ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Deep Learning

ARTISTIC STYLE IMITATION WITH GENERATIVE ARTIFICIAL INTELLIGENCE

CANDIDATE
Tiberio Marras

SUPERVISOR
Prof. Andrea Asperti

Academic year 2024-2025

Session 1st

To my mother, my brother, my dad, my aunt, my grandfather and my friends.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the introduction of ChatGPT in 2022, generative artificial intelligence has spread rapidly[36].

Currently, many generative models are available, and people use them daily.

Among these, image-generative models play a crucial role. This type of models can be text-to-image, image-to-image or both. Image-to-image models take an image as input and generate a modified or transformed version of it. Text-to-image models take a text as input and generate an image based on its indications.

The purpose of this work is to assess the capability of text-to-image models to reproduce the style of the artistic movements from 1500s to the first half of 1900s. In order to do so, I created, together with my colleagues[5], a labeled dataset of images. We generated these images using 12 models: Dall-E 3, Firefly Image 3, Auto-Aesthetics, Midjourney, Stable Diffusion 1.5, Stable Diffusion 3.5-large, Leonardo Phoenix, Omnigen, Ideogram, Kolors 1.5, Flux 1.1 Pro and Flux Schnell.

The models were given a set of prompts describing several artistic movements, in particular each prompt was given to all the models.

Subsequently we created two surveys in order to assess the quality of the images generated and the capability of the models to follow the given prompts.

The thesis is structured as follows. In Chapter 3, I describe how we created the dataset, selected the models, and evaluated them. Chapter 4 provides an overview of text-to-image models and describe the ones used in this work. In Chapter 5, I outline the structure of the dataset and I conduct an analysis of it. Chapter 6 defines the surveys, their purpose, and how they were conducted. In Chapter 7, I discuss the results obtained from the surveys. Finally, in Chapter 8, I present the conclusions based on the results.

Chapter 2

Related works

Recent years have seen remarkable progress in the style transfer of generative models, largely driven by contributions from many researchers[49][50]. Since its introduction, in 2015[14], several architectures have been employed to implement it[46]. This work contributes by providing an evaluation of state-of-the-art models in terms of style adherence and prompt fidelity. Since a research evaluating the state of the art in style transfer is missing, I will analyze studies that propose new approaches for image generation, with a particular focus on the methods used to evaluate them. Subsequently, I will introduce works that present new metrics for the analysis of generative models.

Sidonie Christophe et al. in their study[13] explored the use of Generative Adversarial Networks (GANs)[15] for cartographic style transfer, specifically the application of the visual style of a historical or artistic map to satellite images or modern maps. Among the methods employed for evaluating the produced images is **perceptual assessment**: they compared the generated images with real maps in order to assess how realistic and consistent the style transfer is. However, the limited number of experts involved in applying this method may have resulted in a lack of objectivity.

In their work[23], Justin Johnson et al. aimed to improve style transfer and super-resolution with Convolutional Neural Networks (CNN)[45]. Human evaluation is one of the approaches used to assess the presented model:

they run a user study on Amazon Mechanical Turk to compare the quality of images generated by different techniques.

Jianbo Wang et al. also relied on Amazon Mechanical Turk to evaluate **STyle TRansformer** (STTR), a transformer-based model they presented in [44]. Specifically, 50 participants were shown 20 pairs of content and style images. For every pair the resulting images generated with 7 different methods (including the one proposed by the authors) were presented. The participants were asked to select the image they preferred based on aesthetic criteria and the fidelity of the style transfer.

Dar-Yen Chen et al. adopted a similar approach to analyze the performance of the model they presented in [11]. In particular, they engaged 212 participants to rate images generated by their model (and others) on a scale from 1 to 7, assessing textual accuracy, stylistic fidelity, and overall quality.

Some researchers have preferred to delve deeper into automatic metrics for evaluating generative models.

In their study[39], Mehdi S. M. Sajjadi et al. argue that traditional evaluation metrics like Fréchet Inception Distance (FID)[4] and Inception Score (IS)[10] cannot distinguish between different failure cases of generative models, due to the fact that they provide a single-score evaluation. For this reason they introduce **precision** and **recall** for distributions: the former measures the quality of generated samples while the latter measures how much of the target distribution is covered. The authors propose an algorithm that computes precision and recall for distributions based on sample comparisons. They tested this algorithm on GANs and Variational Autoencoders (VAEs)[24].

More specific to style transfer are the metrics introduced by Mao-Chuang Yeh et al. in [47]. The authors introduced **Effectiveness** and **Coherence**, which respectively measure how much the style has been transferred and how well the content has been preserved. These metrics have been calibrated using human evaluations: a study of 2 rounds involving ~50 users was conducted

and the results were associated with the metrics using logistic regression models.

As previously mentioned, however, none of these studies focus on evaluating the state of the art of text-to-image models in reproducing artistic movements. This represents the main contribution of our work.

Chapter 3

Methodology

In this section I describe how we created the dataset and the methodology we used to select the models and evaluate them.

3.1 Creation of the dataset

The first step in creating the dataset involved using ChatGPT to generate the prompts.

To obtain a prompt, we asked ChatGPT to produce one based on an artistic movement, a historical period and, occasionally, a specific artist.

We then provided the output to the models and fed the generated images back into ChatGPT, asking if it could recognize the style. If the style identified by ChatGPT matched the one specified in the prompt, this was considered ”acceptable.”

Below are a couple of examples:

- ”A portrait of a medieval princess in the baroque style of the first half of the XVII century. You may get inspiration from Diego Velázquez. The princess is adorned in rich clothing, with intricate embroidery and a formal gown typical of the period. The background is dark and muted, allowing the princess to stand out in the center of the composition. The color palette includes soft, royal hues such as deep reds, golds, and

whites, capturing the elegance and poise of the subject. The brushwork reflects Velázquez's masterful technique, with soft blending and detailed textures, creating a lifelike yet majestic presence for the princess."

- "A landscape painting in the style of classical European art from the 17th century, similar to works by Gaspard Dughet. The scene should feature a serene countryside with rolling hills, dense forests, and a prominent rocky mountain in the background. Include detailed trees in the foreground, some showing intricate foliage. The sky should be filled with dramatic clouds, capturing a soft interplay of light and shadow. In the lower part of the painting, include small human figures engaged in pastoral activities, such as shepherds with animals or travelers resting, to give a sense of scale and narrative."

As previously stated, we provided each prompt as input to all the models. To achieve this, we divided the models among ourselves.

Some models had restrictions on the instructions that could be provided as input. For example, most of the models did not accept the presence of nudity. Figure 3.1 illustrates the punishment of Prometheus, an image that many models struggled to generate due to these restrictions.

Figure 3.1: The punishment of Prometheus (Firefly Image 3)

Another restriction we met was in the atmosphere described by the prompt. For example Auto-Aesthetics didn't allow the word "unsettling". In order to overcome this restriction we removed the unaccepted words from the prompt.

Once we generated the image we used a Colab notebook to upload it and add the metadata associated on Kaggle.

3.2 Selection of the models

The models we selected are among the most widely used at present.

In particular, once we found a candidate model, we started generating images using different prompts, then we evaluated it according to the following criteria:

- **Realism**: if the images generated represented detailed subjects without being too realistic.

- **artifact minimization**: if the details in the image were well represented and there were not errors in the depiction (e.g. distorted eyes, third legs etc.)

- **Adherence to the prompt** If the images followed accurately the indications given in the prompt used to generate them.

