# TEXT CONDITIONED IMAGE GENERATION USING DIFFUSION MODELS

**A PROJECT REPORT**

***Submitted By***

**DHANYA SRINIVASAN 205001029**

**MAHESHWARI M R 205001061**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Department of Computer Science and Engineering**

**Sri Sivasubramaniya Nadar College of Engineering**
**(An Autonomous Institution, Affiliated to Anna University)**
**Kalavakkam - 603110**

**May 2024**

## Sri Sivasubramaniya Nadar College of Engineering

**(An Autonomous Institution, Affiliated to Anna University)**

## BONAFIDE CERTIFICATE

Certified that this project report titled **"TEXT CONDITIONED IMAGE GENERATION USING DIFFUSION MODELS"** is the *bonafide* work of "**DHANYA SRINIVASAN (205001029)** and **MAHESHWARI M R (205001061)**" who carried out the project work under my supervision.

Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**Dr. T.T. MIRNALINEE**
**HEAD OF THE DEPARTMENT**
Professor,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

**Dr. P. MIRUNALINI**
**SUPERVISOR**
Associate Professor,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

Place:
Date:

Submitted for the examination held on. . . . . . . . . . . .

**Internal Examiner** **External Examiner**

# ACKNOWLEDGEMENTS

I thank GOD, the almighty for giving me strength and knowledge to do this project.

I would like to thank and deep sense of gratitude to my guide **Dr. P. MIRUNALINI**, Associate Professor, Department of Computer Science and Engineering, for his valuable advice and suggestions as well as his continued guidance, patience and support that helped me to shape and refine my work.

My sincere thanks to **Dr. T.T. MIRNALINEE**, Professor and Head of the Department of Computer Science and Engineering, for her words of advice and encouragement and I would like to thank our project Coordinator **Dr. B. BHARATHI**, Associate Professor, Department of Computer Science and Engineering for her valuable suggestions throughout this project.

I express my deep respect to the founder **Dr. SHIV NADAR**, Chairman, SSN Institutions. I also express my appreciation to our **Dr. V. E. ANNAMALAI**, Principal, for all the help he has rendered during this course of study.

I would like to extend my sincere thanks to all the teaching and non-teaching staffs of our department who have contributed directly and indirectly during the course of my project work. Finally, I would like to thank my parents and friends for their patience, cooperation and moral support throughout my life.

**DHANYA SRINIVASAN** **MAHESHWARI M R**

# ABSTRACT

Image generation systems often lack understanding of textual descriptions, resulting in generated images missing context-specific details. In this research work, a novel method of image generation from text using image diffusion models has been proposed. The system uses the fine-tuned Bootstrapping Language-Image Pre-training (BLIP) model to learn the relationship between image-text pairs by extracting image and text features from the training dataset. The extracted features are used for end-to-end training of Vector-Quantised Variational Auto-encoder (VQ-VAE) model, which consists of an encoder, quantizer and decoder. The encoder encodes the features, and reduces them to a lower dimensional shared-latent space, which is quantized and mapped by the quantizer according to the codebook and fed into the PixelSNAIL model in order to predict the pixels. The decoder takes as input the predicted pixels in order to generate semantically-preserved and visually realistic images. The system achieved a Learned Perceptual Image Patch Similarity (LPIPS) score of 0.124 for the generated images. The system is further evaluated by generating text captions for the generated images, obtaining a BLEU score of 0.47138 and a ROUGE-1 score of 0.4312.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

CHAPTER 1

# INTRODUCTION

A picture is regarded to be worth a thousand words. Nowadays, despite that, with so many images on the internet from different locations and circumstances, it's critical to understand the precise background information of every given image. It is essential in these circumstances that the image be given some context in the form of an accompanying text. As a result, it is evident that language is crucial for gaining a suitable context, improving comprehension of an image, and adding to its clarity.

Text-to-Image (T2I) generation refers to the generation of visually realistic images, that match the context of the given text description. Although visually realistic images are generated by Generative Adversarial Networks (GANs) as in Zhang et al [**zhang2017stackgan**], Zhang et al [**zhang2018photographic**], Xu et al [**xu2018attngan**], and Hong et al [**hong2018inferring**], they struggle in producing images that are semantically aligned with the input text. T2I leverages the power of GAN in using textual conditioning to generate images, instead of starting on noise. GAN based methods utilize a discriminator to distinguish between the generated image-text pair and also the ground truth image-text pair. The discriminator by itself is inefficient in modeling semantic consistency due to the domain gap between the image-text pairs. Attention mechanisms Zhang et al ([**zhang2018photographic**]) are also used to help the generator in focusing on different words, but this also fails in producing global semantic consistency due to the difference in image and text modalities. Image generative models are useful in various applications, such as photo searching, photograph editing and inpainting.

In order to consider modeling the underlying semantics in both domains, the proposed framework leverages the relationship existence between the image-text pairs, in order to generate images that are semantically consistent with the provided text descriptions. The PixelSNAIL generative model is made use of, which helps in achieving images of greater clarity.

## 1.1 MOTIVATION

Text-image understanding facilitates cross-modal understanding by integrating textual and visual information seamlessly. This enables multimodal fusion, where textual descriptions enrich visual content, and vice versa, leading to more comprehensive and expressive image generation.

Deep learning models use generative networks, which lack in the semantic alignment of the generated images with the given textual descriptions. It is very much needed to generate semantically aligned images with the textual descriptions, along with them being visually realistic. Most of the current systems often struggle with nuanced cross-modal understanding, leading to mismatches between textual descriptions and generated images.

To facilitate the generation of semantically aligned as well as visually realistic images, a method to integrate domain-specific knowledge by learning the relationship between image-text pairs using shared representations of the modalities is proposed. To further generate visually realistic images, an autoregressive generation process is used to predict high-quality diverse output images pixel by pixel.

## 1.2 BACKGROUND

In recent years, the integration of textual and visual information has emerged as a promising avenue for advancing machine learning models, particularly in image generation. This integration, enhances various applications such as content creation and multimedia analysis. By combining textual descriptions with visual content, text-image understanding enables multimodal fusion, resulting in more comprehensive image generation.

However, current generative networks struggle to align generated images with textual descriptions, hindering the production of semantically meaningful and visually realistic images. To address this, this research proposes leveraging domain-specific knowledge to enhance alignment during the generation process. The goal of this work is to increase fidelity by adding domain-specific semantics and learning correlations between image-text pairs. An autoregressive generation technique is used to guarantee that high-quality photos are generated.

Current systems often lack context-specific details, and training them with diverse datasets can be time-intensive. The thesis, after exploring limitations of current image generation systems, aiming to overcome them by proposing a methodology focused on generating images by leveraging the BLIP model along with the VQ-VAE model to learn the relationship between image-text pairs.

Building on text-image understanding principles and advancements in image generation, this research aims to leverage domain-specific knowledge to enrich the generation process and produce semantically aligned and visually realistic images. By using models such as VQ-VAE and PixelSNAIL, alongside the BLIP model, enhancement of the understanding of the interplay between text and images is achieved. This approach aims to push the boundaries of text-to-image synthesis and contribute to the development of more sophisticated image generation systems.

## 1.3 PROBLEM DEFINITION

The research work proposes a methodology to generate semantically aligned and visually realistic images by implementing a model facilitating cross-modal understanding to improve the generation process. The work employs BLIP, which is trained to understand if a relationship exists between the text-image pair, to address the challenge faced by many image generating programs, which is having incomplete or a dataset without any caption. In order to produce text and picture features that need to be personalized, the VQ-VAE and PixelSNAIL models are employed to create high-quality images.

VQ-VAEs generate a discrete latent space when compared to the normal VAEs, which generate a continuous latent space. VQ-VAE's prior is trained, preventing it from having a posterior collapse when the posterior distribution over latent variables becomes overly dependent on the observed data.

The quantizer of the VQ-VAE model is used to extract an LMDB of the latent space codes which is used to train the PixelSNAIL model. The PixelSNAIL model is used to generate high quality images in auto-regressive manner, capturing long-range dependencies and incorporating conditional information. The PixelSNAIL model predicts the pixels, which is fed to the decoder of the VQ-VAE. The decoder predicts the image.

Standard metrics are used to evaluate the effectiveness of the proposed model. To analyse the working of the BLIP model metrics such as F1 score, accuracy, mean and median similarity score are used.

The Encoder of the VQ-VAE model is analysed by visualizing the latent space using the t-SNE (t-Distributed Stochastic Neighbor Embedding) and the PCA (Principal Component Analysis) techniques. It is observed in these visualizations that nodes with similar features are closely knit together whereas nodes with dissimilar features are placed away from each other. The decoder of the VQ-VAE model is analysed using MSE (Mean Square Error) and PSNR (Peak Signal-to-Noise Ratio) metrics.

To evaluate the generated images, the caption of the images are generated and compared against the original caption. This comparison is done based on metrics such as the BLEU, METEOR, ROUGE-1, ROUGE-L scores. The LPIPS score is also used to evaluate the perceptual similarity between the ground truth image and generated image.

## 1.4 ORGANISATION OF THE REPORT

The initial section of the following report addresses the motivation for conducting this research and the problem statement. Subsequently, the next section provides an overview of the evolution of text-image generative models over the years, including a discussion on the GANs and state-of-the-art model, CLIP (OpenAI). Research gaps observed in the existing literature within this domain are also highlighted. Following this, the report delves into a discussion about the proposed model in Chapter 3, which aims to address some of the challenges encountered by previous models. The next section outlines the work conducted within the scope of this research. The evaluation results and the performance analysis is presented in the next section. The social impact that the research is subject to in society is then analysed. The report is then concluded with how the work can be further taken forward.

