ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

WORDS AS ART MATERIALS: GENERATING PAINTINGS WITH SEQUENTIAL GENERATIVE ADVERSARIAL NETWORKS

M.Sc. THESIS

Azmi Can ÖZGEN

Department of Computer Engineering

Computer Engineering Programme

JULY 2020

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

WORDS AS ART MATERIALS: GENERATING PAINTINGS WITH SEQUENTIAL GENERATIVE ADVERSARIAL NETWORKS

M.Sc. THESIS

Azmi Can ÖZGEN
(504161535)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Hazım Kemal EKENEL

JULY 2020

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SANAT MATERYALİ OLARAK KELİMELER: SERİ ÜRETİCİ ÇEKİŞMELİ AĞLAR İLE SANATSAL RESİM ÜRETİMİ

YÜKSEK LİSANS TEZİ

Azmi Can ÖZGEN
(504161535)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Hazım Kemal EKENEL

TEMMUZ 2020

Azmi Can ÖZGEN, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504161535 successfully defended the thesis entitled "WORDS AS ART MATERIALS: GENERATING PAINTINGS WITH SEQUENTIAL GENERATIVE ADVERSARIAL NETWORKS", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Hazım Kemal EKENEL** Istanbul Technical University	.....
Jury Members :	**Assist. Prof. Dr. İlkay Öksüz** Istanbul Technical University	.....
	Assoc. Prof. Dr. Aykut Erdem Hacettepe University	.....
		.....

Date of Submission : **14 June 2020**
Date of Defense : **14 July 2020**

To my family,

FOREWORD

I am very thankful to my advisor, Prof. Dr. Hazım Kemal EKENEL for his support and guidance in both this work and in my academic career.

I am grateful to my family for their unconditional support throughout my whole life.

Finally, I would like to thank the members of SiMiT Lab, my teammate Şeymanur Aktı, Alperen Kantarcı, İrem Eyiokur, Doğucan Yaman, Omid Abdollahi Aghdam, Anıl Genç, Deniz Engin, Aydın Ayanzadeh, Emre Kolaç for all the supports and their valuable friendship.

July 2020 Azmi Can ÖZGEN

TABLE OF CONTENTS

Page

ABBREVIATIONS

GAN	**:** Generative Adversarial Networks
CNN	**:** Convolutional Neural Network
RNN	**:** Recurrent Neural Network
WGAN-GP	**:** Wasserstein Generative Adversarial Networks with Gradient Penalty
FID	**:** Fréchet Inception Distance
IS	**:** Inception Score

LIST OF TABLES

Page

LIST OF FIGURES

Page

WORDS AS ART MATERIALS:
GENERATING PAINTINGS
WITH SEQUENTIAL GENERATIVE ADVERSARIAL NETWORKS

SUMMARY

Generative adversarial networks are one of the most popular topics and widely researched area of computer vision. They have been developed a couple of years ago as an unsupervised learning method. These models can easily be utilized as feature learners, and they are used in many interesting applications such as generating realistic images, generating sketch images, image-to-image translation, 3D object generation, image inpainting, generating super-resolution images, voice/music generation, anomaly detection, generating synthetic datasets, adversarial attacks. We focused on text-to-image synthesis in this work. Specifically, we synthesize painting images from a given set of keywords. The aim is to generate unique artificial painting images that reflect given keywords in their contents.

Converting text descriptions into images using Generative Adversarial Networks has become a popular research area. Visually appealing images have been generated successfully in recent years. There is a remarkable amount of studies in the literature regarding this topic. Inspired by these studies, we investigated the generation of artistic images on a large variance dataset. As an important part of this work, we created our dataset. Dataset has 3492 images with classical paintings and also digital paintings. They both provided a mixture of different painting styles for our work and we believe that this dataset will be a good resource for other researchers as well. Therefore, we shared our dataset publicly to be used in artificial art generation studies.

One major characteristic of our work is that we used keywords as image descriptions, instead of sentences. Therefore, our dataset consists of image-keyword pairs and includes images with variations, for example, in shape, color, and content. These variations in images provide originality which is an important factor for artistic essence. However, those variations make learning difficult at the same time. As a result of this difficulty, we designed a sequential deep neural network model.

As our neural network model, we proposed a sequential Generative Adversarial Network model which consists of three separate stages to handle images and keyword sets simultaneously. Keyword and image pairs are processed through the stages of neural network layers. The first stage of this sequential model processes the word vectors and creates a base image whereas the next stages focus on creating high-resolution artistic-style images without working on word vectors. To deal with the unstable nature of GANs, we proposed a mixture of techniques like Wasserstein loss, spectral normalization, and minibatch discrimination. Besides, we set a special hyper-parameter set separately for all the networks used in the model.

Ultimately, we were able to generate painting images, which have a variety of styles. Evaluation of the results is one very critical part of the work. We provided both quantitative and qualitative analyses for them. As a quantitative analysis, we evaluated

our results by using the Fréchet Inception Distance score. For the qualitative analysis, we conducted an extensive user study with 186 participants. Both analysis objectively supported that our painting images can be considered as artworks.

SANAT MATERYALİ OLARAK KELİMELER: SERİ ÜRETİCİ ÇEKİŞMELİ AĞLAR İLE SANATSAL RESİM ÜRETİMİ

ÖZET

Yapay zekanın sanat üretme yeteneği, bilgisayar biliminin tartışmalı konularından biridir. Yapay zeka gerçek sanat yaratabilir mi, yoksa bilgisayarla üretilen eserler sıkıcı sayılar ve denklemlerden oluşan bir matristen daha fazlası değil mi? Bu sorunun cevabı araştırmacılar arasında ve hatta yapay zeka ile ilgilenen insanlar arasında uzun zamandır düşünülüyordu. Bu çalışmaya başlamadan önce bu sorunun cevabını merak ediyorduk ve başlarken en büyük motivasyonumuz bu oldu.

Çoğu zaman sanat, duyguları ve fikirleri aktarma çabası olduğu için bir insan eylemi olarak kabul edilir. Öte yandan, sanatsal üretimin deterministik ve mantıksal süreçlerin sonunda ortaya çıktığını iddia edenler de var. Mantıksal bir süreç olsa bile, sanat üretmenin sadece insan beyninin üstesinden gelebileceği bir şey olabileceği ya da sadece insan beyni tarafından üretildiğinde bir sanat eseri olarak düşünülebileceği de söz konusu olabilir. Başka bir deyişle, bir eseri yalnızca insan zihni, duyguları ve deneyimleri sonucunda üretildiğinde bir sanat eseri olarak değerlendiriyor olabiliriz. Tersine, eğer bir sanat eseri insan aklının bir sonucu değilse, o zaman onu rastgele bir üretim olarak düşünme eğiliminde olmamız olası. Tüm bu soru ve fikirleri düşünmeyi okuyuculara ve uzmanlara bırakıyoruz, çünkü bu fikirler bu çalışmanın felsefeye dokunan kısmı. Tezin amacına odaklanırsak, sanat eserleri olarak kabul edilen insan yapımı resimlerle eğitilmiş bir yapay sinir ağı modeliyle yapay resimler üretmeyi ve bu yapay resimlerin sanat eserleri olup olmadığını sorgulamayı amaçladık.

Diğer amaçlardan biri ise, yazarlar ve ressamlar gibi sanatsal üretim yapmak için ilham alması gereken kişiler tarafından kullanılabilecek bir uygulama prototipi oluşturmaktır. Bu insanlar, modelimiz tarafından üretilen görüntüleri stok fotoğraf internet sitelerinde sunulan görüntülere benzer şekilde kullanabilirler. Çünkü çalışmadaki modelimiz bir anahtar kelime listesini girdi olarak alıyor ve bu kelimeleri sanatsal bir resme dönüştürülebiliyor.

Bu çalışmanın ve benzerlerinin sanatsal üretim yapabilen bir bilgisayar programı olarak kendi başlarına değerli olduğunu düşünmemizin yanı sıra, böyle programların endüstrinin bazı alanlarında kullanabilecek çeşitli uygulamaların kaynağı da olabileceğini düşünüyoruz. Bunlardan bazıları olarak grafik tasarım, oyun tasarımı ve reklamcılık gibi sanatsal tasarımın ürünün önemli bir parçası olduğu alanları sayabiliriz. Yine de, böyle çalışmaların endüstride kullanılabilecek seviyelere gelmesi için daha büyük ekipler ve daha büyük yatırımlarla desteklenmesi gerektiği kanaatindeyiz.

Son olarak, çalışma kapsamında oluşturduğumuz verikümesinin yapay zeka ile sanat üretimi alanında çalışan araştırmacılar için yararlı bir kaynak olacağına inanıyoruz. Bu amaçla bu verikümesini internet ortamında herkese açık olacak şekilde paylaştık. Bu tarz çalışmaların kapsamlarının genişlemesi için böyle verikümelerinin

boyutlarının yeterince büyük olmasının ve sayılarının da çok olmasının önemli olduğunu düşünüyoruz.

Üretici Çekişmeli Ağlar (ÜÇA), bilgisayarlı görünün en çok çalışılan ve en gözde konularından bir tanesi. Bu modeller, birkaç sene öncesinde eğiticisiz öğrenme yöntemi olarak geliştirildiler. Ancak onlar, rahatlıkla öznitelik öğrenici modeller olarak da kullanılabilirler. Bazı uygulamaları olarak; gerçekçi resim üretimi, eskiz üretimi, resimden resme çevrim, 3B obje üretimi, resim tamamlama, yüksek çözünürlüklü resim üretimi, ses/müzik üretimi, anomali tespiti, yapay verikümesi üretimi, yanıltıcı saldırılar vb. Biz bu çalışmada, metinden görsel üretimine odaklandık. Özellikle, verilen bir kelime kümesinden sanat resimleri sentezleme üzerinde çalıştık. Amacımız, içeriğinde verilen anahtar kelimeleri doğru yansıtan, eşsiz yapay sanat resimleri üretmektir.

Metin, görüntü, ses, video gibi veri tiplerini üretmek ve bunlar arasında dönüşüm yapmak, ÜÇA'lar yardımıyla popülerlik kazanmıştır. Son yıllarda literatürde farklı veri türlerini kullanarak yine bu farklı veri türlerini üretmeye çalışan birçok çalışma yapılmıştır. Son yıllarda da ÜÇA'lar bu amaç için kullanılan bir numaralı sinir ağı modeli haline geldi. Biz de bu çalışmada ÜÇA'ları temel sinir ağı mimarisi olarak kullanmayı tercih ettik.

ÜÇA'lar, üretici ve ayırıcı olarak adlandırılan iki ayrı ağı eğiterek, verilen veri dağılımının en iyi temsillerini öğrenmek için tasarlanmıştır. Ayırıcı, hangi girdilerin gerçek veriden ve hangilerinin üretilmiş veriden geldiğini ayırt etmek için güncellenirken; üretici de ayırıcıyı en çok yanıltabilecek sahte veriyi üretmek için güncellenir. Her iki ağ da diğerini yenmeye çalışır ve sonuç olarak üretici tek başına yapay çıktıları sentezlemek için kullanılır. Üreticiye saf bir gürültü kaynağının yanı sıra, istenen diğer veri tipine dönüştürmek için koşullu veri adı verilen anlamlı veriler de verilebilir. Bu koşullu veriler dönüştürülmek istenen orijinal kaynaktır, çıktıda bu kaynakla ilişkilendirilecek yapay bir çıktı alınır. Üretici ağın bu iki veri arasında bir eşleme öğrenmesi beklenirken, ayırıcının da aynı eşlemeyi hem gerçek hem de yapay veri için öğrenmesi amaçlanır. Biz bu çalışmada koşullu veri olarak olarak anahtar kelimeleri kullanırken, saf gürültüyü de bu girdilerin bir parçası yaptık. Girdilerde gürültü kullanmanın, tez boyunca paylaştığımız gibi, birden fazla faydası bulunmaktadır.

Verilen metin açıklamalarından elde edilen görüntü sentezi, önemli miktarda yayınlanmış eser içeren popüler araştırma alanlarından biridir. Bu çalışmaların çoğu, metin içeriğini öğrenmeye ve bunları verikümesindeki benzer görüntülere dönüştürmeye çalıştıkları yaygın verikümelerine odaklanmıştır. Biz bunların aksine, ÜÇA'ları gerçekçi görüntüler yerine sanatsal resim görüntüleri oluşturmak için kullandık. Bu nedenle, verikümemizi daha fazla yaratıcılığa sahip olması için şekil, renk, çizim tekniği ve içerik bakımından çok çeşitli stillerle oluşturduk. Verikümesi, klasik sanat resimleri ve dijital sanat resimleriyle toplamda 3492 tane görsel içeriyor. Ayrıca, resimler için açıklayıcı cümleler kullanmak yerine, verilen resmin özelliklerini yansıtan anahtar kelimeleri tercih ettik. Yalnızca anahtar kelimeler kullanıldığında, kelime sırası önemsiz ve aralarındaki ilişkiler belirsiz hale geldiğinden kelimeleri görüntülerle ilişkilendirmek zorlaşır. Bununla birlikte, bu, modelimize orijinal bir resim yaratma esnekliğini sağlar ve bu da sanat yaratımı için önemli bir faktördür. İnanıyoruz ki bu verikümesi diğer araştırmacılar için de iyi bir kaynak oluşturacaktır.

Bundan dolayı, çalışmamızı yapay sanat üretimi çalışmalarında kullanılması için erişime açtık.

Çalışmamızın karakteristik özelliklerinden bir tanesi, görsellerin tanımları için cümleler yerine anahtar kelimeler kullanmamızdır. Bu yüzden verikümemiz görsel-anahtar kelimeler çiftlerinden oluşuyor ve şekil, renk ve içerik gibi çeşitliliklere sahip görseller içeriyor. Bu çeşitlilik sanatsal öz için önemli bir etken olan özgünlüğü sağlıyor. Ancak, bu çeşitlilik aynı zamanda öğrenmeyi de zorlaştırıyor. Bu zorluğun üstesinden gelebilmesi için derin ve birkaç aşamadan oluşan bir yapay sinir ağı mimarisi tasarladık.

Yapay sinir ağı modeli olarak, görselleri ve anahtar kelimeleri aynı anda işleyebilecek üç ayrı aşamalı Seri Üretici Çekişmeli Ağ modeli önerdik. Anahtar kelime ve görsel çiftleri sinir ağının katmanları boyunca işlenir. Bu seri modelin ilk aşaması kelime vektörlerini işler ve temel bir görüntü yaratırken, bir sonraki aşama kelime vektörleri üzerinde çalışmadan yüksek çözünürlüklü sanatsal tarzda görüntüler oluşturmaya odaklanır.

Seri üretici modelimize beselenebilmeleri için anahtar kelimeler kelime vektörlerine (gerçek sayıların vektörleri) eşlenmelidir. Geleneksel yöntem n-gram modelleri kullanırken, sinir ağları son zamanlarda popüler hale geldi. Sinir ağı olarak Word2Vec algoritmasını kullandık. Kelime girdilerini bir metin grubu olarak alır ve istenen boyutta (genellikle yüzlerce) bir vektör kümesi üretir. Sözlükteki her kelime, kelime uzayındaki benzersiz bir vektöre eşlenir, böylece ilgili kelimeler bu vektör uzayında birbirine yakın olacak şekilde toplanır. Bu kelime vektörlerini, görüntü üretme modelimizin girdileri olarak kullandık. Word2Vec algoritmasını optimize etmek için vektör boyutu, pencere boyutu, dönem sayısı gibi birkaç hiper-parametre vardır.

