A Data Augmentation Approach using Style-Based Generative Adversarial Networks for Date Fruit Classification

by

Donald Ehikioya Ufuah

A Thesis submitted to the Faculty of

Graduate Studies of The University of

Manitoba

In partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Canada

Copyright © 2022 by Ufuah Donald Ehikioya

EXAMINING COMMITTEE

This thesis was examined and approved by the following examining committee on April 27, 2022.

- **Prof. Gabriel Thomas** (advisor)
  Department of Electrical & Computer Engineering
  University of Manitoba

- **Prof. Ahmed Ashraf**
  Department of Electrical & Computer Engineering
  University of Manitoba

- **Prof. Manickavasagan Annamalai**
  Department of Biological Engineering
  University of Guelph

- **Prof. Zahra Moussavi**
  Department of Electrical & Computer Engineering
  University of Manitoba

ABSTRACT

Hardness is important for ascertaining the quality of dried fruits such as dates. In the dates sorting facilities, hardness is determined by human visual inspection resulting in the high accuracy of the sorting process. However, this process is slow, tedious, subjective, and unhygienic. Machine learning techniques have been adopted in agricultural applications for automatic sorting processes and in some scenarios, a large amount of data may not be available for training Deep Learning classification models. Such is the case presented here for sorting dates based on their hardness. The original dataset in this work consists of 1800 images with 600 images per class for hard, semi-hard, and soft classes obtained from different growing regions in Oman. This is a limited number of examples available to improve the classification accuracy of deep learning classification networks. Thus, this thesis proposes data augmentation based on Generative Adversarial Networks (GAN) (StyleGAN2-ADA) for generating date images. It incorporates Residual Neural Networks (ResNet18) for date classification. The goal is to use the original dataset to further generate enough images so that ResNet18 can successfully classify them into their respective classes. As the original dataset can hardly be considered large, the GAN images help during the training part. Utilizing the new dataset from the GAN generation process achieved accuracies of 93%, 99% and 53% for the hard, soft and semi-hard classes by using an expert system in combination with two ResNet18 networks, one network trained for classifying hard and soft cases as well as a second network that classifies the three classes.

By extension, texture features were extracted from the original and generated images and compared using a Kolmogorov-Smirnov test measuring the similarity of the probability density distributions of the extracted texture features. The similarity for hard, semi-hard and soft classes are 82.86%, 100% and 94.29% respectively. This step further validated the GAN generated images indicating that they were of merit.

Index Terms—**Data augmentation, fruit classification, deep learning, expert system, GAN, StyleGAN, neural networks, Kolmogorov test.**

ACKNOWLEDGEMENTS

My outmost gratitude goes to God almighty, my creator and the pilot of my life for his wisdom, favor, and grace upon me. My sincere appreciation is expressed to my advisor Dr. Gabriel Thomas for his guidance, support, and constant encouragement during my M.Sc. program.

I would also like to express my gratitude to my thesis committee, Dr. Ahmed Ashraf, Dr. Zahra Moussavi, and Dr. Manickavasagan Annamalai for their suggestions and comments for improving my thesis.

I sincerely appreciate the financial support from the faculty of graduate studies through the University of Manitoba Graduate Fellowship (UMGF) Award.

To the graduate program advisors of the department of Electrical and Computer Engineering, Samuel Kidd as well as Heather Katrick and Amy Dario of the department of Biomedical Engineering, I am super grateful for your support all through my M.Sc. program.

In conclusion, my heartfelt gratitude goes to my parents, Ufuah Godwin and Ufuah Philomena and my siblings, Ufuah Pamela, Ufuah Kelvin and Ufuah Louisa for their constant support, prayers and encouragement. Without the scarifies of my parents I would have never made it through to Canada.

To my dear friends Benita Akolo and Dawn Idoko, your endless words of encouragement and support are greatly appreciated.

DEDICATION

This thesis is dedicated to God almighty and my lovely parents and siblings who have supported me from inception.

TABLE OF CONTENTS

EXAMINING COMMITTEE....1
ABSTRACT....2
ACKNOWLEDGEMENTS....3
DEDICATION....4
TABLE OF CONTENTS....5
LIST OF FIGURES....7
LIST OF TABLES....8
LIST OF ABBREVIATIONS....9
CHAPTER 1....11
INTRODUCTION....11
1.1 BACKGROUND....11
1.2 OBJECTIVE OF THE THESIS....13
1.3 THESIS OUTLINE....14
CHAPTER 2....15
LITERATURE REVIEW AND INTRODUCTION TO THE TECHNIQUES USED IN THIS WORK....15
2.1 DATE FRUIT CLASSIFICATION....15
2.2 GENERATIVE ADVERSARIAL NETWORKS....17
2.2.1 GAN EVALUATION METRICS....20
2.2.2 GAN REGULARIZATION....21
2.2.3 STYLE TRANSFER....22
2.2.4 STYLE GAN BASED GENERATOR NETWORK....25
2.2.5 STYLE GAN BASED DISCRIMINATOR NETWORK....26
2.3 RESIDUAL NEURAL NETWORKS....27
2.4 EXPERT SYSTEM....29
2.5 VALIDATION OF THE GAN IMAGES....31
2.5.1 EXTRACTION OF TEXTURE FEATURES....31
2.5.2 HISTOGRAM OF ORIENTED GRADIENTS (HOG)....32
2.5.3 SEMIVARIOGRAMS....32
2.5.4 TAMURA....33

2.6 TWO SAMPLE KOLMOGOROV-SMIRNOV TEST 34

CHAPTER 3 36

METHODOLOGY 36

3.1 PROPOSED FRAMEWORK 36

CHAPTER 4 39

DATASET 39

4.1 ORIGINAL AND GENERATED DATASETS 39

CHAPTER 5 41

EXPERIMENTAL RESULTS AND DISCUSSION 41

CHAPTER 6 52

6.1 CONCLUSION 52

6.2 FUTURE WORK 52

6.3 RECOMMENDATIONS 52

REFERENCES 54

LIST OF FIGURES

Figure 1. Block diagram of Generative Adversarial Networks. 20
Figure 2. Architecture of a Traditional GAN Generator Network.....25
Figure 3. Architecture of a StyleGAN2-ADA Generator Network. 26
Figure 4. Architecture of a StyleGAN2-ADA Discriminator Network. 27
Figure 5. Skip connection of ResNet18 architecture. 28
Figure 6. Descriptive block diagram of ResNet18 architecture. 29
Figure 7. Block diagram of the expert system. 31
Figure 8. Block diagram for validating the GAN generated images. 35
Figure 9. (a) Training accuracies. (b) Colormaps tested. Note how the Jet colormap trains faster than the other ones. 36
Figure 10. First row: Original images, Second row: Images after pseudo-coloring using the Jet colormap. First column: hard dates, second column: semi-hard date, third column: soft date. 37
Figure 11. Block diagram of the proposed system. 38
Figure 12. First row: Original images, Second row: GAN generated images. First column: hard dates, second column: semi-hard date, third column: soft date. 40
Figure 13. Training Confusion matrix for 2-Class ResNet18 Model. 43
Figure 14. Training Confusion matrix for 3-Class ResNet18 Model. 44
Figure 15. Test Confusion matrix for 2 class ResNet18 Model. 44
Figure 16. Test Confusion matrix for 3-class ResNet18 model. 45
Figure 17 Confusion matrix of the 18 images out of the 300 that are evaluated at part C of the expert system. 46
Figure 18. (a) Confusion matrix obtained using the expert system and (b) results when using manual inspection and classification of 18 dates. 46
Figure 19. Comparison of the existing methods for the classification of dates based on hardness. 47
Figure 20. Convergence graph of accuracy and loss function for 2-Class model using ResNet18 for 50 epochs. 48
Figure 21. Convergence graph of accuracy and loss function for 3-Class model using ResNet18 for 100 epochs 49
Figure 22. ROC Characteristics Curve of ResNet18 for 2-Class Date Image Classification. 50
Figure 23. ROC Characteristics Curve of ResNet18 for 3-Class Date Image Classification. 50

Figure 24. Plots of F1,n (blue), F2,n (orange) and Dm,n (black line) for the Tamura features. 51

LIST OF TABLES

Table I. Comparison of training and testing accuracy of Resnet18 using standard augmentation and GAN augmentation.....41
Table II. Summary of ResNet18 hyperparameters for training. 42
Table III. Hyperparameters for training the Style-GAN 2 ADA model. 43
Table IV. Performance metrics of the ResNet18 architecture. 49
Table V. Similarity of the Original and GAN generated images based on the KS Test.....51

LIST OF ABBREVIATIONS

GANs: Generative Adversarial Networks

ResNet: Residual Neural Network

ADA: Adaptive Discriminator Augmentation

KS test: Kolmogorov Smirnov test

ReLU: Rectified Linear Unit

FID: Fréchet Inception Distance

FFHQ: Flicker-Faces-HQ

HOG: Histogram of Oriented Gradients

GLCM: Gray Level Co-occurrence Matrix

CDFs: Cumulative Density Functions

IS: Inception Score

KID: Kernel Inception Distance

MS: Mode Score

P&R: Precision and Recall

CNN: Convolutional Neural Networks

IN: Instance Normalization

BN: Batch Normalization

AdaIN: Adaptive Instance Normalization

PPL: Perceptual Path Length

GP: Gradient Penalty

CR: Consistency Regularization

bCR: Balanced Consistency Regularization

zCR: Latent Consistency Regularization

VAE: Variational Autoencoder

DNN: Deep Neural Network

PPV: Positive Predictive Value

FAO: Food and Agricultural Organization of the United States

CHAPTER 1

Introduction

1.1 Background

Recent advances in the field of machine learning and computer vision have been successfully introduced to a myriad of applications in several industries. Some of these applications include autonomous driving vehicles, image generation and manipulation, face recognition, analysis of video surveillance utilized in forensics, robotics, medical diagnosis etc. [1, 2].
In the agricultural sector, the technology is utilized in several applications such as the design of robots for automatic harvesting purposes, automatic sorting and classification of fruits, livestock monitoring, smart farming, smart irrigation, anomaly detection of plant disease etc. [3]. This thesis explores the possibility of using a generative model, Generative Adversarial Networks (GANs), as a data augmentation technique for improving the classification of date fruits based on hardness; an important characteristic of this fruit, as it is a measure for ascertaining the quality of dried fruits. The thesis is projected to have an impact on the classification of date images and by extension dried fruits based on hardness as manually selecting them can be a tedious task. In particular, the original dataset has proven to be quite difficult to classify as it has been reported by researchers in Sultan Qaboos University in Oman. As a result, the thesis is aimed at applying some of the newest techniques to improve the sorting process.

Machine learning has helped agricultural professionals make informed decisions in the area of produce inspection, preservation and transportation. As the quality of fresh fruits plays a critical role to the consumer purchase decision, multinational technology companies such as Amazon utilize computer vision techniques for sorting fruits and vegetables using detection systems incorporated with cameras and sensors [4]. The system is capable of predicting the sweetness of fruits and vegetables and able to infer whether or not they will go bad. Similarly, the Food and Agricultural Organization of the United States (FAO), utilizes artificial intelligence and remote sensing technologies for the protection and conservation of the life span of palm trees such as dates by monitoring and controlling pest such as red palm weevil that destroy palm trees [5]. In addition, mobile applications such as SusaHamra have been developed and deployed to farmers for data acquisition during palm inspection [5].

Date Fruit (Phoenix dactylifera) are rich in calories and have a long-life span. According to the FAO, it is one of the oldest staple fruits that have been cultivated in the Middle East and North Africa for over 5000 years [6], with 91% production of world date emerging from these regions [7]. Egypt, Saudi Arabia, Islamic Republic of Iran, and Algeria in 2019 were the top producers and accounted for an average of 1396863 tons of world date production [6]. Conversely, between 1961 and 2019, the United States of America accounted for an average production of 22059 tons with the highest production of 55700 tons in 2019. Dates are often served as dessert, and it plays an important role in religious and traditional celebrations.
Fresh Dates contain a rich amount of vitamin C and are beneficial to both adults and children of all ages. As the dates loose moisture content and become dry, the vitamin level tends to decrease [33]. However, it is still nutritious and contains Vitamin A, Vitamin k, Vitamin B2, Vitamin B3, Carbohydrates, Sugar, Calcium, Iron, Potassium, Folate, Thiamine and Fiber [8, 9, 10]. As a result of its rich contents, it can potentially alleviate hunger and provide food security thereby achieving the United Nation's Sustainable Development Goal (SDG) 2 [11].

Date fruit are of different types. For example, in Saudi Arabia, in the province of Al-Qassim and Al-Hasa, the Sukkary date and the Khalas dates are the most popular respectively. Ajwa date is famous in Al-Madinah Al-Munawara region and the entire Kingdom of Saudi Arabia (KSA) [8, 10]. Dates, however, are posed with the challenge of misclassification in automatic sorting processes utilized in agricultural industries. Selecting manually the date fruit by human inspection could be unhygienic, inconsistent, expensive and a tedious process prone to human errors. A mechanical process to assess the hardness could have a destructive effect on the date samples and as a result, utilizing computer vision techniques can be an alternative for the sorting process. Besides hardness, dates are graded based on other classification criteria such as shape, type, size, surface defect, maturity level, weight and color.

Deep learning, a type of representation learning [12] for automatically extracting features, has gained much attention in the field of machine learning as there are millions of sensors all over the word doing data acquisition. In the agricultural sector, such data could be limited, difficult to acquire and sometimes unavailable for training deep learning models for robust agricultural applications. Data augmentation is a step that supplements training data by introducing variants such as rotation, reflection, shear, scale, noise etc. This step is particularly encouraged in scenarios where training data is limited [13, 14]. In some cases,

however, applying conventional data augmentation techniques to training data may not offer good classification results for training and testing deep models and could be an underlying challenge in the original dataset. As a result, generative models have gained much attention and used by researchers as a data augmentation technique for generating new data for improving the classification results of deep learning models [14, 15].

This thesis develops a technique that can potentially be applied to other products for enhanced agro-industrial selection process. Results obtained show that GANs can be used for generating date images and classifying them into three hardness classes of hard, semi-hard and soft using a deep Convolutional Neural Network (CNN), particularly a Residual Neural Network (ResNet18), in combination with an expert system. The thesis by extension, could be applied in scenarios where data for example Covid-19 or brain tumour data are not equally represented among different classes or difficult to obtain as one class may have a higher representation in the dataset than the other. This ultimately makes the deep learning classification network's performance on the test dataset poor. Thus, utilizing GANs can potentially alleviate misclassification in scenarios where an unbalanced dataset is available.

1.2 Objective of the Thesis

The use of GANs have become popular as a technique for data augmentation. This approach helps generate more data for classification purposes as deep networks can face the challenge of training with limited data and may not perform optimally for data classification. The goal is to explore the possibility of using GANs to improve the classification of date fruits and by extension, develop a technique that can be applied to other products for enhanced industrial selection process.

This thesis will be focusing on generating images in applications where a big data set is not available, for example, in the food industry where a large number of images can simply be not an option. In particular, classification of dates was explored. As GANs have been applied to generate images of people, exploring the generation of images in a completely different field can have a great impact when using deep networks that require thousands and thousands of images for training. Keep in mind that the generated GAN images are of dates that do not exist. This is the exciting part of this research, as a deep neural network requires many

examples that in this scenario, and many other cases, are not available.

1.3 Thesis Outline

This thesis is divided into 6 chapters as summarized below:

Chapter 1: Highlights the introduction, background information, problem statement and existing solutions, objectives, and major contribution.