CHAPTER 2

# LITERATURE SURVEY

Text-to-image synthesis, situated at the intersection of natural language processing and computer vision, continues to advance significantly. Recent research has seen substantial progress in photorealistic image generation, language-agnostic training, and hierarchical modeling. These have helped us have a more seamless image text understanding. This review explores twelve influential papers in text-guided image generation.

## 2.1 GENERATIVE ADVERSARIAL NETWORKS (GAN) BASED MODELS

The authors of Ramzan et al. (2022) [**ramzan2022text**] focuse on the innovative application of RC-GAN (Recurrent Conditional Generative Adversarial Network) methodology, utilizing the Oxford-102 flower dataset. This dataset comprises 8,189 images across 102 classes, with each image accompanied by 10 textual descriptions. The core of the RC-GAN approach lies in leveraging semantic information from textual descriptions as input for the generator model. This model is tasked with converting characteristic information into pixels to synthesize images that are coherent with the provided descriptions. Subsequently, these generated images are fed into the discriminator model alongside both correct and incorrect textual descriptions, as well as real sample images from the dataset, to evaluate the authenticity of the generated images. The effectiveness of this text-to-image generation process is quantified using the inception score and the Peak Signal-to-Noise Ratio (PSNR), which serve as the primary evaluation metrics.These metrics help in assessing the quality and the fidelity of the images generated by the RC-GAN, providing a quantitative measure of its performance in creating realistic images from textual descriptions.

Reed et al. (2016) [**reed2016generative**] introduces the utilization of Deep Convolutional Generative Adversarial Networks (DC-GANs) in the context of text-to-image synthesis. The study employs diverse datasets, including the Caltech-UCSD Birds dataset, Oxford-102 Flowers dataset, and the MS COCO Dataset. The DC-GAN methodology involves a dual process where the discriminator distinguishes between real training data and synthetic images, while the generator endeavors to deceive the discriminator. Both the generator network (G) and the discriminator network (D) perform feed-forward inference conditioned on the text feature, aiming to create images that align with the provided textual descriptions. However, the DC-GAN model has certain limitations, as it tends to not noticeably reflect single-word changes and may generate scenes that lack coherence. These limitations highlight areas for improvement and refinement in the context of text-guided image generation using DC-GANs.

Qiao et al. (2019) [**qiao2019mirrorgan**] introduces an approach to text-to-image generation through three key modules: STEM, GLAM, and STREAM. The dataset employed for this study includes the CUB Bird Dataset and the MS COCO Dataset. In the MirrorGAN framework, STEM is responsible for generating word/sentence embeddings, which subsequently serve as input for GLAM. GLAM, a cascaded architecture, generates target images in a progressive manner, moving from coarse to fine scales. It leverages both local word attention and global sentence attention to enhance diversity and ensure semantic consistency in the generated images. Additionally, STREAM attempts to regenerate the text description from the generated image, aligning semantically with the provided text description. Despite these advancements, MirrorGAN has limitations. The modules, including STREAM, are not jointly optimized with complete end-to-end training due to computational constraints. Furthermore, the basic methods employed for text embedding in STEM and image captioning in STREAM could benefit from further refinement for improved performance.

The domain of text-guided image generation, specifically focusing on diverse face images is employed in the Xia et al. (2021) [**xia2021tedigan**]. The dataset used for this research is the Multi-Modal CelebA-HQ dataset. The methodology employed is a GAN inversion technique capable of mapping multi-modal information into a common latent space of a pretrained StyleGAN. This allows for learning instance-level image-text alignment, enabling the model to generate diverse face images guided by textual descriptions.Despite its innovative approach, TediGAN exhibits limitations. Notably, some unrelated attributes may be unintentionally altered when manipulating a given image based on a text description. Additionally, the use of a simple text encoder can lead to insufficient disentanglement of attributes and mismatches in image-text alignment, highlighting areas for potential improvement.

## 2.2 DIFFUSION MODELS

The development of visual models that leverage natural language supervision for learning is seen in the Radford et al. (2021) [**radford2021learning**] . The model presented is one of the state of art models. Utilizing the WebImage Text (WIT) dataset, the methodology centers around contrastive learning with natural language processing (NLP) supervision. This approach employs a contrastive objective that aims to bring closer the representations of semantically related image-text pairs while pushing apart those that are unrelated. To achieve this, both images and texts are projected into a shared embedding space, allowing for a direct alignment between the two modalities.One of the primary limitations of CLIP is its substantial demand for computational resources. This requirement can make it difficult for researchers with limited access to computational power to train or experiment with the model. Additionally, fine-tuning CLIP for specific tasks presents its own set of challenges, as this process necessitates a careful selection and tuning of hyperparameters in accordance with the characteristics of the dataset being used.

Kocasari et al. (2022) [**kocasari2022stylemc**] explores efficient methods for text-guided image generation and manipulation. The datasets employed in this study include FFHQ, LSUN Car, Church, Horse, AFHQ Cat, Dog, Wild, and Metfaces datasets, showcasing a diverse range of visual content. The methodology involves passing latent codes through the generator, with a focus on optimizing the global manipulation direction corresponding to the given text prompt. This optimization is achieved by minimizing both CLIP loss and identity loss. However, the model exhibits limitations, as the manipulation capabilities are highly dependent on the datasets on which CLIP and StyleGAN2 were trained.

Additionally, the joint representation capabilities of CLIP are noted to be limited and biased, suggesting potential areas for improvement in future work.

The authors of Nichol et al. (2021) [**nichol2021glide**] focuse on advancing image generation and editing through text-guided diffusion models. The AVA Dataset is used for experimentation. The methodology revolves around the exploration of diffusion models for text-conditional image synthesis, incorporating two distinct guidance strategies: CLIP guidance and classifier-free guidance. The study emphasizes the comparison of these strategies to understand their impact on the quality of generated images. Despite the promising results, the model exhibits limitations. It fails to accurately capture certain prompts describing highly unusual objects or scenarios. Furthermore, an unoptimized model takes 15 seconds to sample a single image on a single A100 GPU, making it comparatively slower than related GAN methods. This reduced speed may impact its practicality for real-time applications, highlighting an area for potential optimization in future research.

In the paper Zhou et al. (2022) [**zhou2022towards**] the authors introduce an approach to training text-to-image generation models without language supervision. The dataset used for experimentation is CC3M (Conceptual-Caption 3 Million). The methodologies employed involve the use of pretrained CLIP and state-of-the-art (SOTA) image captioning model VinVL. This approach offers an affordable solution for constructing text-to-image generation models, particularly in scenarios with limited image-text data pairs. The Translator T in the proposed methodology serves a dual purpose. When only images are provided (language-free text), a pseudo text-feature generation process is considered, denoted as T:x→h'. Conversely, if image-text pairs are available in a fully supervised setting, the model encodes ground-truth text, represented as T: t → h. However, the methodology has its limitations. The text features generated may not always be relevant to the image as a whole, and inherent limitations of CLIP are acknowledged, suggesting areas for potential improvement and further research.

Ramesh et al. (2022) [**ramesh2022hierarchical**] introduces a model referred to as DALLE2, which leverages a diffusion model, decoders, and both autoregressive and diffusion model priors for text-conditional image generation. The dataset used for experimentation is AVA. The model operates in two stages: the first stage involves a prior that generates a CLIP image encoding given a text caption, and the second stage utilizes a decoder to generate an image conditioned on the image embedding using a diffusion model. However, the model has its limitations. UnCLIP, as part of DALLE2, exhibits challenges in binding attributes to objects, as CLIP embedding itself does not explicitly bind attributes to objects. Additionally, DALLE2 struggles to produce coherent text and encounters difficulties in generating detailed scenes, raising concerns about its risk profile over GLIDE in terms of deceptive and harmful content.

Gu et al. (2022) [**gu2022vector**] presents the VQ-VAE model, which is applied to the CUB-200 and MSCOCO datasets for text-to-image synthesis. The methodologies involve an autoregressive model, encoder and decoder architectures, and a mask and replace strategy. Unlike GAN models that generate images in a left-right, top-down manner, diffusion methods, such as VQ-VAE, generate images in a global manner. VQ-VAE specifically employs a mask and replace strategy, aiming to avoid accumulation errors during the synthesis process. However, the model exhibits certain limitations, including low text coherence, low image resolution, and occasional production of incoherent images due to a perceived lack of proper feature extraction mechanisms.

The authors of Razavi et al. (2019) [**razavi2019generating**] explore the VQ-VAE-2 model's application on the FFHQ dataset for image generation. The methodologies employed include an autoregressive model, encoder and decoder architectures, and the introduction of multi-scale latent maps to enhance the resolution of the generated images. Unlike its predecessor, VQ-VAE-2 introduces a multi-scale latent map strategy for addressing issues related to image resolution. However, the model has several limitations, including low text coherence, a high-risk profile, and an inability to produce very complex images, highlighting the need for improved feature extraction mechanisms.

An approach to unify vision-language understanding and generation using the COCO dataset is presented in Li et al. (2022) [**li2022blip**] . It employs a bootstrapped pre-training strategy that encompasses masked language modeling, masked image modeling, and image-text matching techniques. To enhance the model's learning curve, a curriculum learning strategy is applied, gradually increasing the complexity of tasks in alignment with the model's growing proficiency. This methodology benefits from the integration of Transformer-based components within the model architecture, allowing for the simultaneous processing of complex image and text inputs.However, the implementation of BLIP is not without its challenges. The training process demands substantial computational resources, making it a significant consideration for those looking to replicate or build upon this work. Additionally, the effectiveness of BLIP is intricately tied to the quality and diversity of the pre-training dataset. This dependency underscores the importance of having access to high-quality, varied datasets to achieve optimal performance, highlighting a potential limitation for applications with less diverse data availability. An overview of the papers reviewed is given below in Table 2.1.