Görüntü oluşturma modeli için çoğunlukla evrişimsel ve evrişimsel olmayan katmanlar içeren ÜÇA'lar tasarladık. Görüntülerin büyük çeşitliliğini öğrenmek için, ÜÇA'ların üç aşamadan oluşan sıralı bir mimarisini önerdik. Her aşama, daha ayrıntılı ve daha yüksek çözünürlüklü görüntüler üretir ve bir öncekinden daha sanatsal stiller ekler. Bu çok aşamalı yapı, her aşama görüntüler üzerinde ayrı ayrı çalıştığı için daha soyut öğrenme sağlar.

ÜÇA'larda karşılaşılan en yaygın zorluk, ağırlıkları birbirini yenecek şekilde güncellendiğinden üreticiyi ve ayırıcıyı aynı anda güncellemektir. Bu çoğu zaman dengesiz bir eğitim eğrisine neden olmaktadır. ÜÇA'ların dengesiz doğası, böyle bir sıralı modelle çok hızlı biriktiğinden, dengesiz öğrenme eğrilerini ve mod çöküşü olarak bilinen fenomeni önlemek için, Wasserstein kaybı, spektral normalizasyon ve küçükküme ayrılması gibi bazı dengeleyici optimizasyon tekniklerini kullandık. Ayrıca, modelde kullanılan tüm ağlar için ayrı bir özel hiper-parametre kümesi belirledik.

Sonuçta, çeşitli stillere sahip resim görüntüleri üretebildik. Sonuçların değerlendirilmesi çalışmanın kritik bölümlerinden bir tanesi. Bunun için niceliksel ve niteliksel analiz yaptık. Niceliksel analiz olarak, sonuçlarımızı Frechet Başlangıç Mesafesi puanını kullanarak değerlendirdik. Niteliksel analiz olarak, 186 katılımcı ile bir kullanıcı anketi gerçekleştirdik. Bu iki analizin de sonuçlarını tablolar ve grafiklerle paylaştık. İki analiz de ürettiğimiz resimlerin sanat eseri olabileceğini destekledi. Ayrıca modelimizin ürettiği resimlerden bazılarını girdi olarak kullanılan anahtar

kelimelerle birlikte en son kısımda paylaştık. Bu görsel sonuçların değerlendirilmesini okuyucuya bırakıyoruz.

1. INTRODUCTION

Generating various data types such as texts, images, audio, videos, and conversion between those have gained popularity with the aid of Generative Adversarial Networks (GANs) [1]. In recent years, many studies have been conducted in the literature trying to produce these different types of data using different datasets. GANs became the number one neural network model used for this purpose. In this study, we preferred to use GANs as the basic neural network architecture as well.

GANs were designed to learn the best representations of given data distribution by training two separate networks called generator and discriminator. The generator is updated to generate similar outputs to given input data while discriminator is updated to distinguish which inputs come from real training data and which come from the generated data. Both networks try to beat the other one and as a result, the generator is used to synthesize artificial outputs. The generator might be given a noisy input as well as meaningful data called conditions to convert them into the other desired data type. Converting text (specifically words) to images is the one that we have focused on this work.

Image synthesis from given text descriptions is one of the trending research area, which has a considerable amount of published works. Most of these works focused on common datasets, where they try to learn text content and convert them to similar images in the dataset. On the contrary, we utilized GANs to generate artistic painting images instead of photo-realistic images. Therefore, we have formed our dataset with a large variety of styles in shape, color, drawing technique, and content to have more creativity. Moreover, instead of using descriptive sentences for images, we preferred keywords that reflect the features of the given image. By using only keywords, the word order is insignificant, and relations between them are obscure, so associating the words with the images becomes harder. However, this provides our model the flexibility to generate images more freely which is an important factor for art creation. The overview of the proposed pipeline is shown in Figure 1.1.

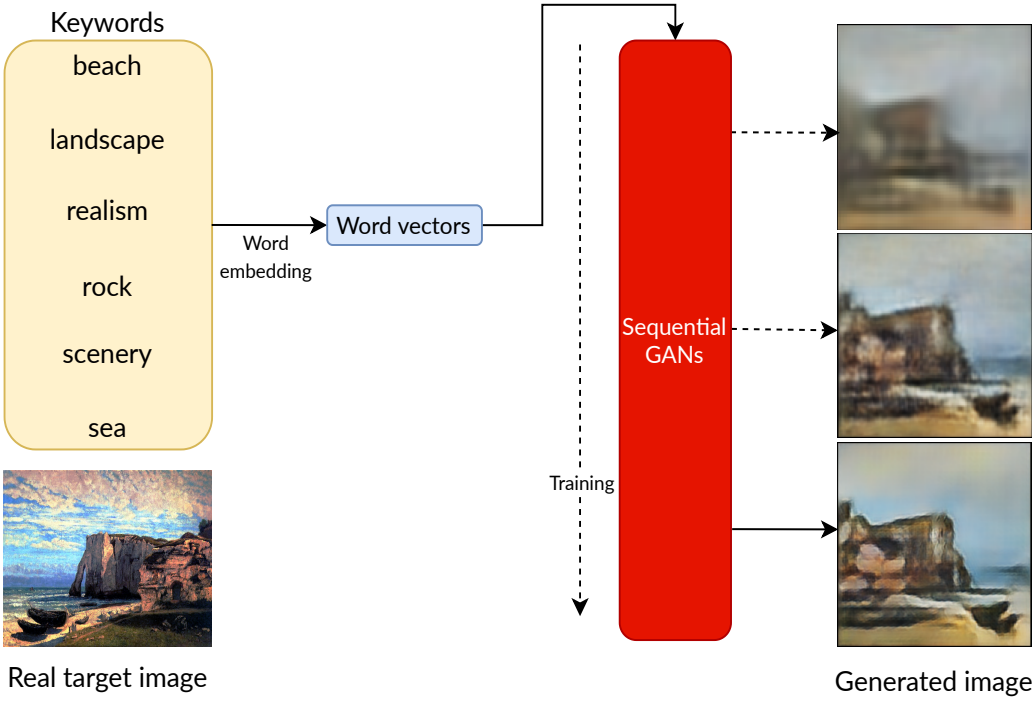

Figure 1.1 : Overview of the proposed pipeline. Keyword lists are converted into word vectors, and they are given to our sequential GAN model. At each stage, the model proceeds from the output of the previous stage, and more detailed and higher resolution images are generated. Notice that the generated image contains most of the keywords, and relations between them are creatively connected.

As can be seen from Figure 1.1 keywords must be mapped into word vectors (vectors of real numbers) via a process called word embedding. The traditional method uses n-gram models, whereas neural networks became popular recently. We used the off-the-shelf Word2Vec [2, 3] algorithm, which uses neural networks as well. It takes word inputs as a corpus of text and produces a vector space with the desired dimension (usually hundreds). Each word in the corpus is mapped to a unique vector in the space so that the related words will also be close to each other in this vector space. We used these word vectors as the inputs of our image generation model. There are several hyper-parameters to optimize the Word2Vec algorithm such as vector size, window size, epochs, etc. as we shall see in the following chapters.

For the image generation model, we designed GANs that mostly contain convolutional and deconvolutional layers. To learn large variance of images, we proposed a sequential architecture of GANs, which consists of three stages. Each stage generates more detailed and higher resolution images as well as adds more artistic styles than the

previous one. This multiple staged structure provides more abstract learning as each stage works on the images separately. The detailed mechanism of the model will be shared in the following sections as well.

The most common difficulty for the GANs is that converging the generator and the discriminator at the same time since their weights are updated to beat each other. This is causing an unstable training curves most of the time. As the unstable nature of GANs accumulates too fast with such a sequential model, to prevent unbalanced learning curves and the well-known phenomenon called *mode collapse*, we utilized some stabilizing optimization techniques such as Wasserstein loss [4, 5], spectral normalization [6] and minibatch discrimination [7] which will be discussed in the following chapters as well.

Our contributions can be summarized as follows:

- We have created a new keywords-to-painting dataset. Dataset contains 3492 images with 1746 digital art images and 1746 classical painting images. There are 738 unique keywords.
- We have used keywords for image descriptions, where word order is insignificant and word relations are more obscure unlike full sentence descriptions, leading to more artistic image generations.
- We have designed a sequential GAN architecture that handles both word vectors and real-fake image pairs.
- We have employed a mixture of optimization techniques —Wasserstein loss, spectral normalization, and minibatch discrimination— to train all the networks stably and efficiently.

The remainder of the thesis is organized as follows. Chapter 2 is on our proposed pipeline, network architecture, and optimization methods. Chapter 3 gives details about the dataset that we created. In Chapter 4, we present training techniques, and experimental results. Finally, Chapter 5 concludes the thesis.

1.1 Purpose of Thesis

The ability of artificial intelligence to produce artwork is one of the controversial topics of computer science. Can artificial intelligence create true art, or it is no more than a matrix formed by dull numbers and equations? The answer to this question has been long-awaited among researchers and even among people interested in artificial intelligence. Before we started this study, we were wondering the answer to this question and this was our biggest motivation when starting this study.

Often, art is considered to be a human act, as it is an effort to convey emotions and ideas. On the other hand, some argue that artistic production comes out at the end of deterministic and logical processes. Even if it is a logical process, producing art can be something that only the human brain can overcome, or it can be considered as a work of art only if it is produced by the human brain. In other words, we can only evaluate a work as a work of art when it is produced as a result of the human mind, emotions, and experiences. Conversely, if a work of art is not a result of the human mind, then we think of it as just a randomly generated production. We leave thinking of all these questions and ideas to readers and experts, as this is the part of this work that touches philosophy. If we focus on the purpose of thesis, we aimed to produce artificial paintings by a complete mathematical model, trained with human-made paintings, which are considered to be works of art and to question whether these artificial paintings are works of art.

Another aim is to create a prototype of an application that can be used by people who need the inspiration to produce artistic production, such as writers and painters. These people can use the images produced by our model in a similar way to the images presented on stock photo websites. Because the model we use takes the keyword set as input and these words can be fed into our model by people who want to use such an application. We anticipate that such an application can be used in various fields of the industry such as graphic design, game design, and advertising besides pure artistic production. The last but not least goal is; we believe that the dataset we created will be a useful resource for researchers working in the field of artwork generation with artificial intelligence.

1.2 Literature Review

1.2.1 Deep generative learning models

Deep generative learning models with a convolutional layer structure have shown remarkable performance for generating images. Earlier works were conducted as unsupervised feature learning models [1, 8–11]. Mirza et al. [12] introduced a conditional version of this generative model and opened a way for data translation tasks. All these models use either classic GANs or a variant of their old or newer variants.

1.2.2 Image generation

Image generation with labels is extensively studied. Oord et al. [13], Yan et al. [14], and Odena et al. [15] have proposed conditional models to generate images where conditions can be class labels or descriptive image attributes. Brock et. al. [16] generated class-conditional samples using a large scale GAN training and acquired visually high-quality images. There are also progressive training approaches [17, 18], in which generated output resolutions continually increase and better image qualities are acquired.

1.2.3 Image-to-image translation

Image-to-image translation studies use images as conditions (inputs to the generator) to generate new images (outputs from the discriminator). Isola et al. [19] approached this translation as a general-purpose solution and modeled a conditional adversarial network that learns a mapping from an input image to an output image as well as learns a loss function to train the mapping. Choi et al. [20] proposed a single GAN model that can perform image-to-image translations for multiple domains. Zhu et al. [21] extended the translation idea for the datasets where the paired input and output samples are absent. Similarly, Yi et al. [22] devised an unsupervised general-purpose image-to-image translation network which enables learning for two sets of unlabeled datasets from two different domains.

1.2.4 Style transfer

Style transfer is another research area that is related to our work since they mostly deal with the translation of artistic essence (colors, styles, etc.) of images. [21] can be mentioned here as well because their work experimented as photo-generation from paintings and vice versa. Gatys et al. [23] designed a CNN-based neural network where they separate and recombine the content and style of images.

1.2.5 Text-to-image synthesis

Text-to-image synthesis studies use text descriptions as conditions to GAN models. Dash et al. [24] synthesized images from conditioned text descriptions. Xu et al. [25] devised an attention-driven mechanism through multi-stage architecture. Similarly, Mansimov et al. [26] designed another attentional mechanism with bidirectional recurrent neural networks to encode the image captions. Zhang et al. [27, 28] proposed a decomposed architecture where the first stage sketches basic attributes such as color and shape from given text descriptions, and the second stage adds more details and yields better image quality. Qiao et al. [29] designed a text-to-image-to-text framework to learn generations by modifying text descriptions. Some other approaches focus on object locations as well as text descriptions while generating images from captions [30, 31]. Similarly, Park et al. [32] proposed a model that focuses on foreground and background objects at the same time for given text descriptions. For better learning of text descriptions, using mismatching text is another idea [33].

2. METHODOLOGY

Our methodology consists of two main parts: Word embedding and three stages of GANs. Word embedding is realized relatively more straightforward since it utilizes the off-the-shelf method: Word2Vec [2, 3]. Three-staged sequential GAN, on the other hand, is a fundamental part of the work and one of the contributions of this work. The proposed method's workflow is illustrated in Figure 2.1, and it is explained in the following sections.

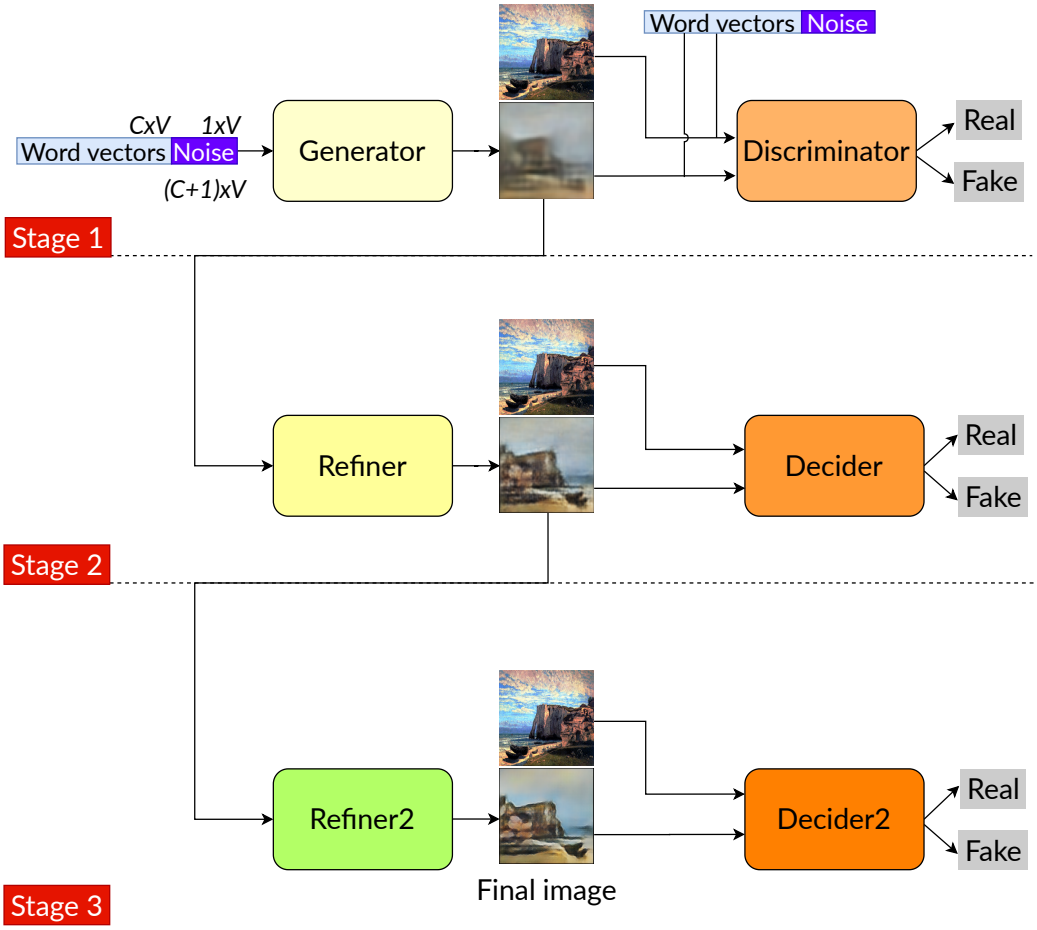

Figure 2.1 : Basic model workflow with three stages of GANs. Word vectors are processed only in the first stage. The second stage continues where the first stage left off and image resolution increases. The third stage continues where the second stage left off, and the final image is created in this stage.