Chapter 2: Literature review on date fruit classification using computer and non-computer vision-based techniques. GANs and its variants are presented as well as the concept of style transfer. This chapter also gives an insight into Residual Neural Networks (ResNets), expert systems, and the validation of the GAN generated images.

Chapter 3: A discussion of the proposed framework for generating and classifying the date images is presented.

Chapter 4: The dataset utilized for the generation and classification of the date images is presented in this chapter.

Chapter 5: Discussion on the experimental results obtained are elaborated on this chapter.

Chapter 6: A conclusion of this thesis, recommendations and discussion on future work is summarized.

CHAPTER 2

Literature Review and Introduction to the Techniques Used in this Work

2.1 Date fruit classification

Image acquisition and processing of such images to perform tasks automatically by a computer-based platform is referred to as computer vision [16]. In the context of this thesis, the technique entails the acquisition, processing, and analysis of images to perform inspection of the physical attributes of agricultural and food products. Computer vision is utilized in agricultural industries for automatic sorting processes and ultimately provides enhanced value for agricultural products.

Fruit grading systems proposed by researchers employed in the agricultural industry varies from one fruit to the other. Hence, most systems designed for this purpose are usually focused on a particular fruit or product. However, efforts have been made to design computer-based systems for automatic sorting of fruits of more than one category [17]. For example, the authors in [18] designed a computer-based fruit grading robot capable of sorting fruits such as apples, peaches, and pears.

Kavdir *et al.* in [19] utilized fuzzy logic (FL) for the sorting of apples by extracting five features: size, surface defect, color, weight and shape. The authors in [20] proposed a technique for the detection of fruit (towel gourd, pumpkin, citrus, bitter gourd, mango and tomato) for harvesting purposes based on contour features rather than color as the latter is sensitive to illumination changes. The technique was tested on a Support Vector Machine (SVM) for classification. Computer vision techniques have also been applied for the automatic sorting of fruits in dedicated categories such as apples in [21, 22], blueberries [23], grapes [24], peach [25, 26], orange [27, 28], mango [29] and banana [30].

Several techniques have been explored for the grading and classification of date fruits using both computer and non-computer vision-based techniques. While the latter could be unhygienic and laborious as a result of human interaction with the date fruits, the former is more desirable as it is fast, consistent and can offer better results. Besides hardness, dates are graded based on other classification criteria. For example, the authors in [9] classified dates into four different classes: Ajwah, Sagai, Sellaj and Sukkary, using size, shape and local texture descriptors as features using a SVM for classification. A similar approach using SVM

was implemented in [31] to estimate type, maturity level, and weight. The classification of dates into seven different categories: Sukkari, Nubot Saif, Berhi, Sefri, Khlass, Seqei and Serri using color and physical characteristics was explored in [32]. The sorting of dates based on surface defects using image analysis was investigated in [33]. The authors in [34] classified date types such as in [9] by extracting color, shape and texture features using an Artificial Neural Network (ANN). Similarly, the work by Ohali in [17] classified dates using an ANN into three categories (grade 1, grade 2, and grade 3) by extracting the size, surface defect, flabbiness, intensity and shape manually determined by human visual inspection. The classification of dates using a Probabilistic Neural Network (PNN) by extracting color features from five date types (Khalas, Berhi, Fard, Lolo and Bomaan) was investigated in [35]. The research by Khalid *et al.* in [36] investigated the use of computer vision system (CVS) for the classification of dates by extracting color and shape features utilizing a Multi-Layer Perceptron (MLP) network incorporated with backpropagation and a Radial Basis Function (RBF) network respectively. Altaheri *et al.* in [7] proposed a technique for real-time harvesting of date fruits by agricultural robots based on maturity level, type, and harvesting decision. Two pre-trained deep learning models (AlexNet and VGGNet) were utilized for classifying the dates based on type and maturity level. The output of the respective models was utilized as input to a binary classifier for estimating the harvesting decision criteria based on defined harvesting rules. The authors in [37] developed a computer vision system for the automatic classification of Medjool dates based on their size and percentage of skin separation into four classes (jumbo, extra fancy, fancy and utility) using digital reflective near-infrared imaging techniques. A similar technique utilizing Near Infra-Red (NIR) was developed in [38] for the sorting of 30,000 Barhi and Hayani dates based on their maturity level.

Research on date fruit classification based on hardness using computer vision-based techniques is limited in the literature. For example, classification of dates based on hardness into three classes; hard, semi-hard and soft classes using ANNs and Linear Discriminant Analysis (LDA) was explored in [39]. Texture and histogram features were extracted from the date images. The overall accuracies obtained was 77% for the hard and soft classifier and 71% for the three class hard, semi-hard, and soft classifier. Similarly, the authors in [40] proposed several edge detection techniques for extracting features from date images and classifying them using an ANN into three hardness classes; hard, semi-hard and soft classes. Fourier-based features were also extracted in the frequency domain for the classification of the dates with overall accuracies of 87% for the two class hard and soft classifier and 75% for

the three class hard, semi-hard, and soft classifier.

2.2 Generative Adversarial Networks

Generative models are one of the techniques where the training data is unlabeled as opposed to models that train on supervised learning [12]. Generative models comprise of three groups: Generative Adversarial Networks (GAN) [41], Variational Autoencoder (VAE) [42], and Autoregressive Networks [43].

VAEs are generative models that tend to model the probability density of data but is somewhat limited as it generates more blurred data compared to GANs as a result of its output having a certain bias [15]. Autoregressive models such as PixelRNN and PixelCNN are used for the generation of images sequentially i.e., pixel by pixel. While the latter reduces computational time as a result of its convolutional layers, the former offers better results [44]. However, GANs process the pixels of an image at once thereby making it a better and faster option.

GANs have gained attention in the data generation world as they have been utilized by researchers to improve the accuracies of their models [15]. Recent works in this field include speech emotion recognition as reported in [45, 46, 47]. The authors in [48] utilized GANs for the synthesis of spectrograms for identifying Parkinson's diseases in a patient's voice print. The work in [49] employed GANs for the generation of electroencephalogram signals to improve the classification results of brain-computer interfacing systems. The detection of anomalies in medical images using GANs was investigated in [50]. Similarly, the authors in [51] utilized GANs for the generation of DNA sequences. Artificial human faces have been generated using GANs [52, 53, 54] and the face generation technique has immensely improved with the inception of Nvidia's Style-GAN2 ADA model which significantly stabilizes training using a small dataset as training GANs with a small dataset leads to discriminator overfitting [55]. Furthermore, the generation of video signals using GANs was explored in [56] and recent advances in the field of computer and network security utilize GANs for creating security loopholes in order to hack intrusion detection systems [57]. In addition, the generation of adversarial malware examples using GANs was explored in [58].

GANs are generative models that utilize a game-theoretic approach for training the network. They consist of two networks: the generator and the discriminator network. The generator is an up sampling network while the discriminator is a down sampling network. The goal of the

generator is to generate fake data from random noise whereas the discriminator aims at distinguishing real from fake data. As the generator gets better at generating fake data and the discriminator better at telling fake from real data, an optimal state is reached where the discriminator is unable to distinguish real from fake data. This optimal state is called Nash equilibrium [15]. At this point, the generator is said to be generating data from the real dataset.

GANs are of different variations. For example, Deep Convolutional Generative Adversarial Networks (DCGAN) which substitutes the fully connected layers present in the generator network with deconvolution layers was proposed by Radford *et al.* in [59]. Conditional Generative Adversarial Networks (CGAN) which imposes conditions to both the generator and discriminator network using additional knowledge in order to prevent training divergence was introduced by Mirza and Osindero in [60]. Similarly, variants of CGAN such as Information GAN (InfoGAN) and Auxiliary Classifier GAN (ACGAN) were proposed by Chen *et al.* [61] and Odena *et al.* [62] respectively. Broke *et al.* [54] proposed BigGAN and its variant BigGAN-deep.

The generator and discriminator network have loss functions represented by $L^{(G)}$ and $L^{(D)}$ respectively.

The Loss function for training the discriminator is:

$$L^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim pdata}\log D(x) - \frac{1}{2}\mathbb{E}_{z\sim \boldsymbol{p}(\boldsymbol{z})}\log(1 - D(G(z))) \quad (1)$$

where x is the original data, z is the random noise (uniform or Gaussian), $G(z)$ is the output of the generator network and expectation is represented by $\mathbb{E}$. The probability that the discriminator network classifies x as being a real sample is represented by $D(x)$. The probability of the discriminator ascertaining the data produced by the generator network is given by $D(G(z))$.

The authors in [41] defined the discriminator network as a binary classifier that distinguishes real and fake data. The discriminator aims at ascertaining correctly the origin of the data; its goal is to make $D(G(z))$ to approach 0 while the generator aims at making it 1, hence the adversarial phenomenon between both networks. Consequently, the generator's loss function is derived from the discriminator as shown next:

$$L^{(G)} = -L^{(D)} \tag{2}$$

Both networks are optimized in a minimax game represented by:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x\sim pdata(x)}}[\log D(x)] + \mathbb{E}_{\boldsymbol{z\sim p(z)}}[\log(1 - D(G(z)))] \tag{3}$$

As training progresses, the parameters of the network are updated simultaneously. The point at which $D\big(G(z)\big) = 0.5$ signifies the Nash equilibrium and as a result, the discriminator cannot ascertain the differences between the real and fake data. In practice, training GANs using the Binary Cross Entropy Loss (BCE) as represented in equation 3 where the generator network minimizes the probability that the discriminator network classifies the data produced by the generator as real could result in instability in the training of the generator network. As a result, a modification in the loss where the generator maximizes the probability that the discriminator classifies the data synthesized by the generator as fake prevents saturation of the network and improved performance in training of the generator network. The equation for the modified generator loss is represent by:

$$\max_G \left\{\mathbb{E}_{\boldsymbol{z\sim p(z)}}\left[\log\Big(D\big(G(z)\big)\Big)\right]\right\} \tag{4}$$

GANs are challenged with discriminator overfitting when a dataset with limited number of samples is used for generating new data. This leads to training divergence. To overcome this challenge, the authors in [55] proposed Adaptive Discriminator Augmentation (ADA), a technique that dynamically augments the training set by adapting to the variation of overfitting. The goal of ADA is to curb overfitting during training when the dataset available is limited with few images as in this project.

We know that one of the ways to prevent overfitting is by augmenting the training set with variants like rotation, scaling, reflection, noise etc. Also, any augmentation applied to the training data is automatically replicated by the generator and if not properly controlled, the generator learns to generate the augmented distribution. For example, a noise augmentation leads to noisy results even though there is none (noisy images) in the training dataset. This is known as leaky augmentation, and it is undesirable for the generated samples. As a result, ADA prevents leaky augmentations in the generator network by combining balanced consistency regularization (bCR) and latent consistency regularization (zCR) a technique

originally proposed by Zhao *et al.* [63]. The block diagram of the GAN network used for generating the date images is shown in Figure 1.

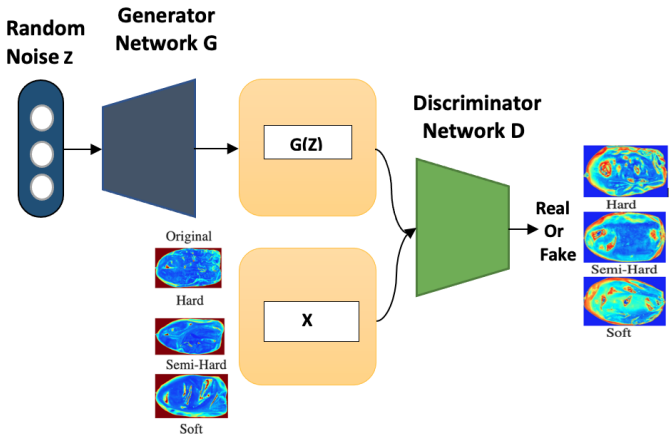


Figure 1. Block diagram of Generative Adversarial Networks.

2.2.1 GAN Evaluation Metrics

Several metrics have been proposed in the literature for evaluating the performance of GANs. For example, the Fréchet inception distance (FID) also referred to as Wasserstein-2 distance proposed by Heusel *et al.* [64] which computes the density difference between two distributions in the multi-dimensional feature space of an InceptionV3 classifier [65] is one of the metrics for evaluating the similarity of real and generated data using GANs. FID is used to measure the distance between the mean and covariance of the gaussian distribution of the real and generated samples to evaluate the quality of generated data [15]. A low FID score means the GAN is generating high quality data and higher scores imply the generation of low quality data in comparison to the original dataset. The authors in [55] improved FID of the CIFAR 10 dataset from 5.59 to 2.42.

The equation for computing the FID score of real and generated data is represented by:

$$FID(X_R, X_G) = \|\mu_R - \mu_G\|^2 + Tr\left(C_R + C_G - 2(C_R C_G)^{\frac{1}{2}}\right) \quad (5)$$

where X_R and X_G are the images from the original and generated datasets respectively. μ_R and μ_G represents the mean of the original and generated images respectively. Similarly, C_R and C_G represents the covariance matrix of the real and generated images respectively. Tr is

the trace operation which computes the sum of the elements in the leading diagonal of the square covariance matrix.

The Kernel Inception Distance (KID), Inception Score (IS), Precision and Recall (P&R), Mode Score (MS), and Perceptual Path Length (PPL) are some of the metrics that are also used for evaluating the performance of GANs. IS scores, proposed by Salimans *et al.* [66], indicate that the GAN is generating high quality samples. The KID distance was proposed by Bińkowski *et al.* [67] to address the challenge of overfitting where IS and FID are limited. Nowozin *et al.* [68] proposed the MS which simultaneously evaluates the category and visual quality of the generated data. The metric P&R proposed by Sajjadi *et al.* [69] precisely evaluates the proportion of generated data that are of similar distribution to the original dataset and the proportion of data that can be generated by the GAN respectively. Perceptual Path Length (PPL) enables for a smooth change in the semantics of the generator network as small perturbations are initiated to the input latent vector z [70]. However, the metric PPL tends to be biased to the texture features of the generated images. Consequently, a low PPL score will generate an image that is of less quality as observed from visual inspection whereas a high PPL will generate an image that is of higher quality but is not totally correlated to the original image [71].

Ascertaining the ideal metric for the evaluation of GANs is somewhat challenging to researchers. However, the author in [72] suggests various quantitative and qualitative techniques that could help researchers determine the appropriate metric for GAN evaluation.

An ideal evaluation metric should differentiate between real and generated data, spot overfitting, and be sensitive to mode dropping and mode collapse [73].

2.2.2 GAN Regularization

Regularization is a technique that imposes constraints by the addition of a regularization term to the loss functions of neural networks and machine learning algorithms to improve the generalization of the models [74] i.e., it improves the performance of the machine learning models on unseen data as it prevents them from overfitting. Popular methods of regularization include early stopping, tangent propagation, weight sharing, curvature-driven smoothing etc. [75, 76].

