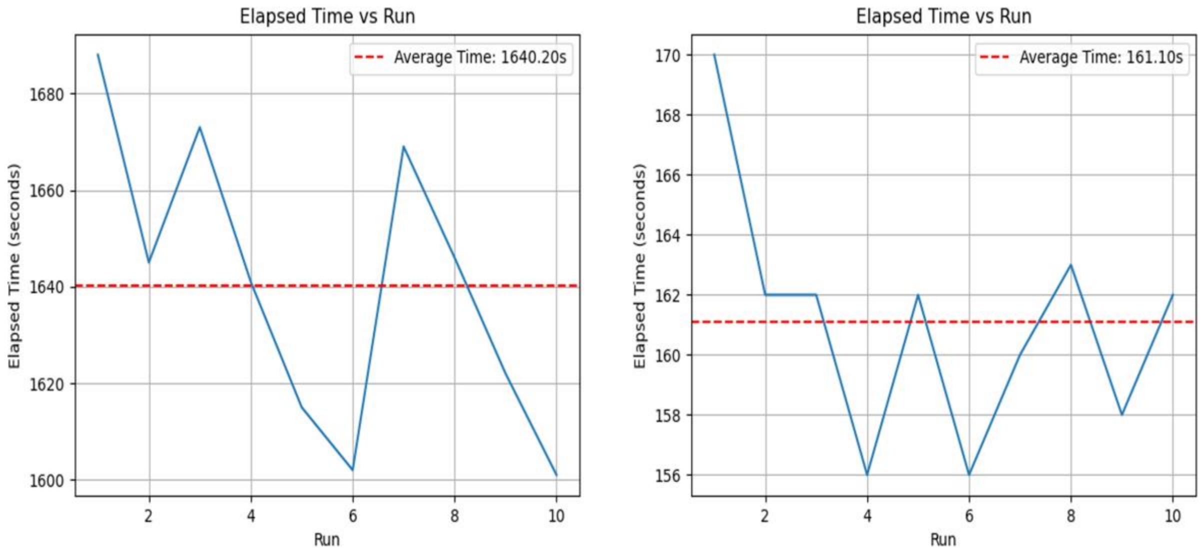

Figure 9: Learning Time (C-GAN on the left and C-VAE on the right)

4.3 Monitoring Progress and Loss Functions

During the training process of GAN and VAE models, it is crucial to keep track of progress and loss functions to guarantee convergence, as well as to identify and address potential issues. This monitoring helps to optimize the models' performance and provides valuable insights into the training dynamics.

1. **LFID Metric**: Periodically calculating the LFID metric allows for the assessment of the quality of the generated images. This computationally efficient metric provides real-time feedback on the performance of the models, which can be used to fine-tune the training process and make necessary adjustments to improve image generation quality.
2. **Model Losses:** Tracking different loss components for each model type is essential for understanding how well the models are learning and adapting during the training process.
    - **C-VAE** it's important to track both the reconstruction loss and the KL divergence loss.

The reconstruction loss measures the difference between the input images and their reconstructions, while the KL divergence loss enforces a smooth and meaningful latent space representation by ensuring it follows a specified prior distribution, typically a standard Gaussian distribution.

- **C-GAN** monitoring the generator and discriminator losses is vital. The generator loss quantifies how well the generator is able to create realistic images, and the discriminator loss measures how accurately the discriminator can distinguish between real and generated images.

By consistently monitoring progress and loss functions, the training process can be optimized, ensuring that the models converge and achieve the desired performance in generating synthetic images for OCR tasks. This monitoring process enables the identification of potential issues and allows for adjustments to the models or training parameters to improve overall performance.

4.4 Early Stopping:

Incorporating early stopping criteria based on the LFID metric or validation dataset performance can be employed to prevent overfitting and reduce training time if the models converge faster than anticipated. This strategy helps optimize the models' performance by stopping the training process when further training would not lead to significant improvements, ensuring efficient use of computational resources.

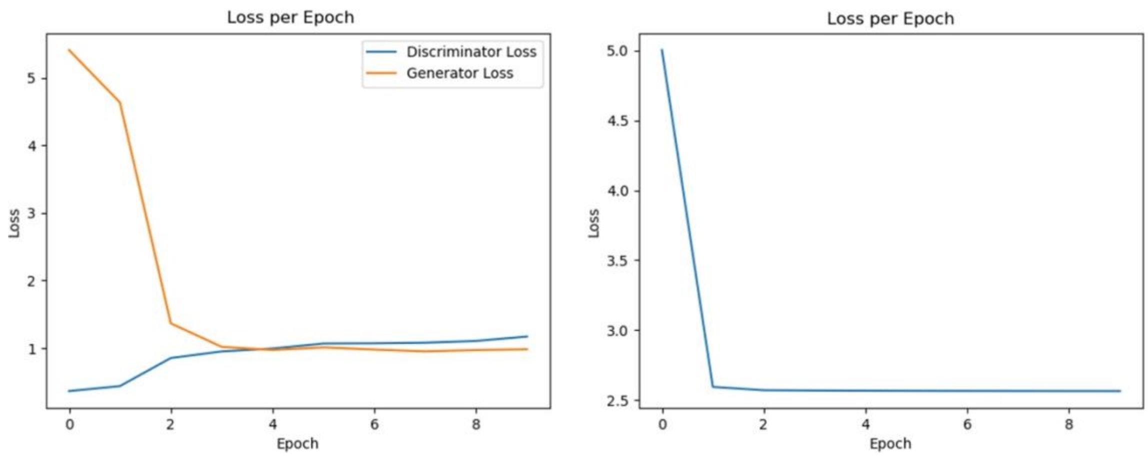

Figure 10: Loss Functions (C-GAN on the left and C-VAE on the right)

In summary, the detailed training procedure for GAN and VAE models focuses on optimizing their performance in generating synthetic images for OCR tasks. By incorporating mini-batch preparation, noise and label sampling, generator/encoder and discriminator/decoder training, latent space regularization, training iterations, monitoring progress, and employing early stopping criteria, the training process aims to effectively train the models, prevent overfitting, and achieve the desired balance between performance and computational efficiency.

4.5 OCR

A Convolutional Neural Network (CNN) is employed as the OCR model for Arabic handwritten digit recognition. The architecture of the CNN consists of two convolutional layers, each followed by a max-pooling layer, a dropout layer, and a fully connected layer for classification. The CNN is designed to learn and extract important features from the input images, facilitating accurate digit recognition.

