Advancements in Text-to-Image Generation: A Comparative Study of Model Architectures, Datasets, and Performance Metrics

Kaveesh Khattar, Tejas Goyal, KP Dhruv, Aditya Hombal, and Dr. Mamatha H R[1]

Department of Computer Science and Engineering,
PES University, Bengaluru, Karnataka, India
[1]mamathahr@pes.edu

Abstract. Text-to-image creation is a fast expanding topic that has received a lot of attention in the last few years. This study provides a thorough comparative examination of cutting-edge text-to-image generation models, with the goal of providing an overview of their improvements and capabilities. The investigation focuses on the various model architectures, datasets utilised for training and assessment, and performance measures used to assess picture creation quality. Researchers and practitioners may get significant insights into the strengths and shortcomings of different techniques by comparing and contrasting these models, allowing informed decision-making for picking the best text-to-image generating model for certain applications.

Keywords: Image Models · Text-to-Image · Generative AI.

1 Introduction

The goal of this comparison analysis is to identify the architectural designs along with the advantages and disadvantages of various text-to-image creation techniques. We look at the datasets such as as COCO and CUB used by these models for training and assessment in addition to architectural comparisons. The diversity and quantity of these datasets, as well as any preprocessing techniques used, have a significant impact on model performance. Various performance indicators have been used in the field to analyse the quality of produced photographs. Our study incorporates human assessments, user research, and additional qualitative evaluations that the analyzed models employed. This allows for a thorough assessment of each model's visual accuracy and realism. We hope that this comparative analysis will give scholars and practitioners a full grasp of the various text-to-image creation techniques. We give a deep study of the model designs, datasets, and performance indicators in the next sections of this work, as well as a comprehensive comparison analysis. We end by summarising the important findings and outlining potential future research avenues in text-to-image creation.

2 Datasets

YFCC100M (Yahoo Flickr 100 Million Creative Commons): is a huge dataset that contains 100 million Flickr photographs and videos. It is freely distributed under the Creative Commons licence, making it an excellent resource for computer vision and multimedia research. The dataset has been utilised for picture categorization, object identification, and deep learning applications, allowing for breakthroughs in visual perception and analysis. This can be found in [1].

MS-COCO (Microsoft Common Objects in Context): The mscoco benchmark dataset is commonly used for object identification, segmentation, and captioning tasks. It includes almost 200,000 photos that have precise annotations such as object bounding boxes, segmentation masks, and image descriptions. MS COCO has made major contributions to computer vision research by generating cutting-edge models for a variety of visual comprehension problems. This can be found in [2].

CUB dataset (Caltech-UCSD Birds-200-2011): The dataset is commonly utilised in computer vision for fine-grained bird species detection. It includes 200 bird species and 11,788 photos in total. Each image in the collection has bounding boxes, part positions, and characteristics labelled on it. The CUB dataset has been used to create and test algorithms for fine-grained classification, attribute prediction, and other bird species recognition tasks. This can be found in [3].

Oxford-102 Flowers: The Oxford-102 Flowers dataset is a well-known benchmark dataset for fine-grained flower categorization in the field of computer vision. It has 102 flower categories with a total of 8,189 photos. Each photograph is labelled with the flower species it depicts. The dataset contains a wide variety of floral photos, allowing researchers to create and test algorithms for flower detection, classification, and other tasks. This can be found in [4].

KTH Action Recognition: The KTH Action Recognition dataset is a popular benchmark dataset for recognising human actions in videos. It comprises of six separate films of people walking, jogging, running, boxing, hand waving, and clapping. The collection includes numerous sequences for each activity done by many people and captured from various perspectives. This can be found in [5].

UCF Sports: The UCF Sports activity dataset is a well-known benchmark dataset for recognising activity in sports videos. It is a broad collection of videos that capture numerous athletic activities such as basketball, soccer, diving, horseback riding, and more. The dataset provides a diverse variety of action classes captured from various perspectives and under variable settings. This can be found in [6].
The dataset and their corresponding size are highlighted in Table 1.

