Text-Image Correlation in Generative-AI: An In Silico Study of Their Adaptivity

Md Solaiman[1*], Md Aumit Hasan[1], and Afsana Begum[1]

[1] Department of Software Engineering, Daffodil International University, Dhaka-1216, Bangladesh
solaiman35-1107@diu.edu.bd*, aumit35-738@diu.edu.bd, afsana.swe@diu.edu.bd

Abstract. Generative artificial intelligence (GA) has the potential to revolutionize several industries, including the arts, entertainment, and content creation, by facilitating data synthesis and improving creativity through the use of techniques such as variational autoencoders (VAEs) and generative adversarial networks (GANs). The visual attractiveness of AI generated images and their relationship to the text prompts used to generate them are not entirely evident, though. We are here to demonstrate that, although no one has demonstrated this in any previous work, in practice, we use three two-stage neural-network pipelines: BLIP, GIT, and CLIP ResNet architectures. With a cosine-similarity scale ranging from -1 to 1, we obtained 0.45 similarities from the CLIP architecture, 0.46 from the BLIP architecture, and 0.36 from the microsoft GIT. In that regard, the findings suggest that while generative AI (GA) demonstrates an impressive correlation between image-textual signals, it is unable to mimic the contextual knowledge and nuanced creativity that are fundamental to humans. And for the upcoming research in this field, we will also make available a combine dataset of three generative AI (GA) models images—Stable Diffusion, DALL-E 3, and Midjourney—along with their quality ratings and aesthetics assessed by OpenAI ImageGPT-small, microsoft Swin-Transformer, and Google ViT.

Keywords: Generative AI (GA), CLIP, BLIP, microsoft GIT, Stable Diffusion, Midjourney, and DALL-E 3

1 Introduction

The field of generative AI (GA) [1][2] focuses on developing models [3][4][5][6] and systems that can produce new content, such as text-dependent graphics, audio, and video with only simple inputs. Generative AI (GA) models generate new and realistic outputs that closely resemble the original data distribution by learning patterns and structures within the data through training on huge datasets. By enabling data synthesis and enhancing creativity through methods like variational autoencoders (VAEs)[1] and generative adversarial networks (GANs), generative AI (GA) has the potential to completely transform the human creativity fields. Computer scientist Ian Goodfellow[2] came up with the idea for GANs one night in a bar in 2014. Where two models are placed against one another, the first network, known as a generator[7], attempts to produce realistic visuals and the second one is the discriminator[2]; that is a different type of model, and it's given both created and real images, and its initial task is to determine which of them are real.

In order to generate fresh, comparable data, a generative deep learning model called a variational autoencoder (VAEs)[1] compresses input into a continuous, probabilistic latent space and then learns to decode it. In contrast, a VAE maps input data to a probability distribution in a hidden space (the "latent space") rather than a single, fixed point, as is the case with a standard autoencoder[8]. The generative[7] ability of VAEs is derived from this probabilistic methodology. In what way does a VAE produce fresh data? The encoder is no longer required for data creation after training: A sample of a random vector is taken from a normal distribution. A new data sample that closely resembles the initial training text and graphics is then produced by the trained decoder network after receiving this random vector.

Stable Diffusion[3] generates images from text prompts by progressively transforming random noise into coherent visuals through an iterative process. Text prompts are iteratively denoising a latent representation that is conditioned on the prompt's semantic features, which is called the text-conditioned latent diffusion process. The model uses a neural network (typically a U-Net) that's been trained to understand how different words and phrases relate to visual features. So if your prompt says "a cat wearing sunglasses", the model nudges the noise toward patterns that resemble that concept. DALL-E 3 [4] transforms detailed textual descriptions into high-fidelity images by encoding the prompt into a latent representation and then using a transformer network to synthesis and iteratively refine the visual output. Leveraging a transformer network trained on massive amounts of image–text pairs, DALL-E

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Advances in Intelligent Systems Research 206,
https://doi.org/10.2991/978-94-6239-664-7_75

3[4] associates the textual information with corresponding visual features. As it processes the encoded prompt, the model generates an image representation that progressively aligns with the described scene. Enhanced training on diverse data allows it to capture subtle and sophisticated details. And finally Midjourney[5][6] transforms users' textual prompts into artistic images by employing a multi-step, iterative process. Operating primarily via a discord-based[9] interface, users submit prompts using commands (typically/imagine), which the model then interprets by mapping the descriptive text into a latent space. From there, an evolving algorithm drawing on techniques reminiscent of generative adversarial networks (GANs) and refined through human feedback iteratively transforms random initial patterns into coherent, high-quality images.

In this paper, we are going to showcase the following:

1. We will first use three image rating models to display the image aesthetics and quality ratings on a (0–10) scale of three well-known generative AI (GA) models Stable Diffusion, Midjourney, and DALL-E 3 generated images via the ViT ("google/vit-base-patch16-224") [10], the Swin-Transformer ("microsoft/swin-base-patch4-window7-224") [11], and the well-known image reward model based on ImageGPT ("openai/imagegpt-small") [12].

2. Subsequently, we will present a combine dataset of three generative AI (GA) models image-text pairs along with their corresponding aesthetics and quality ratings of DALL-E 3 (ProGamerGov/synthetic-dataset-1m-dalle3-high-quality-captions) [13], first N-sample 10435 images from Synthetic Dataset 1M, Midjourney (ava-space/MidJourney) [14], all the 3039 images, and Stable Diffusion (HighCWu/diffusiondb-2m-first-5k-canny) [15], 5000 images created using the canny algorithm from DiffusionDB 2M.

3. Lastly, we attempt to demonstrate whether or not AI generated images are visually appealing and how they correspond with the word prompts that were used to create the images. ResNet50x4 and ResNet101, two variations of the CLIP[16] Resnet architecture, will be use to assess similarity using the cosine similarity metric. Additionally, we will be employ two more vision language models: Generative Image-to-text Transformer[17] ("microsoft/git-base") and Bootstrapping Language-Image Pre-training[18] ("Salesforce/blip-image-captioning-base") to cross check the similarity scores validity.