2.1 Word Embedding

A shallow neural network algorithm Word2Vec [2, 3] was used for word embedding. Word2Vec is a language processing model which transforms a text corpus into word vectors with desired dimensions. In the process, similar words come closer to each other in the vector space. There are two methods that Word2Vec uses: Continuous Bag of Words (CBOW) and Skip-Gram. CBOW tries to predict a word *w* by looking at the whole context. Conversely, Skip-Gram tries to predict the next word by looking at the word *w*. We preferred the CBOW method. The reason for the word 'continuous' is that the vectors are represented by float values and not only binary values.

The size of the context is limited by choosing a window. We have chosen window size as 10. If the sentence length is smaller than 10, the window size is bounded by the length of sentence length for the given keyword set.

The basic architecture is a two-layer neural network with V dimensional output layer. V also represents the size of word vectors, which is taken as 64. The hidden layer size N is also 64. Context size (number of words representing one image) is fixed to six, and one noise vector is concatenated to the end of context as the same size of a real word vector size. Finally, our vocabulary size is 738 (total unique words) for the training dataset. A detailed schema is given in Figure 2.2.

painting	**scenery**	**realism**	**female**
paint	landscape	romanticism	sexy
religious	cliff	impressionism	woman
jesus	nature	peasant	bed
christ	day	painting	beautiful

impressionism	**portrait**	**animal**	**male**
realism	famous	hunt	man
expressionism	stylish	predator	beard
paint	thoughtful	prehistoric	famous
umbrella	self	reptile	book

horse	**dark**	**sea**	**fantasy**
equine	black	ocean	horn
stallion	red	wave	evil
gallop	darkness	turtle	magic
fast	evil	underwater	wing

Table 2.1 : Some top words (bold ones) and their most related top four words (below) from our dataset

We utilized the Gensim library [34][1] which provides an efficient, powerful, and well-documented Word2Vec implementation and it is widely used by the natural language processing community. Some other hyper-parameters used in WordVec model are window size is 10, minimum word count is 3 (ignoring all words with frequency lower than this), the initial learning rate is 0.01, minimum learning rate is 0.00001, epoch is 500, negative sampling is 20 (how many noise words will be used). As a result of our Word2Vec model, we presented some of the top words and their most relevant words all selected from our keywords of the dataset in Table 2.1. Notice that, these relations reflect the context of the images. Therefore, if some words used more in classical paintings or digital paintings, then its most relative words are reflecting the context of that image respectively.

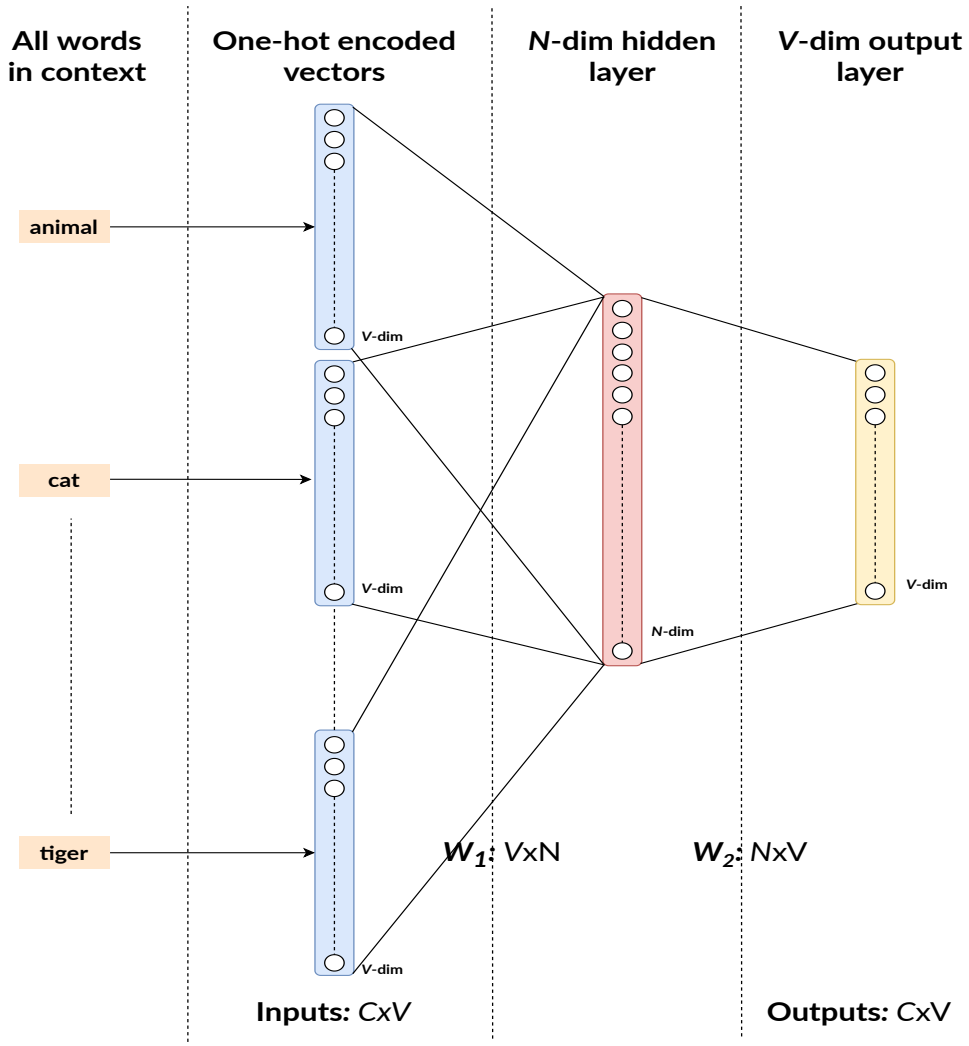

Figure 2.2 : Shallow neural network architecture of word embedding model with a N dimensional one hidden layer, $C \times V$ shaped inputs and $N \times V$ shaped outputs. C and V represent context size and word vector size respectively.

[1] https://radimrehurek.com/gensim/index.html

2.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) is considered to be one of the most interesting ideas of our decade. After the Deep Convolutional Neural Networks proposed by Krizevsky et al. [35] in 2012, it was seen by many as the second most innovative work in the field of artificial intelligence in the last decade. GANs quickly became popular after it is proposed by Goodfellow et al. [1] in 2014, and was applied to popular computer vision applications besides all other artificial intelligence problems. Considering our application, GANs are the first network model that comes to mind. Although there are many variations of GANs, we will just give details about its basic form and equations, then we will explain convolutional GANs. After that, we will start explaining our sequential GAN model.

A GAN is composed of two separate neural networks. One is the generator which is designed to generate a data distribution of interest from a given latent space. Here, the latent space term is used to define randomly chosen inputs. These random inputs lead to unpredictably generated outputs in the end. On the other hand, the discriminator tries to distinguish those generated outputs from the real data distribution. In other words, a discriminator works as a simple binary classifier. While the generator tries to deceive the discriminator by its fake data distribution, the discriminator is learning the real and fake data. This process leads to a competitive learning model between two different neural networks. The basic schema of a GAN is given in Figure 2.3.

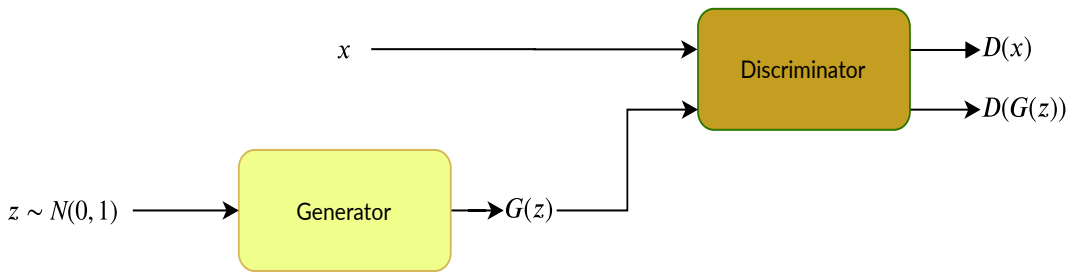

Figure 2.3 : The basic schema of GAN. The generator is fed with latent variable z, randomly chosen inputs, and generates fake data $G(z)$. Discriminator is both fed with real data x and fake data $G(z)$, outputs $D(x)$ and $D(G(z))$ respectively.

Real data distribution is $p_d(x)$ and fake data distribution is $p_g(z)$. The objective for the discriminator is given Equation 2.1 which is a maximization condition for the discriminator.

$$\max_D \mathbb{E}_{x\sim p_d(x)}[\log(D(x))] + \mathbb{E}_{z\sim p_g(z)}[\log(1-D(G(z)))] \tag{2.1}$$

In Equation 2.1 the discriminator tries to make fake inputs $D(G(z)) = 0$ and real inputs $D(x) = 1$ (discriminator output is bounded to an interval $[0,1]$ by a sigmoid function). In the end, the discriminator tries to minimize the classification error. Similarly, the objective of the generator is a minimization condition that is given in Equation 2.2.

$$\min_G \mathbb{E}_{z\sim p_g(z)}[\log(1-D(G(z)))] \tag{2.2}$$

The generator tries to make $D(G(z)) = 1$ and minimize Equation 2.2. This means the generator tries to minimize Equation 2.1 indirectly and maximize classification error. Both equations are combined into one in Equation 2.3

$$\min_G \max_D \mathbb{E}_{x\sim p_d(x)}[\log(D(x))] + \mathbb{E}_{z\sim p_g(z)}[\log(1-D(G(z)))] \tag{2.3}$$

where both the generator and the discriminator have different objectives (minimization and maximization) over the same equation.

Equation 2.3 also defines a minimax game where the generator tries to minimize and the discriminator tries to maximize the objective. A minimax game is a well-known concept in game theory where two-players compete with each other by making simultaneous moves to optimize a condition without knowing the moves of the other player. In the game theory, the solution of a minimax game is equal to Nash equilibrium where both players gain nothing by changing their strategy. Namely, the game reaches a point where the players do their best and their total gain against each other is 0 (this leads to a zero-sum game). Training a GAN should ideally reach a Nash equilibrium where the discriminator has only a 50% chance at its binary predictions since the fake outputs are indistinguishable from the real ones. However, as we shall see this case is practically impossible for the training of GANs.

2.3 Deep Convolutional Generative Adversarial Networks

Before delving into our model and its training stabilization techniques we will talk about Deep Convolutional GANs which is a variation of a classical GAN architecture to specifically generate images from given latent space. This architecture is also the building block of our model.

Deep Convolutional Generative Adversarial Networks (DC-GAN) is firstly proposed by Radford et al. [11]. These networks are an extension of GANs and use convolutional and deconvolutional (convolutional-transpose) layers inherently in both the generator and the discriminator parts. The strided deconvolutional layers transform the latent vectors into a desired image shape by upsampling it in the generator. Conversely, convolutional layers downsample the image in the discriminator to classify it as real or fake. Both the generated image and the real image is processed through the same layers in the discriminator. In the end, data is reshaped to 1D vector by fully connected layers, then a sigmoid layer squeezes it into $[0, 1]$ interval. Both networks can be updated by Equation 2.3 as well as by other objective functions such as Wasserstein loss as we shall see in the following sections. The simple schema of DC-GAN is given in Figure 2.4.

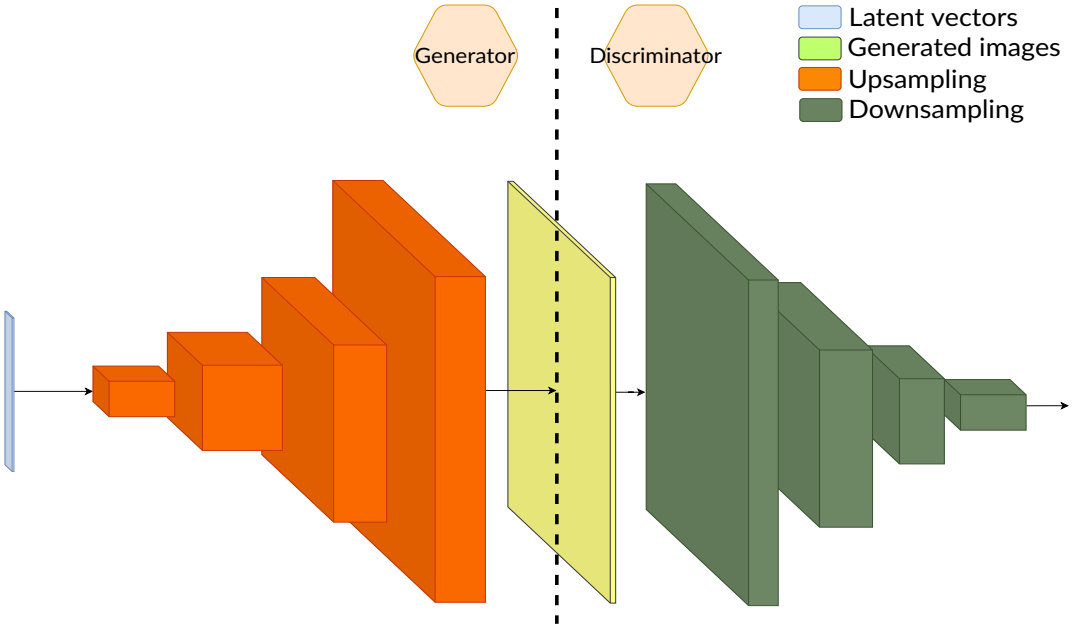

Figure 2.4 : The basic schema of DC-GAN. The generator is fed with latent vectors. This vectors are upsampled through deconvolutional layers in the generator. Conversely, the generated image (or real image) is downsampled through the convolutional layers of discriminator.