Training the OCR Model: To investigate the effects of data augmentation on the performance of the OCR model, the CNN is trained separately on the original dataset and each augmented dataset. The training process involves using mini-batch training with a suitable batch size and training the model for a predefined number of epochs. This allows for a comparison of the OCR model's performance when trained on different datasets, providing insights into the impact of the C-GAN and C-VAE-generated synthetic images on the model's accuracy and generalization capabilities.

By augmenting the dataset with synthetic images and training the OCR model on the original and augmented datasets separately, the study aims to assess the effectiveness of different data augmentation strategies and their impact on the performance of the OCR model. This approach helps to understand the potential benefits of using GAN and VAE-generated images for OCR tasks and identify the most effective augmentation strategy for improving model performance.

4.6 Evaluation

Evaluating the OCR Model: After training the OCR model on the original and augmented datasets, its performance is evaluated on a separate test dataset to assess generalization capabilities and compare the impact of different data augmentation strategies. Accuracy is used as the primary evaluation metric, but other metrics like precision, recall, and F1-score can also be reported to provide a more comprehensive assessment of the model's performance, as shown in the Figure 14.

Investigating the Impact of Synthetic Data: To understand the relationship between the quality of synthetic images, as measured by the LFID metric, and the improvement in OCR performance, the correlation between LFID scores and OCR model accuracy on the test dataset is investigated. A strong correlation would indicate that the LFID metric is effective in evaluating the quality of generated images and that higher quality images lead to better OCR performance.

Comparing OCR Features with Arabic Handwritten Digit Features: Another way to assess the impact of synthetic data on OCR performance is to compare the most critical features learned by the OCR model with the unique features of Arabic handwritten digits. This analysis can help identify whether the synthetic images generated by GAN and VAE models capture essential features needed for accurate digit classification. It also provides insights into the effectiveness of different generation loss functions and data augmentation strategies in preserving and enhancing these features for better OCR performance.

By evaluating the OCR model on test data and investigating the correlation between LFID scores and model accuracy, as well as comparing important OCR features with the unique features of Arabic handwritten digits, the study aims to provide a comprehensive understanding of the impact of synthetic data on OCR performance. This approach helps to identify the most effective augmentation strategies and generation loss functions for improving model performance in OCR tasks.

4.7 FID Scores

The Fréchet Inception Distance (FID) score is a metric used to evaluate the quality of images generated by generative models. In essence, a lower FID score suggests that the distribution of features extracted from the generated images is closer to the distribution of features extracted from real images, indicating better quality. This study confirms that a lower dimensional FID (LFID) score correlates positively with the perceived quality of the images generated by the Conditional Variational Autoencoder (C-VAE) as compared to the Conditional Generative Adversarial Network (C-GAN).

Two primary pieces of evidence support this claim. First, human perception of the similarity between generated and real images aligns with the LFID scores. Generated images from the C-VAE model, which have lower LFID scores, are perceived as more similar to the real images than those generated by the C-GAN model with higher LFID scores. This perception of likeness suggests that the C-VAE model is more successful at creating images that closely mimic the properties of the real images.

Second, the comparison of LFID scores provides quantitative support to this observation. The C-VAE model's lower LFID score signifies a smaller statistical distance between the real and generated images, indicating that it has successfully learned a more accurate representation of the original data distribution. Conversely, the C-GAN model's higher LFID score suggests a greater distance, indicating a less accurate representation. Thus, the LFID scores not only substantiate the human perception of image quality but also provide a quantifiable measure of the quality and realism of the images generated by these models.

Adding to the argument of the effectiveness of lower dimensional Fréchet Inception Distance (LFID) over the traditional high-dimensional FID scores, our experimental results bring out an- other crucial aspect. It was observed that images generated through the Conditional Variational Autoencoder (C-VAE) significantly improved the performance of the Optical Character Recognition (OCR) system, as opposed to those generated by the Conditional Generative Adversarial Network (C-GAN), which, interestingly, had an adverse effect on OCR performance.

The generated images from the C-VAE model, associated with lower LFID scores, contributed to enhancing the OCR's ability to recognize and interpret the characters accurately. The generated images effectively enriched the training dataset and facilitated the OCR model in learning a more robust and generalized representation of the digit classes.