We tried 14 models, but only 12 were considered suitable for our work.

The models we discarded were DeepFloydIF and RunwayML. This because they produced images that were unsatisfactory both in terms of quality and adherence to the prompt.

In figure 3.2 I show two images generated by DeepFloydIF with a romanticist and a renaissance prompt.

(a) Romanticist (b) Renaissance

Figure 3.2: A romanticist and a renaissance image generated by DeepFloydIF.

3.3 Evaluation

Once we created the dataset we designed two surveys to evaluate the performance of the chosen models.

In particular we used the surveys to assess the models using two principles: **authenticity** and **adherence to the prompt**.

3.3.1 Authenticity

We consider an image authentic if it can be misclassified as human-made. This characteristic is crucial in assessing the effectiveness of generative models, particularly in the generation of paintings.

Several factors can influence the authenticity of an image:

- **Presence of brushstrokes**: if the image contains visible brushwork it is more likely that a person would consider it made by a human.
- **Realism**: if the artwork is not overly realistic is less likely that a person would classify it as AI-generated, due of the presence of ”human” errors.
- **Lack of artifacts**: AI models often make mistakes in generating body parts such as limbs, faces, and hands, which can reduce perceived authenticity.
- **Familiarity with Traditional Artistic Techniques**: if an artwork replicates established painting techniques, it might be mistaken for a real painting.

3.3.2 Adherence to the prompt

The second principle we considered in assessing the models is their ability to follow the given prompt. A model could be able to generate high quality images but fail in following the indications given in input. This classification task is more complex than the previous one, as it needs a deep understanding of the prompt.

When evaluating adherence to the prompt, three sub-criteria can be considered:

- **Adherence to the style**
- **Adherence to the content**

- **Technical adherence** (absence of distortions and artifacts)

We also considered to use CLIP, or similar models, for an automated evaluation, but we weren't sure about their ability to address nuanced factors, such as style, historical period and other attributes. We are planning to explore these methods in future research.

Chapter 4

The models

In this chapter I present the functionality and architecture of the most widely used image generative models, then I describe the models selected for this work.

4.1 Overview

T2I (text-to-image) models have seen rapid advancements in recent years[9][33], leveraging deep learning techniques such as diffusion models and transformers. They are widely used in art, design, content creation, and scientific visualization.

4.1.1 Diffusion models

In 2015 there has been the introduction of diffusion models. They are a class of generative models that create high-quality images (and other types of data[19]) by progressively refining random noise [27].

The generic pipeline in diffusion models is generally specified by the **forward process**, the **reverse process**, and the **sampling procedure**[22].

- **Forward process** The forward process gradually perturbs a training sample x_0 into the sequence $\{x_t\}_{t=1}^{T}$ as the timestep t progresses. Each

forward transition $p(x_t|x_{t-1})$ defines this perturbation, with a small amount of noise e_t introduced at each step. As the process advances along the chain, increasing noise is added through $p(x_t|x_{t-1})$, causing the perturbed sample x_t to become progressively noisier. Since the forward process exclusively adds noise throughout the chain, it does not involve any trainable parameters. Specifically, it is structured as a sequence of forward transitions:

$$p(x_T|x_0) := p(x_1|x_0)...p(x_t|x_{t-1})...p(x_T|x_{T-1}) = \prod_{t=1}^{T} p(x_t|x_{t-1})$$

where t represents the timestep, T denotes the total number of timesteps and x_0 is the original training sample at $t = 0$ which is gradually perturbed until reaching x_T after T timesteps. The term $p(x_t|x_{t-1})$ describes the transition distribution between two consecutive timesteps. The forward process shares both similarities and distinctions with VAE. Like VAEs, it typically transforms $p(x_0)$ into the isotropic Gaussian distribution, setting $p(x_T) = \mathcal{N}(0, I)$ as the terminal distribution. However, unlike VAEs, the forward process does not include trainable parameters and solely perturbs x_0 by adding noise.

- **Reverse process** The reverse process trains a denoising network to progressively remove noise step by step. Unlike GANs, which eliminate all noise in a single step, the denoising network is trained to iteratively denoise the sample between consecutive timesteps. The reverse process proceeds backwards along the multi-step chain as t decreases from T to 0. This iterative noise removal is referred to as the reverse transition $p_\theta(x_{t-1}|x_t)$, which is learned by optimizing the trainable parameters θ within the denoising network. The reverse process is mathematically expressed as a sequence of reverse transitions:

$$p_\theta(x_0) := p(x_T)p_\theta(x_{T-1}|x_T)...p_\theta(x_{t-1}|x_t)...p_\theta(x_0|x_1)$$

$$= p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

where θ represents the parameters of the denoising network, and $p_\theta(x_{t-1}|x_t)$ denotes the reverse transition distribution. Typically, the reverse process is parameterized as:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t))$$

where $\mu_\theta(x_t, t)$ and $\sigma_\theta(x_t, t)$ are the Gaussian mean and variance, respectively, estimated by the network θ. The denoising network is trained by optimizing the usual variational bound on negative log likelihood [17]:

$$L = \mathbb{E}[D_{KL}(p(x_T|x_0)||p(x_T))$$

$$+ \sum_{t\geq 1} D_{KL}(p(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) - log p_\theta(x_0|x_1)]$$

where $D_{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) divergence, which quantifies the difference between two probability distributions. Basically, minimizing the objective L means reducing the discrepancy between $p_\theta(x_0)$ and $p(x_0)$.

- **Sampling procedure** In the sampling procedure the denoising network θ^* generates new data x_0. It progresses backward along the chain, recursively applying the trained network θ^*. Specifically, the process begins by drawing a sample x_T from the terminal distribution $p(x_T)$. The trained network is then employed iteratively to remove noise through the sampling transition $p_{\theta^*}(x_{t-1}|x_t)$. By successively applying the transition along the chain, the procedure ultimately produces new data, where $x_0^* \sim p_{\theta^*}(x_0) \approx p(x_0)$. More formally, the sampling procedure is represented as a sequence of sampling transitions:

$$p_{\theta^*}(x_0) := p(x_T) p_{\theta^*}(x_{T-1}|x_T) ... p_{\theta^*}(x_0|x_1) = p(x_T) \prod_{t=1}^{T} p_{\theta^*}(x_{t-1}|x_t)$$

where θ^* represents the optimized parameters of the denoising network, $p(x_T)$ is the terminal distribution and $p_{\theta^*}(x_{t-1}|x_t)$ is the sampling transition.

Architecture

The core of diffusion models is the U-Net, a convolutional neural network with an encoder-decoder structure. It is composed of three main parts: **encoder**, **bottleneck** and **decoder**.

- **Encoder**: it compresses the image reducing the resolution. Every layer of the encoder contains:
  - **convolutions 3x3 with ReLU**.
  - **batch normalization**: to stabilize the training.
  - **downsampling with strided 2x2 convolutions or pooling 2x2**: to reduce the dimension.
- **Bottleneck**: it is the deepest part of the network, where the noisy signal is processed. In the bottleneck the image is highly compressed. Additional convolutions are applied and in some architectures self-attention layers are used.
- **Decoder**: it takes the compressed representation from the encoder and reconstructs a clearer image, gradually removing noise. It has three main characteristic:
  - **transposed convolutions**: used to increase the resolution.
  - **skip connections**: they take in input the features from the encoder, in order to avoid loss of informations during the compression of the image.
  - **3x3 convolutions**: to refine the details.