2.4 Mode Collapse

The well-known problem of Equation 2.3 is the vanishing gradient for the generator. Generally, discriminator classifies better as opposed to generator generates images. This leads to gradients approach zero very quickly for the generator. In this case, the discriminator dominates the training, and the generator can never improve itself. Conversely, if the generator is arranged to have many more parameters as opposed to a discriminator with fewer parameters, then the generator quickly learns to deceive discriminator with the same meaningless images. Either case GAN falls to a situation called *mode collapse*. In a typical mode collapse case, the fake outputs are plain one-color images even though the inputs are different from each other. These meaningless inseparable images either deceives the discriminator all the time and the generator dominates, or these outputs are an indication that the generator fell too much behind by the discriminator since the discriminator learned to classify very quickly and started to say "fake" everything that the generator produced.

There are several reasons for the mode collapse occurred. One is if the model capacity of the one network outweighs the other one, then this network can easily dominate. Therefore, model sizes (number of parameters) must be tuned so that the balance between the generator and the discriminator should be secured. One other problem might be the loss function. Finding a suitable loss function is the first step to resolve the unbalanced training problem. Other common solutions are stabilizing by tuning the hyper-parameters such as the learning rate and regularization strength. Tuning the hyper-parameters may slow one down and gives the other a chance to catch it.

Once a mode collapse has occurred, practically the objective cannot be realized and training must be stopped. From the engineering point of view, training is restarted until you are sure that it can continue without observing "mode collapse". Training stabilization must be guaranteed before starting the training by tuning the hyper-parameters. And at some point, the best hyper-parameters can only be found by trial and error. We performed a mixture of optimization techniques to prevent mode collapse which is presented in the following sections.

2.5 Model Architecture

The very basic schema of our model is given in Figure 2.1 earlier. This model architecture consists of three staged GANs. Each stage is a Convolutional GAN in itself and has one generator and one discriminator. The model architecture is similar to StackGAN proposed by Zhang et al. [27,28]. But our architecture has one extra refiner stage, and there are some other differences at the inner structure of GANs. Also, training methods and optimization techniques are radically different from StackGAN since we try to learn the artistic features of the images and try to provide originality in the outputs. These differences will be discussed in the upcoming sections. First, we will explain the details of the GAN stages.

The first stage processes the word vectors in the generator and produces a relatively lower resolution image. This image forms a base for the upcoming stages. On the other hand, the discriminator processes real image-word vectors and fake image-word vectors pairs and gives unbounded outputs. Unbounded output without sigmoid function at the end is used since we use Wasserstein loss as we will discuss in Section 2.6.

The second stage takes the output of the first stage generator, refines it through the convolutional layers, and outputs a higher resolution image. This stage's networks are called as refiner and decider even though they are generator and discriminator essentially. At this stage, no word vector is processed by neither refiner nor decider.

The third stage is quite similar to the second stage. It takes the output of the second stage and gives the final image with the highest resolution.

The advantage of using this type of sequential staged model is that separating base image creation with word vectors and image refinement processes. Therefore, we can design more specialized networks and increasing image resolution through the layers becomes easier. To generate higher resolution images, more refinement stages can be added. However, each refiner stage requires more an more computation power as well as memory allocation. In our experiments, we made it feasible to train with 3 stages (2 refiner stages) with our equipment. But, we highly encourage the researchers to try the same architecture with more refiner stages.

2.5.1 The first stage

The first stage is a classical GAN with one generator and one discriminator, and its architecture is given in Figure 2.5. Word vector inputs are fixed in the shape of 7×64 with six words and one additional noise row. If given words for the image is not filling the context of six words, then one of the closest words from vocabulary is chosen randomly. On the contrary, if there are more words than six, then a random subset is chosen. This way inputs become noisier and varied so that it provides stable learning curves as proposed by Salimans et al. [7].

The generator consists of a series of convolutional and convolutional upsample layers. In the middle of the generator, there is a series of ResNet [36] blocks to extract features better as the generator also works as a feature learner. In the end, the generator outputs a 64×64 image. The same word vectors are given to discriminator paired with real and fake images separately. In the discriminator, word vectors firstly pass through a fully connected layer to be able to concatenate them with the images channel-wise.

At the end of the discriminator, there is a minibatch discrimination layer in which its functions will be explained later. And it gives unbounded outputs.

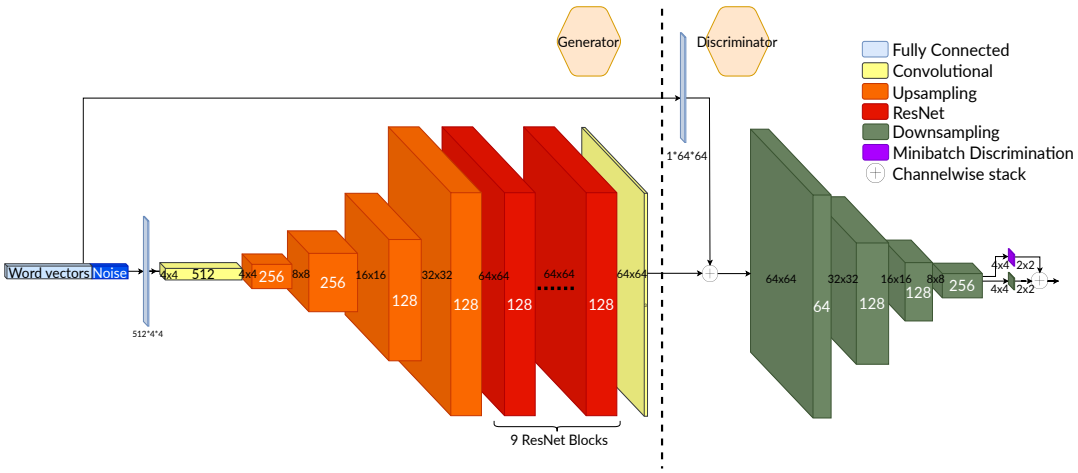

Figure 2.5 : The first stage GAN architecture. Generator has one fully connected layer to process word vectors, convolutional upsample layers and ResNet blocks. Discriminator has one fully connected layer to process word vectors, convolutional downsample blocks and minibatch discrimination layer. Batch normalization is applied after convolutional layers except the last layer and the first layer of the generator and discriminator.

2.5.2 The second stage

The second stage also has one generator (refiner) and one discriminator (decider) and the architecture are given in Figure 2.6. It proceeds from the output of the first stage and refines it through its generator.

The generator consists of a series of convolutional downsample and upsample layers and works like an autoencoder essentially. There are skip connections between those downsample and upsample layers to provide smoother gradient flows and faster learning. In the end, the generator outputs a 128×128 image. Since this stage proceeds where the first stage left off, word vectors are not processed at this stage but only the output of the first stage.

Similarly, the discriminator consists of series of convolutional downsample layers and only processes the output of the refiner paired with corresponding real images. The real images are resized to 128×128 the shape of output images of the refiner so that both the fake images and real images can be fed to the first layer of the discriminator. In the end, there is a minibatch discrimination layer in which its detail will be explained in the upcoming sections.

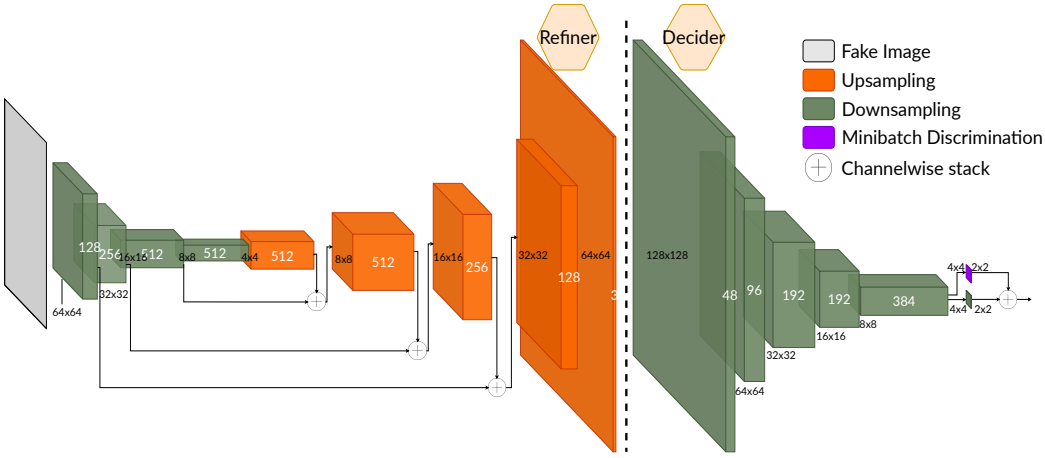

Figure 2.6 : The second stage GAN architecture. Generator has convolutional upsample and downsample blocks. There are skip connections between layers that has same resolution images. Discriminator has convolutional downsample blocks and minibatch discrimination layer. Batch normalization is applied after convolutional layers except the last layer and the first layer of the generator and discriminator.

2.5.3 The third stage

The third stage architecture is almost identical to the second stage as given in Figure 2.7 except that it is fed with the output of the second stage and generates a 256×256 image.

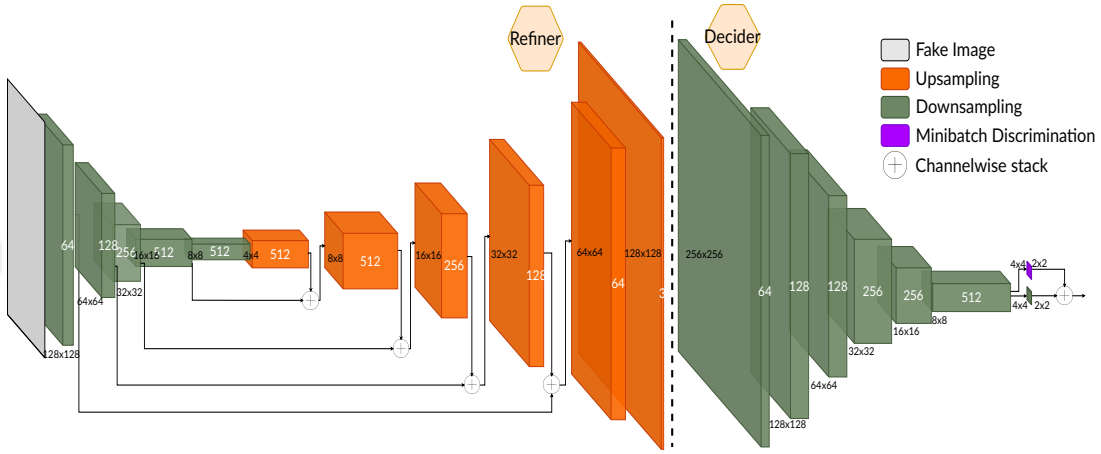

Figure 2.7 : The third stage GAN architecture. Generator has convolutional upsample and downsample blocks. There are skip connections between layers that has same resolution images. Discriminator has convolutional downsample blocks and minibatch discrimination layer. Batch normalization is applied after convolutional layers except the last layer and the first layer of the generator and discriminator. Notice that the filter sizes are slightly different than the second stage.

2.6 Model Optimization

Stable convergent training with GANs is quite challenging. As mentioned before, when either the generator or the discriminator starts to dominate training, *mode collapse* is quite likely to occur. In this case, saving training from falling apart is almost impossible. Therefore, prior adjustments are required. Choosing a good loss function, setting the most suitable hyper-parameters, arranging network layers are the most important features for these adjustments to apply on.

Our architecture has three separate GANs, and the first two feed the others by their outputs. Therefore, we tried out a lot of stabilizing optimization techniques and to perfect the training curves. We will first talk about our loss function. Then, we will give details about minibatch discrimination and spectral normalization. Finally, we will conclude this chapter with our hyper-parameter strategy.

2.6.1 Wasserstein loss with gradient penalty (WGAN-GP)

Wasserstein GAN (WGAN) was proposed by Arjovsky et al. [4]. It is an improved way of training GANs with a different loss function to prevent vanishing gradient for both the generator and the discriminator and to avoid mode collapse. WGAN replaces the discriminator with a critic and instead of classifying as real or fake (binary) it evaluates the inputs and puts on a scale. According to this scale, the most real inputs are grouped at the top and the most fake inputs are grouped at the bottom. This scale is not bounded between $[0,1]$ as it would be in the classical GAN. To achieve this unlimited output, no sigmoid function is used at the end of discriminator. Ultimately, in Wasserstein loss the discriminator tries to maximize $D(x) - D(G(z))$ and the generator tries to maximize $D(G(z))$.

Since the loss is not bounded, gradients should be clipped to avoid gradient explosion. Generally, weights can be clipped between $[c, -c]$ where $c \in \mathbb{R}$ in WGAN. Besides, in [4] discriminator is updated more than the generator and RMSProp is used for optimization to stabilize training.

However, enforcing stabilization by clipping weights is not an effective method according to Arjovsky et al. [4] since it limits the model capacity. To overcome this Wasserstein loss with gradient penalty (WGAN-GP) is proposed by Gulrajani et al. [5]. Equation 2.3 transforms into Equation 2.4 with a gradient penalty term.

$$\mathbb{E}_{\tilde{x}}[D(\tilde{x})] - \mathbb{E}_{x}[D(x)] + \lambda \mathbb{E}_{\tilde{x}}[(||\nabla_{\tilde{x}} D(\tilde{x})||_2 - 1)^2] \tag{2.4}$$

where x and $\tilde{x}$ represent real and fake data instances respectively and λ is constant multiplier for gradient penalty term.

As Gulrajani et al. stated WGAN-GP provides more stable convergence in the training. Therefore we selected it for optimization and obtained the best training curves in our experiments. The final form of our loss function is given in Equation 2.4 and λ is taken as 10.0. Wasserstein loss provided us more stable converging curves as opposed to BCE (Binary Cross Entropy) loss and LSGAN loss [37].

2.6.2 Minibatch discrimination

For our task, one of the most important conditions for producing good quality images is to produce them with great variety. The basic idea is evaluating the variety of batches of outputs instead of just individual samples. Another term used to express this variety is entropy. Minibatch discrimination was developed by Salimans et al. [7] to prevent the generator from producing outputs with lower entropy than the real dataset. Therefore, minibatch discrimination is essentially intended to increase the entropy of the generator.

Minibatch discrimination layers are shown in Figures 2.5 and 2.6, which forces the generator to generate a wide variety of images and prevent mode collapse. It is applied to the final dense layer of the discriminator by concatenating the similarity of input image x with all other images in the batch. The similarity for the minibatch b is given as

$$o\left(\boldsymbol{x}_i\right)_b = \sum_{j=1}^{n} c_b\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \in \mathbb{R} \tag{2.5}$$

where $c_b\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = \exp\left(-\left\|\boldsymbol{M}_{i,b} - \boldsymbol{M}_{j,b}\right\|_{L_1}\right) \in \mathbb{R}$ is L_1-distance between i^{th} and j^{th} image. Here, $\boldsymbol{f}\left(\boldsymbol{x_i}\right) \in \mathbb{R}^A$ is the vector of features of input $\boldsymbol{x}_i$, $\boldsymbol{T} \in \mathbb{R}^{A \times B \times C}$ is learnable parameter tensor of minibatch discrimination layer and $\boldsymbol{M}_i = \boldsymbol{f}_i \cdot \boldsymbol{T}$. Similarity output of the all batches is given as

$$o\left(\boldsymbol{x_i}\right) = \left[o\left(\boldsymbol{x_i}\right)_1, o\left(\boldsymbol{x_i}\right)_2, \ldots, o\left(\boldsymbol{x_i}\right)_B\right] \in \mathbb{R}^B \tag{2.6}$$

which is concatenated channel-wise with the feature vector $\boldsymbol{f}\left(\boldsymbol{x_i}\right)$.