Table 1. Dataset Information

Dataset Name	Dataset Size
YFCC 100M	15 GB
MS-COCO	25 GB
CUB Dataset	1.1 GB
Oxford-102	0.32 GB
KTH Action Recognition	2.2 GB
UCF Sports	1.7 GB

3 Architecture

3.1 CogView

The tokenizer of CogView, a text-to-image synthesis model, is a vector-quantized variational autoencoder. The model architecture is as follows: The text encoder reads a text caption and generates a sequence of latent codes as mentioned in [7]. The image decoder uses the text encoder's latent codes to generate an image. After training the VQ-VAE to reconstruct pictures, a separate language model is utilised to translate user input text to the VQ-VAE's latent space, where image production happens. The text encoder in this model setup is a unidirectional Transformer that takes a text caption as input and outputs a series of latent codes. A dataset including 1.56 million Chinese text-image pairs makes up the training data. The model is trained with a batch size of 6,144 over a total of 144,000 steps. The model is trained with a combination of 16-bit and 32-bit precision using the Adam optimizer with a learning rate of 3e-4. Interestingly, to guarantee training stability, a method called Precision Bottleneck Relaxation (PB-Relax) is utilized. The architecture is highlighted in Figure 1.

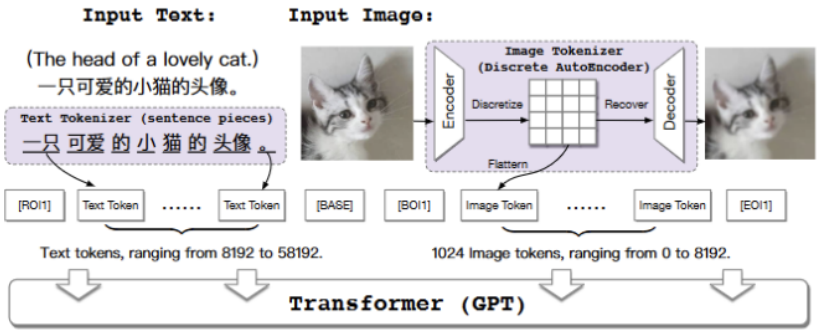

Fig. 1. Cogview architecture.

3.2 dVAE (disentangled Variational Autoencoder)

Disentangled Variational Autoencoder is a text-to-image synthesis model that generates pictures from text descriptions using a disentangled latent space. The following is the model architecture: The text encoder takes a text caption as input and produces a sequence of latent codes. The image decoder takes the latent codes from the text encoder and produces an image. Because the dVAE's latent space is disentangled, the latent codes reflect distinct features of the picture. This enables the model to provide more realistic and varied visuals. To match the produced photos to the written descriptions, the supervised loss is utilised. The reinforcement learning loss is used to encourage the model to create aesthetically attractive pictures.The photos are utilised to train the image decoder, while the text captions are used to train the text encoder. The dVAE model has been demonstrated to be capable of producing realistic visuals from text descriptions. The image decoder is a convolutional neural network that takes the latent codes from the text encoder and produces an image. The latent space of the dVAE is disentangled into three factors of variation: pose, shape, and appearance. The model is trained on a dataset of 100,000 text-image pairs. The model is trained for 100 epochs. The learning rate is decayed using a cosine annealing schedule. The Adam optimizer is used with a learning rate of 3e-4. The model uses a technique called Wasserstein loss to improve the stability of training. The image of the explained architecture is displayed below and is referenced from [8]. The architecture is highlighted in Figure 2.

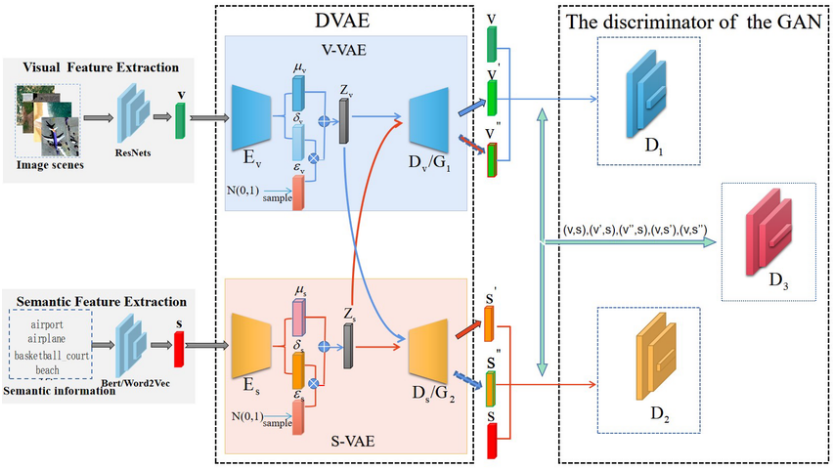

Fig. 2. dvae architecture.