## 2.3 RESEARCH GAPS IDENTIFIED

Understanding the relationships between text-image pairs is crucial for generating images from text data, applicable in various fields. While the image-text interoperable applications keep evolving the technologies for those applications also have been evolving. But there are some limitations that have been identified in the models that have studied in this work. Some limitations of existing models include:

- Existing models like GAN and DALLE are exceptionally large. These not only take a large time to train but also require a very large dataset to be able to learn properly and generate a well defined image.
- Generative models are often unable to generate images that are semantically aligned with the given captions. Hence, they often generate unintended images. GANs make use of a discriminator to distinguish between the generated image-text pair and also the ground truth image-text pair. The discriminator is often inefficient in modeling semantic consistency due to the domain gap between the image-text pairs.

| S.no | Paper Title | About | Dataset | Limitations |
|---|---|---|---|---|
| 1 | Towards lang - Free Training for Text-to- Image Generation (Zhou et al. (2022) [**zhou2022towards**]) | Uses a language free and fully supervised translator . | CC3M (conceptual -caption 3 million) | The text features may not be relevant to the image as a whole |
| 2 | BLIP: Bootstrapping Lang- Image Pre- training for Unified Vision- Language Understanding and Generation (Li et al. (2022) [**li2022blip**]) | CLIP trains with masked language and image modeling, image-text matching. Its architecture uses transformers, enabling simultaneous processing of intricate image and text inputs. | COCO dataset | BLIP's training requires substantial computational resources.Its performance hinges on the quality and diversity of the pre-training dataset. |
| 3 | Text-to-Image Generation Using Deep Learning RC-GAN (Ramzan et al. (2022)[**ramzan2022text**]) | The generator model used textual descriptions to convert characteristics into pixels, creating images. These are used in discriminator to evaluate | Oxford- 102 flower dataset | GAN uses a local generation stratergy and uses a lot of computational power and is slow |

| S.no | Paper Title | About | Dataset | Limitations |
|---|---|---|---|---|
| 4 | Generative Adversarial Text to Image Synthesis DC-GAN (Reed et al. (2016) [**reed2016generative**]) | Deep convolutionalGAN. The discriminator discerns real from synthetic images, as the generator endeavors to deceive it. Both generator abd discriminator network conduct feed-forward inference conditioned on the text feature. | COCO dataset | GAN uses a local generation stratergy |
| 5 | Learning Transferable Visual Models From Natural Language Supervision (Radford et al. (2021) [**radford2021learning**]) | bring semantically related image-text pairs closer and push unrelated ones apart in shared embedding space, enabling direct alignment of images and text. | Web Image Text (WIT) | Requires significant computational power,fine-tuning it requires hyperparameter and dataset consideration. |
| 6 | GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models (Nichol et al. (2021) [**nichol2021glide**]) | Explore diffusion models for the problem of text-conditional image synthesis and compare two different guidance strategies: CLIP guidance and classifier-free guidance. | AVA Dataset | Fails to capture certain prompts which describe highly unusual objects or scenarios. Unoptimized model takes 15 seconds to sample one image on a single A100 GPU. |

| S.no | Paper Title | About | Dataset | Limitations |
|---|---|---|---|---|
| 7 | Hierarchical Text-conditional Image Generation with CLIP Latents (Ramesh et al. (2022) [**ramesh2022hierarchical**]) | 2 stage model is used -a prior : generates a CLIP image encoding given a text caption -a decoder: generates an image conditioned on the image embedding using diffusion model | AVA Dataset | UnCLIP is worse at binding attributes to objects. CLIP embedding itself does not explicitly bind attributes to objects. hard time producing details in complex scene |
| 8 | Bleu: a Method for Automatic Evaluation of Machine Translation ([**papineni2002bleu**]) | evaluating machine translation by comparing the output of machine translation systems with one or more reference translations | N-gram Precision and Weighted precision | Insensitive to Semantic Errors, Insensitive to Synonyms and Paraphrases |
| 9 | Vector Quantized Diffusion Model for Text-to-Image Synthesis (Gu et al. (2022) [**gu2022vector**]) | Encoder and decoder architectures, mask and replace strategy. Diffusion methods generate the image in a global. | CUB-200 and MSCOCO datasets | Resolution of images is low. Produces incoherent images due to lack of proper feature extraction |

| S.no | Paper Title | About | Dataset | Limitations |
|---|---|---|---|---|
| 10 | Generating Diverse High-Fidelity Images with VQ-VAE-2 (Razavi et al. (2019) [**razavi2019generating**]) | VQ-VAE2 model introduces a multi-scale latent map for increasing the resolution of the image generated | FFHQ dataset | High risk profile. Unable to produce very complex images and requires better feature extraction |

TABLE 2.1: Overview of literature review

- Attention mechanisms are used to help the generator in focusing on different words, but this fails in producing global semantic consistency due to the difference in image and text modalities.

- Visually realistic images are often not generated by existing T2I models.

In our work, the above limitations are overcome by making use of the BLIP model. This model is used to precisely understand the relationships between the modalities of the inputs.

Moreover, the VQ-VAE model is used, which, while being smaller than existing models like GAN and DALLE, also has a discrete latent space, which leads to considerably more interpretable and controlled representations. The VQ-VAE model is used in conjunction with the PixelSNAIL model in order to generate highly realistic images.

## 2.4 RESEARCH OBJECTIVES

Image and text go hand in hand for a variety of purposes. Thus, fulfilling any application that combines these two becomes heavily dependent on comprehending the relationship between the image and a related text.

- The understanding of the relationship between image and text is put to use in image generation. With the use of an image generating model, one can create an image when given a text cue. To do this, the model is trained using a dataset that associates each object in an image with a textual description. These descriptions are then combined to create the requested image.

- Additionally, a smaller model is developed that can be trained on a comparatively small dataset, thereby resolving the issue of the lack of models small enough for customization to their respective domains.
- The use of the diffusion based model is explored, which aids in producing an image of higher quality. A deeper comprehension of it is obtained, by making use of the latent space of the VQ-VAE model, which accurately maps the multiple modalities together.

CHAPTER 3

# PROPOSED METHODOLOGY

The proposed system makes use of the BLIP model, along with the VQ-VAE model and the PixelSNAIL model. The BLIP model is used for feature extraction of the text-image pairs of the dataset after learning the nuanced relationships existing between the image and text. Then, using the representations generated from the BLIP, the VQ-VAE model is used for large-scale image generation, enhancing coherence and fidelity with autoregressive priors. Its streamlined architecture prioritizes fast encoding and decoding. By using autoregressive models in the compressed latent space, sampling speed is significantly improved, particularly for large images. Further the PixelSNAIL model implements an autoregressive generation process to predict high-quality output images pixel by pixel using convolutional layers. These pixels are then decoded by the VQ-VAE decoder along with the quantized latent space in order to generate the images. The proposed system is depicted in Figure 4.3.

## 3.1 FINE-TUNING THE BLIP MODEL FOR FEATURE EXTRACTION

The BLIP model [**li2022blip**] is a Vision-Language Pre-training (VLP) framework which has been used to help understand the relationship between multiple modalities. The BLIP model is a unified multimodal mixture of an encoder and decoder with both understanding and generation capabilities. It

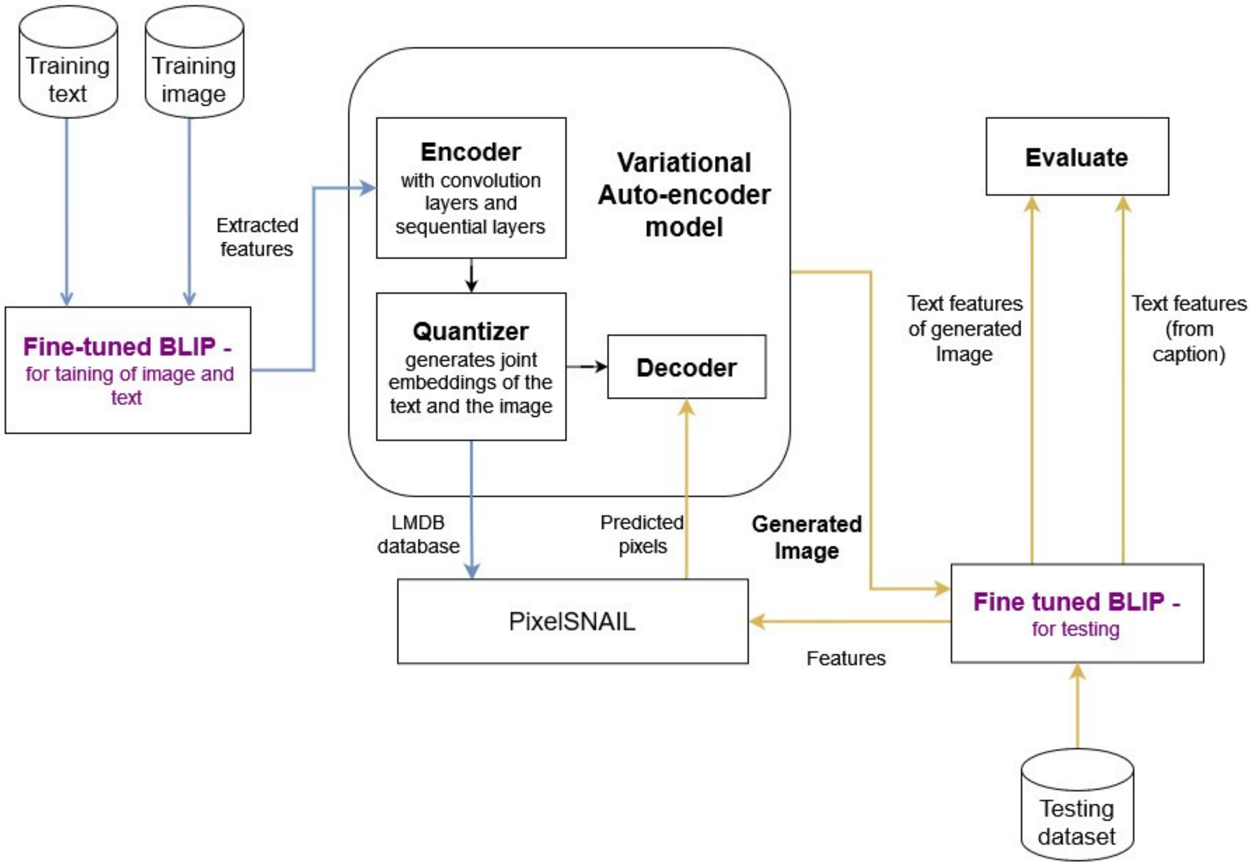

Training text
Training image
Fine-tuned BLIP - for taining of image and text
Extracted features
Encoder with convolution layers and sequential layers
Variational Auto-encoder model
Quantizer generates joint embeddings of the text and the image
Decoder
LMDB database
Predicted pixels
Generated Image
PixelSNAIL
Features
Fine tuned BLIP - for testing
Testing dataset
Evaluate
Text features of generated Image
Text features (from caption)