As our dataset contains a wide variety of images, real inputs to the discriminator will have very low similarity. On the contrary, if the generator starts to output monotonic images, the similarity metric given in Equation 2.6 will be very high. In this case, discriminator should penalize the generator. As we shall see in visual results in Chapter 4, minibatch discrimination provided our model to generate large variety of outputs for given set of keywords.

2.6.3 Spectral normalization

Spectral normalization is firstly proposed by Miyato et al. [6] as an alternative to gradient penalty and other weight normalization techniques. The idea is simply replacing all the weights W in a layer with $W/\sigma(W)$. Here, $\sigma(W)$ is the spectral norm of the weight matrix W. Spectral norm of W is given in Equation 2.7

$$\sigma(W) = ||Wv|| = u^T W v \tag{2.7}$$

where u and v are randomly initialized vectors for given weight W in a layer. In a specific update time t, v_t is given as

$$v_t = \frac{(W^T W)^t v}{||(W^T W)^t v||} \tag{2.8}$$

and at time $t+1$, u_{t+1} and v_{t+1} are computed as

$$\begin{aligned} u_{t+1} &= W v_t \\ v_{t+1} &= W^T u_{t+1} \end{aligned} \tag{2.9}$$

Equation 2.9 defines a power iteration rule over u and v as the weight W changes slowly over time. Since it does not involve any derivative operation, spectral normalization is computationally efficient than applying gradient penalty.

We applied spectral normalization [6] to all convolutional layers as an additional stabilizer of training. As Miyato et al. stated spectral normalization provides more diversely generated samples than conventional weight normalization [6].

Miyato et al. [6] compared their model Spectral Normalization GANs (SN-GANs) to WGAN-GP models and regularization techniques and proposed that the images generated by their models are clearer and more diverse on CIFAR-10 [38] and STL-10 [39] datasets. In this work, we utilized spectral normalization in combination with the gradient penalty as a regularization method.

2.6.4 Hyper-parameter strategy

Planning of the hyper-parameter set is the last of the strategies we applied to optimize the training of our model. These hyper-parameters include general neural networks parameters such as learning rate, regularization strength as well as others such as label flipping probability rate specific to GANs. We will explain parameter separation idea that we used on our sequential GAN architecture to have more stabilized training.

Separating parameters for the generator and the discriminator is another technique to have more stable training. As Heusel et al. introduced [40] using different learning rates (generator: 0.0001, discriminator: 0.0002) benefits more stabilized convergence for the generators against the discriminators. These learning rates are the same in the refiner and the decider networks in the second and third stages. Inspired by this idea, we applied different dropout rates (generator: 0.2, discriminator: 0.65) for all the convolutional and dense layers on all networks. Conversely, we used the same weight decay as 0.00001 on our Adam optimizer [41] on all networks. The details of Adam optimizer will be given in the upcoming sections.

Another strategy for training stabilization is that flipping labels with 0.05 probability (fake images become real and real ones become fake). This strategy slows down the discriminator activity and prevents mode collapse by setting wrong labels on the images and give more chances to the generator to generate different images.

Other hyper-parameters and optimization techniques will be explained in detail in the following sections. Now we will continue with our dataset and its features in the next chapter.

3. DATASET

Our dataset[1] is a painting images collection acquired from *deviantart.com* for digital paintings and *wikiart.org* for classical paintings. 1746 digital paintings and 1746 classical paintings were used in total and 349 images were reserved for testing. As can be seen in Figure 3.1, our dataset contains a large variety of images in shape, color, content, and style. Similarly, our vocabulary size (738) is quite extensive. We believe that this variety is a necessity for artistic essence since it leads to generating unique images for a given set of input words.

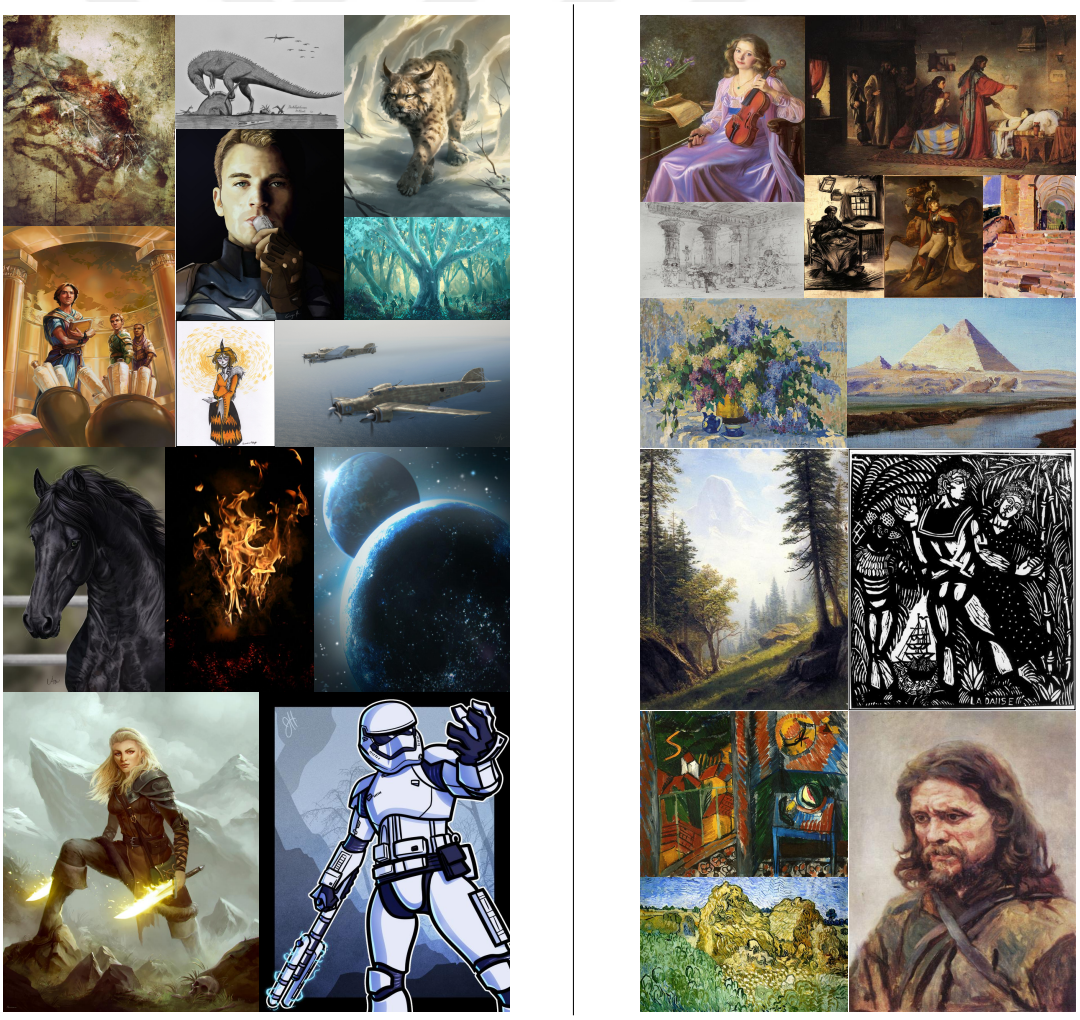

Figure 3.1 : Sample digital art images on the left and classical art images on the right

[1] Accessible from `https://doi.org/10.5281/zenodo.3690752`

3.1 Words Analysis

The number of total unique words is 738. Histogram of the top most used 50 words of our dataset is shown in Figure 3.2.

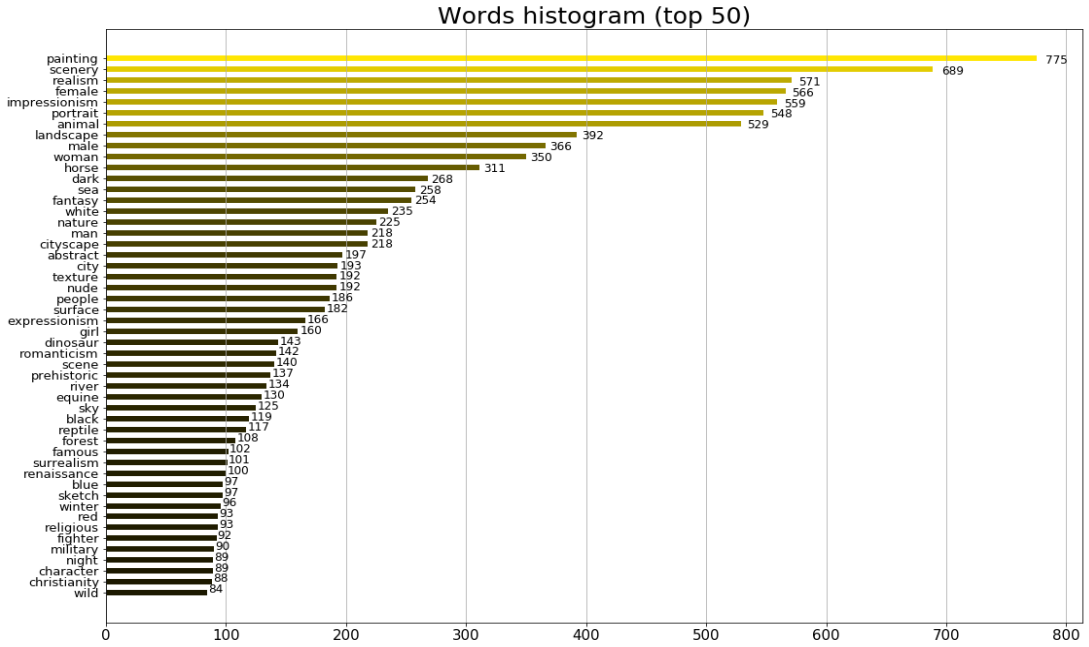

Figure 3.2 : Top most used 50 words in our dataset.

Keyword set for a given image is called a sentence. Although we have used fixed sentence length in the training phase, which is 6, images in the original dataset have not the same sentence length. The histogram of sentence lengths is given in Figure 3.3.

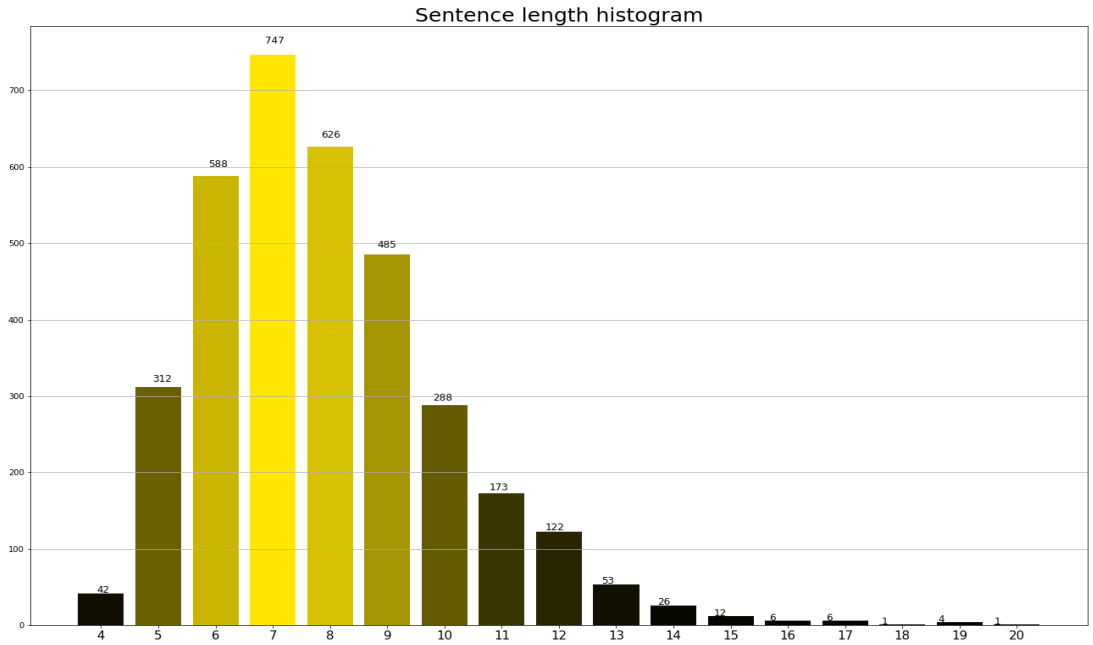

Figure 3.3 : Sentence length histogram of our dataset.

3.2 Image Analysis

All images are 3-channel RGB images. Classical painting images have relatively smaller resolution than digital painting images but have a lower standard deviation in width and height. The detailed image statistics are given in Table 3.3.

	Total	Mean width	Mean height	Std width	Std height
Classical paintings	1746	609.54	561.06	147.99	60.13
Digital paintings	1746	1108.05	1015.45	307.39	367.33
All paintings	3492	858.79	788.25	346.88	347.69

Table 3.1 : Image shape statistics of our dataset

	Red mean	Green mean	Blue mean
Classical paintings	120.91	107.25	90.48
Digital paintings	127.64	121.79	118.51
All paintings	124.88	114.52	104.50

Table 3.2 : Image color channel means of our dataset

	Red std	Green std	Blue std
Classical paintings	55.84	52.45	48.77
Digital paintings	56.10	54.85	53.10
All paintings	55.97	53.65	50.94

Table 3.3 : Image color channel standard deviations of our dataset

3.3 Data Augmentation

Data augmentation plays an important role in our experiments due to the relatively small size of our dataset. We applied a series of image processing methods randomly. One of them is Gaussian blur with a randomly selected radius from the uniformly distributed $(0,3)$ interval. Color jittering was used that randomly factored brightness, contrast, saturation, and hue values in the intervals $(0.6,1.6)$, $(0.6,1.6)$, $(0.6,1.6)$, and $(-0.2,0.2)$ respectively. Additionally, gamma correction, histogram equalization, sharpening, and random resolution changing was applied. We did not use other well-known augmentation methods, such as horizontal flipping, image rotation, channel swapping, or grayscaling since they would have caused incompatibility between the words and content of the images. Images are normalized to $(0,1)$ scale as well as our word vectors. Augmentation effects can be seen in Figure 3.4.

Figure 3.4 : Sample augmented images. On the top: Original image, on the second row from left to right: Gaussian blur, color jitter, gamma correction, on the second row from left to right: histogram equalization, resolution changing, sharpening.

4. EXPERIMENTS

4.1 Training

We did the training with the 3143 image-keyword pair, which is 90% of the dataset. Word vectors are acquired using our Word2Vec model trained previously and are normalized into $[-1,1]$ scale. Training of all three stages was conducted simultaneously. Namely, one forward pass is completed until all three stages output one image. The alternating training method as proposed by [40] was not used. Therefore, all the generators and the discriminators were updated at the end of the same minibatch. Instead of updating generator and discriminator separately, we separated learning rate and dropout rate parameters of them to organize stabilized training as we shall discuss in the upcoming sections.

For all networks, Adam [41] optimizer with $beta = 0.5$ was used. The batch size is selected as 16. Excluding hyper-parameter stabilization experiments, all networks were trained for at least 3000 epochs. Besides, there have been experiments where we have over 3000 epochs to fine-tune more. As we mentioned before, learning rates were initialized to 0.0001 and 0.0002 for the generator and the discriminator, respectively, and decayed to half for every 300 epochs but not below 0.00001. This learning rate strategy covers the refiner and decider networks that belong to the second and the third stages as well.