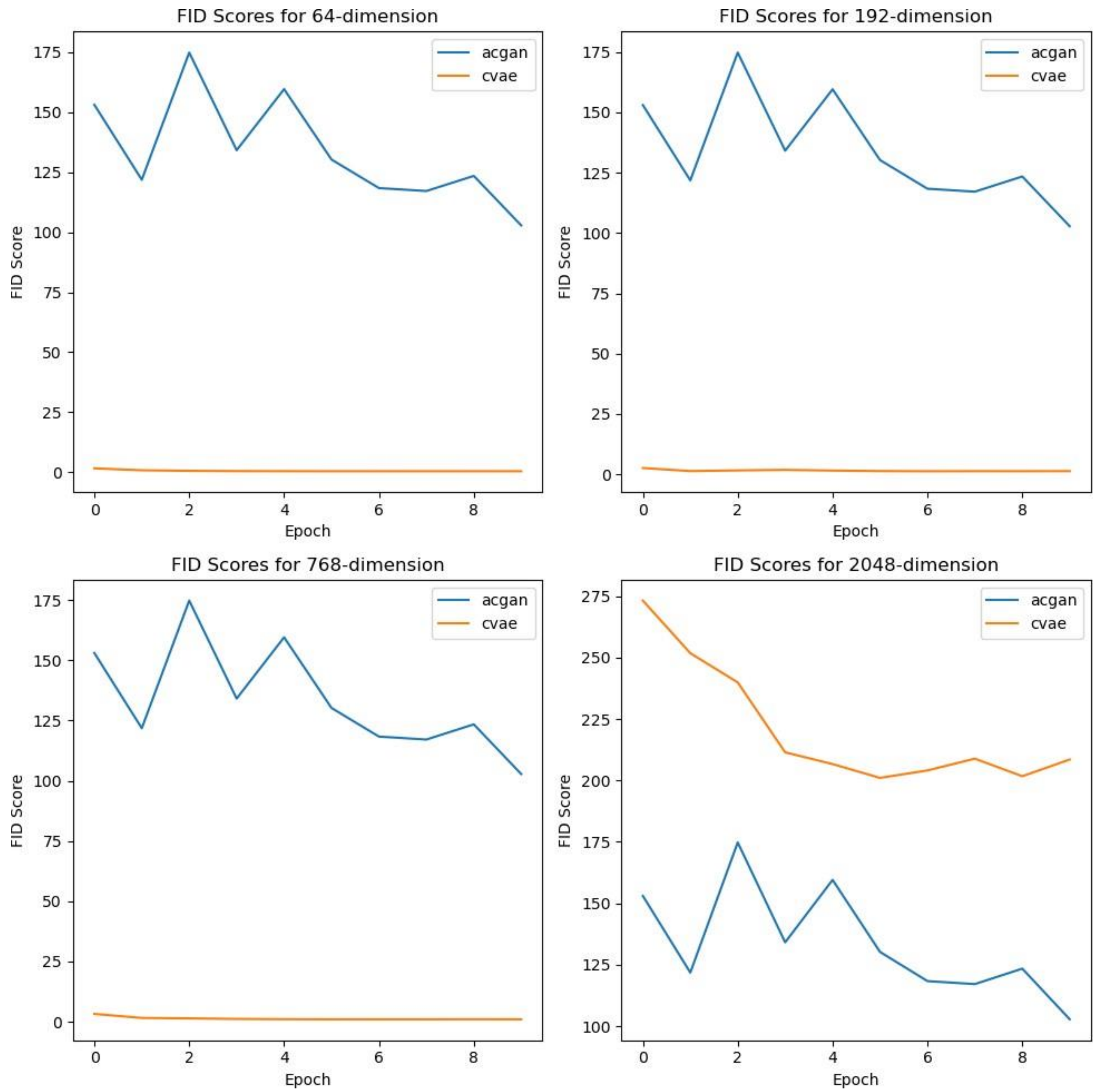

Figure 11: FID Scores

Contrastingly, C-GAN, despite producing sharper images, couldn't contribute positively to OCR performance. C-GAN generated images, which were associated with higher LFID scores, seemed to compromise the OCR's interpretive capacity. This inverse relationship between the LFID scores and the OCR performance reaffirms the credibility of LFID as a more meaningful and insightful measure, especially in lower dimensions, in evaluating the quality and utility of images generated by these generative models.

Therefore, our experimental results add another dimension to the advantages of LFID over high-dimensional FID, emphasizing its effectiveness not just in reflecting the perceptual similarity and the statistical likeness between the generated and real images, but also in predicting the utility of the generated images in improving the performance of downstream tasks, like OCR in this case.

4.8 Saliency Maps Visualization

The results displayed in Figure 12 and 13 highlight the visual interpretation of the learning process our model goes through while performing digit classification, using a technique known as Saliency Maps [111] . These maps illuminate the regions in an image that are most salient or relevant to making a particular classification decision. In our case, the model has to recognize and classify different handwritten Arabic digits [1].

The Saliency Maps thus represent how our model "sees" and interprets these digits, effectively acting as a heatmap of model attentiveness. Bright regions in the Saliency Maps signify areas where small changes in the pixel values significantly affect the classification outcome, denoting high sensitivity. In contrast, darker regions imply areas of lower sensitivity [93].

In the context of digit classification, these maps confirm that the unique, distinguishing attributes of each digit, such as the particular curves, edges, or other specific strokes that differentiate one digit from another, are the key factors influencing the correct identification and classification. These features appear to be the most salient or critical in the eyes of our model [11].

Notably, the successful recognition and highlighting of these unique digit attributes by the Saliency Maps further validate the high accuracy achieved by our Optical Character

Recognition (OCR) model. It was trained on a dataset augmented by the Conditional Variational Autoencoder (C-VAE), as these maps effectively demonstrate that the model correctly focuses on the most important and distinguishing features in its learning process.

This outcome bolsters our confidence in the model's learning process, its understanding of the data, and its subsequent performance. It also provides us with an intuitive way to visually verify and interpret the decision-making process of the model, thereby reaffirming the robustness and reliability of the OCR model's high accuracy.