In the world of GANs, regularization is applied to both the generator and discriminator network ultimately improving the performance of the model. Several techniques have been

proposed in the GAN literature for performing regularization [77]. For example, Gulrajani *et al.* in [78] proposed Gradient Penalty (GP) to improve the performance of Wasserstein GANs (WGAN) [79]. Kodali *et al.* [80] introduced Deep Regret Analytic Generative Adversarial Networks (DRGAN) by modifying the state-of-the-art GP technique and reported to better improve the performance of GANs. Roth *et al.* [81] proposed the Jensen-Shannon Regularizer (JSR). The vanilla Consistency Regularization (CR) method was investigated by Zhang *et al.* [82] and Wei *et al.* [83]. In this technique, the original images from the dataset and their corresponding augmentations are utilized as input to the discriminator network. An auxiliary loss function is used to engender the discriminator to synthesize correlated output images from the original dataset and its corresponding augmentations using prior knowledge. This technique however effective, results in a bias as the augmentations are performed only on the original images and not applied to the generated images. As a result, the generator network learns to apply such augmentations to the generated images which results in the introduction of undesirable artifacts in the generated images.

Zhao *et al.* [63] constrained the mapping from the discriminator network to the generator network and performed CR on the generator network resulting in improved quality of the generated samples. In their technique, two variants; Balanced Consistency Regularization (bCR) and Latent Consistency Regularization (zCR) are investigated. bCR was introduced to add a regularization term to both the discriminator and generator loss function creating a form of balance in the network. In this technique, the augmentation is also applied to the generated samples and used as an input to the discriminator network. In the zCR approach, the augmentation is applied to the latent vector space and utilized as an input to the generator network. Similarly, the regularization term is applied to both generator and discriminator loss function. A combination of both methods has experimentally shown to improve the performance of GANs, a technique referred to as Improved Consistency Regularization (ICR).

2.2.3 Style Transfer

Style transfer is a technique that structures an image (typically called a content image) to have the style features for example color and texture of another image [84, 85, 86]. Several researchers have introduced algorithms to transfer the contents of one image to another. For example, the work by Gatys *et al.* [86] experimented style transfer using Deep Neural Networks (DNN) by matching the extracted features in the convolutional layers of both the

content image and style image respectively. The style transfer of an image to a video sequence was investigated by Ruder *et al.* in [87]. Wang *et al.* [88] explored style transfer utilizing a deep Convolutional Neural Network (CNN) by considering separate representations of color and illumination of the images with different scaling losses. Similarly, Dumoulin *et al.* [89] investigated a Deep Neural Network (DNN) for encoding a variety of 32 artistic styles from several painting images. A deep feed forward neural network for generation of 300 texture features and the transfer of 16 styles respectively was explored by the authors in [90]. A similar approach using a pretrained CNN was developed in [91] for arbitrary style transfer.

Recent advances in the field of GANs have witnessed the use of style transfer for improving the quality of generated images [84]. This technique utilizes Adaptive Instance Normalization (AdaIN), an algorithm that performs style transfer between the content image and style image by adjusting the variance and mean of respective images. Batch Normalization (BN), a regularization technique that conserves the variance of features in data as a network gets deeper [92] is utilized in GANs to improve the training of the discriminator network.

However, AdaIN is applied to the generator network at every instance as it is reported to better improve the style transfer of the generator network [70]. The most effective method in the literature for the normalization of GANs as reported by Miyato *et al.* [93] is spectral normalization and it is used widely by researchers for adjusting the weight vectors of GANs. The authors in [84] defined BN as a technique utilized for the normalization of the expected value and standard deviation of data for respective feature channels.

The Batch Normalization equation is represented by:

$$BN(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta \tag{6}$$

where $x \in \mathbb{R}^{B\times C\times H\times W}$ is the input image, B is the batch size, C represents the number of feature channels e.g., C = 3 for an RGB image, the spatial dimensions of the input image H and W represent the height and width of the input image respectively. $\gamma, \beta \in \mathbb{R}^{C}$ represents the learnable affine parameters from the data. $\mu(x), \sigma(x) \in \mathbb{R}^{C}$ represents the expected value and standard deviation evaluated over the batch size and spatial resolution independently for respective feature channels.

$$\mu_c(x) = \frac{1}{BHW}\sum_{b=1}^{B}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{bchw} \tag{7}$$

$$\sigma_c(x) = \sqrt{\frac{1}{BHW}\sum_{b=1}^{B}\sum_{h=1}^{H}\sum_{w=1}^{W}(x_{bchw} - \mu_c(x))^2 + \epsilon} \quad (8)$$

where ϵ is an arbitrary small constant value that improves numerical stability.

Ulyanov *et al.* [94] improved the performance of style transfer by replacing the batch normalization layers with Instance Normalization (IN) layers after every convolutional layer of the network. This approach effectively decreases the style loss after every iteration of the network.

The IN equation is represented by:

$$IN(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta \quad (9)$$

In this scenario, $\mu(x)$ and $\sigma(x)$ represents the mean and standard deviation computed independently for each training example or sample. i.e., $\mu(x)$ and $\sigma(x)$ is evaluated over the spatial resolution independently for individual feature channels and for each training sample.

$$\mu_{bc}(x) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{bchw} \quad (10)$$

$$\sigma_{bc}(x) = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(x_{bchw} - \mu_{bc}(x))^2 + \epsilon} \quad (11)$$

In this scenario, b is = 1 for every training example or sample.

The authors in [84] modified IN by introducing AdaIN which essentially scales the input x with the non-learnable style spatial statistics.

The AdaIN equation is represented by:

$$AdaIN(x, y) = \sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \mu(y) \quad (12)$$

where x and y represents the input/content and style images respectively. $\sigma(y)$ and $\mu(y)$ are utilized as scaling and shifting parameters respectively evaluated over spatial dimensions similar to IN. Before applying the style statistic $\sigma(y)$ and $\mu(y)$ to the content image, AdaIN undos the original style information present in the content image to effectively transfer the

styles present in the style image y to the content image x. In summary, AdaIN transfers the style y to an image x based on the non-learnable channel-wise expected value $\mu(y)$ and the variance $\sigma(y)$ of the style image y.

The architecture for a traditional GAN generator network is represented in Figure 2.

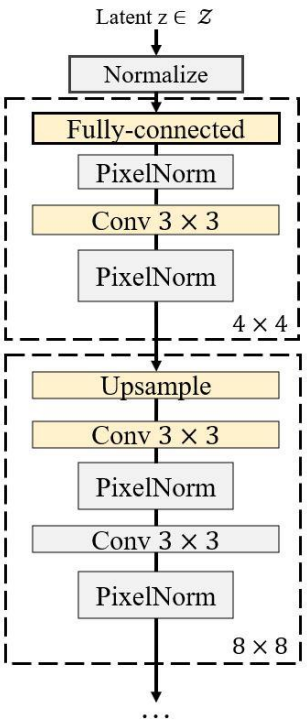


Figure 2. Architecture of a Traditional GAN Generator Network.

2.2.4 Style GAN based Generator Network

The up sampling network consist of latent input z which is usually a style image for example, normalized, encoded, and non-linearly mapped unto a latent space w using a multi-layer perceptron (MLP) with 8 fully-connected layers [70]. The generator network consists of a constant input vector having zero mean and a standard deviation of one utilized at the first layer and serves as the input to the first AdaIN operation. The styles from the encoded latent space w are transferred using the AdaIN operation at respective resolutions of the generator network. The output of the respective convolutional layers is determined by the style introduced by the AdaIN operation. Gaussian distributed noise [95] is added per pixel after respective convolutional layers to effectively alleviate artifacts that may be present in the

image and appear as water droplets in regions such as wrinkles and hair in human faces. This approach effectively introduces stochastic variation [70] to the generated images.

The architecture of a style-based generator network StyleGAN2-ADA is represented by Figure 3.

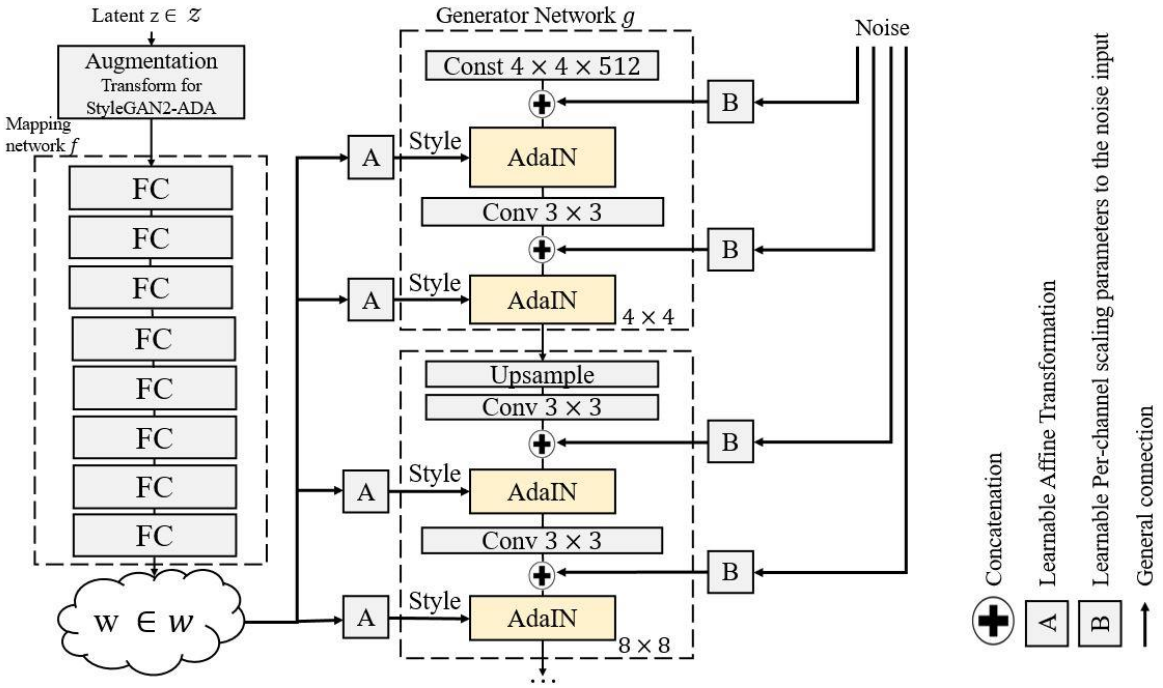


Figure 3. Architecture of a StyleGAN2-ADA Generator Network.

2.2.5 Style GAN based Discriminator Network

The discriminator is a down sampling network where the images dimensions match that of the final output of the generator network. The image serves as an input to the discriminator network which is further augmented using specified augmentation transforms. Furthermore, these input images are down sampled to corresponding resolutions of the generator network using residual or skip connections. A binary classification is made by the discriminator network which distinguishes real from fake images.

The architecture of a style-based discriminator network is represented by Figure 4.

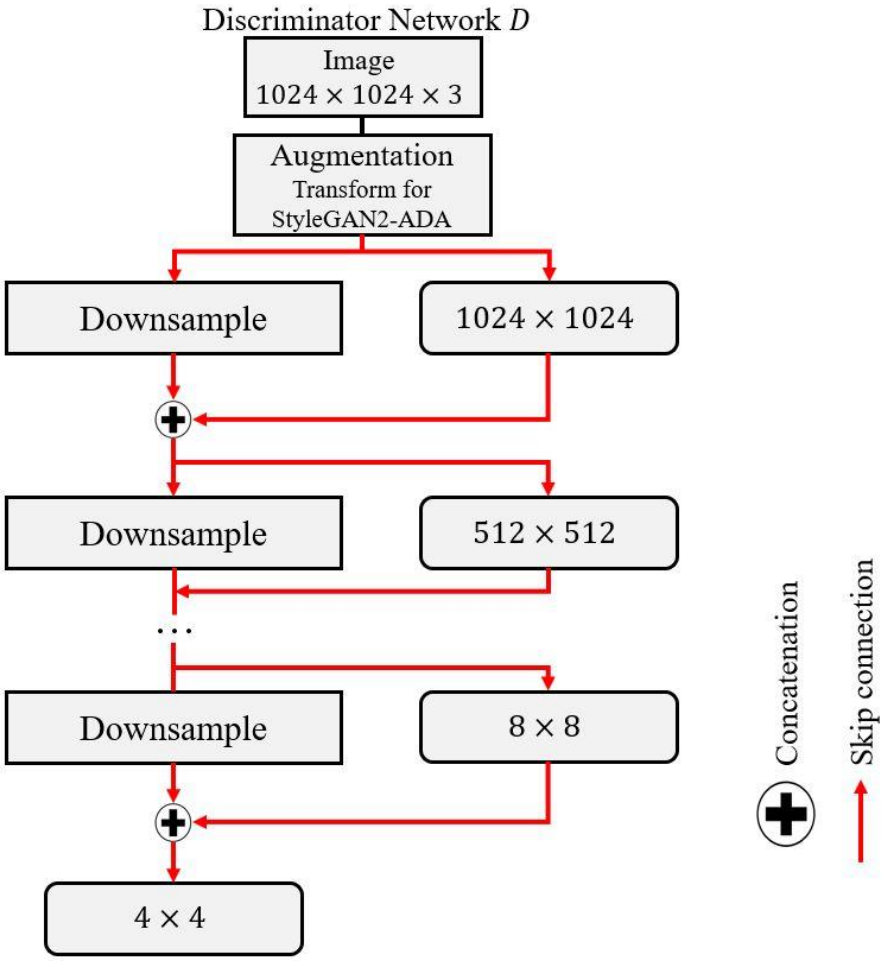


Figure 4. Architecture of a StyleGAN2-ADA Discriminator Network.

2.3 Residual Neural Networks

Residual Neural Networks (ResNet) improve the generalization ability of deep networks as the number of layers increases significantly [96]. The residual connection or skip connection between the input and output layers of the network helps the network to learn the identity function. The intuition behind ResNet is that as the network gets deeper, it becomes difficult to choose parameters and learn the identity function. This difficulty is as a result of exploding or vanishing gradients [97, 98, 99] hence, the network fails to converge. The degradation of the network as more layers are stacked however, is not a result of overfitting as reported in [96]. To overcome this drawback, skip connections are used from the output of one layer to the input of the next layer. As a result, the identity function is added to the output of a layer before passing through the ReLU [100, 101, 102] activation function. Figure 5 shows the details of the skip connection in the residual network.

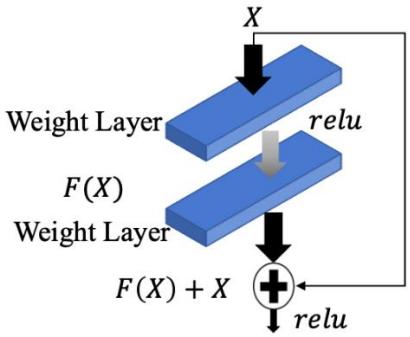


Figure 5. Skip connection of ResNet18 architecture.

The residual learning equation is represented by:

$$y = \mathcal{F}(X, \{W_i\}) + x \tag{13}$$

Where x and y denote the input and output vectors of specified layers respectively. W_i represents the weight vector in the i^{th} layer of the network. $\mathcal{F}(X, \{W_i\}) + x$ denotes the residual mapping that the network is to learn.

Let W_1 and W_2 represent the weight vectors in the first and second layer of the skip connection represented in Figure 5. Then

$$\mathcal{F} = W_2\lambda(W_1\, x) \tag{14}$$

where λ represents the non-linearity i.e., ReLu activation function [103].

The deep network architecture used for classifying the date images used in this thesis is ResNet18. This pre-trained network consists of 18 layers excluding the pooling and batch normalization layers. ResNet18 was chosen because of its performance on the date dataset and the level of complexity of Resnet18 versus the computational time it requires lies in the middle of all the available network architectures in Matlab [104]. The architecture used for classifying the date images consists of 71 layers with 8 residual blocks. These layers consist of convolutions, batch normalization, ReLU, pooling and fully connected layers. The pooling layers consists of max-pooling and global average pooling [105]. The output of the fully connected layer is fed into the SoftMax block which outputs the probability of the input as belonging to a particular class using the cross-entropy loss function [106]. Figure 6 shows the architecture of the ResNet18.