4.2 Testing

For this work, testing means that creating a set of images for a given keyword set and evaluating them using a particular metric or asking the quality of the images to the people. Our evaluation strategy will be shared in the upcoming sections. We did the testing with the 349 image-keyword pair, which is 10% of the dataset. Images are created using only the generator networks (in three stages). All final created images

were compared to ground truth images using Fréchet Inception Distance (FID) [40] metric as we will talk about in the next section.

4.3 Fréchet Inception Distance (FID)

Fréchet Inception Distance (FID) proposed by Heusel et al. [40] is the first method that we used to measure the quality of the generated images. It is an alternative method to Inception Score (IS) that is proposed by Salimans et al. [7]. The FID score simply compares the generated samples to the real samples. Lower FID score corresponds to more similar real and generated samples as measured by the Inception-v3 model [42]. In the case of measuring FID for the identical datasets, the score becomes 0. FID score evaluates the similarity between the real and the generated datasets as well as the diversity of the generated images by measuring the distance of two Gaussian distribution which is given as

$$||\mu_r - \mu_g||^2 + Tr(C_r + C_g - 2(C_r C_g)^{1/2}) \tag{4.1}$$

where (μ_r, C_r), (μ_g, C_g) are the mean and covariance of the real and generated image distributions.

We compared the quality of generated images by the FID score (lower is better) in Table 4.1. Complete Gaussian random noise images are our upper bound and they represent the worst images that can be generated for given keywords. The lower FID bound is missing since the ideal painting image does not exist for given keywords. As can be seen from Table 4.1, more realistic images are generated through the stages of our model. As we shall see in the upcoming sections, Stage 3 presents the best-quality images visually.

	FID score
Random noise	487.66
Stage 1	384.67
Stage 2	234.18
Stage 3 (Final output)	**218.34**

Table 4.1 : FID scores of random noise and stages of our model.

4.4 Survey

The survey is the second method that we used to measure the quality of the images that our model generated. Since the evaluation of artworks is a subjective action, we thought that the survey we conducted would be more successful than a numerical metric in evaluating the quality of the pictures.

The survey [1] was held by querying the participants whether a given painting image is made by our model or by a human. The questions of the survey were short and simple, as the survey was intended to be organized with a sufficient number of images and with a lot of participants from different professions. To make the survey simpler, keywords of the images were not given and all the images were shown one by one. Besides, we asked the participants their confidence level on each image (*Unsure*, *Somewhat confident*, *Very confident*). Results were collected from 186 participants where 98% of the total is at least a university student or has a college degree. We did not ask the participants' professions, but we also know that there are some art critics, and many researchers working in the area of artificial intelligence among the participants.

Participants were shown 48 images. 24 of them are generated by our model and the rest is made by humans (those images were collected from our dataset but were not used in the training). The participants were also asked their relations with AI at a scale of 1 to 5. Besides, other anonymous information such as their age, education status, and work status were collected. Although there was no time limit while answering the questions, participants were also warned not to look at the pictures for more than a few seconds and give quick answers.

We evaluated the survey results by plotting charts of the participant's success concerning their information. Also, we provided some of the most and the least confusing images in the survey and shared our thoughts on them. The survey results according to participant relations with AI can be seen in Figure 4.1. Similarly, Figure 4.2, Figure 4.3, and Figure 4.4 show other results according to participant age, education status, and work status respectively. The height of bars in the groups are normalized as the length of all participant groups will be 1.0.

[1] Accessible from `https://forms.gle/LiWTiMKbLvsx6ZL38`

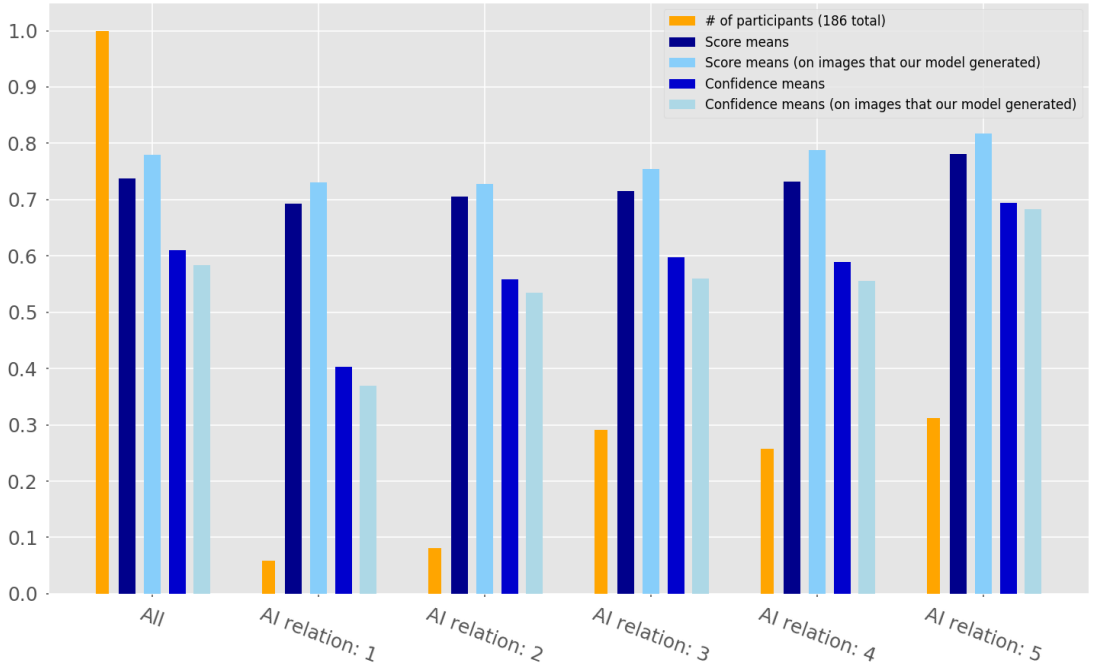

Figure 4.1 : Survey results. From left to right; all participants and groups according to their AI relations from 1 to 5. Blue bars represent the normalized total score of 186 participants that they acquire from 48 images. Light blue bars represent total score on generated 24 images. All bars are normalized to $(0, 1)$ interval.

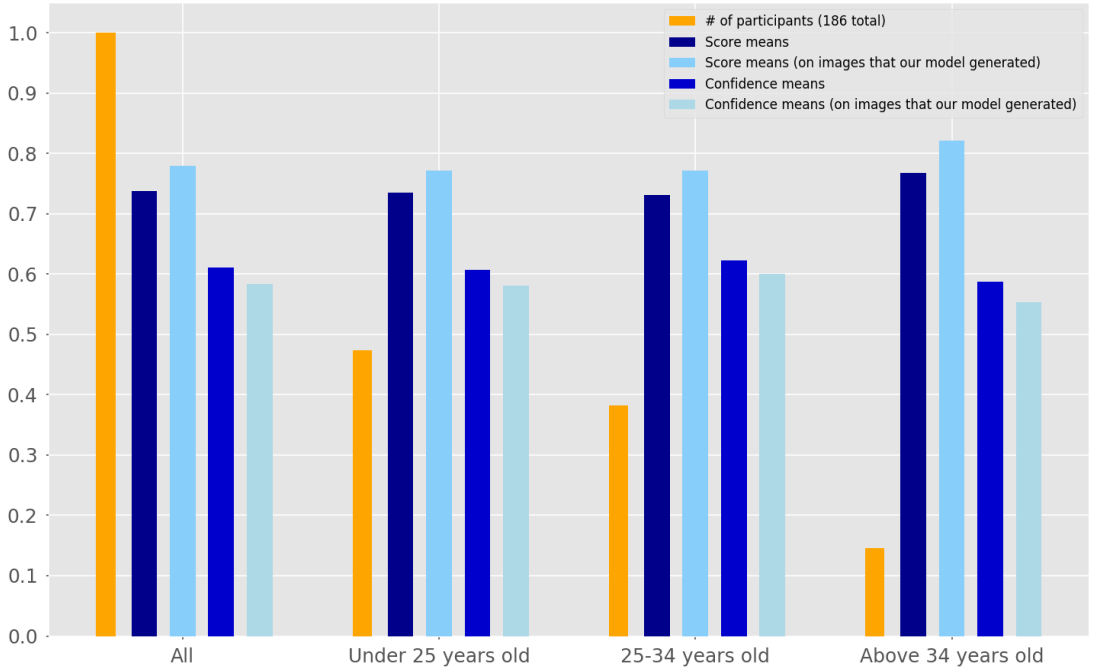

Figure 4.2 : Survey results. From left to right; all participants and groups according to their age groups. Blue bars represent the normalized total score of 186 participants that they acquire from 48 images. Light blue bars represent total score on generated 24 images. All bars are normalized to $(0, 1)$ interval.

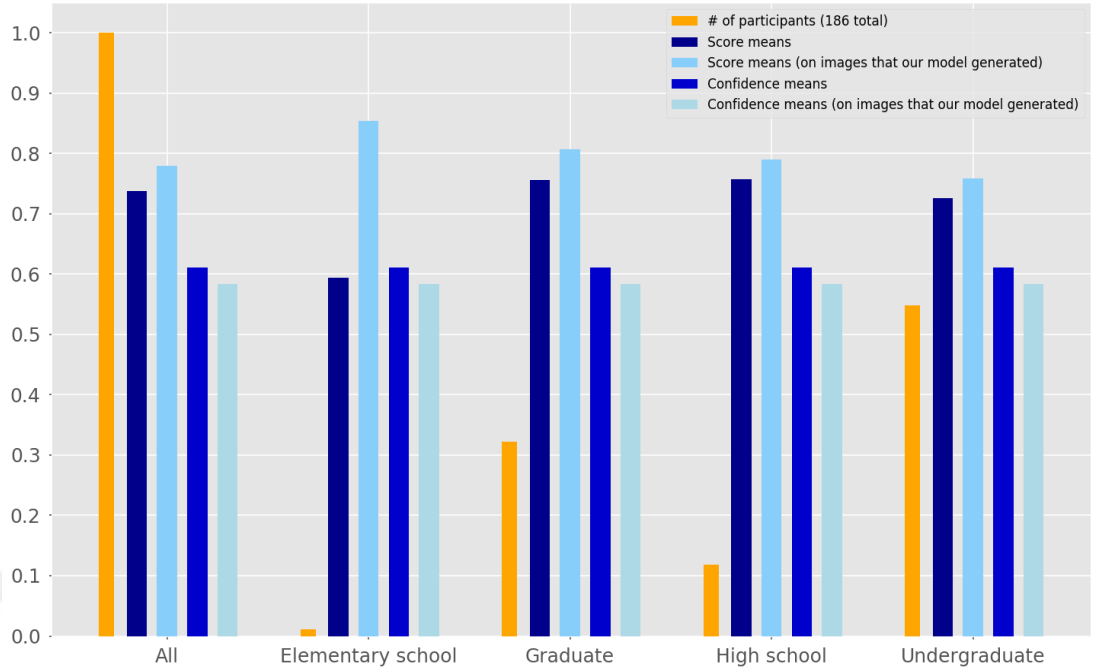

Figure 4.3 : Survey results. From left to right; all participants and groups according to their education status groups. Blue bars represent the normalized total score of 186 participants that they acquire from 48 images. Light blue bars represent total score on generated 24 images. All bars are normalized to $(0, 1)$ interval.

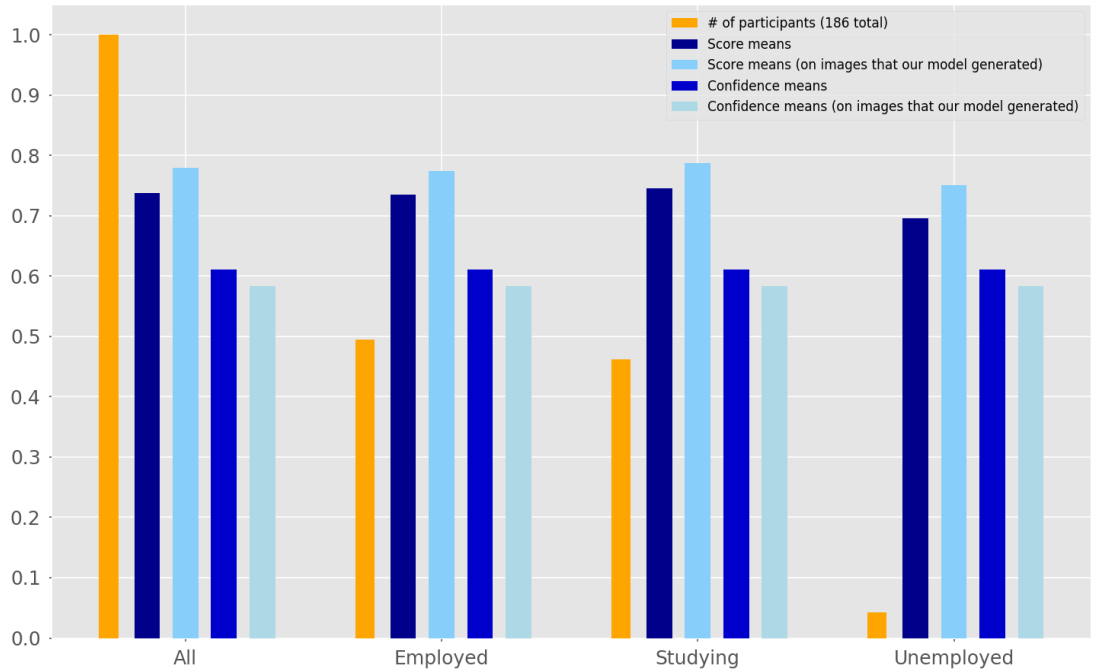

Figure 4.4 : Survey results. From left to right; all participants and groups according to their work status groups. Blue bars represent the normalized total score of 186 participants that they acquire from 48 images. Light blue bars represent total score on generated 24 images. All bars are normalized to $(0, 1)$ interval.

As a result of the survey, the average success rate of guessing whether a given image (of all 48 images) is generated by our model or by a human is found to be 73.8%. The success rate of guessing correctly for all images (24 images) generated by our model is 77.9%. According to survey results, the top five most and the least confusing images created by both our model and humans are given in Figures 4.5, 4.6, 4.7, and 4.8 respectively.

Figure 4.5 : Top 5 the most-confused images that are actually made by our model. From left to right; 46.2%, 40.3%, 39.2%, 38.2% and 30.6% of all participants guessed that they are made by humans. We are informed that most of the participants thought that these images have quite realistic brush strokes and objects are not obscure as the others.

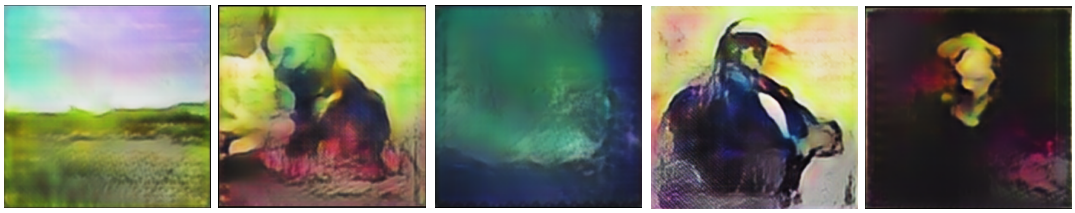

Figure 4.6 : Top 5 the least-confused images that are actually made by our model. From left to right; 6.5%, 10.8%, 11.3%, 11.8% and 12.4% of all participants guessed that they are made by humans. For those images, participants said that the artifacts around the images played an important role in their decisions.