3.3 Multi-Stage AttnGAN

Attention GAN employs attention to regulate the creation of pictures from text descriptions. The model architecture is referenced from [9] and is as follows. The image generator takes the latent codes from the text encoder and produces an image. The image discriminator takes an image as input and produces a probability that the image is real or fake. When producing the picture, the attention mechanism allows the image generator to focus on select sections of the written description.A mix of adversarial and supervised losses is used to train the model.To match the produced photos to the written descriptions, the supervised loss is utilised.The algorithm is trained on a dataset of picture captions and text captions.The pictures are utilised to train the image discriminator, while the text captions are used to train the text encoder and image generator. The text encoder in the model architecture is a bidirectional LSTM, which accepts a text caption as input and outputs a series of latent codes. These latent codes are then used by the image generator, a convolutional neural network, to create an image. Simultaneously, a second convolutional neural network called the image discriminator determines how likely it is that an image is real or fraudulent. The model is trained using a cosine annealing schedule for learning rate decay over a 100 epoch training period. The Adam optimizer is used with a batch size of 64 and a learning rate of 3e-4. The architecture is highlighted in Figure 3.

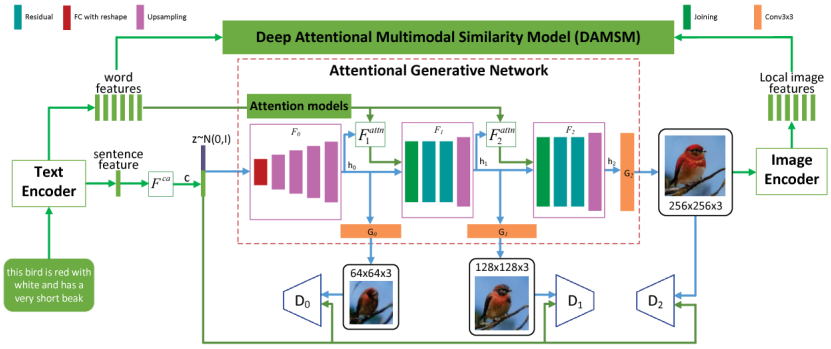

Fig. 3. AttGAN architecture.

3.4 CycleGAN + BERT

In this work, Cycle GAN strategy and the attention GAN technique are combined. With the addition of subtitles and an RNN trained on visual attributes, the model is enhanced. By converting images back into text, the Semantic Text Regeneration and Alignment Module (STREAM) makes sure that the pictures contain the latent data needed to recreate the original captions. Constructed from deep, pre-trained language models, these transformers have shown promise

in numerous natural language processing applications and aid in the enhancement of the CycleGAN architecture.

The image discriminator is a convolutional neural network that takes an image as input and produces a probability that the image is real or fake. The BERT model is a Transformer-based model that takes the text caption as input and produces a sequence of hidden states. The text encoder is a bidirectional LSTM that receives a text caption as input and generates a sequence of latent codes. The image generator model uses a technique called Wasserstein loss to improve the stability of training as decribed by [10]. The model uses a technique called KL annealing to gradually increase the importance of the KL divergence loss during training. The training goes on for 100 epochs and the learning rate is decayed using a cosine annealing schedule, with a size of 64 for each batch. The model uses a technique called label smoothing to regularise the BERT model. The architecture is highlighted in Figure 4.