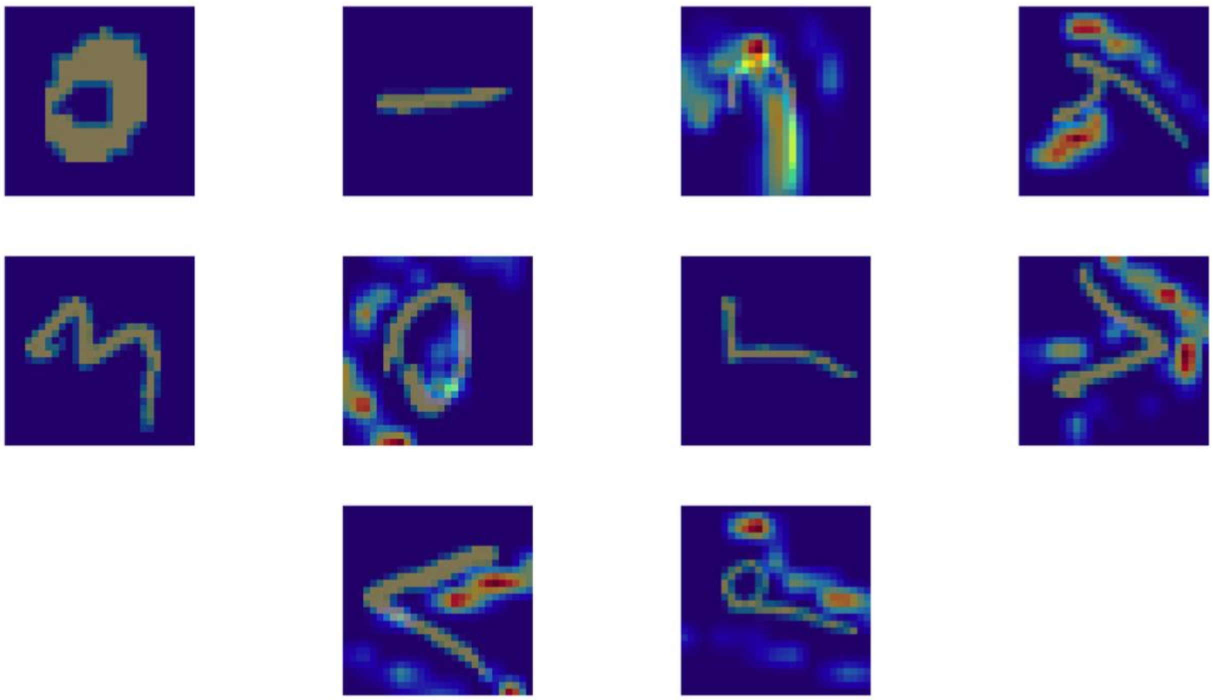

Figure 12: C-GAN Saliency Maps

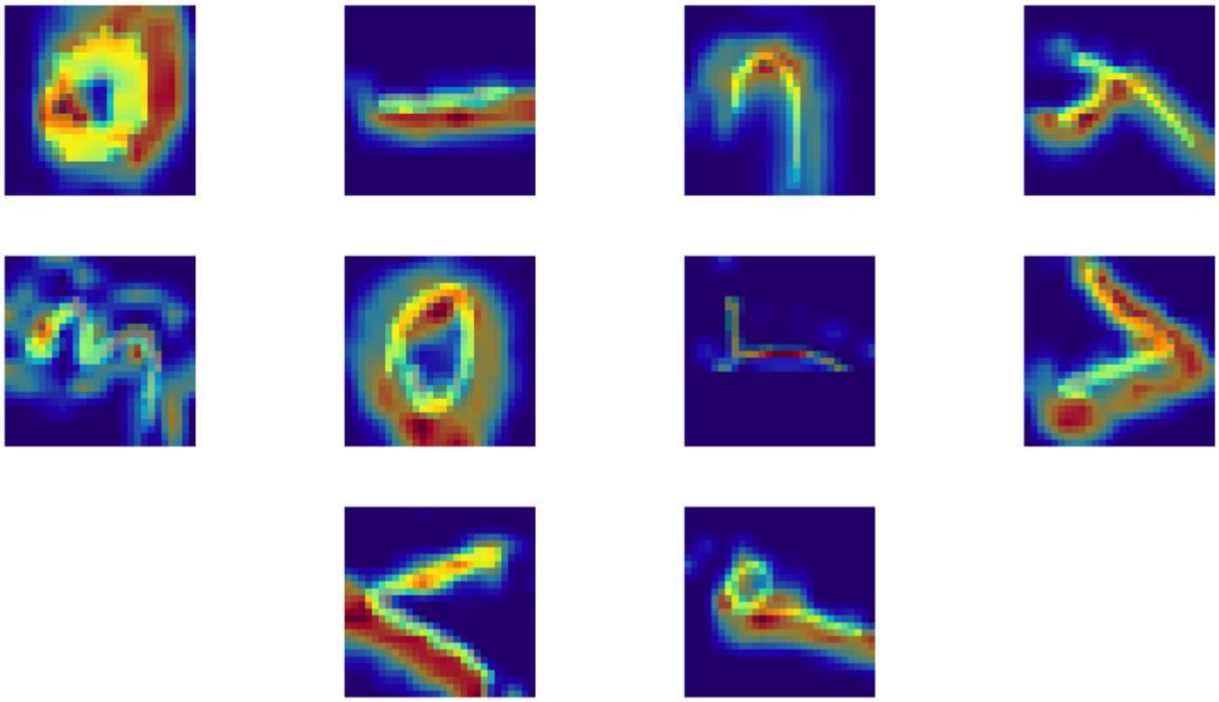

Figure 13: C-VAE Saliency Maps

4.9 Summary

The proposed methodology focuses on optimizing OCR performance for Arabic handwritten digit recognition and examining the impact of synthetic images generated by GAN and VAE models on model accuracy and generalization capabilities. This comprehensive approach includes several steps, such as model compilation, data preprocessing, data augmentation, model training, validation, hyperparameter tuning, and evaluation on the test dataset.

By utilizing the LFID metric to assess the quality of synthetic images and evaluating the OCR model's performance on augmented datasets, the study aims to provide valuable insights into the effectiveness of various generation loss functions and data augmentation strategies. Furthermore, it investigates the correlation between LFID scores and OCR model accuracy, as well as comparing the most critical OCR features with unique features of Arabic handwritten digits.

This in-depth analysis helps identify the most promising techniques for improving OCR performance and reveals the benefits and limitations of different synthetic data generation approaches in the context of OCR tasks. Ultimately, the study contributes to a better understanding of the factors that influence OCR performance and offers guidance for practitioners seeking to develop more accurate and robust OCR models using synthetic data augmentation.

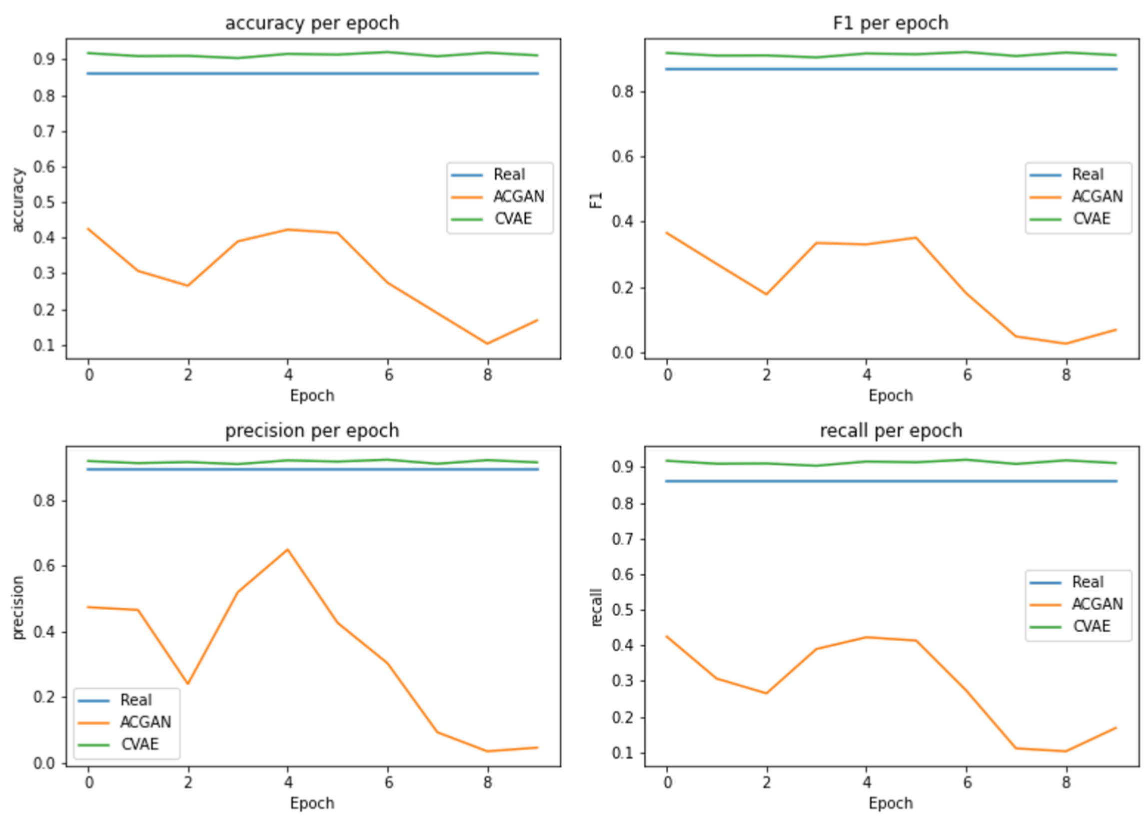

Figure 14: Classification Metrics

CHAPTER 5

CONCLUSION

In conclusion, our study highlighted the advantages of Conditional Variational Autoencoders (C-VAEs) over Conditional Generative Adversarial Networks (C-GANs) for improving real-time OCR performance, especially in the context of Arabic handwritten digit recognition. The VAE model's superior performance can be attributed to its ability to generate synthetic images ten times faster than GANs, resulting in a more efficient training process.