2 Related Work

Yan and Mikolajczyk (2015) [19] used Deep Canonical Correlation Analysis (DCCA) to address image-text matching. Deep RNN and LSTM models (Vinyals et al., 2014) [20] improved on earlier approaches that concentrated on caption synthesis using object detection and language models (Farhadi et al., 2010) [21]. Researchers turned to cross-modal retrieval to get over evaluation issues. Methods such as Kernel CCA (Hodosh et al., 2013) [22], transfer CCA (Gong et al., 2014) [23], and fragment embeddings (Karpathy et al., 2014) [24] made headway but were hindered by their great complexity. In order to achieve state-of-the-art performance on common benchmarks, Yan and Mikolajczyk [19] expanded DCCA, which was initially proposed by Andrew et al. (2013) [25], with GPU acceleration and overfitting control.

SeLIP, a similarity-enhanced contrastive language-image pretraining framework for multi-modal head MRI, was suggested by Liu et al. [26] in 2025. While modifications like ALIGN (Jia et al., 2021) [27], BASIC (Pham et al., 2021) [28], and LiT (Zhai et al., 2022) [29] enhanced scalability and efficiency, earlier research like CLIP [16] (Radford et al., 2021) demonstrated the effectiveness of contrastive learning on large-scale image-text pairs. Techniques including MedCLIP (Wang et al., 2022) [30], GLoRIA (Huang et al., 2021) [31], and ConVIRT (Zhang et al., 2020) [32] applied contrastive learning to radiology reports and pictures in the medical field, but they had trouble with the small datasets and semantic overlap in clinical texts. Recent attempts to handle multi-modal MRI, such as UniBrain (Zhou et al., 2023) [33], were hindered by weak cross-modal alignment and missing modalities. Liu et al. [26] developed a combined syntax–semantic similarity loss to address these problems; it softens targets and more accurately depicts the relationships between radiological reports. Their method greatly enhanced segmentation, classification, and image-text retrieval, emphasizing the value of semantic similarity in medical image pretraining.

The goal of Zaidi et al.'s (2019) [34] study on text-based picture retrieval was to increase the precision of selecting pertinent photos from search engine results. Information retrieval (IR) and information filtering (IF) techniques were first developed in earlier image retrieval research. Latent factor models and

collaborative filtering are two methods that have been used across domains (Hanani et al., 2001 [35]; Zhang et al., 2011 [36]). Two broad approaches have been used in image retrieval research: Content-Based Image Retrieval (CBIR), which makes use of visual aspects including color, texture, and shape (Duan et al., 2011) [37], and Text-Based Image Retrieval (TBIR), which depends on keywords and metadata (Rui et al., 1999) [38]. However, noisy or unclear queries frequently cause TBIR to produce irrelevant responses, whereas CBIR struggles with the semantic gap issue. Zaidi et al. [34] examined Cosine Similarity and Sequence Matcher methods for filtering Bing API image results in order to overcome these difficulties. According to their research, Cosine Similarity outperformed Sequence Matcher in terms of precision and was more successful in locating pertinent photos.

3 Methodology

3.1 Data Collections and Experimental Setup

To create our evaluation benchmark, we used three publicly accessible image-text datasets from Hugging Face: *ava-space/MidJourney*, *HighCWu/diffusiondb-2m-first-5k-canny*, and *ProGamerGov/synthetic-dataset-1m-dalle3-high-quality-captions*. We downloaded every image-caption pair that was available for each dataset exactly as supplied which we shown in "Table 1". Approximately one million synthetic photos with high-quality DALL-E 3 captions make up the ProGamerGov dataset; however, evaluating the entire collection would be computationally expensive. We thus chose the first 10,435 samples (about 1.00 % of the dataset) acquired via WebDataset streaming, in accordance with usual procedure in large-scale vision–language evaluation. Due to the high computing cost of metric computation, this sampling approach (e.g., Li et al. [39], Amazon Science 2024; Lv et al. [40], DiffuseHigh, ICCV 2025) is frequently used in contemporary VLM and diffusion-model research, where assessments are usually carried out on 10k randomly distributed samples.

Table 1. The percentage of samples used in our study, including dataset statistics.

Dataset (HuggingFace)	Total Samples	Used Samples	Used %	Sampling Notes
Synthetic Dataset 1M (ProGamerGov/synthetic-dataset-1m)	1,000,000	10,435	≈1.00%	Streaming WebDataset, first N samples.
DiffusionDB 2M (HighCWu/diffusiondb_2m_first_5k_canny)	5,000	5,000	100%	Full dataset.
MidJourney (ava-space/MidJourney)	3,039	3,039	100%	Full dataset.

Data from "Table 1" shown we utilized all 5,000 photos from the HighCWu/diffusiondb-2m-first-5k-canny subset for the Stable Diffusion data. Only the first 5,000 prompts that were run through the Canny edge detector are included in this subset, which is taken from the original DiffusionDB 2M dataset. Only the particular 5k canny-processed subset supplied by HighCWu is used in our studies; the entire 2-million picture DiffusionDB corpus is not used. And lastly we utilized all 3,039 photos from the ava-space/MidJourney dataset, which includes user-generated prompts, for the MidJourney data. When combined, these datasets provide a variety of pictures produced by DALL-E 3, Stable Diffusion, and Midjourney, allowing for a thorough cross-model comparison under carefully monitored assessment settings.

Following data extraction, we arranged and saved every image in three distinct Google Drive directories, one for each dataset. Three CSV files containing the corresponding text captions were also stored in Google Drive. We utilized Google Colaboratory, a cloud-based Jupyter notebook environment that is popular for machine learning experiments because of its easy interaction with Google Drive and readily available GPU/TPU resources.