The input to this network is of dimension 224 × 224 × 3. The fully connected layer at the output was replaced with one matching the number of classes for the date images. The images were classified into hard, semi-hard and soft dates using a cross validation of 5 folds and batch size of 32 with epochs = 50 for the two-class classifier (classifier that identifies hard and soft cases only) and epoch = 100 for the three-class classifier. The Adaptive Moment Estimation (Adam) optimizer [107] with learning rate = 0.00001 was the desired choice as it significantly improved the training accuracy of the date images compared to Stochastic Gradient Descent with momentum (SDGM) [2]. The Matlab command ImageDataAugmenter was used for further augmenting the training set images with variants of rotation range = [-5, 5], reflection = 1 and shear range = [-0.05, 0.05].

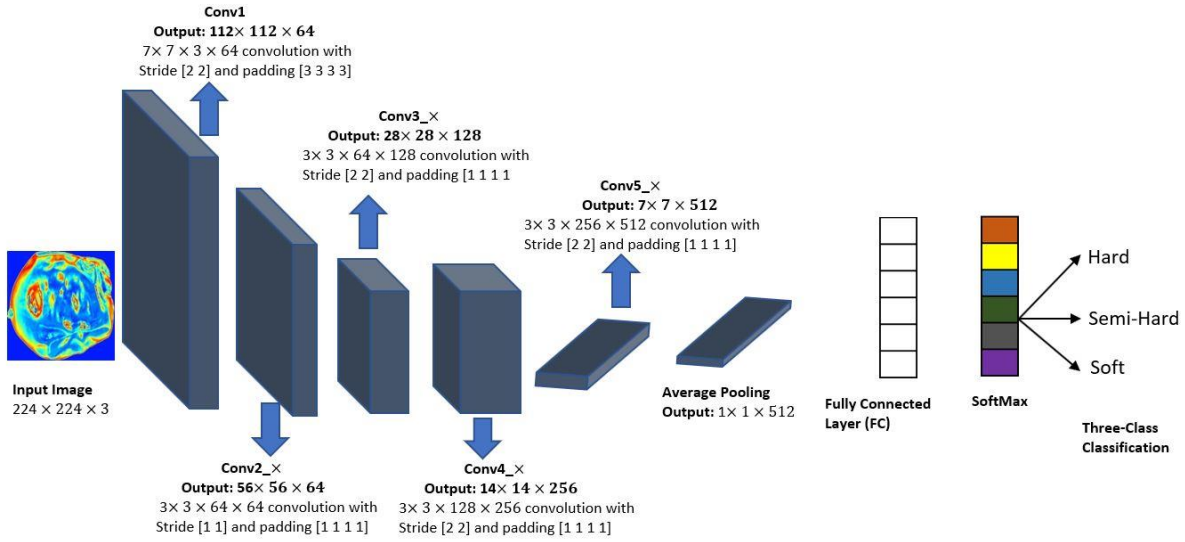


Figure 6. Descriptive block diagram of ResNet18 architecture.

2.4 Expert system

The original date dataset has been proven quite difficult to classify as reported by researchers in Sultan Qaboos University in [39]. Because of the underlying nature of the semi-hard dates having textures similar to both the hard and soft dates, the per class accuracy for the three-class classifier on the semi-hard date is affected. This was not the case with the hard and soft two-class classifier as it performed better on the test set as shown in the results section of this thesis. Expert systems are usually designed to solve complex problems and achieve accuracies similar to that attainable by human experts. These systems are often developed

using if-then rules in combination with logical or Boolean operators [108]. Thus, an expert system was developed to combine the results of the two and three class classifiers.

In the ResNet18 architecture, the probability scores are the output values from the softmax layer before they are passed through the classification layer which computes the cross-entropy loss for classification. These probability scores were used to obtain a benchmark for correct classification of the semi-hard dates based on the trend observed from the predictions and probability scores. A similar idea was developed in [109] for classifying text using K-Nearest Neighbors (KNN) by a combination of logical expression designed with Boolean operators.

Figure 7 is the flow chart describing the expert system used for classifying the date images.

The system acquires the probability scores from the two-class classifier (C_A) and the three-class classifier (C_B) respectively. P_A represents the probability score from the two-class classifier while P_B is the probability score from the three-class classifier. The expert system makes a prediction by logically comparing respective probability scores using the AND Boolean operator as shown in Figure 7. From Figure 7, when both conditions for selecting the prediction of respective classifiers are not satisfied, the system selects the maximum probability score obtained from either classifier as the final prediction. A modification of the system that allows for the manual inspection of the dates by a human expert was incorporated by modifying the selection of $C = \max\{P_A,P_B\}$ as indicated in the results section of this thesis.

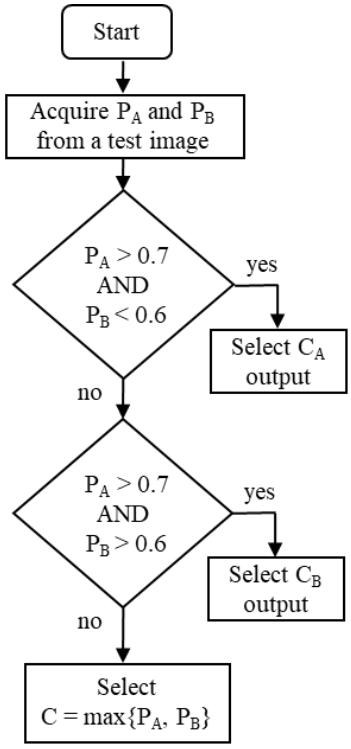


Figure 7. Block diagram of the expert system.

2.5 Validation of the GAN images

2.5.1 Extraction of texture features

The hypothesis here is that the date's texture will be "rough" as they dry and become hard. Thus, texture features included in this analysis were Histogram of Gradients (HOG) [110], as they analyze the histogram of the gradients of the images emphasizing the edges within the dates as well as another feature extraction technique, Tamura features, which also extracts features from the edges [111]. For considering a possible repetition of the texture features, semivariograms were used [112]. Note that the popular feature extraction method using Gray Level Co-occurrence matrix (GLCM) [34] was not considered as these possible repetitions of patterns in the texture images do not offer patterns that can be highlighted within the date images by the position operators in the GLCM approach.

2.5.2 Histogram of Oriented Gradients (HOG)

HOG is a technique utilized in computer vision as feature descriptors by computing the frequencies of the derivative of specified regions of an image for object detection [113]. For example, the authors in [114] utilized HOG for extracting texture features for the classification of Situ leaf. HOG emphasizes the corners and edges of an image, computing the gradients along these paths. In an image, an edge is a region where there is a sudden change in the image intensity function, and it represents the shape of the image [40] as well as rough areas within an object.

The method consists of calculating histograms from the gradient of the pixels in a gray level value image in both directions.

The steps for an input image *I* of size S_x by S_y are:

1. In case of noisy images, apply a low pass filter to reduce the noise increase that usually happens when applying gradient operators. In case of the date images, this step was not implemented as the date images are not noisy.
2. Estimate the gradient of the images in both horizontal and vertical directions G_x and G_y. The Prewitt operators were utilized. $G_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ and $G_y = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$
3. Compute the magnitude of the gradient as
$$G_m = \frac{1}{6}\sqrt{G_x^2 + G_y^2} \quad (15)$$
4. Compute the directions of the gradient as
$$G_d = arctan\left(\frac{G_y}{G_x}\right) \quad (16)$$
5. Calculate the histograms over N_{HOG} bins of both G_m and G_d. For the work presented here, $N_{HOG} = 16$ for a total of 32 features using this HOG technique.

2.5.3 Semivariograms

Semivariograms were introduced in geostatistics as metrics that describe the spatial dependence between geographical information. Semivariograms are suitable for characterizing regular patterns of image variation and can be used as texture metrics. For an image positioned at row *x* and column *y* having N_p pairs of pixel values *I*(*x*, *y*) and *I*(*x*', *y*')

separated by a distance h, the variogram is computed as [112]:

$$\gamma(h) = \frac{1}{2Np}\sum_{i=1}^{Np}[I(x,y) - I(x',y')]^2 \quad (17)$$

Using three different directions for pairs of pixels located in the E-W, N-S, and N-E directions, for example for the N-E the variogram is calculated as

$$\frac{1}{2Np}\sum_{i=1}^{Np}[I(x,y) - I(x+h,y+h)]^2 \quad (18)$$

and using h = 1, 2 we calculated two more features using this technique.

2.5.4 Tamura

Tamura *et al.* [111] defined a coarseness texture features that corresponds to human visual perception. Here coarseness corresponds to the scale and repetition rates of texture. For image patterns with different structures, the bigger its element size, the coarser it is. Coarseness is computed as:

$$Coarseness = \frac{1}{S_x S_y}\sum_{x=0}^{S_x-1}\sum_{y=0}^{S_y-1} S_{best}(x,y) \quad (19)$$

where $S_{best}(x,y) = 2^k$ and k is the value that maximizes $max_{1\le k\le L}\left(E_{k,h}(x,y), E_{k,v}(x,y)\right)$ in the horizontal or vertical direction, $k \in [1, L]$ where $2^L \le \min(S_x, S_y)$. $E_{k,h}(x,y), E_{k,v}(x,y)$ are computed from $A_k(x,y)$ which is the average at each pixel $I(x,y)$ of the neighborhood of size 2^k x 2^k:

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1}\sum_{j=y-2^{k-1}}^{y+2^{k-1}-1}\frac{I(i,j)}{2^{2k}} \quad (20)$$

where

$$E_{k,h}(x,y) = |A_k(x+2^{k-1},y) - A_k(x-2^{k-1},y)| \quad (21)$$

$$E_{k,v}(x,y) = |A_k(x,y+2^{k-1}) - A_k(x,y-2^{k-1})| \quad (22)$$

2.6 Two sample Kolmogorov-Smirnov test

Also known as the KS test, it is a non parametric test that is general in the sense that it does not assumes normality (Gaussian distribution) [115, 116] as for example the t-test does. Working with the Cumulative Density Functions (CDFs) of samples coming from a distribution, it measures the differences of two sample distributions by the differences on the CDFs of the two distributions. The null hypothesis is:

Ho: Two samples drawn from populations have the same distribution.

Let $F_{1,n}$ and $F_{2,m}$ be the empirical distribution functions of the first and the second sample distributions where n and m are the number of realizations. The difference between the first and second sample distributions is measured by the statistic $D_{n,m}$ which computes the greatest vertical distance between $F_{1,n}$ and $F_{2,m}$ represented as:

$$D_{n,m} = \sup\left|F_{1,n} - F_{2,n}\right| \tag{23}$$

where sup is the supremum function. The null hypothesis is rejected at level α if

$$D_{n,m} > \sqrt{-ln\left(\frac{\alpha}{2}\right)\frac{1+\frac{m}{n}}{2m}} \tag{24}$$

The significance level α is the probability of rejecting the null hypothesis when it is true. A significance level of 0.05 indicates a 5% risk of concluding that a difference exists when there is no actual difference. Figure 8 shows the block diagram of the validation of the GAN images.

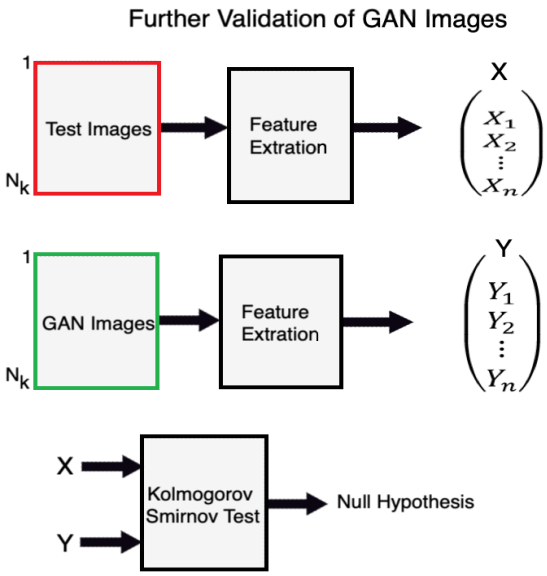


Figure 8. Block diagram for validating the GAN generated images.

CHAPTER 3

Methodology

3.1 Proposed Framework

The proposed framework consists of a Style-Gan2 ADA (Adaptive Discriminator Augmentation) architecture [55, 117] utilized for generating the hard, semi-hard and soft date images in the data augmentation step and used on the training of the ResNet18 networks. The input to this framework is a batch of date images of dimension 1024 ×1024 pre-processed by converting the original gray level images into color ones using the Jet colormap as it has proven to improve the training accuracy of the date images as shown in Figure 9 [118]. Figure 10 shows examples of this pseudo-coloring pre-processing.

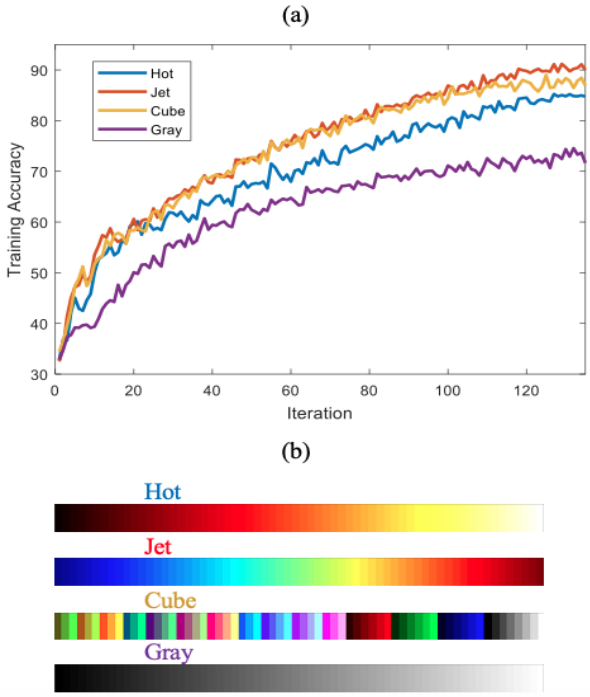


Figure 9. (a) Training accuracies. (b) Colormaps tested. Note how the Jet colormap trains faster than the other ones.

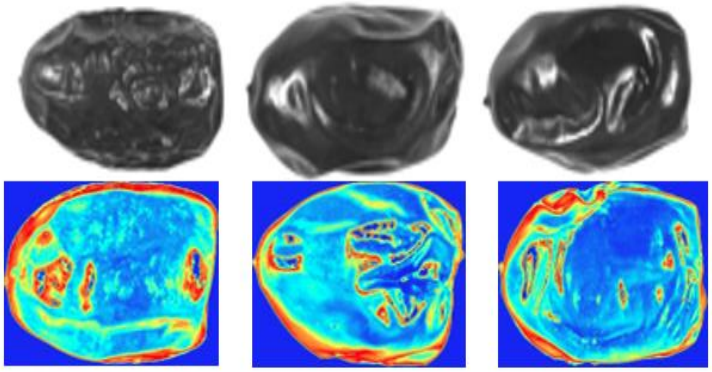


Figure 10. First row: Original images, Second row: Images after pseudo-coloring using the Jet colormap. First column: hard dates, second column: semi-hard date, third column: soft date.