FIGURE 3.1: Proposed system architecture

consists of an unimodal encoder which separately encodes image and text features. The text and image encoders are referred to as the text model and the vision model respectively. The text model is based on a transformer architecture BERT [**devlin2018bert**], which processes textual descriptions present in the image-text pairs. The text model tokenizes the text input and then converts it into token embeddings. The vision model, based on convolutional neural networks (CNNs) and vision transformer architectures (ViT) [**dosovitskiy2020image**] processes visual data such as images. Images are pre-processed, resized, and normalized before being fed into the vision model which learns to represent image tokens. The text and vision models which are pre-trained individually, have been integrated into a joint text-vision model, which combines the representations learned by the text and vision models into a shared multimodal space which helps in cross-modal understanding. During joint training, the model is trained using paired image-text samples from a large-scale dataset. The model learns to align the representations of images and their corresponding text captions through pre-training objectives such as image-text alignment or cross-modal prediction tasks. By fine-tuning the model using MediaEval MUSTI training dataset [**zinnen2022odor**], the model is able to identify image-text relationships on the specific dataset. The architecture of the BLIP is depicted in Figure 3.2, following which the fine-tuning process is explained. Since the VQ-VAE model uses both the text and image representations during its training process, the efficiency of this model is influenced by the features extracted from these tokens. An image and a corresponding text caption are utilized to extract text and image representations which are then fed to the next step. Hence the BLIP is fine-tuned to process the data and extract image and text embeddings which are further normalized and used to train the VQ-VAE model.

## 3.2 VQ-VAE

The image and text features extracted using the fine-tuned BLIP model are used to train the VQ-VAE model. It is a probabilistic model that finds latent, low-dimensional representations of data. The VQ-VAE model consists of residual blocks of text and image encoders, a quantizer and a decoder. The encoders encode images and text into a lower dimensional latent space, which are

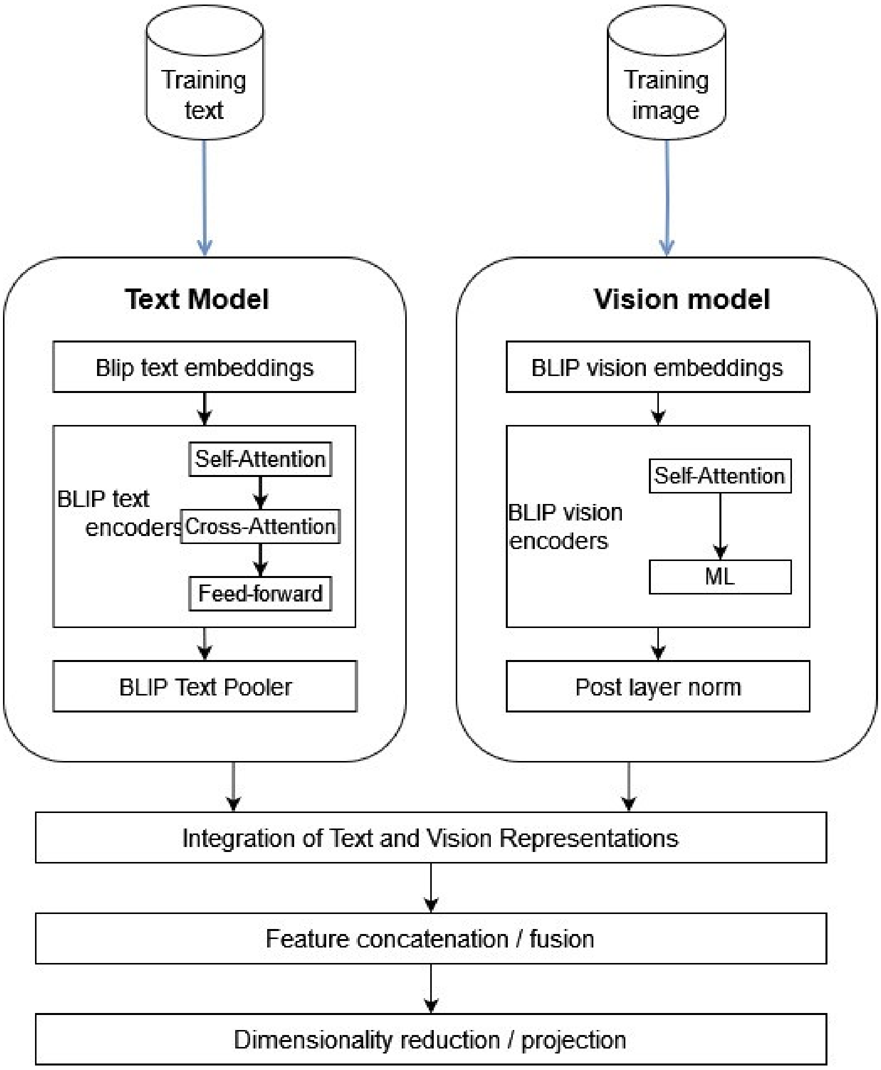


FIGURE 3.2: BLIP model Training process

quantized by the quantizer according to the codebook. At the generation time, the decoder decodes points from the latent space back into the original format. Without a proper VQ-VAE, the model can't properly reconstruct the images. The following subsections explain the working of the VQ-VAE model. The algorithm of the training process is explained in Alg. **??**.

### 3.2.1 ResBlock Module

This module implements a residual block, which is the key component in deep neural networks. It is used in the image encoder and decoder blocks of the proposed VQ-VAE model to perform downsampling and upsampling, based on the stride parameter. This residual block consists of 2 convolutional layers followed by ReLU activation functions.

1. The first convolutional layer processes the input features

2. The second convolutional layer refines the output.

The output of the second convolutional layer is added to the input features thus creating a skip connection that helps in the gradient propagation during the training process.

### 3.2.2 Image Encoder Module

This module encodes the input image features into a lower-dimensional representation. It includes the convolutional layers and residual blocks, which extract the hierarchical features from the input image features. Depending on the stride parameter (2 or 4), the encoder performs down-sampling to capture the spatial information at different resolution.

### 3.2.3 Quantizer Module

This module takes the lower dimensional input feature of text and image from the encoders and quantizes them into joint discrete embeddings. It is done on the basis of the following attributes:

1. **dim -** The dimensionality of the input features
2. **n_embed -** The number of embedding vectors
3. **decay -** The exponential moving average decay rate used for updating embeddings
4. **eps -** A small value added for numerical stability

During initialization, random embeddings of shape (dim, n_embed) are created. Quantization is done by finding the closest embedding vector using Euclidean distance. The quantization error between the quantized features and the input features is calculated. During the training process, the module updates the embeddings using the exponential moving average for better representation of the input distribution.

### 3.2.4 Decoder Module

The decoder module reconstructs the quantized features back into the original features. It mirrors the structure of the encoder in a reverse order, using the transposed convolutional layers for upsampling. The module includes the convolutional layers and the residual block to capture the detailed information during reconstruction.

**Algorithm 1** Training Algorithm for VQ-VAE

```
procedure TRAIN($epoch, loader, model, optimizer, scheduler, device$)
    $criterion \leftarrow$ MSELoss()
    $latent\_loss\_weight \leftarrow 0.25$
    $sample\_size \leftarrow 25$
    $mse\_sum \leftarrow 0$
    $mse\_n \leftarrow 0$
    for $i$ in $[0, \text{len}(loader))$ do
        $model.zero\_grad()$
        $img, label \leftarrow$ loader[$i$]
        $img \leftarrow img.to(device)$
        $text \leftarrow text.to(device)$
        $out, latent\_loss \leftarrow model(img, text)$
        $recon\_loss \leftarrow$ criterion($out, img, text$)
        $latent\_loss \leftarrow$ latent_loss.mean()
        $loss \leftarrow recon\_loss + latent\_loss\_weight * latent\_loss$
        $loss$.backward()
        if $scheduler \neq$ None then
            $scheduler$.step()
        end if
        $optimizer$.step()
        $part\_mse\_sum \leftarrow recon\_loss$.item() $\times$ img.shape[0]
        $part\_mse\_n \leftarrow$ img.shape[0]
        $mse\_sum \leftarrow mse\_sum + part\_mse\_sum$
        $mse\_n \leftarrow mse\_n + part\_mse\_n$
        $lr \leftarrow optimizer$.param_groups[0][”$lr$”]
    end for
end procedure
```

## 3.3 CODE EXTRACTION MODULE

This is a bridging step between the VQ-VAE and the PixelSNAIL module. In this module, the trained VQ-VAE is used to extract latent codes from the images and store them in a Lightning Memory-Mapped Database (LMDB).