A significant contribution of our research is the development of the Synthetic Image Evaluation Procedure (Algorithm 3), an accurate and efficient alternative to traditional FID scores for monitoring the quality of synthetic images generated for OCR applications. The evaluation procedure enables real-time evaluation of model performance and supports early stopping during training, further optimizing the OCR system.

Our analysis, corroborated by Saliency Maps, validated the improvement in OCR performance. We demonstrated that the enhanced OCR system effectively leverages unique features of Arabic digits for classification, confirming the system identifies and classifies the digits based on their true unique features, rather than relying on irrelevant or trivial patterns.

By integrating generative data augmentation techniques like C-VAEs with innovative evaluation metrics such as LFID, our approach sets the stage for substantial advancements in OCR performance. These improvements are particularly valuable for real-time applications, where challenges such as noise, distortions, and limited training data availability often impede system accuracy. The insights derived from our research have the potential to guide the development of more advanced OCR systems capable of addressing a broader range of applications and adapting to various contexts.

Future work could explore other generative models and data augmentation techniques, as well as the application of our approach to other languages or domains. Additionally, further research could focus on improving the LFID metric, making it more robust and adaptable for different tasks and contexts.

REFERENCES

[1] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems*, vol. 31, 2018.

[2] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *International Journal of Information Management Data Insights*, vol. 1, no. 1, 2021, p. 100004.

[3] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved handwritten digit recognition using convolutional neural networks (CNN)," *Sensors*, vol. 20, no. 12, 2020, p. 3344.

[4] M. S. Ahmed, T. Gonçalves, and H. Sarwar, "Improving Bangla OCR output through correc- tion algorithms," *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*. IEEE, 2016, pp. 338–343.

[5] H. Al-Wzwazy, H. Mohammed Albehadili, Y. Saood Alwan, N. Islam, H. A. Alwzwazy, H. M. Albehadili, Y. S. Alwan, N. E. Islam, and M. Student, "Handwritten Digit Recognition Using Convolutional Neural Networks," *Article in International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3297, no. 2, 2016.

[6] R. S. Alkhawaldeh, M. Alawida, N. F. F. Alshdaifat, W. Z. Alma'aitah, and A. Almasri, "Ensemble deep transfer learning model for Arabic (Indian) handwritten digit recognition," *Neural Computing and Applications*, vol. 34, no. 1, 1 2022, pp. 705–719.

[7] H. Alt and M. Godau, "COMPUTING THE FRÉCHET DISTANCE BETWEEN TWO POLYGONAL CURVES," *International Journal of Computational Geometry {\textback- slash}& Applications*, 1995.

[8] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, 1993, pp. 185–196.

[9] O. D. Apuke, "Quantitative research methods: A synopsis approach," *Kuwait Chapter of Arabian Journal of Business and Management Review*, vol. 33, no. 5471, 2017, pp. 1–8.

[10] D. M. Arroyo, J. Postels, and F. Tombari, "Variational transformer networks for layout generation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13642–13652.

[11] N. Arun, N. Gaw, P. Singh, K. Chang, M. Aggarwal, B. Chen, K. Hoebel, S. Gupta, J. Patel, M. Gidwani, and others, "Assessing the trustworthiness of saliency maps for localizing abnormalities in medical imaging," *Radiology: Artificial Intelligence*, vol. 3, no. 6, 2021, p. e200267.

[12] A. Asperti and M. Trentin, "Balancing reconstruction error and kullback-leibler divergence in variational autoencoders," *IEEE Access*, vol. 8, 2020, pp. 199440–199448.

[13] M. A. Awel and A. I. Abidi, "Review on optical character recognition," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 6, 2019, pp. 3666–3669.

[14] J. Bao, L. Li, and A. Davis, "Variational autoencoder or generative adversarial networks? A comparison of two deep learning methods for flow and transport data assimilation," *Mathematical Geosciences*, vol. 54, no. 6, 2022, pp. 1017–1042.

[15] S. Barratt and R. Sharma, "A Note on the Inception Score," *arXiv preprint arXiv:1801.01973*, 2018.

[16] D. Berchmans and S. S. Kumar, “Optical character recognition: an overview and an insight,” *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. IEEE, 2014, pp. 1361–1365.

[17] J. A. Bilbrey, J. P. Heindel, M. Schram, P. Bandyopadhyay, S. S. Xantheas, and S. Choudhury, “A look inside the black box: Using graph-theoretical descriptors to interpret a Continuous- Filter Convolutional Neural Network (CF-CNN) trained on the global and local minimum energy structures of neutral water clusters,” *The Journal of Chemical Physics*, vol. 153, no. 2, 2020.

[18] J. L. Blue, G. T. Candela, P. J. Grother, R. Chellappa, and C. L. Wilson, “Evaluation of pattern classifiers for fingerprint and OCR applications,” *Pattern Recognition*, vol. 27, no. 4, 1994, pp. 485–501.

[19] S. Bock, J. Goppold, and M. Weiß, “An improvement of the convergence proof of the ADAM-Optimizer,” *arXiv preprint arXiv:1804.10587,* 2018.

[20] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, “Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregres- sive Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, 2021, pp. 7327–7347.

[21] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, “In- tegrating structured biological data by kernel maximum mean discrepancy,” *Bioinformatics*, vol. 22, no. 14, 2006, pp. e49–e57.

[22] A. K. Boyat and B. K. Joshi, “A review paper: noise models in digital image processing,” *arXiv preprint arXiv:1505.03489,* 2015.

[23] M. Brisinello, R. Grbić, M. Pul, and T. An\djelić, “Improving optical character recognition performance for low quality images,” *2017 International Symposium ELMAR*. IEEE, 2017, pp. 167–171.

[24] J. Cai, L. Peng, Y. Tang, C. Liu, and P. Li, “TH-GAN: generative adversarial network based transfer learning for historical Chinese character recognition,” *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 178–183.

[25] L. Cai, H. Gao, and S. Ji, “Multi-stage variational auto-encoders for coarse-to-fine image generation,” *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019, pp. 630–638.

[26] Q. Cai, M. Abdel-Aty, J. Yuan, J. Lee, and Y. Wu, “Real-time crash prediction on expressways using deep generative models,” *Transportation research part C: emerging technologies*, vol. 117, 2020, p. 102697.