We trained two classifiers using the ResNet18 architecture utilizing transfer learning for classifying the date images. The classifier denoted C_A = ResNet18$_A$ is a two class classifier that considers the soft and hard classes. The classifier denoted C_B = ResNet18$_B$ is a three class classifier that considers all the different classes.

Out of the total N_T = 1800 images, we separated N_{te} = 300 images for final testing, same number used in [39]. For this testing set, one hundred images per class were randomly selected from the original dataset and separated in order to ensure that the GAN data augmentation images were generated without using this final test set.

As a result of the inherent misclassification in the semi-hard date images when using the ResNet18, we developed an expert system by combining the two-class and three-class classifiers. The inputs to this expert system were the probability scores of the final classifications obtained from the ResNet18 networks. These scores are referred as P_A and P_B for the two class and three class REsNet18s classifiers respectively.

Finally, we investigated the similarity of the original and GAN generated images by extracting texture features and a comparison was done using the Kolmogorov-Smirnov test in order to ascertain the validity of the GAN images. The results obtained further confirmed that the images from the data augmentation step were of merit. The block diagram in Figure 11 provides an overview of the project framework.

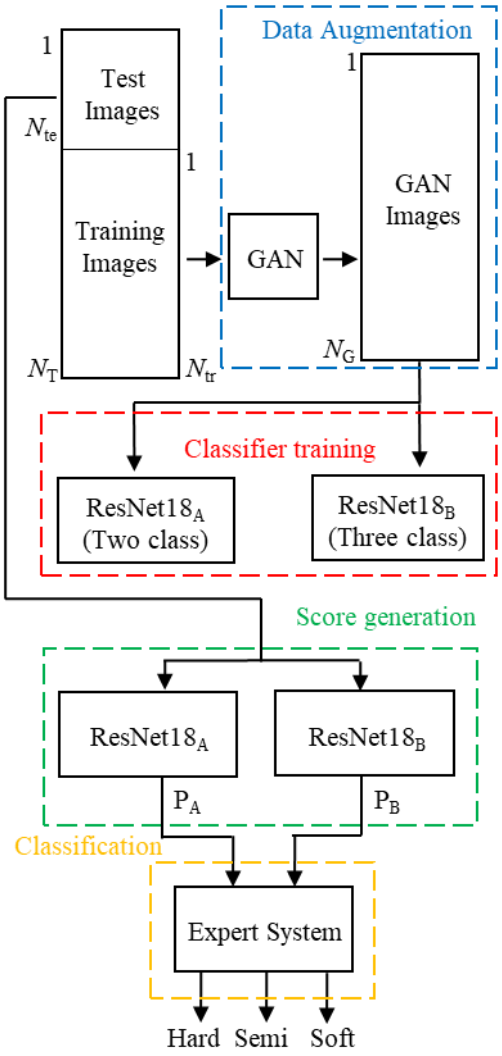


Figure 11. Block diagram of the proposed system.

CHAPTER 4

Dataset

4.1 Original and Generated Datasets

The original dataset in this work consists of 1800 images with 600 images per class. Date fruit of the Fard variety were obtained from three cultivating regions (Al-Dakhliah, Al-Batinah, and Al-Sharqiah) in Oman. From the 600 images per class, 200 images were obtained per region. The standards for the grading of the dates based on hardness (hard, semi-hard, and soft) were defined by the managers of three date sorting factories in Oman (Bright Sun Dates, Samail Dates Factory, and United Date Processing). The sorting of the dates based on the standards was performed by a group of experienced graders in the Bright Sun Dates factory and confirmed by an experienced senior grader in Bright Sun Dates factory. The Dates were compressed on a mechanical plate and the force time curve readings were automatically recorded using a Texture Profile Analyzer (TPA) using the procedure defined in [40]. Uncompressed 8-bit images with resolution 1024×768 pixels were acquired with a monochrome camera (model: XCD-X700, Sony, Japan) incorporated with a charge coupled device (CCD). The camera was connected to a computer (dual-core AMD operator™ processor 2220, 2.80 GHz, 3.50 GB of RAM, HP XW 9400 Workstation). Two fluorescent lights (36 W, model: Dulux L, OSRAM, Italy) were utilized for the artificial illumination of the date fruit samples. The distance between camera and sample was fixed at 1 m in all experiments to simulate the factory feasibilities. A calibration using white and black standard color cards (Digital Kard XL, DGK Color Tools, USA) was implemented on the monochrome camera before obtaining images in respective batches. Image of date fruits were captured remotely through the computer, saved and used for further data analysis. Three hundred testing images were separated from the original dataset with 100 images per class.

Transfer learning on the FFHQ-1024 dataset with 25000 images and a mini-batch size of 4 was used for generating the date images. The original FFHQ dataset consists of 70000 images but with the Style-GAN 2 ADA model, training can be done with less than 30000 images. FFHQ-1024 was chosen because the original date images are of dimensions 1024 ×768 and would be easier to resize to fit the Style-GAN model. From the original dataset, 498 images were utilized after every random shuffle for generating a set of 28 images per class. The

generation of the date images was implemented in python using Tensorflow 1.x framework as the Style-GAN model is not compatible with Tensorflow 2.x. However, recent development in the field now allows for the implementation in PyTorch. The generated GAN dataset consists of 3000 images with 1000 images per class for the hard, semi-hard and soft classes generated using three Style-GAN2 ADA models respectively. Figure 12 represents the original and GAN generated date images for the hard, semi-hard and soft dates respectively.

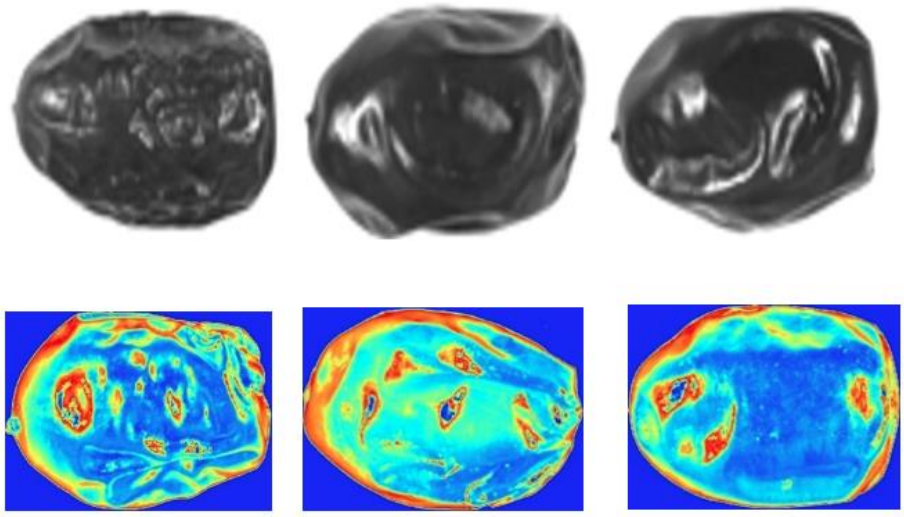


Figure 12. First row: Original images, Second row: GAN generated images. First column: hard dates, second column: semi-hard date, third column: soft date.

CHAPTER 5

Experimental Results and Discussion

The training, evaluation and testing of the generated date images was implemented in MATLAB R2020b using the following computational platform: Intel Core i7-10700 (Hyper-Threaded) @ 2.90 GHz, NVIDIA GeForce GTX 1660 SUPER (6 GB) GPU, 16 GB RAM and 64-bit Windows 10. The batch sizes were initialized to 32 and 4 for ResNet18 and StyleGAN2-ADA models respectively.

The accuracy of the two-class and three-class model using standard augmentation (rotation, reflection and shear) + GAN (S+GAN), GAN and only standard augmentation alone on the original dataset (S+Original) is shown in Table I. We can see that the ResNet model performs poorly on the original dataset with the standard augmentation technique and the GAN augmentation provides better results. The comparison was done using the same number of training and testing images. The data augmentation technique helped overcome overfitting when a limited number of date images was used for training the ResNet18 model. Note, the testing results shown in Table I corresponds to the N_{te} testing images used for the final classification as shown in Figure 11.

Table I. Comparison of training and testing accuracy of Resnet18 using standard augmentation and GAN augmentation.

Model	Augmentation	Training (%)	Testing (%)
2-Class classifier	S+GAN	96.6	90
	GAN	94.1	85.5
	S+Original	96.3	56
3-Class classifier	S+GAN	85.3	62
	GAN	78.9	67.3
	S+Original	72.4	47.7

Table II and Table III is a summary of the hyperparameters used for training the ResNet18 and Style-GAN 2 ADA models respectively. A ratio of 8:2 split was utilized for training and validation respectively.

The performance of the classifier was evaluated based on the testing set. The parameters used for evaluating the performance of the model are summarized below [119]:

1) Sensitivity also referred to as Recall represents the proportion of date images that were correctly classified (True Positive) as belonging to the hard, semi-hard and soft

classes respectively.

2) Specificity represents the proportion of date images that represented the true class but were misclassified (False Positive).
3) Positive predictive value (PPV) defines the ratio of the positive predicted date images that were correctly classified to the overall positive predicted date images.
4) Accuracy represents the ratio of the true predictions to the total number of predictions of the date images.
5) The Area under the Receiver Operating Characteristic (ROC) curve (AUC) represents the performance measure of the model to differentiate between the respective date classes i.e., hard, semi-hard and soft.
6) F1 Score is defined as a measure of the harmonic average of Recall and Precision also called PPV.

Table II. Summary of ResNet18 hyperparameters for training.

No.	Title	VALUE
A. Optimization parameters		
1	Learning rate	0.0001
2	Weight learn rate factor	10 units
3	Bias learn rate factor	10 units
4	Optimizer	'Adam'
5	Mini batch size	32
6	Epoch for 2-class model	50
7	Epoch for 3-class model	100
B. Model specific parameters		
7	Size of input image	$224\times 224\times 3$
8.	Number of Layers	71
9.	Number of Conv. layers	20
10	Fully Connected Layer	Weight- 2×512, Bias- 2×1, Activation- $1\times 1\times 2$

Table III. Hyperparameters for training the Style-GAN 2 ADA model.

No.	Title	VALUE
A.	Optimization parameters	
1	Learning rate	0.002
2	Beta1	0
3	Beta2	0.99
4	Optimizer	'Adam'
5	Mini batch size	4
6	Mini batch gpu	4
7	Epoch	1
B.	Model specific parameters	
8	Dataset for transfer learning	FFHQ1024
9	Number of images	25000
10	Number of Generator Layers	46
11	Number of Discriminator layers	28
12	Generator parameters	30370062
13	Discriminator parameters	29012513

Figure 13 and Figure 14 represents the training confusion matrix for the two-class and three-class classifiers respectively.

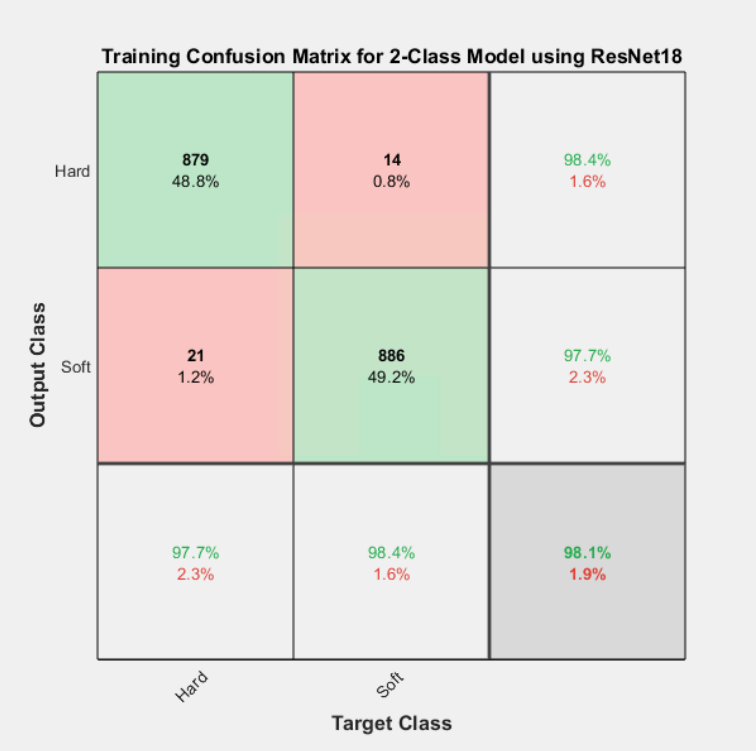


Figure 13. Training Confusion matrix for 2-Class ResNet18 Model.

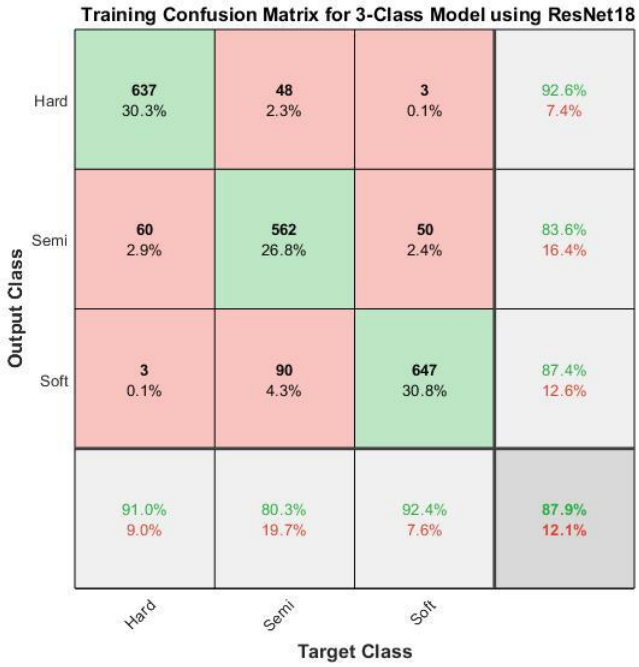


Figure 14. Training Confusion matrix for 3-Class ResNet18 Model.

Figure 15 and Figure 16 represents the confusion matrix on the testing set for the two-class and three-class classifiers respectively.

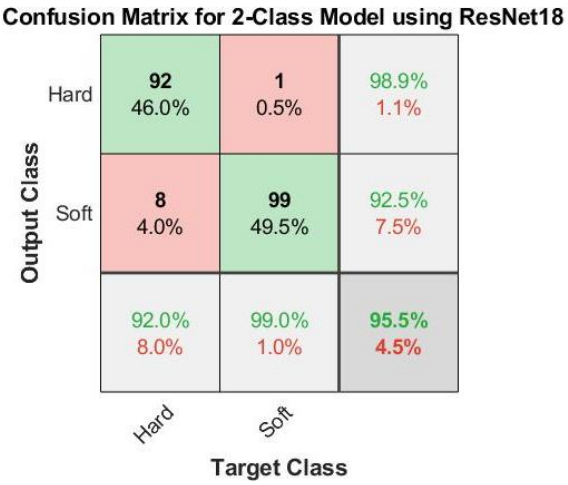


Figure 15. Test Confusion matrix for 2 class ResNet18 Model.