3.2 Image Quality Quantification (scale: 0-10)

We build a function that does inference to obtain the logits after processing each image through the pipeline of the appropriate model, transforming the raw image into a preprocessed tensor. In the "Fig. 1" image rating analysis workflow shown how our model calculate the raw rating of each image by

averaging the logits. Following image processing, we normalize the calculated scores to a standard 0–10 scale. This normalization step is essential to comparing ratings from various models and datasets.

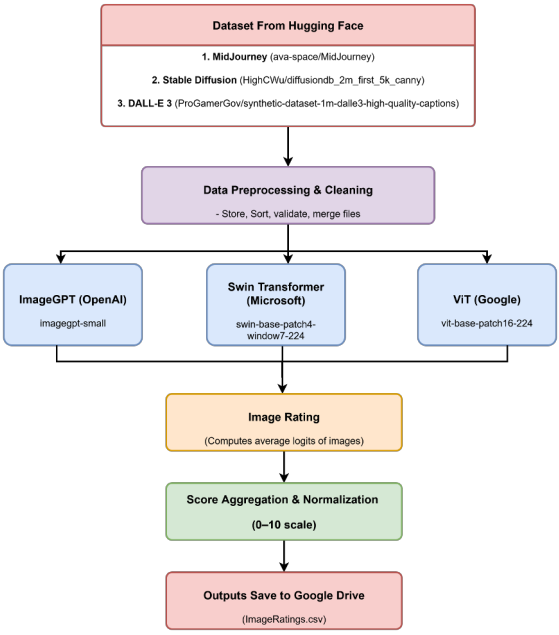

Fig. 1. Image rating analysis workflow.

In order to measure aesthetics and quality, we begin by configuring three different picture categorization algorithms. The image reward model, which is based on ImageGPT ("openai/imagegpt-small"), is loaded first. It uses its internal representations to give each image a score. In order to collect fine-grained features, we then initialize the Swin Transformer("microsoft/swin-base-patch4-window7-224"), utilising its strong hierarchical attention mechanisms. Lastly, we use a patch-based version of the ViT("google/vit-base-patch16-224") to access global picture contexts.

ImageGPT("openai/imagegpt-small"): For image-based scoring, we employ the ImageGPT model. Originally trained for image generation, ImageGPT is a transformer model. It is also used by researchers for picture classification and quality evaluation. A distribution across distinct visual tokens is predicted by ImageGPTForImageClassification. We take its logits output and extract a single quality score.

1. Preprocessing: An input image I is preprocessed using an operator P that resizes and normalizes the image, resulting in $X = P(I)$.
2. Model Inference: The preprocessed image X is fed into the ImageGPT classification model M, producing a logit vector $\ell = M(X) \in R^C$ where C is the number of output logits.
3. Raw Score Computation: A raw score s is calculated by averaging these logits,

$$s = \frac{1}{C} \sum_{i=1}^{C} \ell_i.$$

4. Normalization: With a set of raw scores $\{s_j\}$, determine the minimum and maximum scores: ($s_{min} = \min s_j$ and $s_{max} = \max s_j$). And normalize the score to the range [0-10]

$$r = \left(\frac{S - s_{min}}{s_{max} - s_{min}} \right) \times 10,$$

where r is the final image rating.

Swin Transformer ("microsoft/swin-base-patch4-window7-224"): Swin Transformer is the model that we employ ("microsoft/swin-base-patch4-window7-224"). A multi-scale visual representation is constructed by this paradigm. The input is initially divided into fixed-size patches. It performs self-attention inside each window after grouping patches into non-overlapping windows. After that, it repeats self-attention and moves the window grid by half a window size. Pixels can interact across adjacent windows thanks to this straightforward shift. In order to increase channel depth and decrease spatial size, it merges neighbouring patches in between steps.

1. Mathematically, let I be an input image and P the feature extractor (resize + normalize). We compute
$$X = P(I).$$
2. We feed X into the Swin Transformer function M and get a logits vector
$$\ell = M(X) \in R^C,$$
3. where C is the number of classes. We collapse these logits into a raw score by
$$s = \frac{1}{C}\sum_{i=1}^{C} \ell_i.$$
4. Over all images, let $\{s_j\}$ be the set of raw scores. Define
$$s_{\min} = \min_j s_j \quad and \quad s_{\max} = \max_j s_j.$$
5. We normalize each score to the $[0, 10]$ range
$$r = \frac{s - s_{\min}}{s_{\max} - s_{\min}} \times 10,$$
Where r is the final image rating.

ViT ("google/vit-base-patch16-224"): Each image is divided into a grid of fixed-size patches by Vision Transformer (ViT). Each patch is flattened and linearly projected into an embedding in d dimensions. To preserve spatial information, it incorporates a learnable positional embedding. A unique class token is appended to the patch embeddings. The entire sequence is fed by feed-forward blocks and L layers of multi-head self-attention. The output of the class token is extracted after the last layer, and a linear head is applied to create a logits vector of length C.

1. Input Image & Patch Embedding: An input image I is transformed into a sequence of patches and embedded with positional information using an operator E. The output, $x_0 = E(I)$, includes a class token.
2. Transformer Encoding: The sequence x_0 is processed through a transformer encoder T to produce $x_L = T(x_0)$.
3. Logits Calculation: From x_L, the class token embedding z_{cls} is extracted. The logits vector $\ell = W z_{cls}$ is computed.
4. Score Calculation: The vector ℓ collapses to get a raw score s by averaging across all C logits.
5. Normalization: A set of raw scores $\{s_j\}$ is used to find the minimum $s_{\min}$ and maximum $s_{\max}$. Each score s is linearly mapped to the range [0, 10] to get the final score r.

3.3 Text-Image Similarity Computation

CLIP (Contrastive Language-Image Pre-training): OpenAI's CLIP model is a multi-modal language and vision model that associates text and images with the same latent space. Here, "Fig. 2" architecture of CLIP examined the similarity between AI-generated images and their human-written captions using both image and text queries.