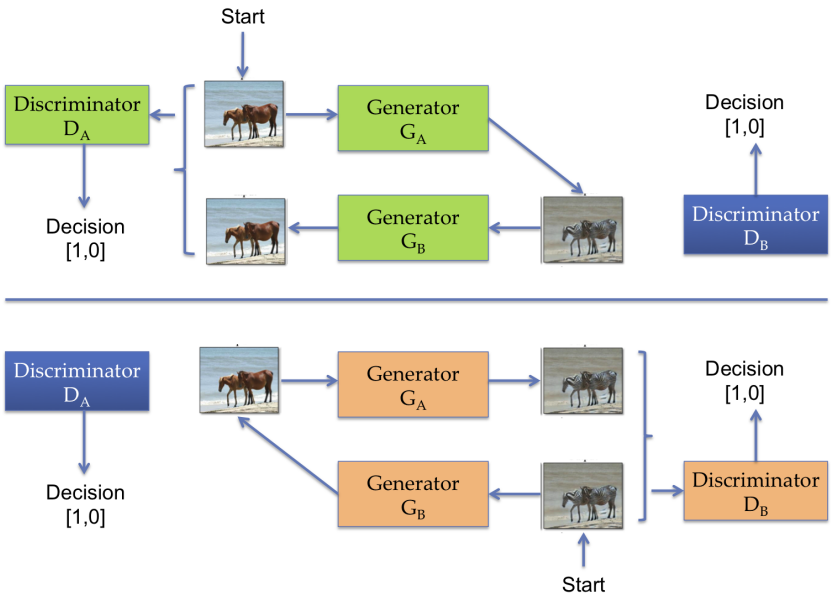

Fig. 4. cycle GAN+BERT architecture.

3.5 DF-GAN

Our architecture is based on the architecture proposed by [12]. A generator, discriminator, and pre-trained text encoder make up DF-GAN. By using a noise vector from a Gaussian distribution and a phrase vector from a text encoder, the generator guarantees visual diversity. To create an image, the noise vector is

processed using convolution layers, fully linked layers, and UP-Blocks (which are made up of DF-GAN, residual, and upsample blocks). DownBlocks are used by the discriminator to transform images into attributes. Semantic coherence and visual realism are assessed using adversarial loss. Semantic vectors are extracted by a bidirectional LSTM text encoder, and AttnGAN's pre-trained model is used directly. Scaling and normalizing are steps in the picture preprocessing process that improve image data processing. By separating fake from real samples, the discriminator helps the generator produce high-quality, semantically consistent images. Tokenizing and vectorizing text is the first step in the comparison study. The generator and discriminator are then frozen. Next, a dataset of actual images is used to train the image encoder, and its weights are updated using a suitable loss function, such as Mean Squared Error. The discriminator and generator are then once more frozen. Vectorized text descriptions are used to train the text encoder, and a suitable loss function is used to update its weights. The generator, discriminator, and encoders are unfrozen in the last stage to produce fictitious images by taking samples from text characteristics and random noise. The architecture is highlighted in Figure 5.

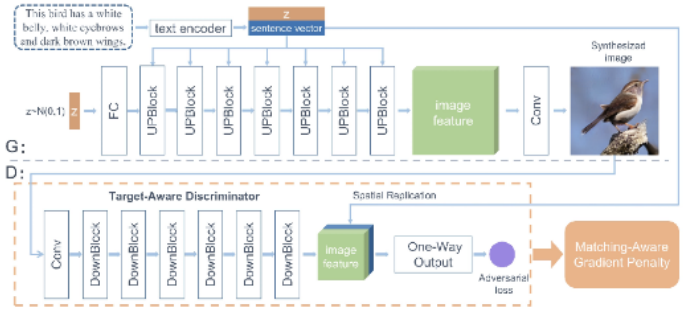

Fig. 5. df-GAN architecture.

3.6 MirrorGAN

MirrorGAN includes a mirror structure that combines text-to-image and image-to-text functions. MirrorGAN's fundamental idea is to use the concept of re-description to train T2I generation. MirrorGAN then reproduces the image's description, successfully matching the created image's underlying semantics with the given text description. The MirrorGAN model is made up of three major components: STEM, GLAM, and STREAM. Each module performs a distinct task in the model's overall operation. The text description is fed into this network, which then encodes it into a fixed-length feature vector. The weights of

the text encoder are updated using an appropriate loss function (such as Cross-Entropy Loss). The generator is trained to produce realistic images from text features and random noise. The discriminator is trained to discern between generated and real images as described in [11]. The weights of the generator and discriminator are updated using suitable loss functions. Update the text encoder's weights using an appropriate loss function after training it with the dataset's text descriptions. The architecture is highlighted in Figure 6.

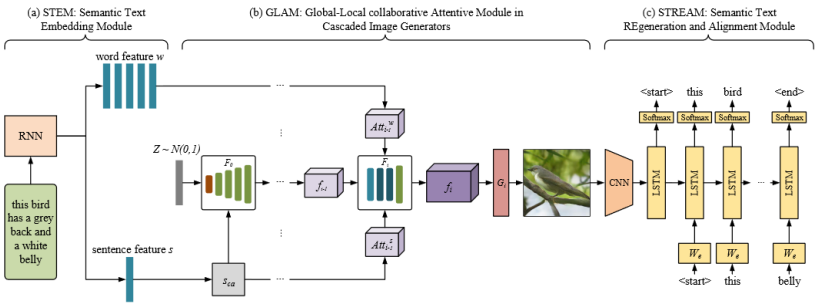

Fig. 6. MirrorGAN architecture.