The LMDB database is B-tree database management library. The entire database is exposed in a memory map, and all data fetches return data directly from the mapped memory. This prevents from any memory copies or useless memory allocations during data fetch. LMDB is extremely high performance and memory-efficient.

In this particular application, LMDB is used as it is highly efficient for data fetching from a large dataset and does not create any garbage data, giving us a highly efficient storage and access mechanism.

For each batch, images and text features are encoded using the VQ-VAE model to obtain latent codes. The latent codes are further converted into NumPy arrays which contain the latent code and filename. It is serialized and then stored in the LMDB database.

## 3.4 PIXELSNAIL MODEL

The latent space generated by the VQ-VAE module stored in the LMDB database is use to train the PixelSNAIL model as seen in [**chen2018pixelsnail**]. The model is an autoregressive generative model that is used in density estimation tasks involving high dimensional data. In this proposed work, it is used to predict pixels in auto-regressive manner while capturing long-range dependencies and incorporating conditional information. The PixelSNAIL model consists of different layers. Each layer has been discussed. The algorithm of the training process is explained in Alg. 2.

### 3.4.1 Weight Normalized Linear Layer

This technique is used to improve the convergence and stability of the neural network training. Weights are normalised according to the L2 norm. This normalisation process helps prevent exploding or vanishing gradients during the training process. It is used for linear transformations of query, key, and value inputs.

### 3.4.2 Convolutional layers

PixelSNAIL uses 2 types of convolutional layers:

1. **Weight Normalized Convolutional layer** It is a standard 2D convolutional layer with weight normalization applied to its weight. This helps in improving training stability and performance.

2. **Causal Convolutional layer** It is a specialized convolutional layer used for autoregressive modelling. Causal convolution ensures that each output pixel only depends on previous input pixels in the sequence, which helps in generating the image sequentially.

### 3.4.3 Shift operations

Shifting operations are performed on input tensors by adding padding. They are used to implement autoregressive behavior, where each pixel prediction depends only on previously generated pixels. They are used to shift the outputs of horizontal and vertical convolutions to prepare for further processing.

### 3.4.4 Gated residual blocks

This block consist of skip connection that allow gradients to flow more easily during training. These residual blocks are gated which helps in controlling the information flow. These are the main blocks used to process input data in each pixel block.

### 3.4.5 Causal self attention

This technique allows the model to focus on different parts of the input sequence while generating the output. It computes the attention scores between different elements of the input sequence and then uses them to weigh the importance of each element.

### 3.4.6 Conditional processing

It involves incorporating conditions into the model to influence the generation process as required. Two techniques are used to do this.

1. **Conditional Residual Network** It processes conditional information using dated residual blocks. Helps the model generates output conditioned on external factors.
2. **Pixel Block** This blocks combines the input data, background information and conditional information to generate output predictions.

The input data is encoded and undergoes horizontal and vertical causal convolutions. The output of horizontal and vertical convolutions is shifted down and right to prepare for further processing. Multiple PixelBlocks are used sequentially, each consisting of several operations including gated residual blocks, causal convolutions, and optionally causal attention mechanisms. Within each PixelBlock, the following operations occur:

- Gated residual blocks process the input data along with conditioning information.
- If attention is enabled, causal attention mechanisms capture long-range dependencies within the block.

The output of each PixelBlock is used as input to the next one in sequence. After processing through all PixelBlocks, the output is passed through a sequence of operations that include gated residual blocks, an ELU activation function, and a final convolutional layer to produce the final output logits.

Overall, the PixelSNAIL model implements an autoregressive generation process by leveraging causal convolutions, gated residual blocks, attention mechanisms, and conditional processing to generate images by predicting pixels using the latent space of image embeddings and relationships learned thus far.

**Algorithm 2** Training Algorithm for PixelSNAIL

```
procedure TRAIN($args, epoch, loader, model, optimizer, scheduler, device$)
    $loader \leftarrow$ tqdm($loader$)
    $criterion \leftarrow$ CrossEntropyLoss()
    for $i$ in $[0, \text{len}(loader))$ do
        $model$.zero_grad()
        $top, bottom, label \leftarrow$ loader[$i$]
        $top \leftarrow top$.to($device$)
        if $args$.hier $=='\ top'$ then
            $target \leftarrow top$
            out, _ $\leftarrow model(top)$
        else if $args$.hier $=='\ bottom'$ then
            $bottom \leftarrow bottom$.to($device$)
            $target \leftarrow bottom$
            out, _ $\leftarrow model(bottom, condition = top)$
        end if
        $loss \leftarrow$ criterion(out, target)
        $loss$.backward()
        if $scheduler \neq$ None then
            $scheduler$.step()
        end if
        $optimizer$.step()
        $pred \leftarrow$ out.max(1)
        $correct \leftarrow$ ($pred ==$ target).float()
        $accuracy \leftarrow$ correct.sum()/target.numel()
        $lr \leftarrow optimizer$.param_groups[0]["$lr$"]
    end for
end procedure
```

## 3.5 IMAGE GENERATION MODULE

The PixelSNAIL model, which is trained using the latent space extracted using the VQ-VAE model, takes the features extracted using BLIP as input and generates samples for both top and bottom hierarchies. The bottom level samples are conditioned on the top level samples to ensure consistency. The PixelSNAIL thus predicts the pixels of the images. It is ensured that the pixel value is within the valid value (-1 to 1). After these samples are generated the VQ-VAE decoder is used to decode the sampled codes into actual images. The decoder of the VQ-VAE model performs the reverse process of the encoder, using convolutional layers and residual block to up-sample and capture details in the pixels generated by the PixelSNAIL to generate image. The generated images are finally saved into specified filenames.

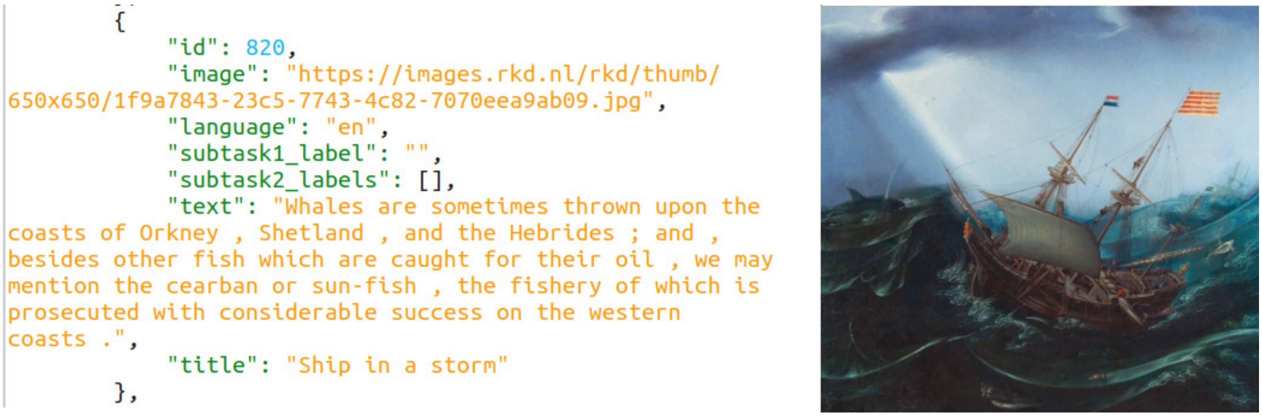
CHAPTER 4

# EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

## 4.1 DATASET DESCRIPTION

The MediaEval MUSTI 2023 dataset from [**zinnen2022odor**] has been used to train and test the proposed system. The dataset consists of artistic paintings such as acrylic and oil paintings, depicting various scenes and still lifes. The training dataset consists of 795 images, each annotated with an id, an elaborate associated text and a title in English.

Similarly, the testing dataset consists of 200 similar images, annotated with ids, associated text and titles. The titles associated with the images are used as the captions and use them to train the BLIP model and the subsequent models. An example of a data sample is given below in Figure 4.1.



```
        {
            "id": 820,
            "image": "https://images.rkd.nl/rkd/thumb/
650x650/1f9a7843-23c5-7743-4c82-7070eea9ab09.jpg",
            "language": "en",
            "subtask1_label": "",
            "subtask2_labels": [],
            "text": "Whales are sometimes thrown upon the
coasts of Orkney , Shetland , and the Hebrides ; and ,
besides other fish which are caught for their oil , we may
mention the cearban or sun-fish , the fishery of which is
prosecuted with considerable success on the western
coasts .",
            "title": "Ship in a storm"
        },
```

FIGURE 4.1: Example of data sample

## 4.2 ECOSYSTEM

### 4.2.1 Hardware specifications:

The NVIDIA GeForce GTX 1080 Ti GPU card along with CUDA version 12.3 was used. For the CPU the INTEL XEON(R) CPU E5-1650 v4 @ 3.60 GHz x 12 was utilized.

### 4.2.2 Software specifications:

The OS, Ubuntu with GNOME version 3.36.8 was used. Python3 environment which facilitates the model's training and testing using the various rich libraries that contain a spectrum of functions of different AI and ML models has been used.

## 4.3 EXPERIMENTS CONDUCTED

In developing the proposed system, several models were taken into consideration. These models were each trained and evaluated. The methods utilized to do the same are explained below.

### 4.3.1 BLIP Fine-tuning

The BLIP model is fine-tuned on the MediaEval MUSTI 2023 training dataset [**zinnen2022odor**]. This fine-tuning is done using the RAdam optimizer, with lookahead logic. The RAdam optimizer is a stochastic optimizer, which helps in rectifying the variance of the adaptive learning rate. The look ahead optimization technique improves optimization performance by effectively smoothing the optimization trajectory.

This BLIP model is fine-tuned rigorously for 40 epochs. After fine-tuning, the BLIP model is tested on the MediaEval MUSTI 2023 testing dataset [**zinnen2022odor**]. The results of the same is discussed in the next section.