In some architectures **time conditioning** is included: the model receives a temporal information on which diffusion step is computing. This is obtained with a **sinusoidal embedding** concatenated to the features.

In text-to-image models the output of an encoder with a text embedding is concatenated to the features of the U-Net.

Latent diffusion models

They are a kind of diffusion models where the U-Net works on a compressed representation of the image, called the latent space[37][7]. In particular, before passing through the U-Net, the input is compressed by an encoder. Once the noise is removed from the latent image by the U-Net, the output passes through a decoder which upscales the resolution back to its original size. The main advantage of these models is that they work faster without losing too much quality.

In Figure 4.1 I show the architecture of Stable Diffusion, a widely used latent diffusion model.

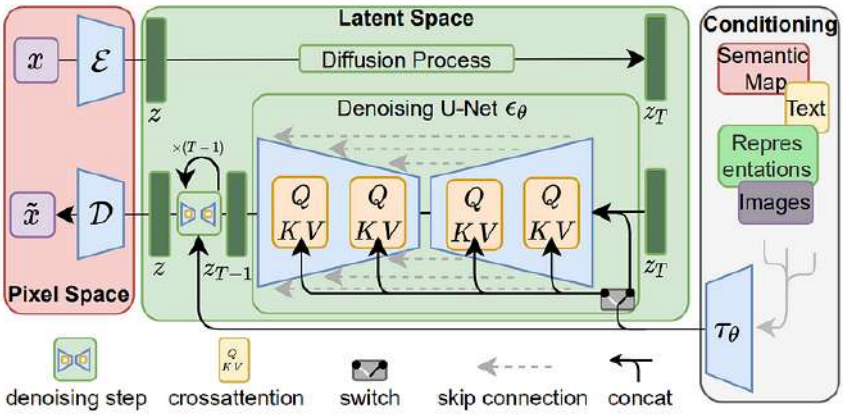

Figure 4.1: The architecture of Stable Diffusion

4.1.2 Transformers

In 2017 there has been a breakthrough in generative models with the introduction of transformers[43].

A transformer is a deep learning model architecture designed primarily for processing sequences of data, such as text[30], images[41][35], and even audio[12]. The architecture remains the same across different kinds of media; only the type of the generated tokens differs. It consists of two main components: **encoder** and **decoder**[28].

Encoder

Its purpose is to transform the input tokens into contextualized representations. Unlike earlier models that processed tokens independently, the transformer encoder captures the context of each token with respect to the entire sequence. The workflow of the encoder can be divided into 4 stages:

- **Input embeddings**: the embedding only happens in the first layer. It converts the input tokens into numerical vectors that represent their semantic meaning.
- **Positional encoding**[38]: Since transformers do not have a recurrence mechanism like RNNs, they use positional encodings added to the input embeddings to provide information about the position of each token in the sequence. In particular the position is encoded using the sine and cosine functions:

$$PE_{(pos,2i)} = sin(pos/N^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = cos(pos/N^{2i/d_{model}})$$

Where pos is the position of the token in the sentence, i is the index of each dimension of the vector, d_{model} is the dimension of the embedding and N is a free parameter that should be significantly larger than the biggest i. The original paper uses $N = 10000$. The sine is used for the even dimensions and the cosine is used for the odd dimensions. The positional encoding dimension matches the embedding dimension,

allowing the embedding layer's output to be directly summed with the positional encoding.

- **Stack of encoder layers**: the transformer encoder includes a stack of identical layers (6 in the original Transformer model). Each layer has the following sublayers:
  - **Multi-headed self-attention mechanism**[48]: it is a set of attention heads. In each of these every token is firstly transformed in 3 matrices: Q, K and V. Then the attention score of each token with respect to other tokens is computed:

$$Scores = \frac{Q \times K^T}{\sqrt{d_k}}$$

    The result is normalized with a softmax and multiplied for V. Finally the output of the heads is concatenated and projected through a linear layer.
  - **Normalization and residual connections**: each sub-layer in an encoder layer is followed by a normalization step. Also, each sublayer output is added to its input (residual connection) to help mitigate the vanishing gradient problem, allowing deeper models.
  - **Feed-forward neural network**[6]: the output of the normalization layer passes through two linear layers with a ReLU in between for additional refinement.
- **Output of the encoder**: it is a set of vectors, each representing the input sequence with a rich contextual understanding.

In Figure 4.2 I show one layer of the encoder.

Decoder

The role of the decoder is to generate sequences. The decoder operates in an autoregressive manner, meaning that each output token depends only on

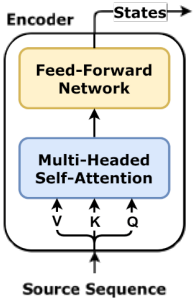

Figure 4.2: One layer of the encoder.

previously generated tokens. Its architecture is similar to that of the encoder. Specifically, the workflow of the decoder consists of four steps:

- **Output embeddings**: the input passes through the embedding layer.
- **Positional encoding**: like in the encoder, the information about the position of the tokens in input is added to the output of the previous layer.
- **Stack of decoder layers**: the output of the positional encoding layer passes through a set of layers (6 in the original transformer). Each layer is structured as follows:
  - **Masked self-attention mechanism**: it is similar to the self-attention mechanism of the encoder, but with a crucial difference: the tokens in the sequence are not influenced by future tokens, only by the previous ones. This is obtained adding a mask matrix M with $-\infty$ where attention must be cut and 0 otherwise:

    $$MaskedScores = M + \frac{Q \times K^T}{\sqrt{d_k}}$$

    The output passes then through a softmax function and is multiplied by V.
  - **Cross attention**: a self-attention layer where Q is computed from the output of the previous layer whereas K and V are computed

from the output of the encoder. Since cross attention works with the output of the encoder, it doesn't need the mask.

– **Feed-forward neural network** like for the encoder, the decoder includes a fully connected feed-forward network in every layer.

- **Linear classifier and Softmax**: the final layer of the decoder is a linear classifier, whose output dimension is the total number of classes involved. The output passes then through a softmax, that transforms it into a range of probability scores, each lying between 0 and 1.

  The index of the highest of these probability scores points to the next token in the sentence.

Each sub-layer of the decoder is followed by a normalization layer and includes a residual connection around it. In the first iteration the decoder takes in input the start token and the output of the encoder. From the second iteration the decoder takes in input the previously generated tokens and the output of the encoder. The cycle repeates until the decoder predicts the end token. In Figure 4.3 I show one layer of the decoder.

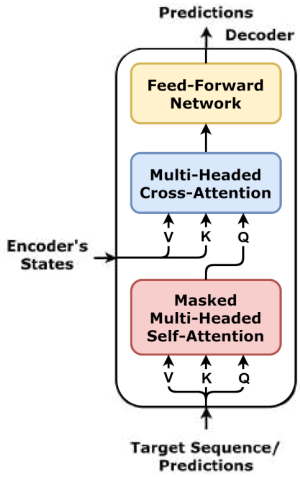

Figure 4.3: One layer of the decoder.

4.2 Our models

As mentioned earlier, we selected the models from among the most widely used at present. A detailed description of each model used in this work is provided below. Subsequently, Table 4.1 summarizes their key characteristics.

4.2.1 Ideogram 2.0

Developed by Ideogram Inc., it was released on August 21, 2024.

It generates images using text-to-image conditioning at a default resolution of 1024x1024. The model allows the generation of images in any aspect ratio and with adherence to a specific color palette. According to the creators [20] it can produce images that can convincingly resemble real photos. The architecture of the model has not been disclosed.