Figure 4.7 : Top 5 the most-confused images that are actually made by humans. From left to right; 76.9%, 59.1%, 58.6%, 58.1% and 51.1% of all participants guessed that they are made by our model. Plain, monochromatic and worn images make participants suppose that there must be an artificial generation.

Figure 4.8 : Top 5 the least-confused images that are actually made by humans. From left to right; 8.6%, 10.8%, 10.8%, 14.0% and 14.5% of all participants guessed that they are made by our model. Color usage on the canvas and theme of the paintings is very critical in the thought process of all the images. Especially, if the participant has experience with artificially generated images before or has an advanced understanding of art, it is more difficult to deceive them. Most of the participants said that these images are more artistic and have more details compared to others.

From the visual results of the survey, we can comment that bright colors make participants think that image is artificially generated as in Figure 4.6. Besides, when the objects are prominently seen and artifacts around the images are absent, participants can be deceived as given in Figure 4.5. Although they are made by humans, there are also pictures thought to be made by AI due to their abstract content and monochromatic use as seen in Figure 4.7. Overall, the participants that have experience with AI, or have an advanced understanding of art are the most difficult ones to deceive. They are quite confident at detecting fake creations as seen in Figure 4.8. This was reflected in the results shared in the Figure 4.1 and Figure 4.3. From these results, we have seen that it is a requirement that the color and brightness balance of the generated images are set correctly and the objects in the images are distinguished clearly to deceive people.

Another remarkable detail that we noticed in the survey was that while they are shown the images, participants became more successful in differentiating images towards the end of the survey, as they understood the style of our model better. Thus, the success of the participants was better in the images shown at the end of the survey compared to the images shown at the beginning. Nevertheless, apart from the statistics, the participants also stated that they had fun during the survey and were surprised at many images made by AI when they saw the answer key.

Here we end the analysis of the results part. In the next section, we will share the visual results and leave the assessment to the reader.

4.5 Visual Results

In this section, we present other visual results other than the ones shared in the survey. We believe that the visual results are the most important factor to evaluate such a research. In the previous section, we made two different evaluations of the pictures produced by our model through the FID score and survey. Here we leave the artistic evaluation of the pictures shared visually to the reader.

The figures that show our visual results include the input keywords as well as the generated images through the stages of our model. From the final generated images we can conclude that the image content of the generated images is compatible with given words generally. Besides, when a very similar word set is given unique styles and contents were generated. Still, the content in the final images should not be expected to be as clear as a human-made painting. Theses paintings as we mentioned before should be seen as productions that will trigger inspiration in artistic production. In other words, rather than criticizing how clear the objects in the images are, the artistic features (originality, color harmonies, abstract features, etc.) of the images should be given more importance.

As a weakness, some details like the face of people are not created very well and there are artifacts around the objects. We believe that those weaknesses would be resolved by expanding the dataset. Detailed visual outputs for given words at different stages are provided in Figure 4.9, 4.10, 4.11, 4.12. Some input keyword-ground truth image pairs may appear more than once. But since corresponding generated images may be different, those pairs were kept in the result figures.

Figure 4.9 : Sample results. From left to right; given words, our outputs through three stages and corresponding ground truth for given words.

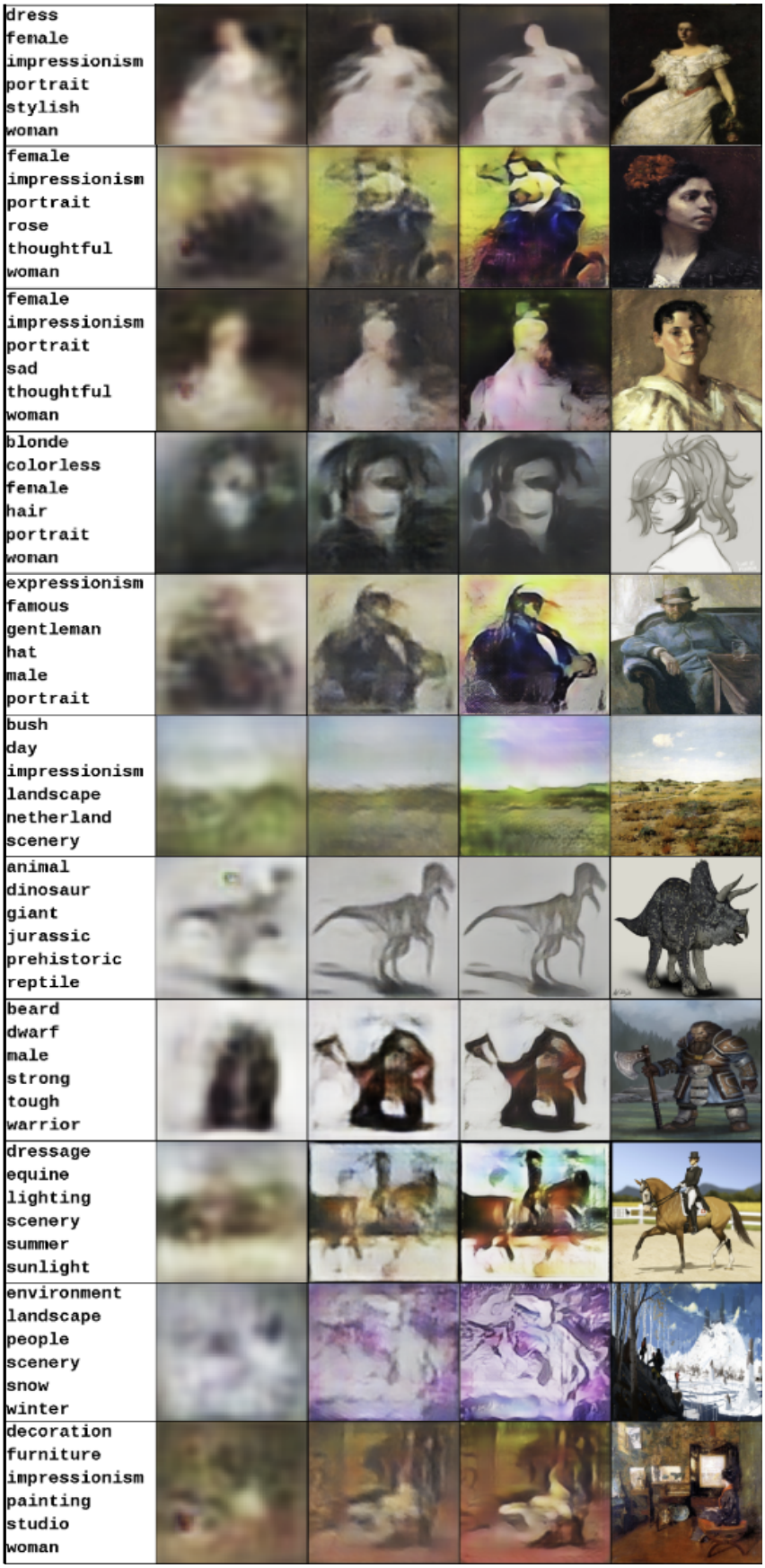

Figure 4.10 : Sample results. From left to right; given words, our outputs through three stages and corresponding ground truth for given words.

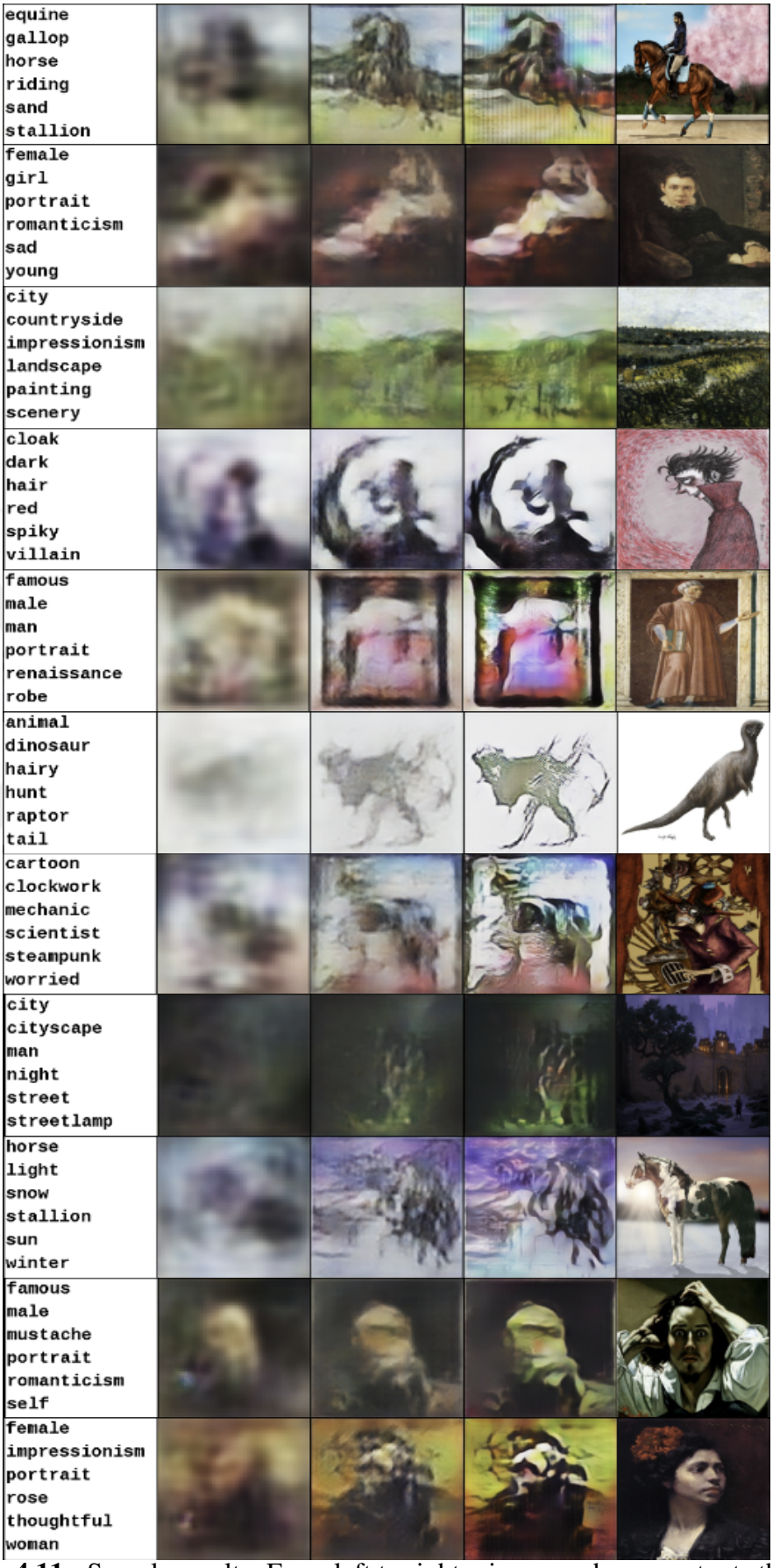

Figure 4.11 : Sample results. From left to right; given words, our outputs through three stages and corresponding ground truth for given words.

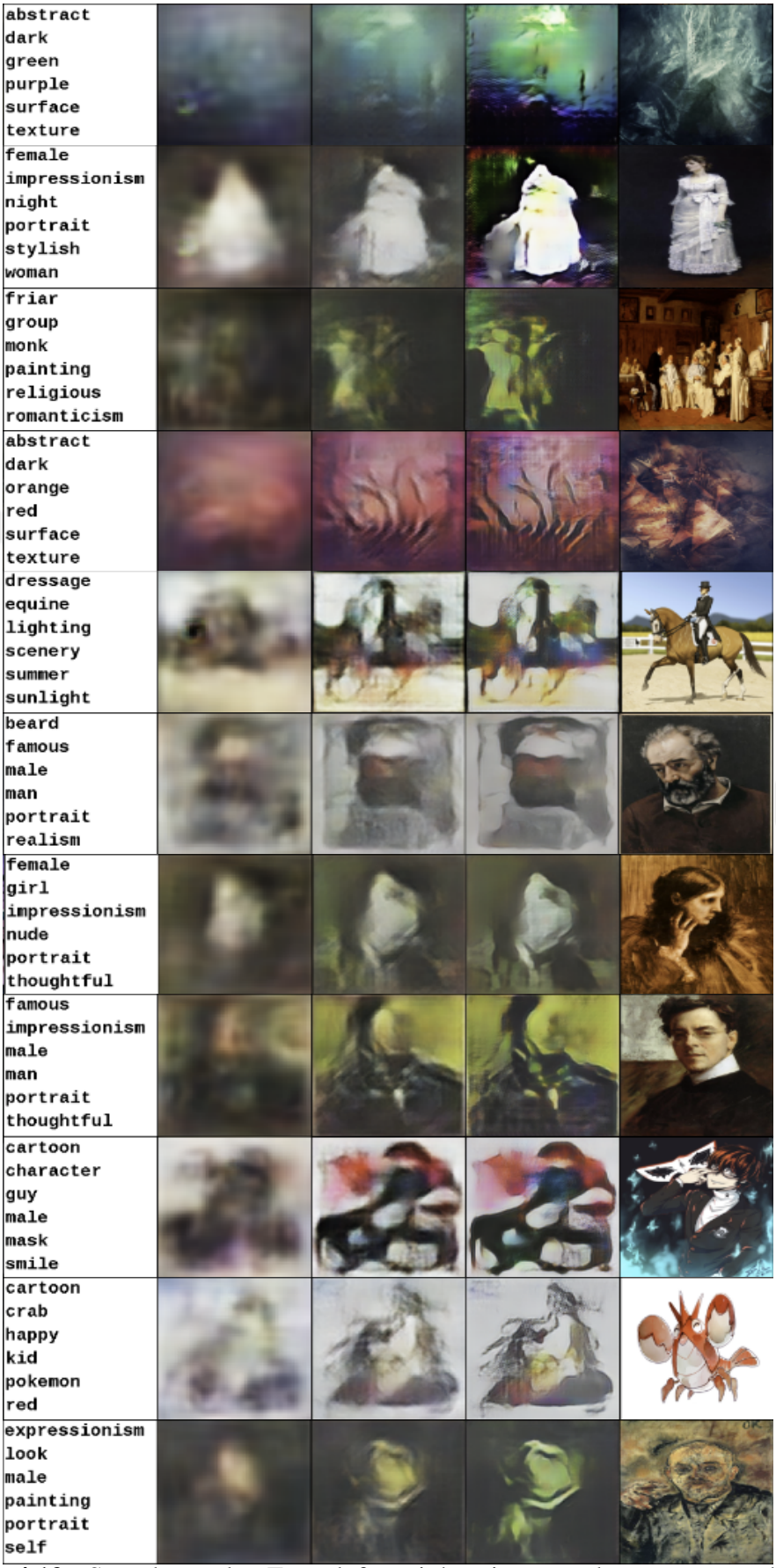

Figure 4.12 : Sample results. From left to right; given words, our outputs through three stages and corresponding ground truth for given words.