### 4.3.2 VQ-VAE Training

The VQ-VAE model is trained rigorously for 560 epochs. After each epoch, the state dictionary of the model is saved as a ".pt" file. During training, the reconstruction loss and latent loss are calculated. The reconstruction loss measures how well the model can reconstruct the input data from the latent representation. It is computed by using Mean Squared Error (MSE) as loss function. The latent loss is a regularization term that encourages the model to learn meaningful and disentangled representations in its latent space. It is computed based on the discrepancy between the latent codes produced by the encoder and a set of discrete latent vectors (codebook) that are learned during training. The latent loss weight is taken as 0.25. The loss is calculated by adding the reconstruction loss and latent loss together. The VQ-VAE model is then tested on the MediaEval MUSTI 2023 testing dataset [**zinnen2022odor**], and the results of the same is discussed in the next section.

### 4.3.3 PixelSNAIL Training

The PixelSNAIL model is trained 2 times, once for the top hierarchy and once for the bottom hierarchy respectively. The top hierarchy of the model is used to generate images at a lower resolution of 32 x 32 pixels. The top hierarchy of the model is built to take up fewer computational resources. The output of the top hierarchy is used as a condition to generate higher resolution images in the bottom hierarchy.

The bottom hierarchy of the model is used to generate images at a higher resolution of 64 x 64 pixels. The bottom hierarchy of the model takes up higher computational resources to train. It takes the output from the top hierarchy as a condition to guide the generation process.

The bottom and top hierarchies of the model are each rigorously trained for 420 epochs. An Adam optimizer is used to optimize the model, with learning rates of 3e-4. The data is sampled from the previously generated LMDB Dataset, which is saved on the system. Depending on the hierarchy specified in the arguments, it generates the target data based on the input data using the PixelSNAIL model. If the hierarchy is 'top', it directly uses top as the target.

A forward pass through the model generates output predictions. The Cross Entropy Loss is calculated between the predicted output and the target data. Cross Entropy Loss, also known as log loss is measures the difference between the predicted probability distribution generated by the model and the true probability distribution of the target. This loss is then backpropagated through the model to compute gradients. The training accuracy is also calculated and printed.

### 4.3.4 Testing process

To test the model, first, the trained VQ-VAE model checkpoints and the PixelSNAIL top and bottom hierarchy model checkpoints are loaded.

The test dataset is loaded one by one and the PixelSNAIL top hierarchy model generates images, which are passed conditionally to the bottom hierarchy model to generate images of a higher resolution.

Finally, the VQ-VAE model decodes the images of the higher and lower resolutions, thus coalescing them into a single representation. This representation is then clamped into the range of [-1,1] to normalize the values. Each image is then saved in the specified output folder.

## 4.4 PERFORMANCE ANALYSIS

### 4.4.1 Performance analysis of the BLIP model

The BLIP model provides insights into the strength and nature of the relationship between an image and its corresponding caption. By testing the BLIP model, how accurately the BLIP model predicts that there is a strong relationship between the image and the text is gauged. To do the same, how often the model predicts that there is a similarity of greater than 50% that the image and the text are related is determined. Various metrics are used to gauge the performance of the fine-tuned BLIP model, These metrics include accuracy, precision, recall, F1 Score and similarity score. The metrics are as explained below:

1. The accuracy measures how often the BLIP model correctly predicts that the image and text are related. This value ranges from 0 to 1.

2. The precision measures the proportion of values that are correctly predicted to be related in the test dataset. This value also ranges from 0 to 1.

3. The recall measures how often the BLIP model correctly identifies the presence of the relationship in the dataset.

4. The F1 score is the harmonic mean of the precision and recall of the BLIP model.

5. The mean calculates the average of the similarity scores calculated by the BLIP model for the testing dataset.

6. The median calculates the middle value of all the similarity scores calculated by the BLIP model for the testing dataset.

7. The standard deviation calculates the dispersion of the similarity scores with respect to the mean of the similarity scores

An accuracy of 1.0, precision of 1.0, recall of 1.0 and an F1 score of nearly 1.0 (0.9996) indicates that the fine-tuned BLIP model is highly efficient in discerning the existence of a relationship between the image and text. Moreover, the similarity scores indicated by the BLIP model has a mean and median of 1.0 and a standard deviation of 0.0, indicating that the BLIP model can now perfectly identify the existence of relationships, and by extension, can identify the relationships as well.

| **Metric** | **Value** |
|---|---|
| Accuracy | 1.0 |
| Precision | 1.0 |
| Recall | 1.0 |
| F1 Score | 0.9996 |
| Mean similarity score | 1.0 |
| Median similarity score | 1.0 |
| Standard deviation of similarity scores | 0.0 |

TABLE 4.1: Evaluation metrics of the BLIP model

### 4.4.2 Performance analysis of the VQ-VAE model

The VQ-VAE model also plays a role in understanding the relationships between the image and text. By testing the VQ-VAE model, how accurately the model is able to understand the image and text features together, and then recreate the image features via the decoder is gauged.

To evaluate the encoder and quantizer, the codes of the testing dataset are extracted. 2 prominent algorithms are used to visualise the same, namely, the t-Distributed Stochastic Neighbor Embedding (t-SNE), Principal Component Analysis (PCA), and eventually, visualize the latent space generated.

- **t-Distributed Stochastic Neighbor Embedding (t-SNE)** -

A highly effective technique for dimensionality reduction and high-dimensional data visualization is t-SNE. The dimensionality of the latent representations was reduced to two dimensions using t-SNE, and matplotlib was used to plot the points. Using clusters of points to represent similarities between data points in the original space, the above visualization aids in the understanding of the latent space's structure. The features of the latent space and the model's ability to discriminate between various categories can be inferred from outliers and the cluster distribution as a whole.

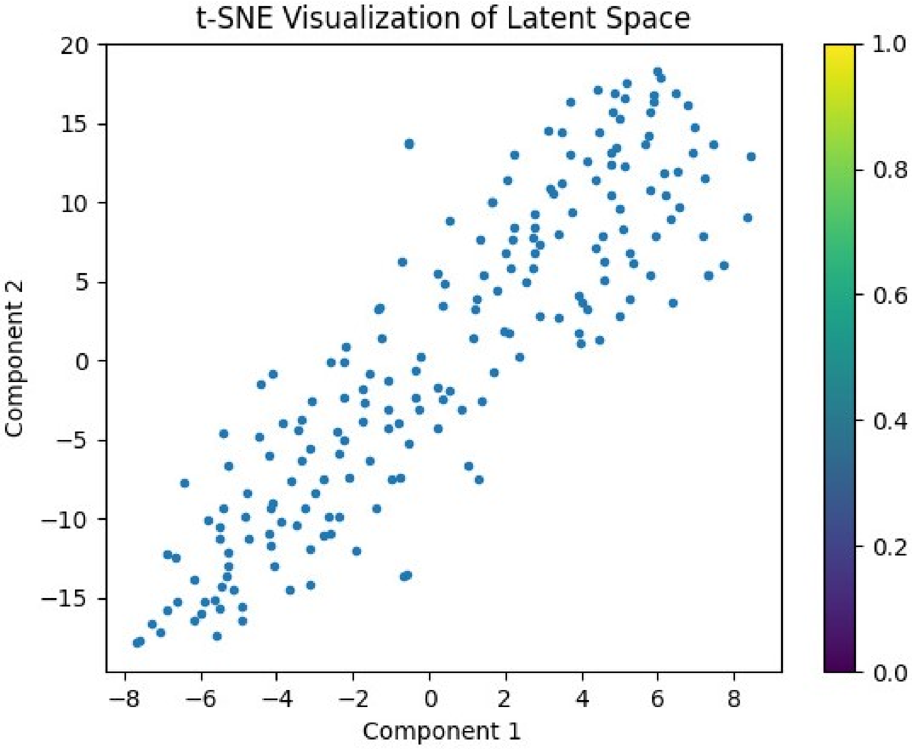


FIGURE 4.2: t-SNE visualization of the latent space

- **Principal Component Analysis (PCA)**

Another method for reducing dimensionality is PCA. In order to minimize dimensionality while retaining as much of the variation as achievable it finds the directions, or principal components, that maximize the variance in the data and projects the data onto these components.

Latent representations' dimensionality was reduced to two dimensions using PCA, and these points were then plotted using matplotlib. With a focus on maintaining the global structure, the final plot displays the high-dimensional data's structure in two dimensions. t-SNE, on the other hand, is more concerned with maintaining local structure.

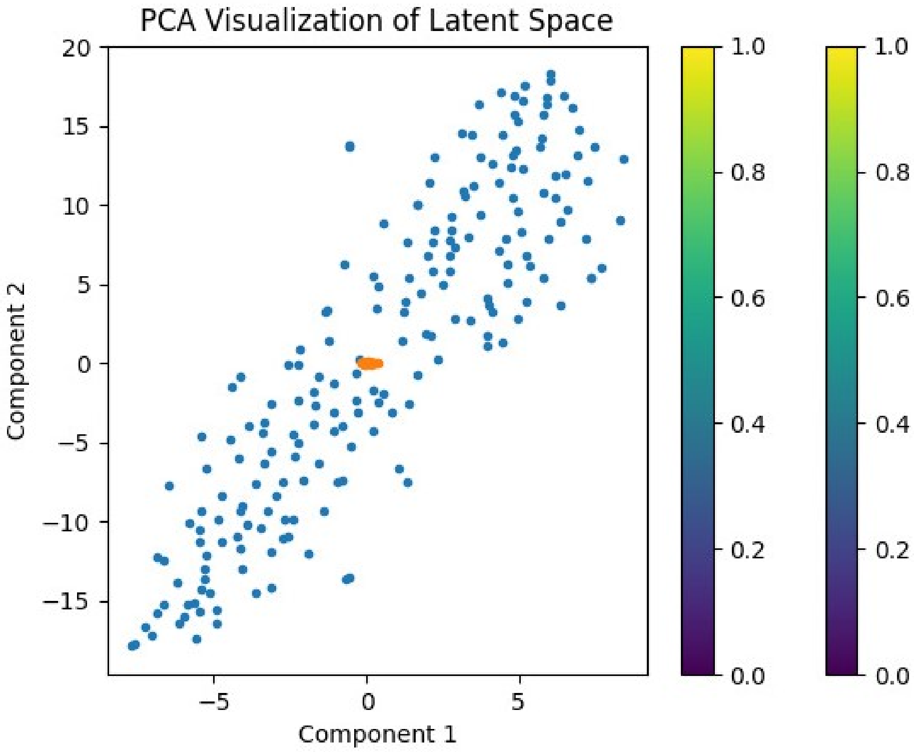


FIGURE 4.3: PCA visualization of the latent space

The directions of the data's largest variance are shown by the plot's axes. The direction of maximum variance is captured by Component 1, while the direction of maximum variance orthogonal to the first is captured by Component 2. Similarity between those data points in the high-dimensional space is indicated by clusters of points that are close to one another. More about the unique characteristics of various groupings in the data can be learned by examining the relative distances between clusters.