4.2.2 Flux 1.1 Pro

Developed by Black Forest Labs, released on October 4, 2024.

According to the creators[26] it is 6 times faster than the previous model and has the highest overall Elo score[8] in the Artificial Analysis Image Arena (as of October 1, 2024). It has a default resolution of 2048x2048 and the output shape is configurable. It is based on a hybrid architecture of multimodal and parallel diffusion transformer blocks with 12B parameters [25]. The model leverages flow matching[18]. In addition it incorporates rotary positional embeddings and parallel attention layers.

4.2.3 Flux Schnell

The faster version of Flux 1.1 Pro, tailored for local development and personal use. It was released on August, 2 2024. It is openly available under Apache 2.0 license. It has been trained using guidance distillation from Flux 1.1 Pro [40],

whose architecture it closely resembles. It generates images with a default resolution of 2048x2048 and has a configurable output shape.

4.2.4 Dall-E

Developed by OpenAI, released on October 2023.

It is built natively on ChatGPT [31]. According to the creators [32] it can reliably render intricate details, including text, hands, and faces and it is particularly good in responding to extensive, detailed prompts. The architecture of the model is composed of two main parts:

- **Discrete Variational Autoencoder** (dVAE)[29]: it compresses the images into a sequence of discrete tokens. It helps to reduce the dimensionality of images while preserving essential details.
- **Transformer-based image generation**: it uses autoregressive modeling. It predicts the next token in the sequence, conditioned on previous tokens.

It has a default output resolution of 1024x1024 and it can support both landscape and portrait aspect ratio.

4.2.5 Firefly Image 3

Developed by Adobe, released on April 23, 2024.

According to the creators [1] it delivers high quality outputs with high variety, giving users control and personalization over the styles of images they generate. It supports text-to-image and image-to-image conditioning. Has a default output resolution of 2048x2048 and it has configurable output shape. The architecture has not been disclosed.

4.2.6 Omnigen

Developed by Beijing Academy, released on October 22, 2024.

It adopts an architecture comprised of a VAE and a pre-trained transformer model. The input to the model can be multi-modal interleaved text and images in free form. The VAE extracts continuous visual features from images, while the tokenizer of Phi-3, a family of large language models developed by Microsoft, processes the text input. It also applies frequency-based positional embedding and appends the timestep embedding. It removes noise from the image in a latent space but, differently from classical diffusion models, it doesn't have a U-Net at its core. It instead leverages a transformer to apply flow matching, thanks to which the model learns the direct transformation between the noisy image and the final image. It has a default output resolution of 2048x2048 and the output shape is configurable.

4.2.7 Leonardo Phoenix

Developed by Leonardo Interactive Pty, released on June 2024.

According to the creators [34], Leonardo Phoenix faithfully follows the prompt and is capable of rendering coherent text in the picture. Has a default output resolution of 1024x1024 and a configurable output shape. It supports text-to-image and image-to-image conditioning. The architecture of the model has not been disclosed.

4.2.8 Midjourney V6.1

It is the latest version of the model developed by Midjourney Inc., released on July 31, 2024.

According to the creators [21] it offers more coherent images, much better image quality and more precise, detailed, and correct small image features compared to the previous version. It supports text-to-image and image-to-image conditioning. The default output resolution is 1024x1024 and has a configurable output shape. The architecture has not been disclosed.

4.2.9 Stable Diffusion 1.5

Developed by Stability AI, released on October 2022.

It supports text-to-image and image-to-image conditioning. It has a default output resolution of 512x512 and a configurable output shape. The architecture of the model has not been disclosed.

4.2.10 Stable Diffusion 3.5-large

Is the upgraded version of Stable Diffusion 1.5, released on October 22, 2024. The model has 8.1 billion parameters. It integrates Query-Key Normalization into the transformer blocks [3]. According to the creators the model excel in the following areas: customizability (it can be fine-tuned), efficient performance (it is optimized to run on standard consumer hardware), diverse outputs (it creates images representative of the world without the need for extensive prompting) and versatile styles (it is capable of generating a wide range of stiles). It supports text-to-image and image-to-image conditioning. Has a default output resolution of 1024x1024 and a configurable output shape. The architecture of the model has not been fully disclosed.

4.2.11 Kolors 1.5

Developed by Kuaishu.

It is a latent diffusion model [42]. Differently from other diffusion models, which use CLIP, it uses GLM (General Language Model) as text encoder. The model was trained on a dataset that was re-labeled using a multimodal language model to enhance the comprehension of textual descriptions. The training was composed of two phases:

- **phase 1**: the model was trained on a public dataset, in this phase it learned the general concepts such as shapes, outlines and proportions.
- **phase 2**: the model was trained on a filtered dataset with high quality

images, in this phase the model learned to generate aesthetically better images.

It performed better than SD3, Dall-E 3 and Playground v2.5 in Multi-Dimensional Preference Score (MPS) and slightly worse in FID. It supports text-to-image and image-to-image conditioning. Has a default resolution of 1024x1024 and a configurable output shape.

4.2.12 Auto-Aesthetics V1

Developed by Neural.love, released on August 14, 2024.

According to the creators[2], the model analyzes the likes and the scores of the user on previously generated images, then uses these informations to adapt and generate images that the user would appreciate. It supports only text-to-image conditioning. Has a default output resolution of 1024x1024 and a configurable output shape.

Model	Creator	Architecture	Conditioning	Resolution (default)	Configurable output shape
Ideogram 2.0	Ideogram AI	Not disclosed.	text-to-image	1024x1024	Yes
Flux 1.1 Pro	Black Forest Labs	Multimodal and parallel diffusion transformer blocks with flow matching.	text-to-image	2048x2048	Yes
Flux.1 Schnell	Black Forest Labs	Fast version of Flux.1.1, trained using latent adversarial diffusion distillation.	text-to-image	2048x2048	Yes
Dall-E 3 (via ChatGPT-4o)	OpenAI	Transfomer with dVAE.	text-to-image	1024x1024	Yes
Firefly Image 3	Adobe	Not disclosed.	text-to-image, image-to-image	2048x2048	Yes
OmniGen	Beijing Academy	Transformer with flow matching.	multimodal-to-image	2048x2048	Yes
Leonardo Phoenix	Leonardo Interactive Pty	Not disclosed.	text-to-image, image-to-image	1024x1024	Yes
Midjourney V6.1	Midjourney	Not disclosed.	text-to-image, image-to-image	1024x1024	Yes
Stable Diffusion 1.5	Stability AI (dismissed)	Not disclosed.	text-to-image, image-to-image	512x512	Yes
Stable Diffusion 3.5-large	Stability AI	Transformer based, not fully disclosed.	text-to-image, image-to-image [16]	1024x1024	Yes
Kolors 1.5	Kuaishou Kolors team - Kling AI	Large-scale latent diffusion based model.	text-to-image, image-to-image	1024x1024	Yes
Auto-Aesthetics V1	Neural.love	Not disclosed.	text-to-image	1024x1024	Yes

Table 4.1: Comparison between the used Models.

Chapter 5

The dataset

This chapter provides an overview of the dataset structure and an analysis of its content.

5.1 Structure

The dataset consists of a collection of 953 images, each accompanied by a corresponding row in a metadata file.

The aforementioned has 6 columns:

- **generative_model**: the model used to generate the image. This is one of the 12 models we selected according to the methodology discussed in Section 3.2.
- **subject**: a list of tags describing the prompt. There are a total of 45 tags, each prompt is associated with a restricted group of them. Tags are stored as a comma-separated list.
- **style**: the style of the artwork. In the dataset there are prompts of 19 different artistic movements.
- **period:** the period of the style of the artwork.
- **prompt**: the prompt used to generate the image.