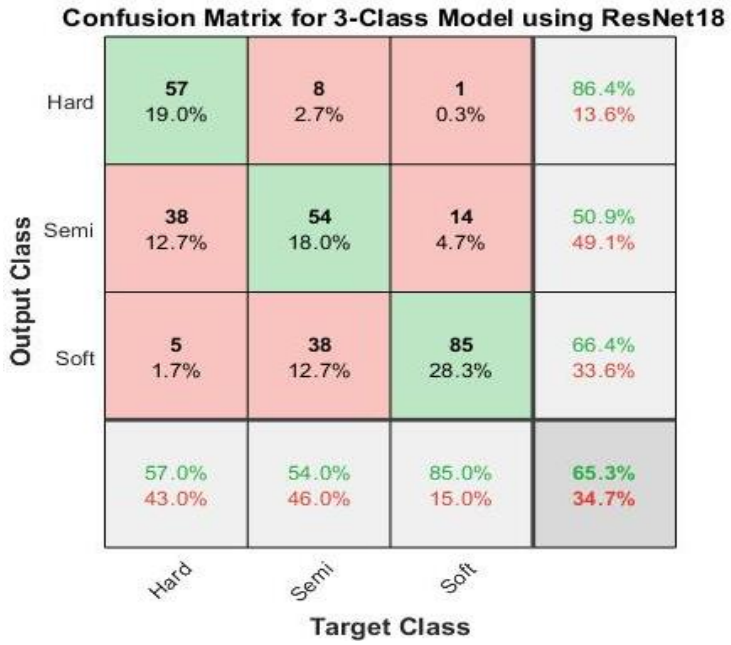


Figure 16. Test Confusion matrix for 3-class ResNet18 model.

As it can be seen in Figure 15 and Figure 16, the three-class classifier offers a rather poor performance, and the two-class classifier accuracy is the one that offers good results. As mentioned before, the scores of these classifiers are used in an expert system described in Figure 11 to combine the results of both. Figure 18 (a) shows the confusion matrix for the results obtained using this step.

Note how the great classification from the 2-class classifier for the soft dates was kept using this expert system. Also note how an improvement of 1% was obtained for the hard class. The not so good performance of the semi-hard class was also kept using the expert system as the one obtained using the 3-class classifier.

A modification in the expert system was made in which the selection of $C = \max\{P_A, P_B\}$ was changed for a decision to manually inspect the date by a human expert. We can note in Figure 17 that the decision for selecting $C = \max\{P_A, P_B\}$ on the 18 images evaluated in this part of the expert system is rather poor.

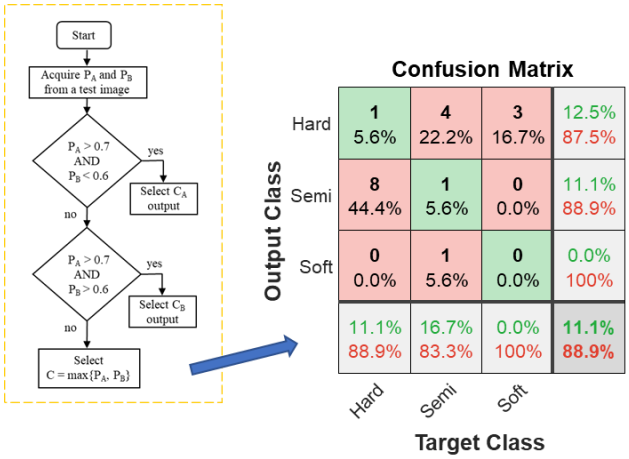


Figure 17 Confusion matrix of the 18 images out of the 300 that are evaluated at part C of the expert system.

Using this modification, the results are shown in the confusion matrix in Figure 18 (b). We can see that the misclassified single soft date in Figure 18 (a) and the seven misclassified hard dates also shown in Figure 18 (a), were selected for manual inspection by the expert system. Thus, after manual inspections of these dates, we can achieve an accuracy of 100% for these two classes. Nine misclassified semi-hard dates were chosen for this manual inspection improving the classification of this class to 61%. Thus, out of 300 test images, 18 are chosen to be manually selected to have this improvement.

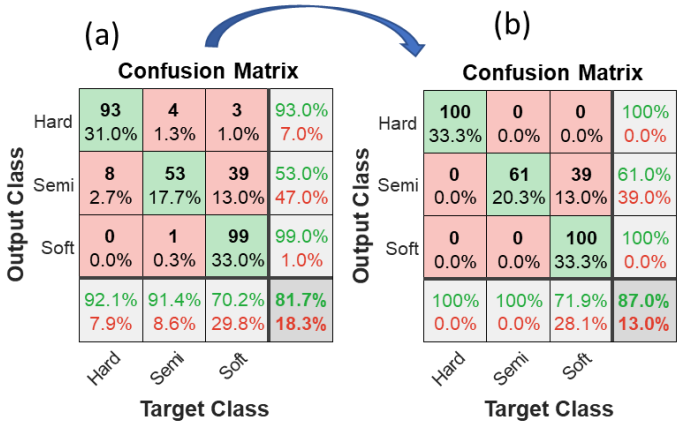


Figure 18. (a) Confusion matrix obtained using the expert system and (b) results when using manual inspection and classification of 18 dates.

The date images are classified with specificity and sensitivity of 98.4% and 97.7% respectively for the 2-class classifier and 86.4% and 91% respectively for the 3-class classifier. Figure 19 compares the overall accuracies of the existing methods available in the literature for the classification of dates based on hardness. We can see that the method

proposed in this thesis improved the classification significantly in the two-class scenario. However, the accuracy was quite poor in the three-class scenario but improved significantly using the expert system. Figure 20 and Figure 21 represents the convergence graph of accuracy and loss for the two class and three class ResNet18 model respectively.

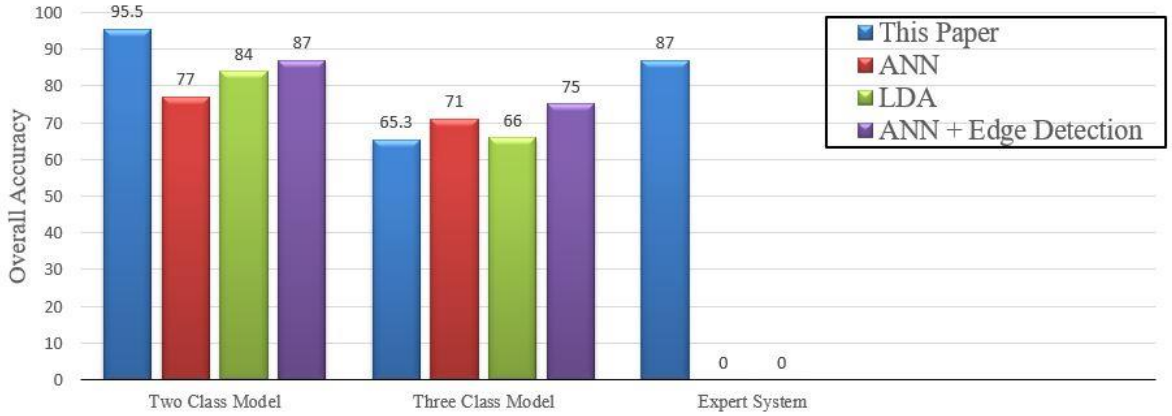


Figure 19. Comparison of the existing methods for the classification of dates based on hardness.

The AUC of 0.9981 and 0.981 is shown in Figure 22 and Figure 23 for the two class and three class model respectively. Table IV summarizes the output values of the performance metrics used for the evaluation of the ResNet18 model for the two class and three class model respectively (TP-True Positive, FP-False Positive, TN-True Negative, FN-False Negative, G-Mean-Geometric mean, P+N-represents the total number of samples)

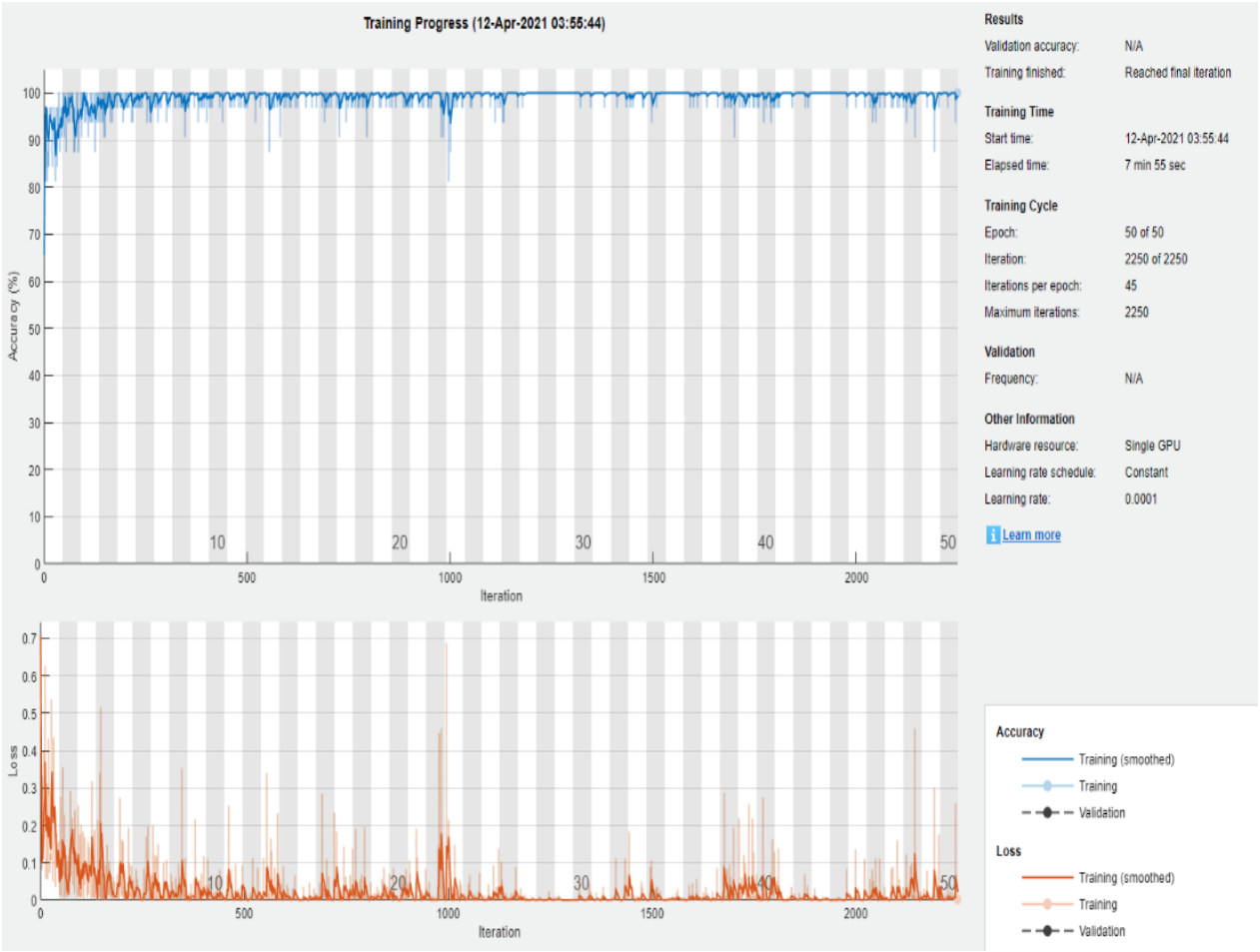


Figure 20. Convergence graph of accuracy and loss function for 2-Class model using ResNet18 for 50 epochs.

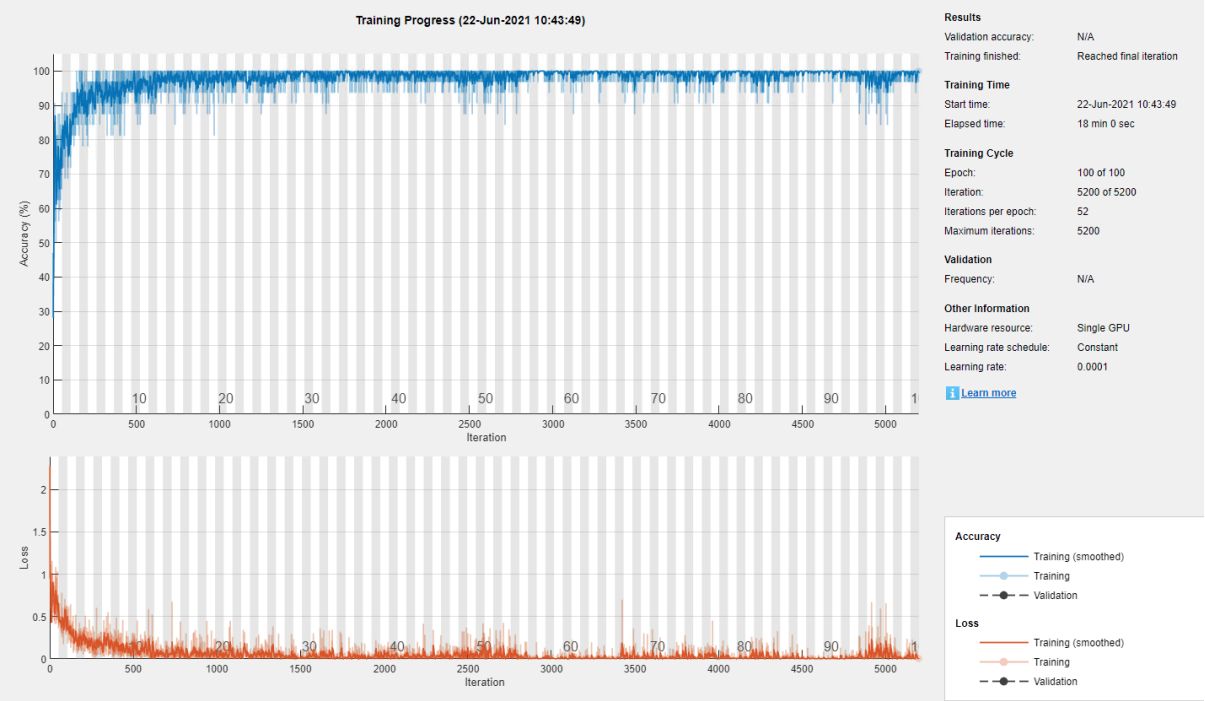


Figure 21. Convergence graph of accuracy and loss function for 3-Class model using ResNet18 for 100 epochs

Table IV. Performance metrics of the ResNet18 architecture.

Classifier	TP	TN	FP	FN	AUC	PRECISION (%) $\frac{TP}{(TP+FP)}$	G-MEAN (%) $\sqrt{\left(\frac{TP}{P}\times\frac{TN}{N}\right)}$	SENSITIVITY (%) $\frac{TP}{(TP+FN)}$	SPECIFICITY (%) $\frac{TN}{(TN+FP)}$	F1-SCORE (%) $\frac{2TP}{(2TP+FP+FN)}$	ACCURACY (%) $\frac{TP+TN}{(P+N)}$
2-Class	879	886	14	21	0.998	98.4	98.1	97.7	98.4	98.1	98.1
3-Class	637	1209	191	63	0.981	76.9	88.7	91	86.4	83.4	87.9

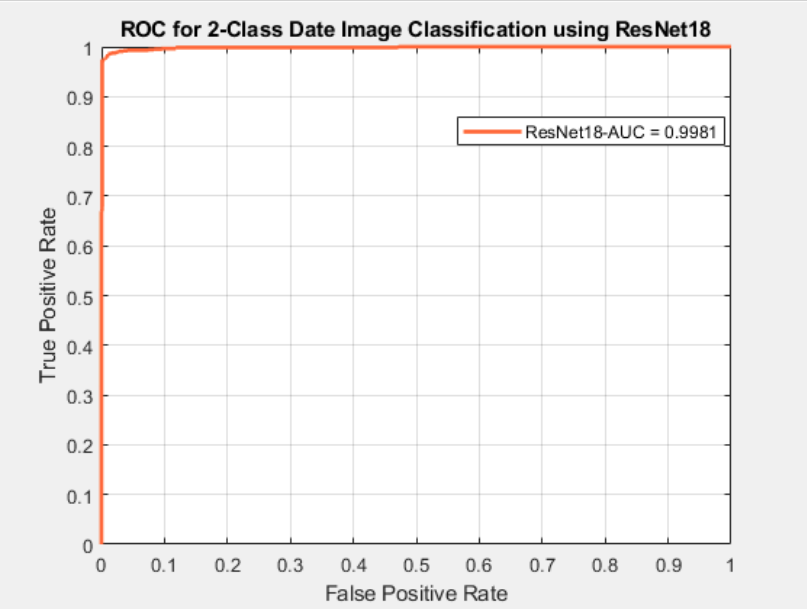


Figure 22. ROC Characteristics Curve of ResNet18 for 2-Class Date Image Classification.