From the eigenvalues, the percentage of variation accounted for by each primary component can be determined. This data aids in the evaluation of the degree of information retention between the original data and the reduced-dimensional space. Since PCA is a linear methodology, it might not be as good at capturing intricate non-linear correlations in the data as t-SNE or other techniques. But it gives us an easy-to-understand method for visualizing high-dimensional data and reducing dimensionality.

To evaluate the decoder, 2 metrics are used, namely, MSE Loss and the PSNR Score. The decoder is evaluated as such, to understand the extent to which the model is able to recreate an image from the latent space. The metrics are explained below:

1. A loss function called Mean Square Error (MSE) or L2 loss determines how much of an error there is between the decoder's prediction and the original data.

2. Peak Signal-to-Noise Ratio (PSNR) is a metric used to assess an image's quality. A signal's quality is determined by dividing its maximum potential value by the noise power.

An MSE Loss of 0.240 is obtained, along with a PSNR Score of 6.5799. As seen, the MSE Loss is not as low as expected. Similarly, the PSNR Score is also only moderately high. This, is due to the composition of the training and testing datasets. The training and testing datasets are uniquely made of paintings and historical drawings, as opposed to the FFHQ face image datasets, which are made out of photographs, with clearly demarcated differences in colors and textures. Moreover, owing to hardware constraints, the model has been reduced in size, which could also contribute to the above results.

### 4.4.3 Evaluation of the images generated by the proposed system

The features of the test dataset are extracted using the BLIP model, and then fed it into the proposed system. The proposed system generates an image of resolution 64 x 64 px. As observed below, the images themselves are slightly pixelated and noisy, in comparison to the original images. This may be due to the smaller size of the PixelSNAIL model adapted to the hardware constraints.

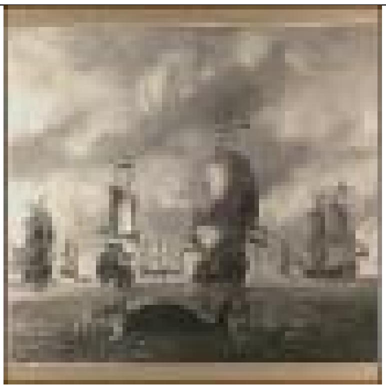
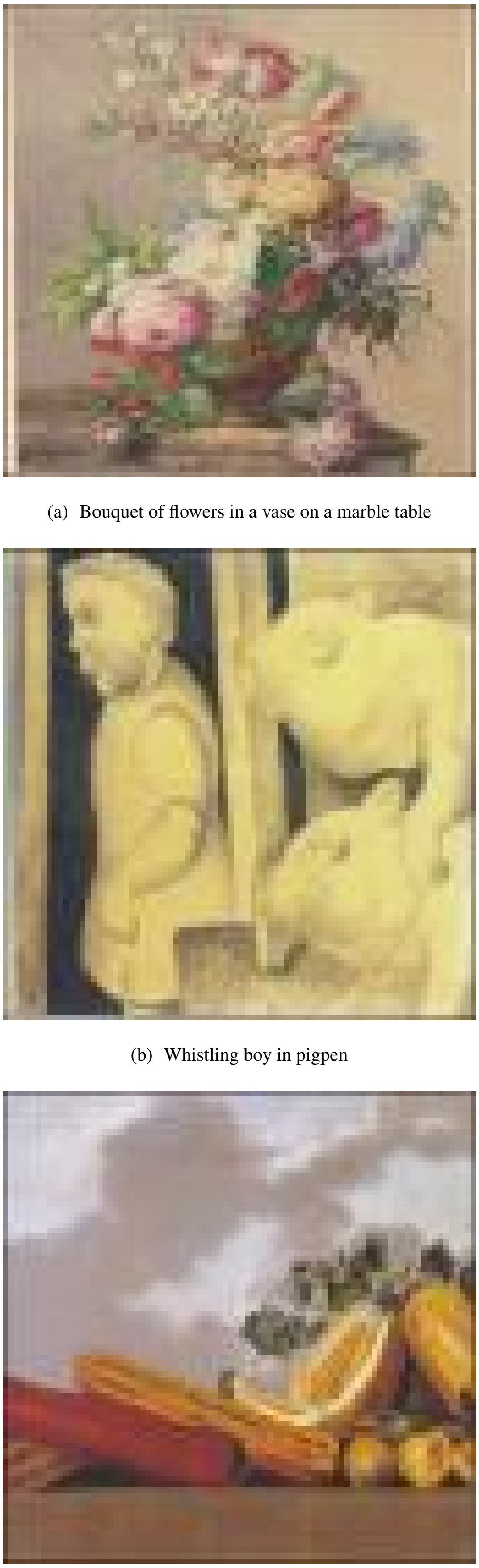
| **Text Prompt** | **Generated Image** | **Caption generated for evaluation** |
|---|---|---|
| Ship in a storm |  | Ship in ocean |

TABLE 4.2: Depiction of the text prompt along with the generated image and a caption generated using this image which is used to evaluate contextual similarity



(a) Bouquet of flowers in a vase on a marble table

(b) Whistling boy in pigpen

(c) Still-life of melons, mango's and a grasshopper

FIGURE 4.4: Image samples generated by the proposed model

To evaluate the above model, several NLP techniques. While this may seem unconventional, it is necessary to evaluate how contextually appropriate the images generated are. To do the same, the generated images are passed through the BLIP model to generate captions for the images that generated.

To evaluate the test results, the BLEU score in [**papineni2002bleu**], the METEOR score in [**banerjee2005meteor**] and the ROUGE scores in [**lin2004rouge**] are used. These metrics are as follows:

1. **The BLEU (Bilingual Evaluation Understudy)** score is a metric used to evaluate the quality of machine-generated text. The text produced by BLIP is compared to the specified reference translations, and a similarity score between the created and reference translations is indicated. In order to determine the precision of sets of n words and, consequently, various gram precisions, the BLEU score computes the n-gram precision and cumulative n-gram scores.

2. **METEOR (Metric for Evaluation of Translation with Explicit Ordering)** is an automatic evaluation metric used to assess the quality of machine-generated translations compared to human-generated references. Unlike BLEU, which focuses primarily on n-gram precision, METEOR takes into account additional factors such as word order and stemming. It calculates precision and recall scores based on the number of matched words and phrases. It normalizes these scores to ensure fairness and consistency

3. **ROUGE-1 (Recall-Oriented Understudy for Gisting Evaluation)** is an automatic metric used in natural language processing (NLP) and specifically

in evaluating the quality of summaries or machine-generated text compared to reference text. It measures the overlap of unigrams between the generated text and the reference text.

4. **ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence)** is an automatic evaluation metric used to assess the quality of machine-produced text or generated summaries in natural language processing (NLP) jobs. The longest common subsequence (LCS) between the reference text and the generated text is measured. Precision in ROUGE-L is defined as the ratio of the generated text's length to the length of the LCS between the generated text and the reference text. Based on the longest common subsequence, it calculates the percentage of the generated text that is relevant to the reference text.

| **Metric** | **Value** |
|---|---|
| BLEU Score | 0.47138 |
| METEOR Score | 0.13567 |
| ROUGE-1 Score | 0.4312 |
| ROUGE-L Score | 0.1563 |

TABLE 4.3: NLP Evaluation metrics of the proposed system

The above scores have been computed as an average for all 200 test data samples. As seen in the above Table 4.3, the BLEU score is fairly good, with a value of 0.47138, while the METEOR score is lower, at 0.13567. This is due to the fact that the METEOR score gives high importance to word order and stemming. The ROUGE scores, similarly, are higher for the ROUGE-1 metric, due to calculation of unigram precision, and lower for the ROUGE-L metric due to the measurement of longest common subsequence.

To evaluate the above model, the Learned Perceptual Image Patch Similarity (LPIPS) metric, introduced in [**zhang2018unreasonable**] is used. LPIPS essentially computes the similarity between the activations of two image patches for the given pre-defined network. This measure has been shown to match human perception well. A low LPIPS score means that image patches are perceptual similar.