- **generated_image**: the name of the file of the image.

In particular, there are 73 unique prompts, each of which has been given to the 12 models to generate the corresponding image. Some prompts have been given multiple times to the same model.

5.2 Analysis

The dataset exhibits certain imbalances, particularly in terms of artistic periods and styles. As shown in Table 5.1, the artworks reproducing artistic movements of XX and XIX centuries are the most frequent.

Table 5.2 shows the most represented styles, with Renaissance and Impressionism being the dominant artistic movements. We plan to explore mitigation strategies to address these imbalances in future work.

Period	Total	%
XX century	307	32.2
XIX century	289	30.3
XVI century	153	16.1
XVII century	117	12.3
XV century	49	5.1
XVIII century	38	4

Table 5.1: Dataset statistics by period.

Style	Total	%
Renaissance	202	21.2
Impressionism	136	14.3
Romanticism	92	9.7
Baroque	86	9.0
Realism	60	6.3
Surrealism	50	5.2
Dadaism	44	4.6

Table 5.2: Most represented styles.

One of the most challenging prompts for the models to adhere was the following:

”A painting of the Baptism of Jesus in the style of Giovanni Bellini. The scene depicts John the Baptist pouring water over Jesus, who stands at the center of the composition in a calm river. The landscape in the background is lush and serene, with rolling hills and a soft blue sky, evoking tranquility. Surrounding them are a few peaceful figures, similar to Bellini’s balanced, serene compositions. The figures are adorned in flowing robes, and the color

palette features Bellini's warm earth tones, golds, and soft greens. The light is soft and diffused, typical of Bellini's religious works, creating a harmonious and contemplative atmosphere. The figures are anatomically precise, and the overall composition emphasizes balance and serenity."

(a) ChatGPT with Dall-E 3

(b) Auto-Aesthetics

(c) Firefly Image 3

(d) Flux 1.1 Pro

(e) Flux Schnell

(f) Ideogram

(g) Kolors

(h) Leonardo Phoenix

(i) Midjourney V6.1

(l) Omnigen

(m) Stable Diffusion 3.5-large

(n) Stable Diffusion 1.5

Figure 5.1: Images generated by the models using the prompt 'Baptism of Christ'.

As shown in Figure 5.1, nearly all the models accurately rendered the color

tones specified in the prompt, but struggled to reproduce the content described. In the images generated by Firefly Image 3 and Kolors (c, g) John the Baptist is absent and in the images produced by Flux 1.1 Pro, Leonardo Phoenix, Omnigen, Stable Diffusion 3.5-large and Stable Diffusion 1.5 (d, h, l, m, n) he is not pouring water on Jesus.

In the images created by ChatGPT with Dall-E 3, Auto-Aesthetics, Flux Schnell and Midjourney V6.1 (a, b, e, i) the water flows from the hand of John the Baptist unnaturally. The most accurate image appears to be the one produced by Ideogram (f), although it exhibits some artifacts, particularly in the rendering of hands.

It is important to note that only the images generated by Kolors and Midjourney 6.1 (g, i) show Jesus Christ with a halo, meaning that the models conveyed the religious essence of the painting.

Interesting is also how the models processed the following prompt:

”A painting in the style of Sandro Botticelli, depicting Saint George and the dragon. The scene features Saint George on horseback, dressed in elegant, flowing medieval armor, slaying a dragon with a lance. The composition is balanced and ornate, set in a detailed, idyllic landscape with soft rolling hills, a serene sky, and distant architectural elements typical of Botticelli’s style. The dragon is depicted as a mythical creature, with intricate detailing on its scales and wings. The artwork emphasizes grace and harmony, with flowing lines, delicate colors, and a sense of movement and spirituality. The lighting is soft and even, enhancing the painting’s serene and elegant atmosphere.”

(a) ChatGPT with Dall-E 3

(b) Auto-Aesthetics

(c) Firefly Image 3

(d) Flux 1.1 Pro

(e) Flux Schnell

(f) Ideogram

(g) Kolors

(h) Leonardo Phoenix

(i) Midjourney V6.1

(l) Stable Diffusion 1.5

(m) Stable Diffusion 3.5-large

(n) Omnigen

Figure 5.2: Images generated by the models using the prompt 'Saint George and the dragon'.

As we can see from Figure 5.2 the models struggled to reproduce both the style and the content described in the prompt. The style of the image generated by Auto-Aesthetics (b) is too realistic and the image produced by Firefly Image 3 (c) has dark tones and strong contrast, which differ from Botticelli's style. In the images created by Firefly Image 3, Kolors, Midjourney V6.1 and Stable Diffusion 1.5 (c, g, i, l) the dragon is absent and in almost all of those where it is present (b, d, e, f, n, m), it appears deformed. The image generated by Leonardo Phoenix (h) faithfully adheres to the prompt (soft tones, elegant

atmosphere, presence of architectural elements) but fails to correctly render the proportions. The most accurate image is the one created by ChatGPT with Dall-E 3 (a) which adheres to the prompt both in terms of style and content. It is noteworthy that all the models, except for Omnigen and Stable Diffusion 3.5-large (n, m), accurately rendered the horse, probably due to its strong presence in the training datasets.

Chapter 6

The surveys

In order to assess the capabilities of the models we created two online surveys.

6.1 First survey

The first survey was distributed to a wide range of people, around 600.

Each user was given 20 images which were randomly extracted from our dataset and a dataset of real artworks, and was asked to classify each one as human-made or AI-generated. This way we collected more than 12000 responses, each one corresponding to a classification of a specific image.

To ensure that the results weren't biased, the survey presented the images without metadata associated.

We created the survey with Google Scripts and we distributed it through the social media and within the university. The results were stored in a table in DynamoDB.

This survey was created to evaluate the ability of the models to generate images that people would classify as human-made.

6.2 Second survey

The second survey was distributed only to our team and some volunteers.

The aim was to classify each image according to its adherence to the prompt used to generate it. Since this task could result hard, we simplified it.

In this survey the user was indeed given each of the 73 prompts with the corresponding images generated by the models and was asked to classify the images as ”good”, ”medium” and ”low”, according to their adherence to the prompt.

This classifications are not meant to be considered absolute, but are relative to the set of images generated with the same prompt. In fact, for each prompt the number of images belonging to each category had to be equal to the total number of images divided by 4.

Due to the limited number of participants, the survey was developed directly on Google Colab. The results were stored in a CSV file in Google Drive.

Chapter 7

Results

In this section I describe the results obtained with the surveys.

7.1 First Survey

In Figure 7.1 I show the confusion matrix of the images in the first survey.