3.7 VQ-SEG - a modified version of the Variational Quantum VAE

A version of the Variational Quantum VAE (VQVAE) architecture designed for image synthesis and segmentation tasks is called VQ-SEG. The architecture of VQ-SEG consists of using an encoder network, incoming images are first transformed into a representation in a lower-dimensional latent space as mentioned in [16]. In order to capture hierarchical information at many scales, this encoder typically has convolutional layers followed by downsampling techniques. The continuous latent coding is discretely quantized by the VQ layer. It uses a preset codebook that it learned during training to match each latent code to the closest codeword.

The VQ-SEG picture segmentation process has an additional branch which produces segmentation masks, pixel by pixel, that show the class labels of different regions inside the input image. VQ-SEG incorporates an extra decoder network to enable picture segmentation. VQ-SEG integrates segmentation and reconstruction losses during training. The reconstruction loss promotes output images that are similar to the original input images, but the segmentation loss penalises differences between expected segmentation masks and ground truth masks. The architecture is highlighted in Figure 7.

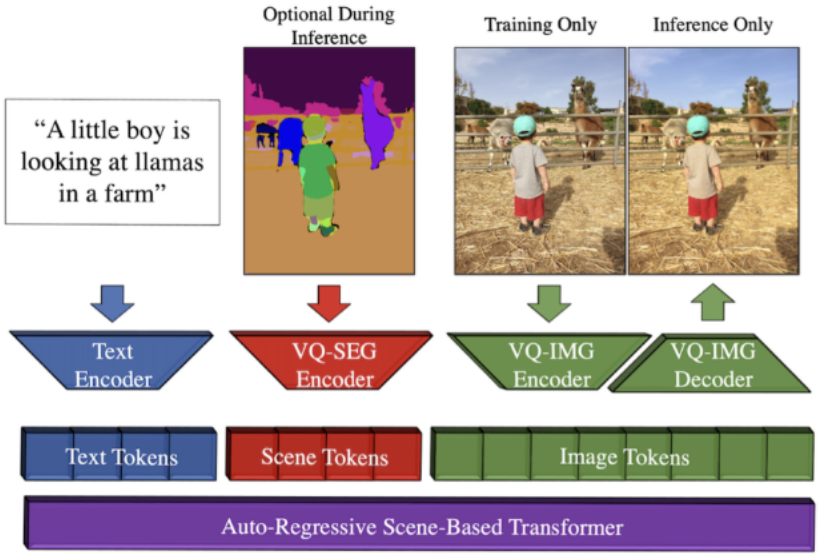

Fig. 7. VQVAE: The architecture of the scene-based method: Images are created from input text with optional layout. Transformer creates tokens that networks then encode and decode.

3.8 StackGAN + Fine-tuned BERT Text Encoding Models

Realistic images are produced from textual descriptions in two steps thanks to the integration of the StackGAN architecture and the BERT model as understod from [14]. Text is converted into low-resolution images in Stage 1, and the improvement of these images into high-resolution ones is the main goal of Stage 2. Efficient interpretation and reflection of textual semantics are made possible by fine-tuning the BERT model using the target dataset. The initial stage of the StackGAN model generates low-resolution images that correspond to color and shape specifications. A discriminator network compares the pictures to actual images. In order to produce high-resolution images that may be compared to actual high-resolution photos by the discriminator, Stage 2 enhances low-resolution images by using them as inputs together with the BERT-based text embedding vector. In order to achieve successful convergence and produce realistic, coherent images that are aligned with text, adversarial training, mini-batches, and carefully selected learning rates are used. The architecture is highlighted in Figure 8.