[27] K. R. Castleman, *Digital image processing,* Prentice Hall Press, 1996.

[28] J. D. B. Castro, S. W. A. Canchumuni, C. E. M. Villalobos, F. C. Cordeiro, A. M. A. Alexandre, and M. A. C. Pacheco, “Improvement optical character recognition for structured documents using generative adversarial networks,” *2021 21st International Conference on Computational Science and Its Applications (ICCSA)*. IEEE, 2021, pp. 285–292.

[29] I. Cetin, M. Stephens, O. Camara, and M. A. G. Ballester, “Attri-VAE: Attribute-based in- terpretable representations of medical images with variational autoencoders,” *Computerized Medical Imaging and Graphics*, vol. 104, 2023, p. 102158.

[30] X. Chang, F. Chao, C. Shang, and Q. Shen, “Sundial-gan: A cascade generative adversarial networks framework for deciphering oracle bone inscriptions,” *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 1195–1203.

[31] P. K. Charles, V. Harish, M. Swathi, and C. H. Deepthi, "A review on the various techniques used for optical character recognition," *International Journal of Engineering Research and Applications*, vol. 2, no. 1, 2012, pp. 659–662.

[32] A. Chaudhuri, K. Mandaviya, P. Badelia, S. K Ghosh, A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K. Ghosh, *Optical character recognition systems*, Springer, 2017.

[33] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional neural network (CNN) for image detection and recognition," *2018 first international conference on secure cyber computing and communication (ICSCCC)*. IEEE, 2018, pp. 278–282.

[34] S. Chen and W. Guo, "Auto-Encoders in Deep Learning—A Review with New Perspectives," *Mathematics*, vol. 11, no. 8, 2023, p. 1777.

[35] Z. Chen, C. Chen, X. Jin, Y. Liu, and Z. Cheng, "Deep joint two-stream Wasserstein auto- encoder and selective attention alignment for unsupervised domain adaptation," *Neural computing and applications*, vol. 32, 2020, pp. 7489–7502.

[36] W. Cheng, G. Darnell, S. Ramachandran, and L. Crawford, "Generalizing Variational Autoencoders with Hierarchical Empirical Bayes," *arXiv preprint arXiv:2007.10389*, 2020.

[37] Y. S. Chernyshova, A. V. Gayer, and A. V. Sheshkus, "Generation method of synthetic training data for mobile OCR system," *Tenth international conference on machine vision (ICMV 2017)*. SPIE, 2018, vol. 10696, pp. 640–646.

[38] R. Child, "Very deep vaes generalize autoregressive models and can outperform them on images," *arXiv preprint arXiv:2011.10650*, 2020.

[39] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending MNIST to handwritten letters," *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2921–2926.

[40] J. E. Contreras-Reyes and R. B. Arellano-Valle, "Kullback–Leibler divergence measure for multivariate skew-normal distributions," *Entropy*, vol. 14, no. 9, 2012, pp. 1606–1626.

[41] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, 2018, pp. 53–65.

[42] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.

[43] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Ieee, 2005, vol. 1, pp. 886–893.

[44] D. Doermann and K. Tombre, *Handbook of document image processing and recognition*, Springer Publishing Company, Incorporated, 2014.

[45] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[46] F. Duffhauss, N. A. Vien, H. Ziesche, and G. Neumann, "FusionVAE: A Deep Hierarchical Variational Autoencoder for RGB Image Fusion," *European Conference on Computer Vision*. Springer, 2022, pp. 674–691.

[47] S. R. Eddy, "Hidden markov models," *Current opinion in structural biology*, vol. 6, no. 3, 1996, pp. 361–365.

[48] M. El-Kaddoury, A. Mahmoudi, and M. M. Himmi, "Deep generative models for image generation: A practical comparison between variational autoencoders and generative ad-

versarial networks," *Mobile, Secure, and Programmable Networking: 5th International Conference, MSPN 2019, Mohammedia, Morocco, April 23–24, 2019, Revised Selected Papers 5*. Springer, 2019, pp. 1–8.

[49] E. A. El-Sherif and S. Abdelazeem, "A Two-Stage System for Arabic Handwritten Digit Recognition Tested on a New Large Database.," *Artificial intelligence and pattern recogni- tion*, 2007, pp. 237–242.

[50] I. Fasel, B. Fortenberry, and J. Movellan, "A generative framework for real time object detection and classification," *Computer Vision and Image Understanding*, vol. 98, no. 1, 2005, pp. 182–210.

[51] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data aug- mentation using GAN for improved liver lesion classification," *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 2018, pp. 289–293.

[52] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," *Class project for Stanford CS231N: convolutional neural networks for visual recognition, Winter semester*, vol. 2014, no. 5, 2014, p. 2.

[53] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Generative adversarial networks and adversarial autoencoders: Tutorial and survey," *arXiv preprint arXiv:2111.13282*, 2021.

[54] L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber, and X. Alameda-Pineda, "Dynamical variational autoencoders: A comprehensive review," *arXiv preprint arXiv:2008.12595*, 2020.

[55] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, 2014, vol. 27, pp. 2672–2680.

[56] M. Guan, H. Ding, K. Chen, and Q. Huo, "Improving Handwritten OCR with Augmented Text Line Images Synthesized from Online Handwriting Samples by Style-Conditioned GAN," *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, vol. 2020-Septe, 9 2020, pp. 151–156.

[57] S. Har-Peled and B. Raichel, "The Fréchet distance revisited and extended," *ACM Transac- tions on Algorithms (TALG)*, vol. 10, no. 1, 2014, pp. 1–22.

[58] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, 2017, pp. 6627–6638.

[59] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE access*, vol. 8, 2019, pp. 4806–4813.

[60] H. Ibrahim, O. M. Fahmy, and M. A. Elattar, "License plate image analysis empowered by generative adversarial neural networks (GANs)," *IEEE Access*, vol. 10, 2022, pp. 30846– 30857.

[61] N. Islam, Z. Islam, and N. Noor, "A survey on optical character recognition system," *arXiv preprint arXiv:1710.05703*, 2017.

[62] S. M. Ismail, S. N. H. S. Abdullah, and S. Norul, "Online Arabic handwritten character recognition based on a rule based approach," *Journal of Computer Science*, vol. 8, no. 11, 1859, p. 2012.

[63] M. Itkina, B. Ivanovic, R. Senanayake, M. J. Kochenderfer, and M. Pavone, "Evidential sparsification of multimodal latent spaces in conditional variational autoencoders," *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 10235–10246.