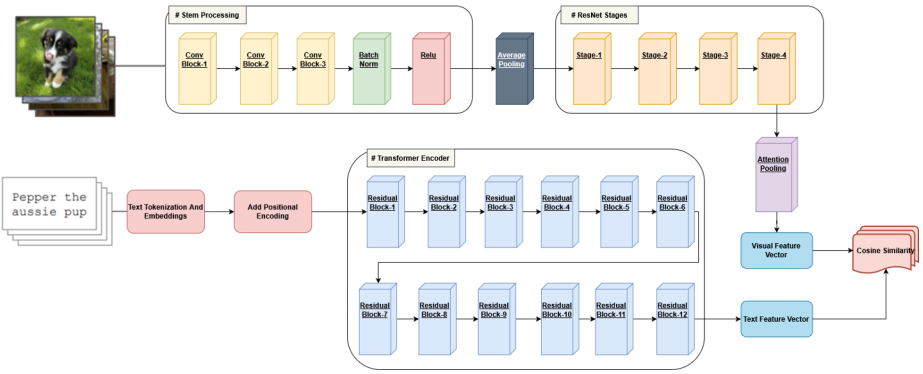

Fig. 2. Architecture of CLIP (Contrastive Language-Image Pre-training)

Image Encoder (Modified ResNet Architecture): An embedding is a numerical vector that can be used to represent an image's semantic content. Effective similarity computing is made possible by comparing these low-dimensional embedding vectors instead of the original images.

1. Initial Convolutions: This network begins with a convolutional layer combined with batch normalization and a nonlinear activation function (typically ReLU). This stage extracts low-level features, such as edges and textures, and a series of residual blocks. These blocks are used to skip connections that allow for the effective training of very deep networks.
2. Pooling Layer: This layer reduces the spatial dimensions of the feature map by helping to retain essential features while reducing computational complexity.
3. Residual Stages (Stages 1–4): The number of bottleneck blocks in each stage of a residual network
   (a) Stage 1: Three (3) bottleneck blocks are used to extract low-level feature representations.
   (b) Stage 2: Four (4) bottleneck blocks are applied, with the first block using a stride of 2 for down-sampling.
   (c) Stage 3: Twenty-three (23) bottleneck blocks are used, with the initial block down-sampling with a stride of 2.
   (d) Stage 4: Three (3) bottleneck blocks are used, beginning with a down-sampling operation through a stride of 2.

   Each bottleneck block uses residual connections to aid in learning deep representations without the issue of vanishing gradients.
4. Attention Pooling: Instead of using global average pooling at the end of the ResNet architecture, CLIP uses this mechanism to aggregate spatial features into a single embedding by learning attention weights over spatial locations.

$$v = \sum_{i \in S} \alpha_i F_i$$

   Where F_i the feature vector is at a spatial location i in the last convolutional feature map, α_i the learnt attention weight for that location is the condition that $\sum_{i \in S} \alpha_i = 1$.
5. Linear Projection: The aggregated features are then passed through a linear projection layer, mapping them into a joint embedding space that is shared with the text encoder.
6. Output Image Embedding: The final stage outputs a compact image embedding that is used for cross-modal comparisons with textual embeddings.

Text Encoder (Transformer-Based Architecture):

1. Tokenization and Embedding: Due to the limitation of the context length of the model (77 tokens), longer texts are segmented into chunks. The input text is first tokenised into a sequence of tokens. Each token is embedded in a continuous vector space. The final score, as $S(I, T)$, is the average of the similarity scores across all fragments, calculated using the formula:

$$S(I,T) = \frac{1}{n}\sum_{i=1}^{n} S(I,T_i)$$

Here, it T represents the set of fragments $\{T_1, T_2, \ldots, T_n\}$.

2. Transformer Layers: The text encoder uses a multilayer transformer architecture (with self-attention and feedforward networks) to capture contextual relationships between tokens. Positional embeddings are added to maintain sequential order to form a single text embedding as T.

Crosine Similarity: The embeddings are computed from both modalities, using a cosine similarity metric to calculate similarity.

The similarity $S(I,T)$ between two embeddings I and T is given by the cosine similarity formula:

$$S(I,T) = \frac{I \cdot T}{\|I\|\|T\|}$$

1. Where I is the visual embedding of the modified ResNet.
2. T is the text embedding from the transformer.
3. And $I \cdot T$ measures the alignment of the vectors.

The norms $\|I\|$ and $\|T\|$ normalise the result to ensure a similarity score in the range $[-1, 1]$. Where the similarity score $S = 1$ indicates perfect semantic alignment, while $S = -1$ indicates opposite meaning.

BLIP (Bootstrapping Language-Image Pre-training): This model turns both the generated caption and a ground-truth caption into fixed-length vectors in the same semantic space. And it computes a cosine similarity between those vectors to tell us how closely the machine-written caption matches the human-written one. By chaining these steps, we are using a vision-language backbone which we thoroughly explain in the "Fig. 3" architecture of BLIP.

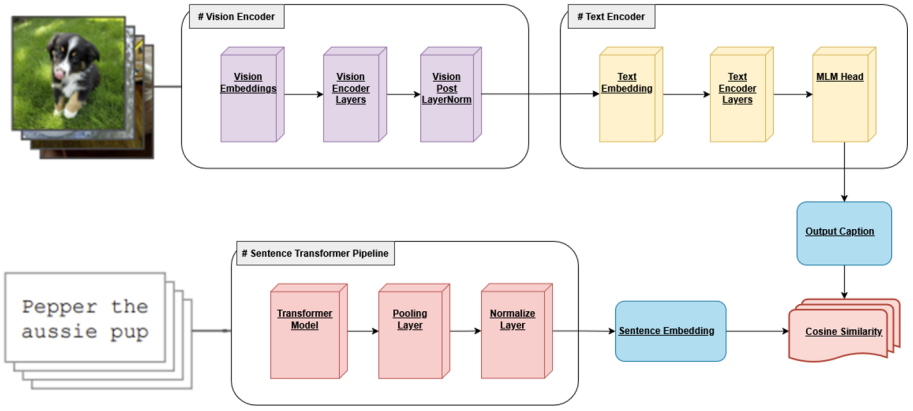

Fig. 3. Architecture of BLIP (Bootstrapping Language-Image Pre-training)

Caption Generation Stream: We transform an input image $x \in R^{3\times H\times W}$ into a textual caption in three main stages.