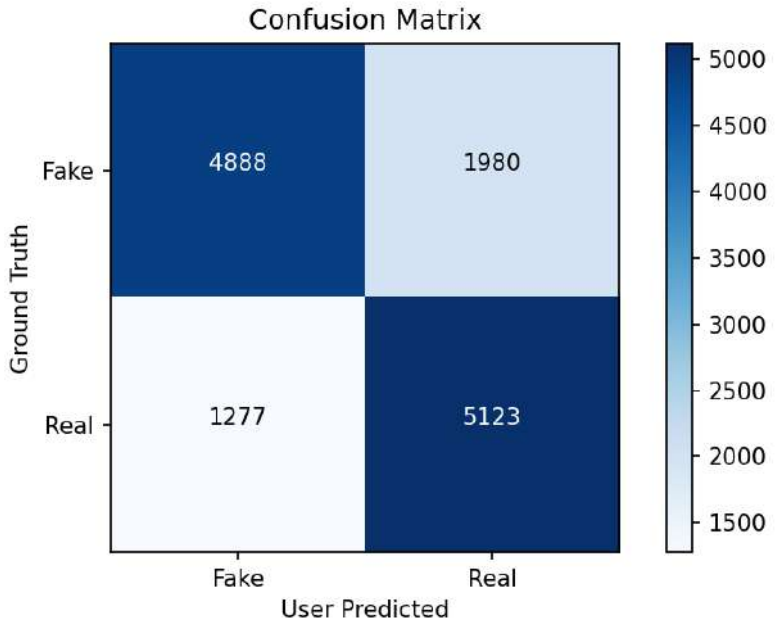

Figure 7.1: Confusion matrix

As we can see fake images have been classified as human-made around

29% of the times.

Surprisingly, real images have been classified as AI-generated around 20% of the times, probably because some original artworks were considered too realistic.

Figure 7.2 shows the ranking of the models by misclassification rate. As we can see Ideogram is the model with the highest performance, followed by Midjourney and Stable Diffusion 3.5.

The model with the lowest performance is instead Auto-Aesthetics V1, with a misclassification rate of less than 0.1.

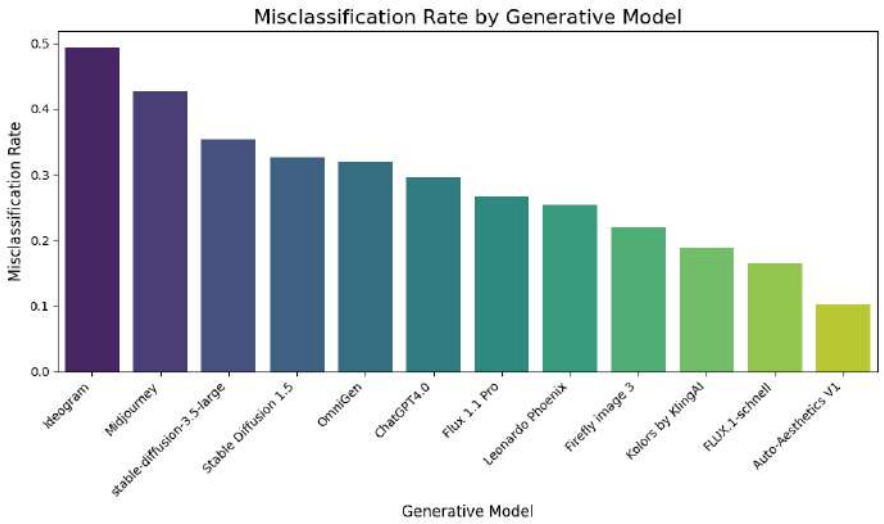

Figure 7.2: Misclassification rate by model

Another interesting result is the one that comes up looking at the periods (Figure 7.3).

As we can see from the Figure, the period with the highest misclassification rate is XX century, followed by XIX century and XVII century.

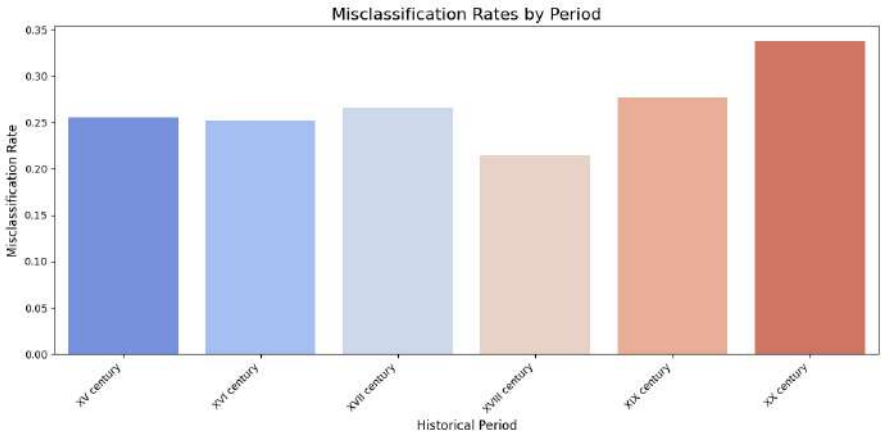

Figure 7.3: Misclassification rate by period

In Table 7.1 we can see the misclassification by style.

As previously mentioned for the periods, the styles corresponding to the most recent ones show the highest performance. In fact, ”art nouveau” is the style that performs best, followed by ”cubism” and ”satirical”.

The style with the worst performance is ”Naive”, probably because users associated the simple style of the paintings with the fact that they were AI-generated.

period	total count	misclassified	ratio
art nouveau	104	49	0.47
cubism	232	92	0.40
satirical	74	29	0.39
impressionism	922	350	0.38
dadaism	320	118	0.37
futurism	114	42	0.37
classicism	273	99	0.36
fauvism	119	40	0.34
expressionism	170	57	0.34
symbolism	302	98	0.32
vedutism	92	26	0.28
renaissance	1458	355	0.24
romanticism	635	154	0.24
abstractionism	91	20	0.22
baroque	574	123	0.21
realism	402	85	0.21
surrealism	334	70	0.21
rococo	157	30	0.20
naive	201	33	0.16

Table 7.1: Misclassification by style

As mentioned in Chapter 5, we assigned a list of tags to each prompt in the dataset.

Our aim was to quantify the impact of each tag on the misclassification rate.

To achieve this, we used a linear regression model in Python.

Specifically, we trained the model using the binarized tags as independent variables and the misclassification rate as the dependent variable.

In Figure 7.4 I present the weights of the tags extracted from the model.

As shown in the Figure, ”boat”, ”angels” and ”family” have the most positive influence on the misclassification rate.

As described in 3.3.1, the presence of brushstrokes positively influences the authenticity of an artwork, and this can be seen by the tag ”brushstrokes” in the Figure.

It is noteworthy that the tags ”animals”, ”landscape”, ”persons”, ”portrait” and ”rain” have no significant influence on the misclassification rate. This is likely because these elements appear frequently across many images, and other factors contribute more to classification decisions.

In contrast, the tag ”moon” have a negative influence on the misclassification rate, possibly indicating that models struggle to handle lighting in nighttime scenes. Similarly, the tags ’female’ and ’child’ have a negative impact on classification. This shows that the models still have difficulties in represent humans.