[64] B. Jähne, *Digital image processing*, Springer Science & Business Media, 2005.

[65] A. K. Jain, *Fundamentals of digital image processing*, Prentice-Hall, Inc., 1989.

[66] M. J. Johnson, D. Duvenaud, A. B. Wiltschko, S. R. Datta, and R. P. Adams, "Structured VAEs: Composing probabilistic graphical models and variational autoencoders," *arXiv preprint arXiv:1603.06277*, vol. 2, 2016, p. 2016.

[67] S. Jung and M. Keuper, "Internalized biases in fréchet inception distance," *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[68] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, and others, "ICDAR 2015 competition on robust reading," *2015 13th international conference on document analysis and recognition (ICDAR)*. IEEE, 2015, pp. 1156–1160.

[69] A. Kebaili, J. Lapuyade-Lahorgue, and S. Ruan, "Deep Learning Approaches for Data Augmentation in Medical Imaging: A Review," *Journal of Imaging*, vol. 9, no. 4, 2023, p. 81.

[70] S. U. Khan, I. Ullah, F. Khan, Y. Lee, and S. Ullah, "Historical Text Image Enhancement Using Image Scaling and Generative Adversarial Networks," *Sensors*, vol. 23, no. 8, 2023, p. 4003.

[71] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," *International Conference on Machine Learning*. PMLR, 2021, pp. 5530–5540.

[72] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2014.

[73] D. P. Kingma and M. Welling, *An Introduction to Variational Autoencoders*, vol. 12, Now Publishers, Inc., 2019.

[74] I. Kissos and N. Dershowitz, "OCR error correction using character correction and feature- based word classification," *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*. IEEE, 2016, pp. 198–203.

[75] B. Köhler, J. Haladjian, B. Simeonova, and D. Ismailović, "Feedback in low vs. high fidelity visuals for game prototypes," 2012 Second International Workshop on Games and Software Engineering: Realizing User Engagement with Game Engineering Techniques (GAS). IEEE, 2012, pp. 42–47.

[76] O. Kolak, B. Byrne, and P. Resnik, "A generative probabilistic OCR model for NLP applications," Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, 2003, pp. 134– 141.

[77] S. Kompalli, S. Nayak, S. Setlur, and V. Govindaraju, "Challenges in OCR of Devanagari documents," *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 2005, pp. 327–331.

[78] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10," *Unpub- lished manuscript*, vol. 40, no. 7, 2010, pp. 1–9.

[79] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolu- tional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, 2012, pp. 1097–1105.

[80] V. Kushwaha, G. C. Nandi, and others, “Study of prevention of mode collapse in generative adversarial network (GAN),” *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*. IEEE, 2020, pp. 1–6.

[81] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, 5 2015, pp. 436–444.

[82] J. Lim, S. Ryu, J. W. Kim, and W. Y. Kim, “Molecular generative model based on conditional variational autoencoder for de novo molecular design,” *Journal of cheminformatics*, vol. 10, no. 1, 2018, pp. 1–9.

[83] T. Lin, S. U. Stich, K. K. Patel, and M. Jaggi, “Don’t use large mini-batches, use local sgd,” *arXiv preprint arXiv:1808.07217*, 2018.

[84] W. Liu, J. Cao, Y. Zhu, B. Liu, and X. Zhu, “Real-time anomaly detection on surveillance video with two-stream spatio-temporal generative model,” *Multimedia systems*, vol. 29, no. 1, 2023, pp. 59–71.

[85] Y. Liu, H. Jiang, Y. Wang, Z. Wu, and S. Liu, “A conditional variational autoencoding generative adversarial networks with self-modulation for rolling bearing fault diagnosis,” *Measurement*, vol. 192, 2022, p. 110888.

[86] M. L. Menéndez, J. A. Pardo, L. Pardo, and M. C. Pardo, “The jensen-shannon divergence,” *Journal of the Franklin Institute*, vol. 334, no. 2, 1997, pp. 307–318.

[87] L. Mescheder, S. Nowozin, and A. Geiger, “Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks,” *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 2391–2400.

[88] S. Mežnar, S. Džeroski, and L. Todorovski, “Efficient generator of mathematical expressions for symbolic regression,” *arXiv preprint arXiv:2302.09893*, 2023.

[89] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” *Technical report*, 11 2014.

[90] R. Mithe, S. Indalkar, and N. Divekar, “Optical character recognition,” *International journal of recent technology and engineering (IJRTE)*, vol. 2, no. 1, 2013, pp. 72–75.

[91] R. Mittal and A. Garg, “Text extraction using OCR: a systematic review,” *2020 second international conference on inventive research in computing applications (ICIRCA)*. IEEE, 2020, pp. 357–362.

[92] M. Moayeri, P. Pope, Y. Balaji, and S. Feizi, “A comprehensive study of image classification model sensitivity to foregrounds, backgrounds, and visual attributes,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19087– 19097.

[93] F. Moosmann, D. Larlus, and F. Jurie, “Learning saliency maps for object categorization,” *International Workshop on The Representation and Use of Prior Knowledge in Vision (in ECCV’06)*, 2006.

[94] S. Mori, C. Y. Suen, and K. Yamamoto, “Historical review of OCR research and develop- ment,” *Proceedings of the IEEE*, vol. 80, no. 7, 1992, pp. 1029–1058.

[95] M. Namysl and I. Konya, “Efficient, lexicon-free OCR using deep learning,” *2019 international conference on document analysis and recognition (ICDAR)*. IEEE, 2019, pp. 295–301.

[96] P. Naveen, “End-to-End Training of VAE-GAN Network for Text Detection,” 2023.

[97] T. T. H. Nguyen, A. Jatowt, M. Coustaty, and A. Doucet, “Survey of post-OCR processing approaches,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, 2021, pp. 1–37.

[98] W. Niblack, *An introduction to digital image processing*, Strandberg Publishing Company, 1985.

[99] W. S. Noble, “What is a support vector machine?,” *Nature biotechnology*, vol. 24, no. 12, 2006, pp. 1565–1567.

[100] C. Olah, A. Mordvintsev, and L. Schubert, “Feature visualization,” Distill, vol. 2, no. 11, 2017, p. e7.

[101] B. Pang, E. Nijkamp, and Y. N. Wu, “Deep learning with tensorflow: A review,” *Journal of Educational and Behavioral Statistics*, vol. 45, no. 2, 2020, pp. 227–248.

[102] V. Pondenkandath, M. Alberti, M. Diatta, R. Ingold, and M. Liwicki, “Historical document synthesis with generative adversarial networks,” *2019 International Conference on Docu- ment Analysis and Recognition Workshops (ICDARW)*. IEEE, 2019, vol. 5, pp. 146–151.