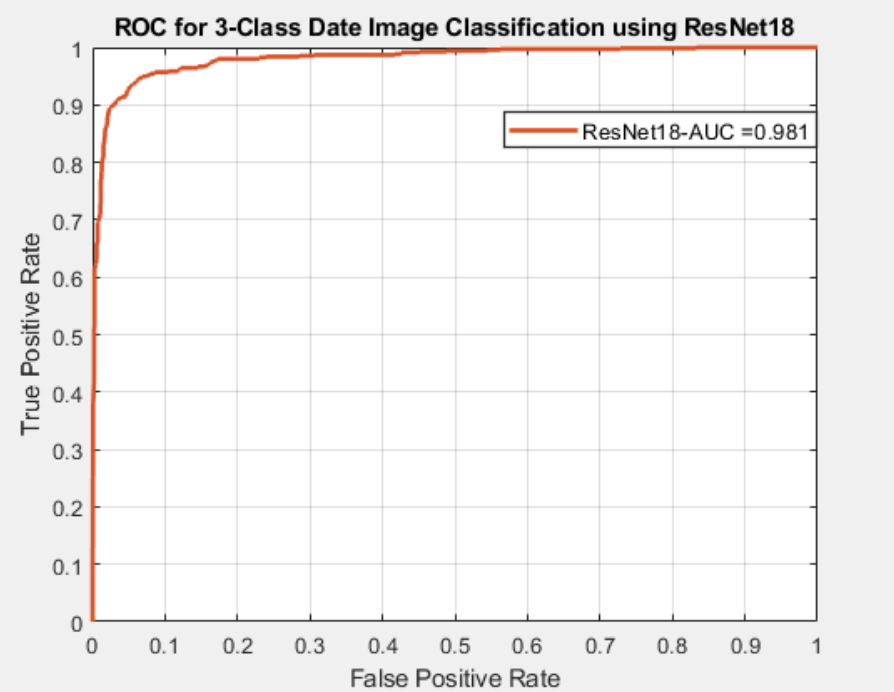


Figure 23. ROC Characteristics Curve of ResNet18 for 3-Class Date Image Classification.

As indicated before, texture features were evaluated from the original and GAN generated images to validate the quality of the GAN images. For $\alpha = 0.05$, the KS test rejected the hard images as being equally distributed. Otherwise for the soft images, they were found similarly

distributed. Figure 24 show these cases for the hard and soft Tamura features. The semi-hard cases were found to be similarly distributed.

Table V summarizes the similarity of the GAN and original images when using the KS test when testing 35 different features using 30 images per class. The percentage was calculated as the number of features found similar over the total number of features used.

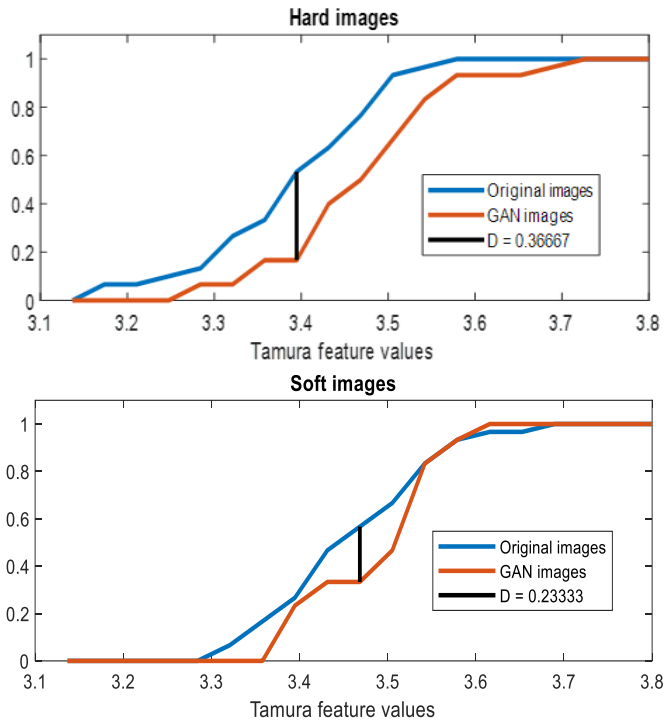


Figure 24. Plots of *F1,n* (blue), *F2,n* (orange) and *Dm,n* (black line) for the Tamura features.

Table V. Similarity of the Original and GAN generated images based on the KS Test.

Class	(%) of similarity
Hard	82.86
Semi-hard	100
Soft	94.29

CHAPTER 6

6.1 Conclusion

In this paper, a state-of-the-art sorting technique using a data augmentation approach based on Generative Adversarial Networks for date fruit classification into hard, semi-hard and soft classes was developed. This technique Used two ResNet18 networks trained with a 5-fold cross validation and an expert system to successfully classify the dates into respective classes based on hardness. A modification in the expert system incorporating manual inspection of the misclassified dates as a final quality control measure, improved the overall sorting technique significantly. Furthermore, 35 texture features were extracted from the original and generated GAN images and compared using a Kolmogorov-Smirnov test as a substitute for the Fréchet inception distance (FID) computing the overall percentage similarity for the hard, semi-hard and soft classes. This test further validated the use of the GAN images. The proposed system significantly outperformed similar systems available in literature.

6.2 Future Work

We plan to use the data augmentation approach with GANs for generating other datasets with different types of fruits and by extension adopt this technique in the agricultural industry for automatic sorting process of fruits. Also, we look forward to exploring Evidential Deep Learning, which seeks to learn the uncertainties what could be present in our pipeline and could lead to failure of our system. These uncertainties could in the data (Aleatoric uncertainties) or as a result of the trained classification models (Epistemic uncertainties). By exploring this area of research, we can correctly ascertain when our model makes a prediction, why it is making that prediction and whether to trust that prediction. In the future, we look forward to using the idea for generating and classifying audio signals. In addition, we look forward to applying this technique for the sorting of dates of other varieties.

6.3 Recommendations

The classification of the date images in this thesis was implemented using one deep learning model, particularly ResNet18 as it performed best on the date dataset. I would recommend the classification of the GAN generated date images to be further experimented with an ensemble of Deep learning classification models such as VGGNet, GoogleNet, AlexNet etc.

using other programming frameworks such as PyTorch and Tensorflow.

Also, metrics such as Fréchet Inception Distance (FID) and Inception Score (IS) could be investigated as it was not considered in the analysis of the date images due to computational power limitations.

Furthermore, the image acquisition process could be investigated using an RGB camera.

REFERENCES

[1] K.O Okokpujie, S.N. John, D. Ufuah, M.Nwagu, E. Noma-Osaghae, C.U. Ndujiuba and I. P. Okokpujie, "Development of an Illumination Invariant Face Recognition System," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 5, pp. 2278-3091, 2020.

[2] Susmita Ray, "A Quick Review of Machine Learning Algorithms," *2019 International Conference on Machine Learning Big Data Cloud and Parallel Computing*, pp. 35-39, Feb. 2019.

[3] A. Khamis, H. Li, E. Prestes and T. Haidegger, "AI: A Key Enabler for Sustainable Development Goals: Part 2 [Industry Activities]," *in IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 122-127, Dec. 2019, doi: 10.1109/MRA.2019.2945739.

[4] M. Darwish, "Fruit Classification Using Covolutional Neural Network," PhD dissertation, Dept. Comp.Eng., Near East University, 2020.

[5] Food and Agriculture Organization Corporate Article. (2019). *FAO-STAT*. Accessed: Nov. 28, 2019. [Online]. Available: http://www.fao.org/fao-stories/article/en/c/1251859/

[6] Food and Agriculture Organization Corporate Statistical. (2016). *FAO-STAT*. Accessed: Oct. 18, 2018. [Online]. Available: http://www.fao.org/faostat/en/#data/QC.

[7] H. Altaheri, M. Alsulaiman and G. Muhammad, "Date Fruit Classification for Robotic Harvesting in a Natural Environment Using Deep Learning," *in IEEE Access*, vol. 7, pp. 117115-117133, 2019, doi: 10.1109/ACCESS.2019.2936536.

[8] E. Saleh, M. Tawfik and H. Abu-Tarboush, "Phenolic Contents and Antioxidant Activity of Various Date Palm (Phoenix dactylifera L.) Fruits from Saudi Arabia," *Food and Nutrition Sciences*, Vol. 2 No. 10, 2011, pp. 1134-1141. doi: 10.4236/fns.2011.210152.

[9] G. Muhammad, "Date fruits classification using texture descriptors and shape-size features," *Eng. Appl. Artif. Intell.*, vol. 37, pp. 361-367, 2015.

[10] A. Al-Abbad, M. Al-Jamal, Z. Al-Elaiw, F. Al-Shreed and H. Belaifa, "A study on the economic feasibility of date palm cultivation in the Al-Hassa Oasis of Saudi Arabia," *J. Develop. Agricult. Econ.*, vol. 3, pp. 463-468, 2011.

[11] A. Kavvada and A. Held, "Analysis-Ready Earth Observation Data and the United Nations Sustainable Development Goals," *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 434-436, doi: 10.1109/IGARSS.2018.8519405.

[12] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798-1828, Aug. 2013, doi: 10.1109/TPAMI.2013.50.

[13] A. Antoniou, A. Storkey and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1701.04340*, 2017.

[14] K.O Okokpujie, A. Reuben, J.C. Ofoche and I. P. Okokpujie, "A Comparative Analysis Performance of Data Augmentation on Age-Invariant Face Recognition using Pretrained Residual Neural Network," *Journal of Theoretical and Applied Information Technology,* vol. 99, no. 6, pp. 1817-3195, 2021.

[15] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan and Y. Zheng, "Recent Progress on Generative Adversarial Networks (GANs): A Survey," *in IEEE Access,* vol. 7, pp. 36322-36333, 2019, doi: 10.1109/ACCESS.2019.2905015.

[16] J. Gill, P.S. Sandhu and T. Singh, "A Review of Automatic Fruit Classification using Soft Computing Techniques," *International Conference on Computer Systems and Electronics Engineering (ICSCEE'2014),* pp. 91-98, 2014.

[17] Y. A. Ohali, "Computer vision based date fruit grading system: Design and implementation," *J. King Saud Univ. Comput. Inf. Sci.,* vol. 23, no. 1, pp. 29-36, 2011.

[18] N. Kondo, "Fruit grading robot," *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, 2003, pp. 1366-1371 vol.2, doi: 10.1109/AIM.2003.1225542.

[19] Ismail Kavdir and Daniel E. Guyer, "Apple Grading Using Fuzzy Logic," *Turkish Journal of Agriculture and Forestry 2003*, vol. 27, no. 6, pp. 375-382, August 2003.

[20] G. Lin, Y. Tang, X. Zou, J. Cheng and J. Xiong, "Fruit detection in natural environment using partial shape matching and probabilistic Hough transform," *Precis. Agricult.*, vol. 21, no. 1, pp. 160-177, Feb. 2020.

[21] B. Zhu, L. Jiang, Y. Luo and Y. Tao, "Gabor feature-based apple quality inspection using kernel principal component analysis," *J. Food Eng.*, vol. 81, no. 4, pp. 741-749, Aug. 2007.

[22] D. Unay, B. Gosselin, O. Kleynen, V. Leemans, M. Destain and O. Debeir, "Automatic grading of Bi-colored apples by multispectral machine vision," *Computers and Electronics in Agriculture*, vol. 75, no. 1, pp. 204-212, 2011.

[23] G. Leiva, G. Mondragon, D. Mery, and J. M. Aguilera, "The automatic sorting using image processing improves postharvest blueberries storage quality," *in Proceedings of the International Congress on Engineering and Food,* 2011.

[24] Kim DaeGwan, Thomas F. Burks and Duke M. Bulanon Jianwei Qin, "Classification of grapefruit peel diseases using color texture feature analysis," *International Journal on Agriculture and Biological Engineering*, pp. 41-50, Sept 2009.

[25] A. Esehaghbeygi, M. Ardforoushan, S.A.H. Monajemi and A.A. Masoumi, "Digital image processing for quality ranking of saffron peach," *International Agrophysics*, vol. 24, no. 2, pp. 115-120, 2010.

[26] A. Alipasandi, H. Ghaffari and S. Z. Alibeyglu, "Classification of Three Varieties of Peach Fruit Using Artificial Neural Network Assisted with Image Processing Techniques," *International Journal of Agronomy and Plant Production*, vol. 4, no. 9, pp. 2179-2186, 2013.

[27] A. S. Simoes *et al.,* "Appling Neural Networks to Automated Visual Fruit Sorting," *Transaction of the ASAE,* 2002.

[28] J. P. Mercol, M. J. Gambini and J. M. Santos, "Automatic classification of oranges using image processing and data mining techniques," 2008.

[29] P. Yimyam, T. Chalidabhongse, P. Sirisomboon and S. Boonmung, "Physical properties analysis of mango using computer vision," *Proc. ICCAS*, pp. 1-5, Jun. 2005.

[30] E. Llobet, E. L. Hines, J. W. Gardner and S. Franco, "Non-destructive banana ripeness determination using a neural network-based electronic nose," *Meas. Sci. Technol.*, vol. 10, pp. 538-548, 1999.

[31] M. Faisal, F. Albogamy, H. Elgibreen, M. Algabri and F. A. Alqershi, "Deep Learning and Computer Vision for Estimating Date Fruits Type, Maturity Level, and Weight," *in IEEE Access*, vol. 8, pp. 206770-206782, 2020, doi: 10.1109/ACCESS.2020.3037948.

[32] I. H. Ali, M. T. AmarNishad and A.M. A. Khaled, "Development of a Neural Network Classifier for Date Fruit Varieties Using Some Physical Attributes," *Res. Bult.,* Agric. Res. Center, King Saud Univ., no. 126, pp. 5-18, 2003.

[33] D. Wulfsohn, Y. Sarig and R. V. Algazi, "Defect sorting of dry dates by image analysis," *Can. Agricult. Eng.,* vol. 35, no. 2, pp. 133-139, Apr. 1993.

[34] A. Haidar, Haiwei Dong and N. Mavridis, "Image-based date fruit classification," *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems,* 2012, pp. 357-363, doi: 10.1109/ICUMT.2012.6459693.

[35] M. Fadel, "Date fruits classification using probabilistic neural networks," *Agric. Eng. Int. CIGR J.*, vol. 9, pp. 1989, Dec. 2007.

[36] K. M. Alrajeh and T. A. Alzohairy, "Date Fruits Classification using MLP and RBF Neural Networks", *International Journal of Computer Applications,* vol. 41, no. 10, pp. 36-41, 2012.