The CycleGAN introduced in [**zhu2017unpaired**] uses adversarial training, training two networks simultaneously: a generator and a discriminator. The generator learns to transform images from one domain to another, while the discriminator learns to distinguish between real images from the target domain and fake images produced by the generator. Domain-Specific Variational Information Bound (DSVIB), [**kazemi2018unsupervised**] on the other hand, learns separate latent spaces for each domain involved in the image translation task. These latent spaces capture the unique characteristics and features specific to each domain. The average LPIPS score is calculated for the 200 test images, attaining an LPIPS score of **0.124**, which is an improvement over the above mentioned models. The results are given in Table 4.4

Based upon the BLEU, Meteor, ROUGE-1, ROUGE-L and LPIPS scores obtained above, it can be said that the proposed system achieves a high level of semantic alignment and visual realism in even this dataset consisting of artistic images. Hence, if this system is trained on real images, of homogeneous subjects, this system can perform even better and produce better metrics.

| **Model** | **Value** |
|---|---|
| CycleGAN ([**zhu2017unpaired**]) | 0.134 |
| DSVIB ([**kazemi2018unsupervised**] | 0.129 |
| Proposed system | 0.124 |

TABLE 4.4: LPIPS Score Comparison

CHAPTER 5

# SOCIAL IMPACT AND SUSTAINABILITY

As this study has demonstrated, image production has a wide range of uses that can be highly beneficial for individuals in a variety of professional circumstances. The following are some areas where image generation can be widely used:

1. **Movies and media :** Image generation can help create stunning and creative visuals in movies and other media like MVs. It brings out more creativeness in fantasy objects in the media. This can help us do more in depth scenes and create more live action movies and go above and beyond the boundaries of realism.

2. **Art :** It can help artists visualize their creative misgivings which can hold back one's imagination. It gives us more creative liberties and express more innovative ideas. Image generation also helps artists experiment their ideas on irreplacable objects and reduce their costs in creating art.

3. **Educational :** Image generation can be used to visualize educational images and videos that can help in easier understanding of a topic than just plain words. It can help more visually inclined learners to become more capable in their memory retention.

4. **Medicine and scientific :** It can not only help us visualize things that hard to imagine and look at but also a perfect image generation model maybe help scientists and researches to pinpoint many things. Image generation especially help scientists in deep space adventures and research.

5. **Visual communication :** Image generation can help us reach a more vast audience especially in professions such as journalism where infographics, data visualizations, and multimedia presentations can impart more knowledge to common folk and important information are reported effectively in an understandable manner.

6. **Social media :** With the popularity of instagram and the advent of the influencer era, people tend to stress about being more perfect and creative. Image generation gives them a platorm to do all these people to go above and beyond in their creativity.

7. **Social and cultural norms :** As a continuation on the previous point where extensive use of image generation in social media can change it could be said that it can create a norm of how things are done in social media. Therefore it can create cultural norms and societal perceptions of beauty, identity, and social roles.

While creating images with a model trained on millions of different photos covering a wide range of themes has many advantages, it also has disadvantages. When used improperly, very realistic image production can result in potentially dangerous situations.Here is a list of a few of them:

1. **Misinformation and fake content :** Image generation tools can be used to create fake images and videos, making it challenging to discern real from fake content. This can be used to spread misinformation and fake news. In case of some fake images it can effect very sensitive topics like crime scene related evidences in case it is used to create fabricated lies and create a false trial for the detectives.

2. **Privacy violations :** As image generation can be used by people to generate fake images of other people without their consent it violates the person's privacy and can cause harm to the person's image in the society if the wrong kind of image is generated and circulated in the media.

3. **Deep fakes and impersonation :** It is a technology where highly realistic photos or videos of people are created to impersonate some other person. There have already been a lot of incidents where such deepfakes where used to tarnish the reputation on some celebrities. While it can also be used in the correct manner for example to substitute an actor to his stunt double, this technology in the wrong hands can cause grave harm to people.

4. **Cultural and social norms :** As discussed above about extensive use of image generation in social media platforms can create cultural and social norms. The fact people tend to show a more beautiful and perfect side of themselves they can loose touch of reality which can lead other people to doubt themselves and have mental distress because of this. Image generation can also spread harmful stereotypes which can lead people to marginalisation.

5. **Security threats :** Image generation can be exploited and used in security issues. People can create forged documents, fake identification, or counterfeit currency which all have adverse which cause security threats. It can slowly corrode the security of the nation's security and the public's safety.

6. **Losing trust :** With increase in fake images and videos, people can lose their trust in anything that appears on social media including important news. It

also affects the trust of important institutes trust in sources which can lead to delay in taking vital decisions

The extensive use of image generation raises questions about consent, privacy, and intellectual property rights. Conducting thorough risk analysis on the generated images is essential to mitigate these risks effectively. Implementing content moderation apps post-generation or keyword filtering mechanisms for the input text can aid in identifying and addressing inappropriate content. While image generation is very useful one must make sure to use it for the right cause.Being a good citizen and a human is of utmost importance.

CHAPTER 6

# CONCLUSIONS AND FUTURE WORK

To sum up, this work presents a complex system to create graphics in this thesis. The model focuses on using multimodal approaches with two already-existing generative models: PixelSNAIL and Vector-Quantized Variational Auto Encoder. More significantly, text-image comprehension on the MediaEval MUSTI dataset is made easier with the use of the BLIP model. The BLIP model's exceptional efficiency in identifying connections between text and images has been demonstrated.

It is also possible to achieve multimodal comprehension of the relationship between text and images by training the VQ-VAE model on the training dataset, which has been updated to accept both text and image attributes. The latent space codes are extracted using the VQ-VAE model and then used to regenerate images. Images are created pixel by pixel using the PixelSNAIL model. The latent codes that are extracted using the VQ-VAE model are used to train it. An overall LPIPS score of 0.124 and a BLEU score of 0.47138 are achieved by the proposed system. Considering the complex and varied training and testing dataset, the suggested approach works effectively overall.

A processing unit with additional computational capability can be to run the model in the future. By doing this, the tradeoff made with the smaller PixelSNAIL can be abandoned. More complex training and testing datasets can also be used, and the models can be trained with larger batch sizes in the future. Additionally, in order to more accurately map the semantic and physical links between text and images, more advanced multimodal approaches can be employed.

# REFERENCES

1. Banerjee, S., and Lavie, A. (2005) 'METEOR: An automatic metric for MT evaluation with improved correlation with human judgments', In Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp. 65-72.

2. Chen, X., Mishra, N., Rohaninejad, M., and Abbeel, P. (2018) 'Pixelsnail: An improved autoregressive generative model', In International conference on machine learning, pp. 864-872.

3. Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2018) 'Bert: Pre-training of deep bidirectional transformers for language understanding', arXiv preprint arXiv:1810.04805.

4. 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.

5. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L. and Guo, B. (2022) 'Vector quantized diffusion model for text-to-image synthesis', In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10696-10706.

6. Hong, S., Yang, D., Choi, J., and Lee, H. (2018) 'Inferring semantic layout for hierarchical text-to-image synthesis', In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7986-7994.

7. Kazemi, H., Soleymani, S., Taherkhani, F., Iranmanesh, S., and Nasrabadi, N. (2018) 'Unsupervised image-to-image translation using domain-specific variational information bound', Advances in neural information processing systems, 31.

8. Kocasari, U., Dirik, A., Tiftikci, M., and Yanardag, P. (2022) ’Stylemc: Multi-channel based fast text-guided image generation and manipulation’, In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer vision, pp. 895-904.

9. Li, J., Li, D., Xiong, C., and Hoi, S. (2022) ’Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation’, In International conference on machine learning, pp. 12888-12900.

10. Lin, C. Y. (2004) ’Rouge: A package for automatic evaluation of summaries’, In Text summarization branches out, pp. 74-81.

11. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I. and Chen, M. (2021) ’Glide: Towards photorealistic image generation and editing with text-guided diffusion models’, arXiv preprint arXiv:2112.10741.

12. Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002), ’Bleu: a method for automatic evaluation of machine translation’, In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311-318.

13. Qiao, T., Zhang, J., Xu, D., and Tao, D. (2019). ’Mirrorgan: Learning text-to-image generation by redescription’, In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1505-1514.

14. 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) ’Learning transferable visual models from natural language supervision’, In International conference on machine learning (pp. 8748-8763).

15. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022) Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 1(2), 3.

16. Ramzan, S., Iqbal, M. M., and Kalsum, T. (2022) ’Text-to-Image Generation Using Deep Learning. Engineering Proceedings’, 20(1), 16.

17. Razavi, A., Van den Oord, A., and Vinyals, O. (2019) ’Generating diverse high-fidelity images with vq-vae-2’, Advances in neural information processing systems, 32.

18. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016) ’Generative adversarial text to image synthesis’, In International conference on machine learning, pp. 1060-1069.

19. Xia, W., Yang, Y., Xue, J. H., and Wu, B. (2021) ’Tedigan: Text-guided diverse face image generation and manipulation’, In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2256-2265.

20. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. (2018) ’Attngan: Fine-grained text to image generation with attentional generative adversarial networks’, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1316-1324.

21. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. (2017) ’Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks’, In Proceedings of the IEEE international conference on computer vision, pp. 5907-5915.

22. Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018) ’The unreasonable effectiveness of deep features as a perceptual metric’, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 586-595.

23. Zhang, Z., Xie, Y., and Yang, L. (2018) ’Photographic text-to-image synthesis with a hierarchically-nested adversarial network’, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6199-6208.

24. Zhou, Y., Zhang, R., Chen, C., Li, C., Tensmeyer, C., Yu, T., Gu, J., Xu, J. and Sun, T. (2022) 'Towards language-free training for text-to-image generation', In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17907-17917.

25. Zhu, J. Y., Park, T., Isola, P., and Efros, A. A. (2017) 'Unpaired image-to-image translation using cycle-consistent adversarial networks', In Proceedings of the IEEE international conference on computer vision, pp. 2223-2232.

26. Zinnen, M., Madhu, P., Kosti, R., Bell, P., Maier, A., and Christlein, V. (2022) 'Odor: The icpr2022 odeuropa challenge on olfactory object recognition', In 2022 26th International Conference on Pattern Recognition (ICPR), pp. 4989-4994.