1. Patch Embedding: Divide the image into non-overlapping 16×16 patches and project each patch into a d-dimensional vector ($d = 768$):

$$E_p = \text{Conv2d}(x; W_p, b_p) \in R^{\frac{H}{16}\times\frac{W}{16}\times d}.$$

2. Transformer Encoder: Pass the patch embeddings E_p through $L = 12$ identical transformer layers. In layer ℓ, let the input be $X^{(\ell)}$. Here, we compute:

$$Q = X^{(\ell)}W_q, K = X^{(\ell)}W_k, V = X^{(\ell)}W_v, A = \text{softmax}\Big(QK^\top\sqrt{d}\Big)\,V,$$

$$\widehat{X} = \text{LayerNorm}\big(X^{(\ell)} + A\big),$$

$$X^{(\ell+1)} = \text{LayerNorm}\Big(\widehat{X} + W_2\,\text{GELU}(W_1\,\widehat{X} + b_1) + b_2\Big)$$

After $L = 12$ layers and a final LayerNorm, we obtain

$$E_v = X^{(L)} \;\in\; R^{\frac{H}{16}\times\frac{W}{16}\times d}$$

3. Auto-regressive Decoder: We generate a caption $y = (y_1, \ldots, y_T)$ token by token. At step t:

$$e_t = \text{WordEmb}(y_t) + \text{PosEmb}(t), h_t = \text{DecoderLayer}(e_{<t},\, E_v),$$

$$\ell_t = W_{\text{dec}}\,h_t + b_{\text{dec}}, p(y_t \mid y_{<t}, E_v) = \text{softmax}(\ell_t)$$

And finally we train by minimizing the cross-entropy loss.

$$\mathcal{L}_{\text{CE}} = -\sum_{t=1}^{T} \log p\big(y_t \mid y_{<t}, E_v\big)$$

Sentence Transformer Pipeline: We denote the generated caption by $\hat{y}$ and the ground-truth caption by y. We embed each token sequence $s \in \{\hat{y},\, y\}$ as follows:

1. Transformer Encoding: We feed s into a BERT-based encoder and collect its hidden states

$$\{h_i\}_{i=1}^{L} = \text{BERTEncoder}(s), \quad h_i \in R^{d'},\; d' = 768$$

2. Mean-Token Pooling: We compute the mean of all token embeddings to obtain a single vector

$$u = \frac{1}{L}\sum_{i=1}^{L} h_i$$

3. Projection and Normalisation: We project u down to $d'' = 384$ dimensions and then ℓ_2– normalise:

$$\tilde{u} = W_{\text{proj}}\,u + b_{\text{proj}}, \bar{u} = \frac{\tilde{u}}{\tilde{u}_2}$$

Cosine Similarity: We are evaluating semantic alignment between two normalised embeddings **u** (from $\hat{y}$) and **v** (from y).

$$\cos(u, v) = \frac{u^\top v}{\|u\|_2\|v\|_2} = u^\top v \in [-1, 1].$$

Between the image embedding (u) and the text embedding (v), cosine similarity ranges from –1 to 1, where 1 is a perfect match, while –1 indicates no match.

GIT (Generative Image-to-text Transformer): This study evaluates the semantic alignment between machine-generated captions and human-authored captions via a two-stage neural pipeline: (i) image-to-text generation using microsoft GIT and (ii) text-to-text similarity scoring using Sentence Transformer. We shown in the "Fig. 4" architecture of GIT, how well the GIT-generated captions match human-written captions by passing each through a shared text encoder and computing cosine similarity between their embeddings.

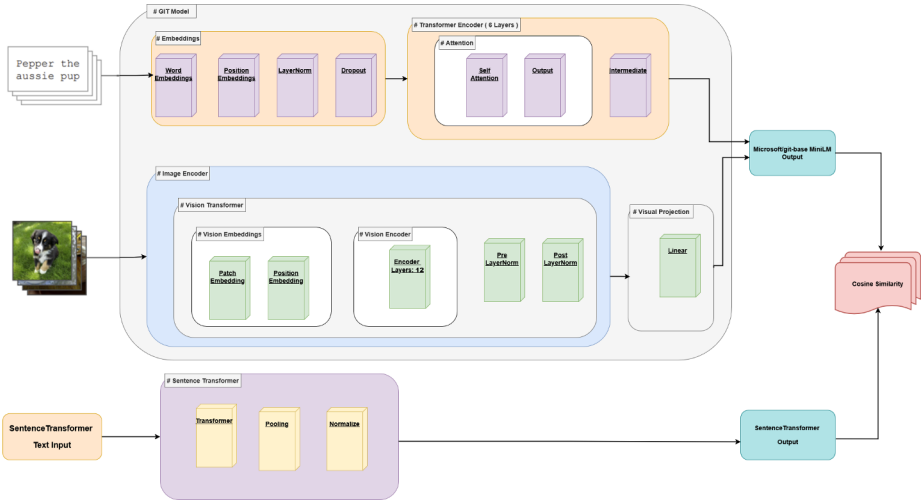

Fig. 4. Architecture of GIT (Generative Image-to-text Transformer)

Caption Generation: We use caption generation to train the microsoft GIT model for image captioning. For each training we,

1. First, take an input colour image, which we denote by I. Then provide a human-written ground-truth caption, which we denote by T.
2. We then apply our GIT model g to the image and obtain a generated caption

$$\hat{T} \;=\; g(I)\,.$$

3. During training, we optimise the model by maximising the conditional log-likelihood of the ground-truth caption given the image:

$$\log P(T \mid I) = \sum_{t=1}^{L} \log P\big(T_t \mid I,\; T_{<t}\big)\,,$$

Where L is the number of tokens in the caption and $T_{<t} = (T_1, \ldots, T_{t-1})$.