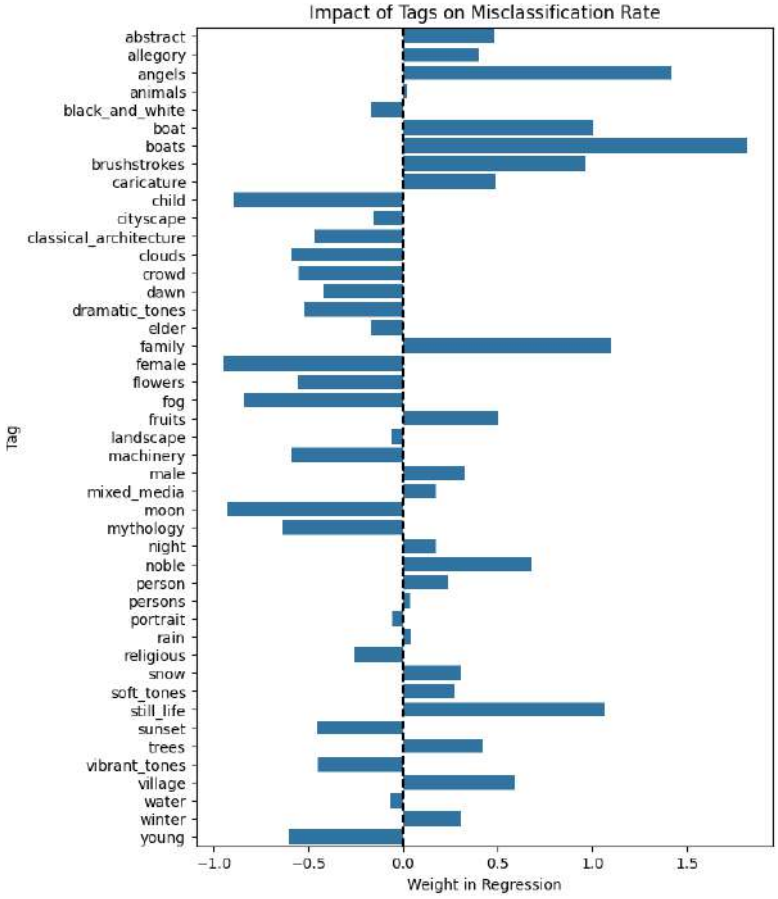

Figure 7.4: Impact of tags on misclassification rate

In Figure 7.5 we can see the best performing images with the associated model.

(a) Ideogram

(b) Midjourney

(c) Stable Diffusion 3.5 large

(e) Ideogram

(f) Ideogram

(g) Midjourney

Figure 7.5: The best performing fake images with associated model.

7.2 Second Survey

As already pointed out in Section 6.2, in the second survey the user was asked to classify an image as ”good”, ”medium” or ”low” according to its adherence to the prompt used to generate it.

In Table 7.2 we can see a ranking of the models by a weighted average of the classifications. In particular we assigned 1 to the ”good” classifications, 0 to the ”medium” classifications and -1 to the ”low” classifications.

We did the sum and divided by the number of classifications.

generative model	weighted average
Leonardo Phoenix	0.321353
ChatGPT 4.0	0.312088
Ideogram	0.307536
Midjourney	0.278351
Stable Diffusion 3.5 large	0.197938
Flux 1.1 Pro	0.081264
Kolors by KlingAI	-0.178330
OmniGen	-0.257367
Firefly Image 3	-0.274841
FLUX.1 Schnell	-0.277014
Stable Diffusion 1.5	-0.457256
Auto-Aesthetics V1	-0.576659

Table 7.2: Weighted average by model.

As we can see from the table, the model with the best performance is Leonardo Phoenix, followed by ChatGPT 4.0 and Ideogram.

The model with the worst performance is instead Auto-Aesthetics V1, with a weighted average of around -0.58.

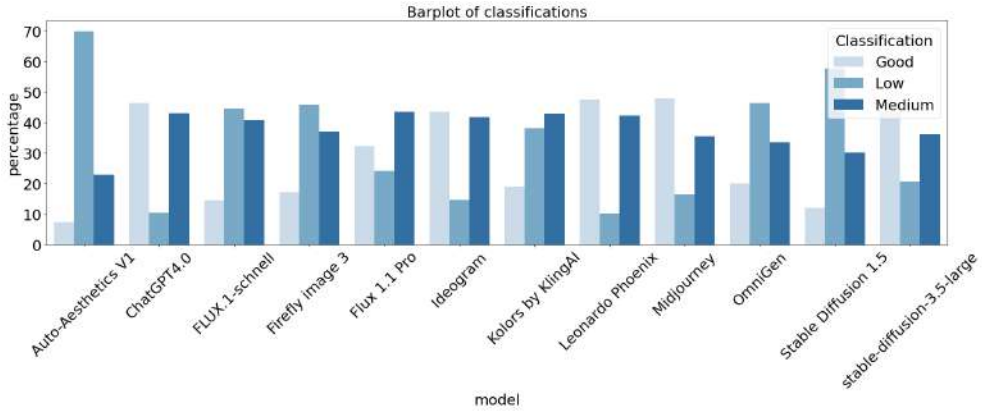

Figure 7.6: Classifications by model

In Figure 7.6 we can see the percentages of classifications by model.

Chapter 8

Conclusions

In this work, we conducted an analysis of the ability of modern text-to-image models to reproduce artistic movements from the 1500s to the first half of the 1900s.

In particular, we created a labeled dataset of images generated by 12 of the best models currently available.

The dataset we created could be used for future research in the field.

We also conducted two surveys to assess authenticity (the model's ability to generate an image that can be classified as human-made) and adherence to the prompt in the generated images.

As seen from the results, some models prioritize aesthetic quality over strict adherence to the prompts, while others sacrifice authenticity for greater accuracy. In particular, Ideogram is the model that performs best in terms of image quality, while Leonardo Phoenix follows the instructions in the prompt most accurately. The fact that ChatGPT performs so well in prompt adherence could be biased because of the fact that we used it to generate the prompts. It is noteworthy that Ideogram performs good also in prompt adherence. On the other hand, Auto-Aesthetics V1 performs the worst in both authenticity and adherence to the prompt.

However, although generative models have reached a high level of quality, there is still significant room for improvement. This is evident from the results

of the first survey: a generated image was classified as human-made less than 30% of the time.

Another conclusion we can draw from the results is that the models are more capable of reproducing artworks from more recent artistic movements than those from older styles. However, this result may be biased due to the dataset being unbalanced toward the more recent styles.

Bibliography

[1] Adobe. *Adobe Introduces Firefly Image 3 Foundation Model to Take Creative Exploration and Ideation to New Heights*. Accessed: 2025-03-11. 2023. URL: https://news.adobe.com/news/news-details/2024/adobe-introduces-firefly-image-3-foundation-model-to-take-creative-exploration-and-ideation-to-new-heights.

[2] N. L. AI. *Ai art revolution: Introducing auto-aesthetics for personalized gen AI experience*. Accessed: 2025-02-13. Aug. 2024. URL: https://neural.love/blog/auto-aesthetics-v1-ai-art-revolution.

[3] S. AI. *Introducing Stable Diffusion 3.5*. Accessed: 2025-03-12. 2024. URL: https://stability.ai/news/introducing-stable-diffusion-3-5.

[4] S. K. Alhabeeb and A. A. Al-Shargabi. "Text-to-Image Synthesis With Generative Models: Methods, Datasets, Performance Metrics, Challenges, and Future Direction". In: *IEEE Access* 12 (2024), pp. 24412–24427. DOI: 10.1109/ACCESS.2024.3365043.

[5] A. Asperti et al. *A Critical Assessment of Modern Generative Models' Ability to Replicate Artistic Styles*. 2025. arXiv: 2502.15856 [cs.CV]. URL: https://arxiv.org/abs/2502.15856.

[6] G. Bebis and M. Georgiopoulos. "Feed-forward neural networks". In: *Ieee Potentials* 13.4 (1994), pp. 27–31.

[7] A. Blattmann et al. “Align your latents: High-resolution video synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 22563–22575.

[8] M. Boubdir et al. “Elo Uncovered: Robustness and Best Practices in Language Model Evaluation”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 106135–106161. URL: `https://proceedings.neurips.cc/paper_files/paper/2024/file/bfba8efb806a970455b83b852c9cf846-Paper-Conference.pdf`.