[103] M. Prakash, A. Krull, and F. Jug, “Fully unsupervised diversity denoising with convolutional variational autoencoders,” *arXiv preprint arXiv:2006.06072*, 2020.

[104] N. V. Rao, A. Sastry, A. S. N. Chakravarthy, and P. Kalyanchakravarthi, “OPTICAL CHARACTER RECOGNITION TECHNIQUE ALGORITHMS.,” *Journal of Theoretical & Applied Information Technology*, vol. 83, no. 2, 2016.

[105] M. Razghandi, H. Zhou, M. Erol-Kantarci, and D. Turgut, “Variational autoencoder gener- ative adversarial network for Synthetic Data Generation in smart home,” *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 4781–4786.

[106] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, “Stabilizing training of generative adversarial networks through regularization,” *Advances in neural information processing systems*, vol. 30, 2017.

[107] N. Sarika, N. Sirisala, and M. S. Velpuru, “CNN based optical character recognition and applications,” *2021 6th International conference on inventive computation technologies (ICICT)*. IEEE, 2021, pp. 666–672.

[108] A. E. Sawy, H. El-bakry, M. Loey, and H. El-bakry, “CNN for handwritten arabic digits recognition based on LeNet-5,” *Advances in Intelligent Systems and Computing*, vol. 533, 2017, pp. 565–575.

[109] G. Sharma, F. Jurie, and C. Schmid, “Discriminative spatial saliency for image classifica- tion,” *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3506–3513.

[110] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,” *Physica D: Nonlinear Phenomena*, vol. 404, 2020, p. 132306.

[111] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visu- alising image classification models and saliency maps,” *arXiv preprint arXiv:1312.6034*, 2013.S. Singh, “Optical character recognition techniques: a survey,” *Journal of emerging Trends in Computing and information Sciences*, vol. 4, no. 6, 2013.

[112] K. Sohn, H. Lee, and X. Yan, “Learning Structured Output Representation using Deep Conditional Generative Models,” *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[113] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder variational autoencoders," *Advances in neural information processing systems*, vol. 29, 2016.

[114] M. A. Souibgui and Y. Kessentini, "De-gan: A conditional generative adversarial network for document enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, 2020, pp. 1180–1191.

[115] D. Sporici, E. Cuşnir, and C.-A. Boiangiu, "Improving the accuracy of Tesseract 4.0 OCR engine using convolution-based preprocessing," *Symmetry*, vol. 12, no. 5, 2020, p. 715.

[116] V. Storchan and J. Beauschene, "Data augmentation via adversarial networks for optical character recognition/conference submissions," *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 184–189.

[117] X. Su, H. Xu, Y. Kang, X. Hao, G. Gao, and Y. Zhang, "Improving text image resolution using a deep generative adversarial network for optical character recognition," *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 9 2019, pp. 1193–1199.

[118] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *IEEE conference on computer vision and pattern recognition.*, 9 2015.

[119] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[120] K. Taghva, J. Borsack, and A. Condit, "Evaluation of model-based retrieval effectiveness with OCR text," *ACM Transactions on Information Systems (TOIS)*, vol. 14, no. 1, 1996, pp. 64–93.

[121] W. Tang, J. Su, and Y. Gao, "Hardware Trojan Detection Method Based on Dual Discrimi- nator Assisted Conditional Generation Adversarial Network," *Journal of Electronic Testing*, 2023, pp. 1–12.

[122] C. Tian, C. Li, G. Zhang, and Y. Lv, "Data driven parallel prediction of building energy consumption using generative adversarial nets," *Energy and Buildings*, vol. 186, 2019, pp. 230–243. A. Tkach, M. Pauly, and A. Tagliasacchi, "Sphere-meshes for real-time hand modeling and tracking," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, 2016, pp. 1–11.

[123] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," *arXiv preprint arXiv:1711.01558*, 2017.

[124] O. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition-a survey," *Pattern recognition*, vol. 29, no. 4, 1996, pp. 641–662.

[125] M. S. Uddin, R. Pamie-George, D. Wilkins, A. Sousa-Poza, M. Canan, S. Kovacic, and J. Li, "Ship Deck Segmentation In Engineering Document Using Generative Adversarial Networks," *2022 IEEE World AI IoT Congress (AIIoT)*. IEEE, 2022, pp. 207–212.

[126] A. Ulhaq, N. Akhtar, and G. Pogrebna, "Efficient diffusion models for vision: A survey," *arXiv preprint arXiv:2210.09292*, 2022.

[127] A. Vahdat and J. Kautz, "NVAE: A deep hierarchical variational autoencoder," *Advances in neural information processing systems*, vol. 33, 2020, pp. 19667–19679.

[128] S. S. Vallender, “Calculation of the Wasserstein distance between probability distributions on the line,” *Theory of Probability & Its Applications*, vol. 18, no. 4, 1974, pp. 784–786.
[129] Y. Wang, F. Su, and Y. Qian, “Text-attentional conditional generative adversarial network for super-resolution of text images,” *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 1024–1029.
[130] Y. Wang, Y. Wei, and H. Wang, “A class imbalanced wafer defect classification framework based on variational autoencoder generative adversarial network,” *Measurement Science and Technology*, vol. 34, no. 2, 2022, p. 24008.
[131] Y. Wu and P. Natarajan, “Self-organized text detection with minimal post-processing via border learning,” *proceedings of the IEEE international conference on computer vision*, 2017, pp. 5000–5009.
[132] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, “Improved variational autoen- coders for text modeling using dilated convolutions,” *International conference on machine learning*. PMLR, 2017, pp. 3881–3890.
[133] Y. Yao, L. Rosasco, and A. Caponnetto, “On early stopping in gradient descent learning,” *Constructive Approximation*, vol. 26, 2007, pp. 289–315.
[134] Y. Yu, W. Zhang, and Y. Deng, “Frechet inception distance (fid) for evaluating gans,” *China University of Mining Technology Beijing Graduate School: Beijing, China*, 2021.
[135] D. Zeng, Categorizing the dispersion of open chromatin usage patterns of Treg related genes using an variational autoencoder based algorithm, doctoral dissertation, Harvard Medical School, 2023.
[136] Z. Zhang, “Improved adam optimizer for deep neural networks,” *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*. Ieee, 2018, pp. 1–2.

VITA

Graduate School

Southern Illinois University Carbondale

Majid Memari

memari@siu.edu

Southern Illinois University Carbondale

Master of Science, Computer Science, May 2018

Dissertation Title: A New Approach to Synthetic Image Evaluation

Major Professor: Dr. Khaled R Ahmed