Text Embedding: We encode both the generated caption $\hat{T}$ and the reference caption T using a pretrained SentenceTransformer model E.

1. In our experiments, we perform:

$$v_{gen} = E(\hat{T}), \quad v_{ref} = E(T),$$

where $v_{gen}, v_{ref} \in R^d$ with $d = 384$.

2. We then normalize each embedding to unit length:

$$\bar{v}_{gen} = \frac{v_{gen}}{v_{gen2}}, \quad \bar{v}_{ref} = \frac{v_{ref}}{v_{ref2}}.$$

Cosine Similarity: We compute the cosine similarity between the normalized embeddings $\bar{v}_{\text{gen}}$ and $\bar{v}_{\text{ref}}$:

$$s = \cos\big(\bar{v}_{\text{gen}}, \bar{v}_{\text{ref}}\big) = \frac{\bar{v}_{\text{gen}}^{\top} \bar{v}_{\text{ref}}}{\|\bar{v}_{\text{gen}}\|_2 \, \|\bar{v}_{\text{ref}}\|_2} \in [-1, 1].$$

We interpret it $s = 1$ as perfect semantic alignment and $s = -1$ as complete semantic opposition.

4 Results and Explanations

We perform a thorough analysis of the model's performance in this section. We examined the performance of various models in our extensive evaluation of image quality ratings "Fig. 1" and image-text cosine similarity "Figs. 2,3 and 4" using three main datasets: DALL-E 3(ProGamerGov/synthetic-dataset-1m-dalle3-high-quality-captions) 10435 images, where we took the first N-sample (1% of the dataset) card from Synthetic Dataset 1M, we took Midjourney(ava-space/MidJourney), all the 3039 image-text pairs, and Stable Diffusion (HighCWu/diffusiondb-2m-first-5k-canny), 5000 image-text pairs created using the canny algorithm from DiffusionDB 2M, as shown in our dataset table "Table 1".

4.1 Image Ratings Analysis

Table 2. Perceptual image quality scores of three transformer models (ImageGPT, Swin Transformer, and Vision Transformer) and their overall mean.

Datasets (Hugging-Face)	No. of Im-ages	Image GPT	Swin Trans-former	Vision Trans-former	Dataset Avg.
MidJourney (ava-space/MidJourney)	3039	5.07	6.10	4.18	∼5.11
Stable Diffusion (HighCWu/. . . canny)	5000	4.19	5.99	4.57	∼4.91
DALL-E3 (ProGamer-Gov/. . . high-quality)	10435	5.62	5.34	4.95	∼5.30
	Model Avg. =	∼4.96	∼5.81	∼4.57	

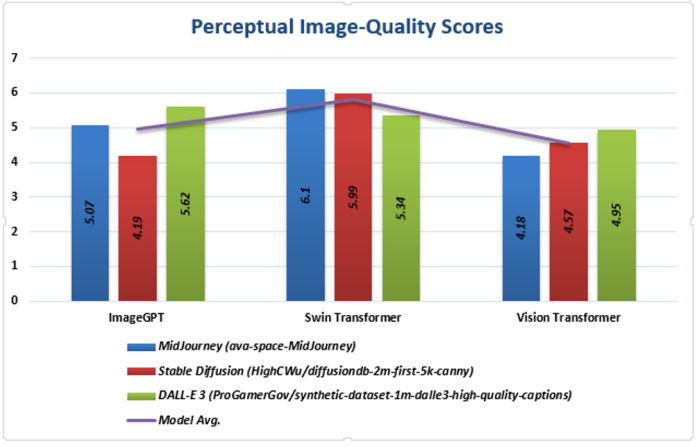

Fig. 5. Perceptual image quality scores comparisons of ImageGPT, Swin Transformer, and Vision Transformer models.

Our initial goal was to gauge how high-quality the photos produced by generative AI (GA) were. In order to measure the image quality ratings on a (0–10) scale, we used three transformer models. First, we used the popular image reward quality ratings model, which is based on ImageGPT(openai/imagegpt-small"). Next, we used a Swin-Transformer (microsoft/swin-base-patch4-window7-224"), and lastly, we used a Vision-Transformer (google/vit-base-patch16-224") to assess the image quality and aesthetics of

the AI generated (MidJourney, Stable Diffusion, and DALL-E 3) Datasets. Following a comprehensive analysis, we obtained a (0–10) scale for image aesthetics and quality evaluations of 5000 Stable Diffusion photos, 3039 Midjourney photos, and 10435 DALL-E 3 photos. We can see from the figure "Fig. 5" ImageGPT and ViT models appear to perform worse than Swin Transformer. For the corresponding (MidJourney, Stable Diffusion, and DALL-E 3) datasets, we can see from our perceptual image quality scores table "Table 2" ImageGPT received aesthetics and quality scores of 5.07, 4.19, and 5.62, while Swin Transformer received scores of 6.10, 5.99, and 5.34, which are similarly higher than Vision Transformer (ViT) model values of 4.18, 4.57, and 4.95.

4.2 Cosine-Similarity Explanations

Table 3. Cosine similarity between image and text as determined by the three distinct models CLIP, GIT, and BLIP.