[37] D.J. Lee, R. Schoenberger, J. Archibald and S. Mccollum, "Development of a machine vision system for automatic date grading using digital reflective near-infrared imaging," *J. Food Eng*, vol. 86, no. 3, pp. 388-398, 2008.

[38] Z. Schmilovitch, A. Hoffman, H. Egozi, R. Ben-Zvi, "Machine for automatic sorting 'barhi' dates according to maturity by near infrared spectrometry," *In Proc. of the 4th International Conference on Postharvest Science,* Jerusalem, Israel, 2001, pp. 481-486.

[39] A. Manickavasagan, N. H. Al-Shekaili, N. K. Al-Mezeini, M. S. Rahman and N. Guizani, "Computer vision technique to classify dates based on hardness," *Journal of Agricultural and Marine Sciences [JAMS],* vol. 22, no. 1, pp. 36-41, 2018.

[40] A.Manickavasagan, H.N. Al-Shekaili, G. Thomas, M.S Rahman, N.Guizani and D.S Jayas, "Edge Detection Features to Evaluate Hardness of Dates Using Monochrome Images," *in Food and Bioprocess Technology,* vol. 7, no. 8, pp. 2251-2258, 2014, doi: 10.1007/s11947-013-1219-0.

[41] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair *et al.*, "Generative adversarial nets," *In Advances in Neural Information Processing Systems,* pp. 2672-2680, 2014.

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *in Proc. Int. Conf. Learn. Represent.,* 2014, pp. 1-14. [Online]. Available: https://arxiv.org/abs/1312.6114.

[43] H. Akaike, "Fitting autoregressive models for prediction," *Ann. Inst. Stat. Math.*, vol. 21, no. 1, pp. 243-247, Dec. 1969. doi: 10.1007/BF02532251.

[44] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *in Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 1747-1756. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045390.3045575.

[45] L. Yi and M. Mak, "Adversarial Data Augmentation Network for Speech Emotion Recognition," *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC),* 2019, pp. 529-534, doi: 0.1109/APSIPAASC47483.2019.9023347.

[46] G. Rizos, A. Baird, M. Elliott and B. Schuller, "Stargan for Emotional Speech Conversion: Validated by Data Augmentation of End-To-End Emotion Recognition," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2020, pp. 3502-3506, doi: 10.1109/ICASSP40776.2020.9054579.

[47] D. Wan, H. Dong-Yan, L. Danqing and Z. Yuexian, "Speech Emotion Recognition using Spectral Normalized CycleGAN," *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW),* 2019, pp. 93-99, doi: 10.1109/ACIIW.2019.8925058.

[48] Z. -J. Xu, R. -F. Wang, J. Wang and D. -H. Yu, "Parkinson's Disease Detection Based on Spectrogram-Deep Convolutional Generative Adversarial Network Sample Augmentation," *in IEEE Access,* vol. 8, pp. 206888-206900, 2020, doi: 10.1109/ACCESS.2020.3037775.

[49] F. Fahimi, S. Dosen, K. K. Ang, N. Mrachacz-Kersting and C. Guan, "Generative Adversarial Networks-Based Data Augmentation for Brain-Computer Interface," *in IEEE Transactions on Neural Networks and Learning Systems,* doi: 10.1109/TNNLS.2020.3016666.

[50] T. Schlegl, P. Seeböck, S. M.Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker

discovery," *in Information Processing in Medical Imaging.* Berlin, Germany: Springer, 2017, pp. 146-157.

[51] N. Killoran, L. J. Lee, A. Delong, D. Duvenaud, and B. J. Frey, "Generating and designing DNA with deep generative models," *in Proc. Adv. Neural Inf. Process. Syst.,* 2017, pp. 1-19.

[52] J. Deng, G. Pang, Z. Zhang, Z. Pang, H. Yang and G. Yang, "cGAN Based Facial Expression Recognition for Human-Robot Interaction," *in IEEE Access,* vol. 7, pp. 9848-9859, 2019, doi: 10.1109/ACCESS.2019.2891668.

[53] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis," *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV),* Oct. 2017, pp. 2439-2448.

[54] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," *in Proc. Int. Conf. Learn. Represent., 2019,* pp. 1-35. [Online]. Available: https://openreview.net/forum?id=B1xsqj09Fm.

[55] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen and T. Aila, "Training generative adversarial networks with limited data," *Proc. NeurIPS,* 2020.

[56] S. Tulyakov, M. Liu, X. Yang and J. Kautz, "MoCoGAN: Decomposing Motion and Content for Video Generation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2018, pp. 1526-1535, doi: 10.1109/CVPR.2018.00165.

[57] I. K. Dutta, B. Ghosh, A. Carlson, M. Totaro and M. Bayoumi, "Generative Adversarial Networks in Security: A Survey," *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON),* 2020, pp. 0399-0405, doi: 10.1109/UEMCON51285.2020.9298135.

[58] W. Hu and Y. Tan. (Feb. 2017). "Generating adversarial malware examples for black-box attacks based on GAN," [Online]. Available: https://arxiv.org/abs/1702.05983.

[59] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deepconvolutional generative adversarial networks," *in Proc. Int. Conf. Learn. Represent.,* 2016, pp. 1-16. [Online]. Available: https://arxiv.org/abs/1511.06434.

[60] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, [Online]. Available: https://arxiv.org/abs/1411.1784.

[61] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," *in Proc. 29th Adv. Neural Inf. Process. Syst.,* 2016, pp. 2172-2180. [Online]. Available: http://papers.nips.cc/paper/6399-infogan-interpretable-representationlearning-by-information-maximizing-generative-adversarial-nets.pdf.

[62] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," *in Proc. 34th Int. Conf. Mach. Learn.,* vol. 70, Aug. 2017, pp. 2642-2651. [Online]. Available: http://proceedings.mlr.press/v70/odena17a.html.

[63] Z. Zhao, S. Singh, H. Lee, Z. Zhang, A. Odena and H. Zhang, "Improved Consistency Regularization for GANs," *CoRR*, vol. abs/2002.04724, pp. 1-3, 2020.

[64] 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," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 6626-6637, 2017, [online]. Available: http://papers.nips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-equilibrium.pdf.

[65] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen and T. Aila, "Analyzing and improving the image quality of StyleGAN," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 8110-8119, Jun. 2020.

[66] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *in Proc. 29th Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234-2242. [Online]. Available: http://papers.nips.cc/paper/6125-improved-techniques-fortraining-gans.pdf.

[67] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," *in Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: https://openreview.net/forum?id=r1lUOzWCW.

[68] S. Nowozin, B. Cseke and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," *in Advances in Neural Information Processing Systems, Red Hook, NY, USA: Curran Associates*, pp. 271-279, 2016. [Online]. Available: http://papers.nips.cc/paper/6066-f-gan-training-generativeneural-samplers-using-variational-divergence-minimization.pdf.

[69] M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, "Assessing generative models via precision and recall," *in Proc. Neural Inf. Process. Syst.*, 2018, pp. 1-6.

[70] T. Karras, S. Laine and T. Aila, "A style-based generator architecture for generative adversarial networks," *CoRR*, vol. abs/1812.04948, 2018.

[71] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," *Proc. Int. Conf. Learn. Represent. (ICLR)*, pp. 1-22, 2019.

[72] A. Borji, "Pros and Cons of GAN evaluation measures," Feb. 2018, [Online]. Available: http://arxiv.org/abs/1802.03446.

[73] Q. Xu et al., "An empirical study on evaluation metrics of generative adversarial networks," *arXiv:1806.07755*, 2018, [Online]. Available: https://arxiv.org/abs/1806.07755.

[74] Yaochu Jin, T. Okabe and B. Sendhoff, "Neural network regularization and ensembling using multi-objective evolutionary algorithms," *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, 2004, pp. 1-8 Vol.1, doi: 10.1109/CEC.2004.1330830.

[75] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, Oxford, UK, 1995.

[76] C. M. Bishop, "Curvature-driven smoothing: a learning algorithm for feedforward networks," *in IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 882-884, Sept. 1993, doi: 10.1109/72.248466.

[77] M. Lee, and J. Seok, "Regularization methods for generative adversarial networks: An overview of recent studies," *in arXiv:2005.09165*, 2020, [online] Available: https://arxiv.org/abs/2005.09165.

[78] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. C. Courville, "Improved training of Wasserstein GANs", Proc. NIPS, pp. 5767-5777, 2017.

[79] M. Arjovsky, S. Chintala and L. Bottou, "Wasserstein generative adversarial networks," *Proc. Int. Conf. Mach. Learn.*, vol. 70, pp. 214-223, 2017.

[80] N. Kodali, J. Abernethy, J. Hays and Z. Kira, "On convergence and stability of GANs" *in arXiv:1705.07215*, 2017, [online] Available: https://arxiv.org/abs/1705.07215.

[81] K. Roth, A. Lucchi, S. Nowozin and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 2018-2028, 2017.

[82] H. Zhang, Z. Zhang, A. Odena, and H. Lee, "Consistency regularization for generative adversarial networks," *in International Conference on Learning Representations (ICLR)*, pp. 1-3, 2020.

[83] X. Wei, B. Gong, Z. Liu, W. Lu and L. Wang, "Improving the improved training of wasserstein gans: A consistency term and its dual effect," *International Conference on Learning Representation(ICLR)*, 2018.

[84] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1510-1519, 2017.

[85] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu and M. Song, "Neural Style Transfer: A Review," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3365-3385, 1 Nov. 2020, doi: 10.1109/TVCG.2019.2921336.

[86] L. A. Gatys, A. S. Ecker and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2414-2423.

[87] M. Ruder, A. Dosovitskiy and T. Brox, "Artistic style transfer for videos," *Proc. German Conf. Pattern Recognit.*, pp. 26-36, 2016.

[88] X. Wang, G. Oxholm, D. Zhang and Y.-F. Wang, "Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 5239-5247, Jul. 2017.

[89] V. Dumoulin, J. Shlens and M. Kudlur, "A learned representation for artistic style," *Proc. Int. Conf. Learn. Representations*, 2017.

[90] Y. Li, F. Chen, J. Yang, Z. Wang, X. Lu and M.-H. Yang, "Diversified texture synthesis with feed-forward networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3920-3928, 2017.

[91] Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," *Proc. NIPS Workshop Constructive Mach. Learn.*, pp. 1-9, 2016.

[92] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proc. ICML*, pp. 448-456, 2015.

[93] T. Miyato, T. Kataoka, M. Koyama and Y. Yoshida, "Spectral normalization for generative adversarial networks," *Proc. Int. Conf. Learn. Represent.*, 2018, [online] Available: https://arxiv.org/abs/1802.05957.

[94] D. Ulyanov, A. Vedaldi and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 6924-6932, 2017.

[95] N. R. Goodman, "Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction)," *Ann. Math. Statist.*, vol. 34, no. 1, pp. 152-177, 1963.

[96] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on ComputerVision and Pattern Recognition*, pp. 770-778, 2016.

[97] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157-166, Mar. 1994.

[98] X. Glorot, A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks," *Proc. Int. Conf. Artificial Intelligence and Statistics*, pp. 315-323, 2011.

[99] A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," in *IEEE Access*, vol. 7, pp. 53040-53065, 2019, doi: 10.1109/ACCESS.2019.2912200.

[100] P. Ramachandran, B. Zoph and Q. V. Le, "Searching for Activation Functions," *Proc. Int. Conf. Learn. Representations*, pp. 1-13, 2017.

[101] B. Xu, N. Wang, T. Chen and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv:1505.00853, 2015, [Online]. Available: http://arxiv.org/abs/1505.00853.

[102] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," arXiv:1803.08375, Feb. 2019, [Online]. Available: http://arxiv.org/abs/1803.08375.

[103] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th International Conference on International Conference on Machine Learning ser. ICML'10. USA: Omnipress*, pp. 807-814, 2010.

[104] A. Canziani, A. Paszke and E. Culurciello, “An analysis of deep neural network models for practical applications,” *arXiv:1605.07678,* 2016, [Online]. Available: http://arxiv.org/abs/1605.07678.

[105] S. Ahuja, B. Panigrahi, N. Dey, T. Gandhi and V. Rajinikanth, “Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices,” *Techrxiv,* 2020.

[106] Y. Zhou, X. Wang, M. Zhang, J. Zhu, R. Zheng and Q. Wu, “MPCE: A Maximum Probability Based Cross Entropy Loss Function for Neural Network Classification,” *in IEEE Access,* vol. 7, pp. 146331-146341, 2019, doi: 10.1109/ACCESS.2019.2946264.

[107] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *in CoRR,* Dec. 2014.

[108] M. S. Kurzyn, “Expert systems and neural networks: a comparison,” *Proceedings 1993 The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, 1993, pp. 222-223, doi: 10.1109/ANNES.1993.323038.

[109] J. Villena-Roman, S. Collada-Perez, S. Lana-Serrano and J. C. G. Cristobal, “Hybrid approach combining machine learning and a rule-based expert system for text categorization,” *Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference* May 18–20 2011, 2011.

[110] M. Sharma and H. Ghosh, “Histogram of gradient magnitudes: A rotation invariant texture-descriptor,” *2015 IEEE International Conference on Image Processing (ICIP),* 2015, pp. 4614-4618, doi: 10.1109/ICIP.2015.7351681.

[111] H. Tamura, S. Mori and T. Yamawaki, “Textural Features Corresponding to Visual Perception,” *in IEEE Transactions on Systems, Man, and Cybernetics,* vol. 8, no. 6, pp. 460-473, June 1978, doi: 10.1109/TSMC.1978.4309999.

[112] X. Wu, J. Peng, J. Shan and W. Cui, “Evaluation of semivariogram features for object-based image classification,” *Geo-spatial Information Science,* pp. 159-170, 2015.

[113] S. Naik and B. Patel, “Machine Vision based Fruit Clssification and Grading - A Review,” *International Journal of Computer Applications*, vol. 170, no. 9, pp. 22-34, July 2017.

[114] A. Olsen, S. Han, B. Calvert, P. Ridd and O, Kenny, “In Situ Leaf Classification Using Histograms of Oriented Gradients,” *In Proc. of the 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2015.

[115] F. J. Massey, “The Kolmogorov-Smirnov test for goodness of fit,” *Journal of the American Statistical Association,* vol. 46, no. 253, pp. 68-78, 1951.
[116] Z. W. Birnbaum, “Numerical Tabulation of the Distribution of Kolmogorov's Statistic for Finite Sample Size,” *Journal of the American Statistical Association,* Vol. 47, pp: 425-44, 1952.

[117] Z. Situ, S. Teng, H. Liu, J. Luo and Q. Zhou, “Automated Sewer Defects Detection Using Style-Based Generative Adversarial Networks and Fine-Tuned Well-Known CNN

Classifier," *in IEEE Access*, vol. 9, pp. 59498-59507, 2021, doi: 10.1109/ACCESS.2021.3073915.

[118] G. Thomas, S. Balocco, D. Mann, A. Simundsson and N. Khorasani, "Intelligent Agricultural Machinery Using Deep Learning," *in IEEE Instrumentation & Measurement Magazine*, vol. 24, no. 2, pp. 93-100, April 2021, doi: 10.1109/MIM.2021.9400957.

[119] C. Nuzzi, S. Pasinetti, M. Lancini, F. Docchio and G. Sansoni, "Deep learning-based hand gesture recognition for collaborative robots," *in IEEE Instrumentation & Measurement Magazine*, vol. 22, no. 2, pp. 44-51, April 2019, doi: 10.1109/MIM.2019.8674634.