[9] M. Brack et al. “LEDITS++: Limitless Image Editing using Text-to-Image Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2024, pp. 8861–8870.

[10] D. A. Chan and S. P. Sithungu. “Evaluating the Suitability of Inception Score and Fréchet Inception Distance as Metrics for Quality and Diversity in Image Generation”. In: *Proceedings of the 2024 7th International Conference on Computational Intelligence and Intelligent Systems*. 2024, pp. 79–85.

[11] D.-Y. Chen, H. Tennent, and C.-W. Hsu. “Artadapter: Text-to-image style transfer using multi-level style encoder and explicit adaptation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 8619–8628.

[12] K. Chen et al. “iCNN-Transformer: An improved CNN-Transformer with Channel-spatial Attention and Keyword Prediction for Automated Audio Captioning.” In: *INTERSPEECH*. 2022, pp. 4167–4171.

[13] S. Christophe et al. “Neural map style transfer exploration with GANs”. In: *International Journal of Cartography* 8.1 (2022), pp. 18–36.

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

[15] I. Goodfellow et al. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[16] A. Henry et al. “Query-Key Normalization for Transformers”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*. Ed. by T. Cohn, Y. He, and Y. Liu. Vol. EMNLP 2020. Findings of ACL. : Association for Computational Linguistics, 2020, pp. 4246–4253. DOI: 10.18653/V1/2020.FINDINGS-EMNLP.379.

[17] J. Ho, A. Jain, and P. Abbeel. “Denoising Diffusion Probabilistic Models”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[18] V. T. Hu et al. “Latent space editing in transformer-based flow matching”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 38. 3. 2024, pp. 2247–2255.

[19] R. Huang et al. “Make-An-Audio: Text-To-Audio Generation with Prompt-Enhanced Diffusion Models”. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by A. Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023, pp. 13916–13932. URL: https://proceedings.mlr.press/v202/huang23i.html.

[20] Ideogram. *Ideogram 2.0*. Accessed: 2025-03-11. 2024. URL: https://about.ideogram.ai/2.0.

[21] M. Inc. *Version 6.1*. Accessed: 2025-03-12. 2024. URL: https://updates.midjourney.com/version-6-1/.

[22] G. Jeanneret, L. Simon, and F. Jurie. “Diffusion Models for Counterfactual Explanations”. In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*. Dec. 2022, pp. 858–876.

[23] J. Johnson, A. Alahi, and L. Fei-Fei. “Perceptual losses for real-time style transfer and super-resolution”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer. 2016, pp. 694–711.

[24] D. P. Kingma and M. Welling. “An Introduction to Variational Autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. ISSN: 1935-8245. DOI: 10.1561/2200000056. URL: http://dx.doi.org/10.1561/2200000056.

[25] B. F. Labs. *Announcing Black Forest Labs*. Accessed: 2025-01-11. 2024. URL: https://blackforestlabs.ai/announcing-black-forest-labs/.

[26] B. F. Labs. *Announcing FLUX1.1 [pro] and the BFL API*. Accessed: 2025-01-12. 2024. URL: https://blackforestlabs.ai/announcing-flux-1-1-pro-and-the-bfl-api/.

[27] A. C. Li et al. “Your Diffusion Model is Secretly a Zero-Shot Classifier”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023, pp. 2206–2217.

[28] T. Lin et al. “A survey of transformers”. In: *AI open* 3 (2022), pp. 111–132.

[29] G. Lorberbom et al. “Direct Optimization through \arg \max for Discrete Variational Auto-Encoder”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/1a04f965818a8533f5613003c7db243d-Paper.pdf.

[30] R. Luo et al. “BioGPT: generative pre-trained transformer for biomedical text generation and mining”. In: *Briefings in bioinformatics* 23.6 (2022), bbac409.

[31] OpenAI. *Dall-E 3*. Accessed: 2025-03-11. 2023. URL: `https://openai.com/index/dall-e-3/`.

[32] OpenAI. *DALL·E 3 is now available in ChatGPT Plus and Enterprise*. Accessed: 2025-03-11. 2023. URL: `https://openai.com/index/dall-e-3-is-now-available-in-chatgpt-plus-and-enterprise/`.

[33] J. Oppenlaender. “The creativity of text-to-image generation”. In: *Proceedings of the 25th international academic mindtrek conference*. 2022, pp. 192–202.

[34] L. I. Pty. *Introducing Phoenix by Leonardo.Ai*. Accessed: 2025-03-12. 2024. URL: `https://leonardo.ai/phoenix/`.

[35] Z. Raisi et al. “Transformer-Based Text Detection in the Wild”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2021, pp. 3162–3171.

[36] B. Ramdurai and P. Adhithya. “The impact, advancements and applications of generative AI”. In: *International Journal of Computer Science and Engineering* 10.6 (2023), pp. 1–8.

[37] R. Rombach et al. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695.

[38] J. Rosendahl et al. “Analysis of Positional Encodings for Neural Machine Translation”. In: *Proceedings of the 16th International Conference on Spoken Language Translation*. Ed. by J. Niehues et al. Hong Kong: Association for Computational Linguistics, Nov. 2019. URL: `https://aclanthology.org/2019.iwslt-1.20/`.

[39] M. S. M. Sajjadi et al. “Assessing Generative Models via Precision and Recall”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/f7696a9b362ac5a51c3dc8f098b73923-Paper.pdf.

[40] A. Sauer et al. “Adversarial Diffusion Distillation”. In: *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXVI*. Accessed: 2025-02-13. 2024, pp. 87–103. URL: https://doi.org/10.1007/978-3-031-73016-0%5C_6.

[41] R. Sortino et al. “Transformer-based image generation from scene graphs”. In: *Computer Vision and Image Understanding* 233 (2023), p. 103721.

[42] K. Team. “Kolors: Effective Training of Diffusion Model for Photorealistic Text-to-Image Synthesis”. In: *arXiv preprint* (2024).

[43] A. Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[44] J. Wang et al. “Fine-grained image style transfer with visual transformers”. In: *Proceedings of the Asian conference on computer vision*. 2022, pp. 841–857.

[45] J. Wu. “Introduction to convolutional neural networks”. In: *National Key Lab for Novel Software Technology. Nanjing University. China* 5.23 (2017), p. 495.

[46] Y. Xu et al. “Style Transfer Review: Traditional Machine Learning to Deep Learning”. In: *Information* 16.2 (2025), p. 157.

[47] M.-C. Yeh et al. “Improving style transfer with calibrated metrics”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 3160–3168.

[48] K. Zaman et al. “Transformers and Audio Detection Tasks: An Overview”. In: *Digital Signal Processing* (2024), p. 104956.

[49] Z. Zhang et al. “Artbank: Artistic style transfer with pre-trained diffusion model and implicit style prompt bank”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 7. 2024, pp. 7396–7404.

[50] Z. Zhang et al. “Towards Highly Realistic Artistic Style Transfer via Stable Diffusion with Step-aware and Layer-aware Prompt”. In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*. Ed. by K. Larson. AI, Arts & Creativity. International Joint Conferences on Artificial Intelligence Organization, Aug. 2024, pp. 7814–7822. DOI: 10.24963/ijcai.2024/865. URL: https://doi.org/10.24963/ijcai.2024/865.

Acknowledgements

I would like to thank Prof. Asperti for his guidance throughout this research. I am also thankful to my colleagues for their constructive collaboration during our work.

Finally, I am deeply grateful to my family and friends for their unwavering emotional support throughout this journey.