Datasets (HuggingFace)	No. of image caption pairs	CLIP (RN50x4)	CLIP (RN101)	BLIP	microsoft GIT
MidJourney (ava-space/MidJourney)	3039	0.37	0.44	0.45	0.36
Stable Diffusion (HighCWu/...canny)	5000	0.38	0.45	0.33	0.31
DALL-E3 (ProGamerGov/...high-quality)	10435	0.37	0.44	0.46	0.36

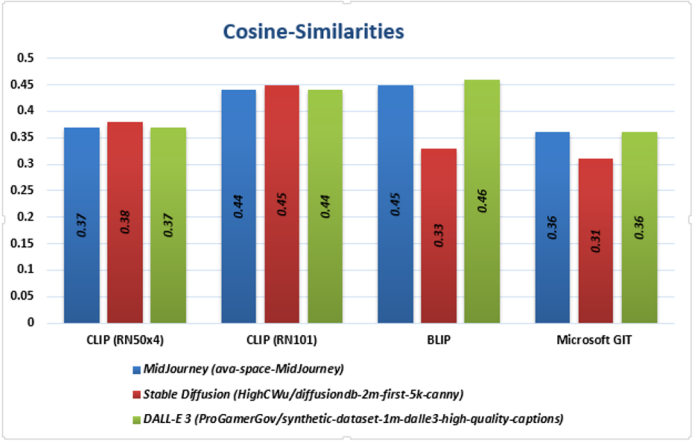

Fig. 6. Cosine similarity comparison between image and their associated text captions using CLIP, BLIP, & GIT.

Regarding the measurement of image-text similarity, we are using two different types of models. One is a semantic alignment model between machine-generated and human-authored captions using a two-stage neural network, and the other is a joint image and text embedding model called CLIP (Fig. 2), which stands for Contrastive Language-Image Pre-training. We used the CLIP ResNet architecture's two variants, ResNet50x4 and ResNet101, to evaluate CLIP's efficacy. Additionally, we used two unified vision-language models, namely GIT (Generative Image-to-text Transformer) and BLIP (Bootstrapping Language-Image Pre-training, respectively (Fig. 4 and 3) to ensure semantic alignment between the generated caption and a ground-truth caption. To obtain non-exaggerated ground truth results from our multi-modal analysis, we employed the cosine-similarity metric. We calculate the similarity scores after

a thorough study, which are essentially a measure of the cosine similarity between the text embedding (from the linguistic branch) and the image embedding (from the visual branch). The cosine similarity scale goes from -1 to 1, where -1 denotes no match and 1 denotes a perfect fit.

Our models' performance in the generative AI (GA) datasets was quantifiable and the cosine similarity between image and text results are summarized in "Table 3". We can see data from the "Table 3", CLIP ResNet101 architecture outperforms CLIP ResNet50x4. For Stable Diffusion image-text pairs, we found 0.45 similarities, while ResNet50x4 acquired 0.38. For the remaining two datasets, DALL-E 3 and MidJourney, it yields 0.37 similarities, whereas the ResNet101 variant yielded 0.44 cosine-similarities. On the BLIP architecture, we outperform the GIT among the vision-language models as clearly shown in "Fig. 6". For the DALL-E 3 dataset, we found 0.46 similarities, and for the MidJourney dataset, we found 0.45 similarities, while GIT found 0.36 similarities for both datasets. Additionally, for the Stable Diffusion dataset, both topologies yield slightly lower results, with 0.33 and.30 image-text pair similarities, respectively. Among all the models, the CLIP ResNet101 variation on the necked eye in "Fig. 6" offers high and stable cosine-similarities across all datasets. Unless a baseline is established, there is no definite "good" or "bad" threshold. Random image-caption couples, for instance, may receive scores that are close to zero or even negative. There a score range of plus 0.30 to 0.50 in that situation indicates a quantifiable measure.

4.3 Proposed Dataset & Future Work

For next studies in that area, we want to make our image aesthetics and quality score evaluation available as a multi-modal dataset. Combination with the three image folders (MidJourney, Stable Diffusion, and DALL-E 3) on Kaggle, we will post two CSV files. Where the Filename, File Directory, imageGPTRatings, SwinTransformerRatings, and ViTRatings are contained in one CSV file, while the filename, file directory, and original captions are contained in another CSV file. To connect data across the CSV files, anyone can use Filename as the primary key and File Directory as the sub-key. Dataset download link here; https://www.kaggle.com/datasets/solaiman1107/synthetic-dataset-of-generated-ai-text-to-images

5 Conclusion

Critical questions concerning the future of graphic design and the changing nature of creative work are being raised by the generative AI (GA) revolution, which is drastically changing the creative industries. This study examines whether traditional graphic design occupations are in danger from sophisticated generative models that can create graphics from textual descriptions. Our research is centered on two analyses: first, we use Cosine metrics (like semantic consistency and perceptual quality) to measure how well current state-of-the-art generative AI (GA) systems convert captions into visually coherent images, and second, we investigate the wider effects of these technological developments on the design workforce. Our results indicate that although generative AI (GA) shows remarkable capacity to match textual signals, it is unable to replicate the contextual awareness and subtle ingenuity that are essential to human designers.

References

1. Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.
2. Goodfellow, I.J., 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, 27.
3. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10684-10695).
4. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y. and Manassra, W., 2023. Improving image generation with better captions. Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf, 2(3), p.8.
5. Don-Yehiya, S., Choshen, L. and Abend, O., 2023. Human learning by model feedback: The dynamics of iterative prompting with midjourney. arXiv preprint arXiv:2311.12131.
6. Simonen, H., Kiviniemi, A. and Oppenlaender, J., 2025. An Initial Exploration of Default Images in Text-to-Image Generation. arXiv preprint arXiv:2505.09166.
7. Hinton, G.E., Dayan, P., Frey, B.J. and Neal, R.M., 1995. The" wake-sleep" algorithm for unsupervised neural networks. Science, 268(5214), pp.1158-1161.

8. Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986. Learning representations by back-propagating errors. nature, 323(6088), pp.533-536.
9. Midjourney, "Discord quick start," https://docs.midjourney.com/hc/en-us/articles/32631709682573-Discord-Quick-Start
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
11. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 10012-10022).
12. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J. and Dong, Y., 2023. Imagereward: Learning and evaluating human preferences for text-to-image generation. Advances in Neural Information Processing Systems, 36, pp.15903-15935.
13. ProGamerGov, "synthetic-dataset-1m-dalle3-high-quality-captions," Hugging Face Datasets, 2025, accessed: 2025-10-06. [Online]. Available: https://huggingface.co/datasets/ProGamerGov/synthetic-dataset-1m-dalle3-high-quality-captions
14. AVA-Space, "Midjourney," Hugging Face Datasets, 2025, accessed: 2025-10-06. [Online]. Available: https://huggingface.co/datasets/ava-space/MidJourney
15. HighCWu, "diffusiondb 2m first 5k canny," Hugging Face Datasets, 2025, accessed: 2025-10-06. [Online]. Available: https://huggingface.co/datasets/HighCWu/diffusiondb
16. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. and Krueger, G., 2021, July. Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PmLR.
17. Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C. and Wang, L., 2022. Git: A generative image-to-text transformer for vision and language. arXiv preprint arXiv:2205.14100.
18. Li, J., Li, D., Xiong, C. and Hoi, S., 2022, June. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In International conference on machine learning (pp. 12888-12900). PMLR.
19. Yan, F. and Mikolajczyk, K., 2015. Deep correlation for matching images and text. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3441-3450).
20. Vinyals, O., Toshev, A., Bengio, S. and Erhan, D., 2015. Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).
21. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J. and Forsyth, D., 2010, September. Every picture tells a story: Generating sentences from images. In European conference on computer vision (pp. 15-29). Berlin, Heidelberg: Springer Berlin Heidelberg.
22. Hodosh, M., Young, P. and Hockenmaier, J., 2013. Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Artificial Intelligence Research, 47, pp.853-899.
23. Gong, Y., Ke, Q., Isard, M. and Lazebnik, S., 2014. A multi-view embedding space for modeling internet images, tags, and their semantics. International journal of computer vision, 106(2), pp.210-233.
24. Karpathy, A., Joulin, A. and Fei-Fei, L., 2014. Deep fragment embeddings for bidirectional image sentence mapping. Advances in neural information processing systems, 27.
25. Andrew, G., Arora, R., Bilmes, J. and Livescu, K., 2013, May. Deep canonical correlation analysis. In International conference on machine learning (pp. 1247-1255). PMLR.
26. Liu, Z., Yang, D., Zhang, M., Sun, H., Wu, H., Wang, H., Shen, W., Chai, C. and Xia, S., 2025. SeLIP: Similarity Enhanced Contrastive Language Image Pretraining for Multi-modal Head MRI. arXiv preprint arXiv:2503.19801.
27. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z. and Duerig, T., 2021, July. Scaling up visual and vision-language representation learning with noisy text supervision. In International conference on machine learning (pp. 4904-4916). PMLR.
28. Pham, H., Dai, Z., Ghiasi, G., Kawaguchi, K., Liu, H., Yu, A.W., Yu, J., Chen, Y.T., Luong, M.T., Wu, Y. and Tan, M., 2021. Combined scaling for open-vocabulary image classification. arXiv preprint arXiv:2111.10050, 1(2), p.4.
29. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A. and Beyer, L., 2022. Lit: Zero-shot transfer with locked-image text tuning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 18123-18133).
30. Wang, Z., Wu, Z., Agarwal, D. and Sun, J., 2022, December. Medclip: Contrastive learning from unpaired medical images and text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing (Vol. 2022, p. 3876).
31. Huang, S.C., Shen, L., Lungren, M.P. and Yeung, S., 2021. Gloria: A multimodal global-local representation learning framework for label-efficient medical image recognition. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 3942-3951).
32. Zhang, Y., Jiang, H., Miura, Y., Manning, C.D. and Langlotz, C.P., 2022, December. Contrastive learning of medical visual representations from paired images and text. In Machine learning for healthcare conference (pp. 2-25). PMLR.

33. Lei, J., Dai, L., Jiang, H., Wu, C., Zhang, X., Zhang, Y., Yao, J., Xie, W., Zhang, Y., Li, Y. and Zhang, Y., 2025. Unibrain: Universal brain mri diagnosis with hierarchical knowledge-enhanced pre-training. Computerized Medical Imaging and Graphics, 122, p.102516.
34. Zaidi, S.A.J., Buriro, A., Riaz, M., Mahboob, A. and Riaz, M.N., 2019. Implementation and comparison of text-based image retrieval schemes. International Journal of Advanced Computer Science and Applications, 10(1), pp.611-618.
35. Hanani, U., Shapira, B. and Shoval, P., 2001. Information filtering: Overview of issues, research and systems. User modeling and user-adapted interaction, 11(3), pp.203-259.
36. Zhang, S., Wang, W., Ford, J. and Makedon, F., 2006, April. Learning from incomplete ratings using non-negative matrix factorization. In Proceedings of the 2006 SIAM international conference on data mining (pp. 549-553). Society for Industrial and Applied Mathematics.
37. Duan, L., Xu, D., Tsang, I.W.H. and Luo, J., 2011. Visual event recognition in videos by learning from web data. IEEE Transactions on pattern analysis and machine intelligence, 34(9), pp.1667-1680.
38. Rui, Y., Huang, T.S. and Chang, S.F., 1999. Image retrieval: Current techniques, promising directions, and open issues. Journal of visual communication and image representation, 10(1), pp.39-62.
39. Li, H., Zou, Y., Wang, Y., Majumder, O., Xie, Y., Manmatha, R., Swaminathan, A., Tu, Z., Ermon, S. and Soatto, S., 2024. On the scalability of diffusion-based text-to-image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9400-9409).
40. Lv, Z., Pan, T., Si, C., Chen, Z., Zuo, W., Liu, Z. and Wong, K.Y.K., 2025. Rethinking Cross-Modal Interaction in Multimodal Diffusion Transformers. arXiv preprint arXiv:2506.07986.