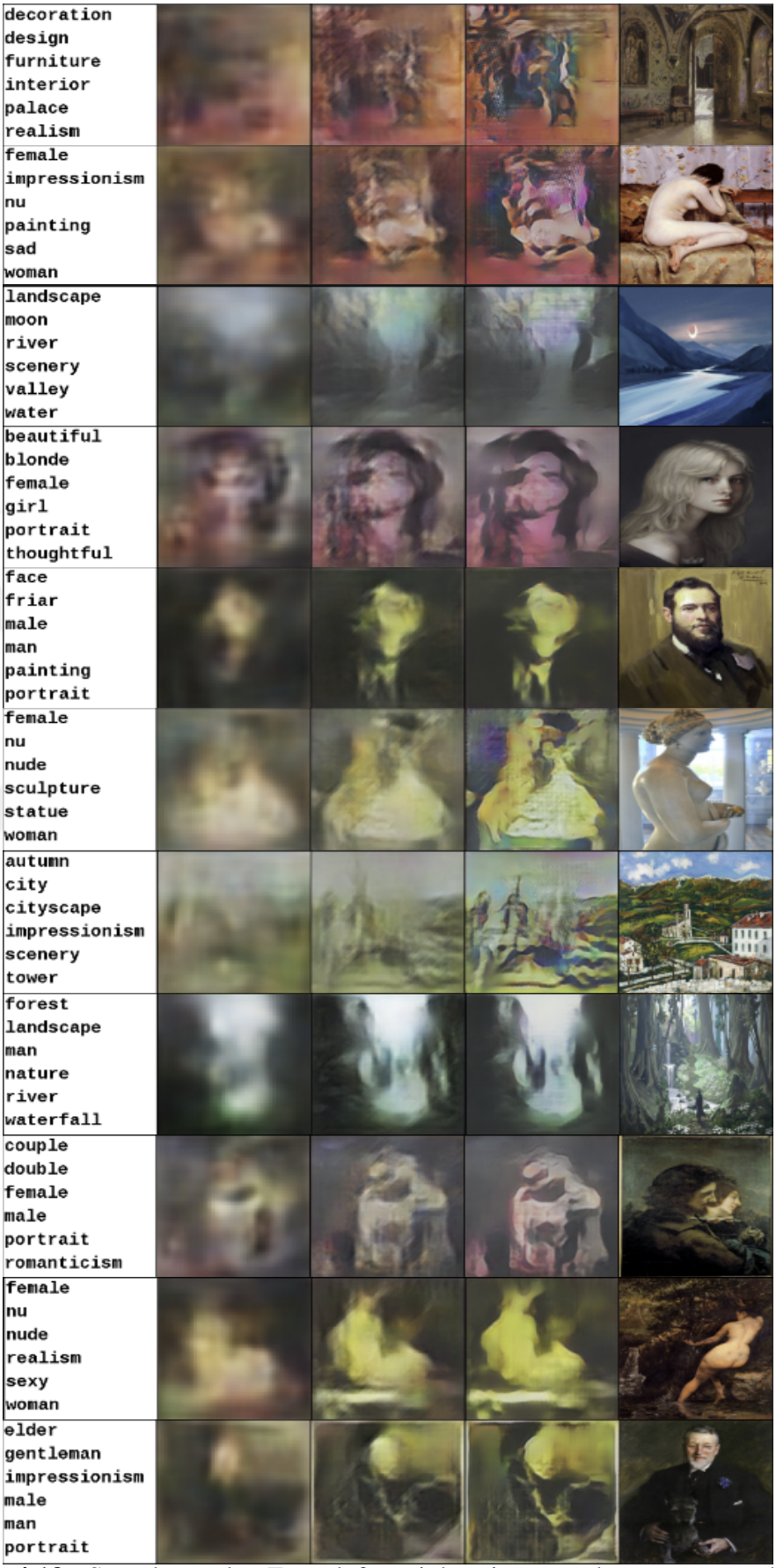

Figure 4.13 : Sample results. From left to right; given words, our outputs through three stages and corresponding ground truth for given words.

5. CONCLUSION

Art generation is one of the most controversial topics of Artificial Intelligence studies. Is the art only the specialty of humans, or can a machine learn it through observation and experiment? In this work, we investigated this question as our motivation. For this purpose, we designed our generative neural network model and created our keyword-painting dataset.

Literature about this field has just begun to form. Although there are hundreds of studies about image generation and tens of studies about text-to-image generation, text-to-art studies are still not too many. The field of art generation is generally concentrated on the neural image synthesis topic. In this topic, researchers tried to create artistic images by merging an artistic image style into a target image, instead of creating an artistic painting from scratch. Likewise, text-to-image generations did not focus on generating scratch art images but focused on creating photo-realistic images from text. Our work differs from most of the art generation studies by its particular sequential generative model and its specially prepared keyword-painting dataset.

In this work, we proposed a sequential GAN model to generate paintings from keywords. For this purpose, we created a keyword-to-painting dataset. Our model acquired 218.34 FID score, and at the survey with 48 images (24 images were made by our model, and the rest were made by humans) 26.2% of all 186 participants mistaken about who made it. Currently, our model capability should be evaluated by its visual outputs even though we have provided an FID score for quantitative measurement and a survey. Please note that we cannot compare the FID score to the other works in literature due to the use of different datasets. However, we believe visual outputs show promising results as well. Please note that with larger datasets, variety in inputs would increase, and this would in return lead to producing better images. Nevertheless, visual

results show that our model is quite successful at generating visually appealing, mixed style, fairly suited to text descriptions painting images.

REFERENCES

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

[2] **Mikolov, T.**, **Chen, K.**, **Corrado, G. and Dean, J.** (2013). Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781*.

[3] **Mikolov, T.**, **Sutskever, I.**, **Chen, K.**, **Corrado, G.S. and Dean, J.** (2013). Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems*, pp.3111–3119.

[4] **Arjovsky, M.**, **Chintala, S. and Bottou, L.** (2017). Wasserstein GAN, *arXiv preprint arXiv:1701.07875*.

[5] **Gulrajani, I.**, **Ahmed, F.**, **Arjovsky, M.**, **Dumoulin, V. and Courville, A.C.** (2017). Improved training of wasserstein GANs, *Advances in Neural Information Processing Systems*, pp.5767–5777.

[6] **Miyato, T.**, **Kataoka, T.**, **Koyama, M. and Yoshida, Y.** (2018). Spectral normalization for generative adversarial networks, *arXiv preprint arXiv:1802.05957*.

[7] **Salimans, T.**, **Goodfellow, I.**, **Zaremba, W.**, **Cheung, V.**, **Radford, A. and Chen, X.** (2016). Improved techniques for training GANs, *Advances in Neural Information Processing Systems*, pp.2234–2242.

[8] **Chen, X.**, **Duan, Y.**, **Houthooft, R.**, **Schulman, J.**, **Sutskever, I. and Abbeel, P.** (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets, *Advances in Neural Information Processing Systems*, pp.2172–2180.

[9] **Denton, E.L.**, **Chintala, S.**, **Fergus, R.** *et al.* (2015). Deep generative image models using a laplacian pyramid of adversarial networks, *Advances in Neural Information Processing Systems*, pp.1486–1494.

[10] **Liu, M.Y.**, **Breuel, T. and Kautz, J.** (2017). Unsupervised image-to-image translation networks, *Advances in Neural Information Processing Systems*, pp.700–708.

[11] **Radford, A.**, **Metz, L. and Chintala, S.** (2015). Unsupervised representation learning with deep convolutional generative adversarial networks, *arXiv preprint arXiv:1511.06434*.

[12] **Mirza, M. and Osindero, S.** (2014). Conditional generative adversarial nets, *arXiv preprint arXiv:1411.1784*.

[13] **Van den Oord, A.**, **Kalchbrenner, N.**, **Espeholt, L.**, **Vinyals, O.**, **Graves, A.** *et al.* (2016). Conditional image generation with pixelcnn decoders, *Advances in Neural Information Processing Systems*, pp.4790–4798.

[14] **Yan, X.**, **Yang, J.**, **Sohn, K. and Lee, H.** (2016). Attribute2image: Conditional image generation from visual attributes, *European Conference on Computer Vision*, Springer, pp.776–791.

[15] **Odena, A.**, **Olah, C. and Shlens, J.** (2017). Conditional image synthesis with auxiliary classifier GANs, *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, pp.2642–2651.

[16] **Brock, A.**, **Donahue, J. and Simonyan, K.** (2019). Large scale GAN training for high fidelity natural image synthesis, *International Conference on Learning Representations*.

[17] **Karnewar, A. and Iyengar, R.S.** (2019). MSG-GAN: Multi-Scale Gradients GAN for more stable and synchronized multi-scale image synthesis, *arXiv preprint arXiv:1903.06048*.

[18] **Karras, T.**, **Aila, T.**, **Laine, S. and Lehtinen, J.** (2017). Progressive growing of GANs for improved quality, stability, and variation, *arXiv preprint arXiv:1710.10196*.

[19] **Isola, P.**, **Zhu, J.Y.**, **Zhou, T. and Efros, A.A.** (2017). Image-to-image translation with conditional adversarial networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1125–1134.

[20] **Choi, Y.**, **Choi, M.**, **Kim, M.**, **Ha, J.W.**, **Kim, S. and Choo, J.** (2018). StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.8789–8797.

[21] **Zhu, J.Y.**, **Park, T.**, **Isola, P. and Efros, A.A.** (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks, *Proceedings of the IEEE International Conference on Computer Vision*, pp.2223–2232.

[22] **Yi, Z.**, **Zhang, H.**, **Tan, P. and Gong, M.** (2017). DualGAN: Unsupervised dual learning for image-to-image translation, *Proceedings of the IEEE International Conference on Computer Vision*, pp.2849–2857.

[23] **Gatys, L.A.**, **Ecker, A.S. and Bethge, M.** (2015). A neural algorithm of artistic style, *arXiv preprint arXiv:1508.06576*.

[24] **Dash, A.**, **Gamboa, J.C.B.**, **Ahmed, S.**, **Liwicki, M. and Afzal, M.Z.** (2017). Tac-gan-text conditioned auxiliary classifier generative adversarial network, *arXiv preprint arXiv:1703.06412*.

[25] **Xu, T.**, **Zhang, P.**, **Huang, Q.**, **Zhang, H.**, **Gan, Z.**, **Huang, X. and He, X.** (2018). AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1316–1324.

[26] **Mansimov, E.**, **Parisotto, E.**, **Ba, J.L. and Salakhutdinov, R.** (2015). Generating images from captions with attention, *arXiv preprint arXiv:1511.02793*.

[27] **Zhang, H.**, **Xu, T.**, **Li, H.**, **Zhang, S.**, **Wang, X.**, **Huang, X. and Metaxas, D.N.** (2017). StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks, *Proceedings of the IEEE International Conference on Computer Vision*, pp.5907–5915.

[28] **Zhang, H.**, **Xu, T.**, **Li, H.**, **Zhang, S.**, **Wang, X.**, **Huang, X. and Metaxas, D.N.** (2018). StackGAN++: Realistic image synthesis with stacked generative adversarial networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(8), 1947–1962.

[29] **Qiao, T.**, **Zhang, J.**, **Xu, D. and Tao, D.** (2019). MirrorGAN: Learning text-to-image generation by redescription, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1505–1514.

[30] **Hinz, T.**, **Heinrich, S. and Wermter, S.** (2019). Generating multiple objects at spatially distinct locations, *arXiv preprint arXiv:1901.00686*.

[31] **Reed, S.E.**, **Akata, Z.**, **Mohan, S.**, **Tenka, S.**, **Schiele, B. and Lee, H.** (2016). Learning what and where to draw, *Advances in Neural Information Processing Systems*, pp.217–225.

[32] **Park, H.**, **Yoo, Y. and Kwak, N.** (2018). Mc-GAN: Multi-conditional generative adversarial network for image synthesis, *arXiv preprint arXiv:1805.01123*.

[33] **Reed, S.**, **Akata, Z.**, **Yan, X.**, **Logeswaran, L.**, **Schiele, B. and Lee, H.** (2016). Generative adversarial text to image synthesis, *arXiv preprint arXiv:1605.05396*.

[34] **Řehůřek, R. and Sojka, P.** (2010). Software Framework for Topic Modelling with Large Corpora, *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, ELRA, Valletta, Malta, pp.45–50, `http://is.muni.cz/publication/884893/en`.

[35] **Krizhevsky, A.**, **Sutskever, I. and Hinton, G.E.** (2012). Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, pp.1097–1105.

[36] **He, K.**, **Zhang, X.**, **Ren, S. and Sun, J.** (2016). Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.770–778.

[37] **Mao, X.**, **Li, Q.**, **Xie, H.**, **Lau, R.Y.**, **Wang, Z. and Paul Smolley, S.** (2017). Least squares generative adversarial networks, *Proceedings of the IEEE International Conference on Computer Vision*, pp.2794–2802.

[38] **Krizhevsky, A.**, **Hinton, G.** *et al.* (2009). Learning multiple layers of features from tiny images.

[39] **Coates, A.**, **Ng, A. and Lee, H.** (2011). An analysis of single-layer networks in unsupervised feature learning, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp.215–223.

[40] **Heusel, M.**, **Ramsauer, H.**, **Unterthiner, T.**, **Nessler, B. and Hochreiter, S.** (2017). GANs trained by a two time-scale update rule converge to a local nash equilibrium, *Advances in Neural Information Processing Systems*, pp.6626–6637.

[41] **Kingma, D.P. and Ba, J.** (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.

[42] **Szegedy, C.**, **Vanhoucke, V.**, **Ioffe, S.**, **Shlens, J. and Wojna, Z.** (2016). Rethinking the inception architecture for computer vision, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.2818–2826.

CURRICULUM VITAE

Name Surname: Azmi Can Özgen

Place and Date of Birth: Istanbul, Turkey, 4th of August, 1993

E-Mail: ozgenaz@itu.edu.tr

EDUCATION:

- **B.Sc.:** 2011-2016, Istanbul University, Faculty of Engineering, Department of Electronics and Communication Engineering
- **B.Sc.:** 2012-2016, Istanbul University, Faculty of Engineering, Department of Physics Engineering
- **M.Sc.:** 2017-2020, Istanbul Technical University, Faculty of Computer and Informatics, Department of Computer Engineering

PROFESSIONAL EXPERIENCE

- Sep 2019 - Present, Researcher at ITU Computer Engineering Dept., SiMiT Lab, Istanbul
- Apr 2018 – Sep 2019, Machine Learning Engineer at Miletos Inc., Istanbul
- Sep 2017 - May 2018, Researcher at ITU Computer Engineering Dept., SiMiT Lab, Istanbul
- Jan 2015 – Sep 2015, Research Scientist at Miletos Inc., Istanbul

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Özgen, A.C., Aghdam, O. A. and Ekenel, H.K., 2020, May. Text-to-Painting on a Large Variance Dataset with Sequential Generative Adversarial Networks. In 2020 28th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Özgen, A.C., Fasounaki, M. and Ekenel, H.K., 2018, May. Text detection in natural and computer-generated images. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.
- Özgen, A.C., Fasounaki, M. and Ekenel, H.K., 2018, May. Generalized circle agent for geometry friends using deep reinforcement learning. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.