3.9 DALLE-2

DALL-E 2 generates high-resolution images conditioned on optional text descriptions and CLIP image embeddings by using diffusion models as explained in [15]. Four context tokens created from CLIP embeddings concatenated with the output sequence of the GLIDE text encoder are included, and CLIP embeddings

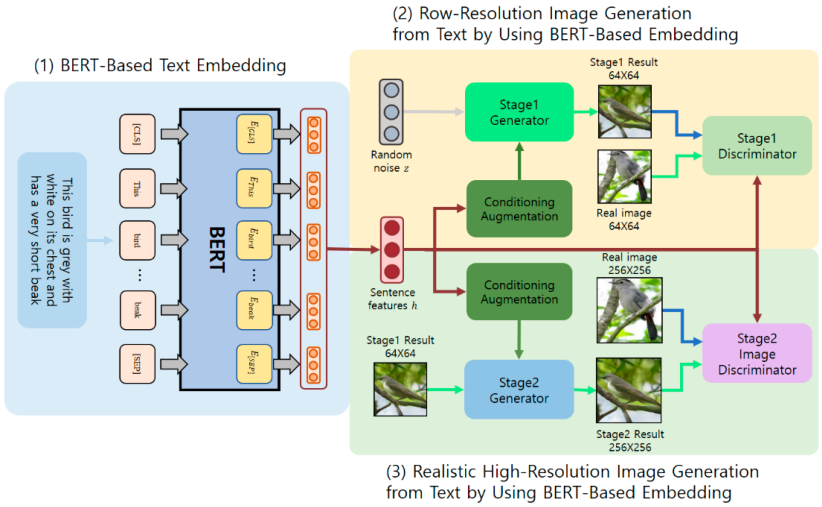

Fig. 8. StackGAN + Fine-tuned BERT architecture.

are projected and added to the current timestep embedding. The text conditioning pathway, however, only marginally improves sample quality; instead, text captions are randomly removed, and occasionally zeroing or applying learnt embeddings is applied during training. There are two trained diffusion upsampler models used: the first focuses on increasing resolution from 64x64 to 256x256, while the second goes one step farther and upsamples to 1024x1024. By using methods like Gaussian blur and variable BSR degradation during conditioning image pollution, robustness is strengthened. With only spatial convolutionals and no attention layers, DALL-E 2 exhibits generalization to higher resolutions during inference, demonstrating enhanced image quality and resilience through upsampling and conditioning. The architecture is highlighted in Figure 9.

3.10 LSTM+GAN

The architecture explained in this section is referred from [13]. Capturing long-range dependencies is an area in which the LSTM (Long Short-Term Memory) paradigm shines. It is made up of cell state (ct), input activation vectors for cells, forget gates (ft), output gates (ot), and input gates (it). The model processes inputs using composite functions built from input and previous hidden states, utilizing logistic sigmoid and hyperbolic tangent functions. Rather than using the estimated gradient calculation of the original LSTM, this research uses backpropagation across time. Every time step, the LSTM unit receives external inputs and updates its hidden state and internal cell state appropriately.
Recurrent Neural Networks (RNNs) with LSTM units are used in the LSTM Autoencoder Model for unsupervised learning. The model is made up of two LSTMs: an encoder and a decoder. It analyzes sequences of input, usually pic-

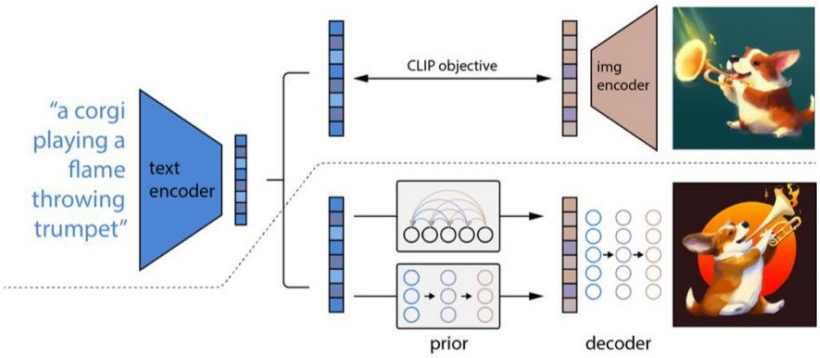

Fig. 9. DALLE 2 architecture.

ture patches or feature sets. After reading the input sequence, the encoder LSTM transfers control to the decoder LSTM. The target sequence, which is frequently the input sequence in reverse order, is predicted by the decoder LSTM. The final output frame feeds into a conditional decoder, bypassing the unconditioned decoder, and the model supports both conditioned and unconditioned decoders. Similar to the Autoencoder Model, the Future Predictor Model differs in its decoder LSTM. The Future Predictor predicts frames that are beyond the input sequence and into the future, in contrast to the Autoencoder, which predicts the target sequence that matches the input. The architecture is highlighted in Figure 10.

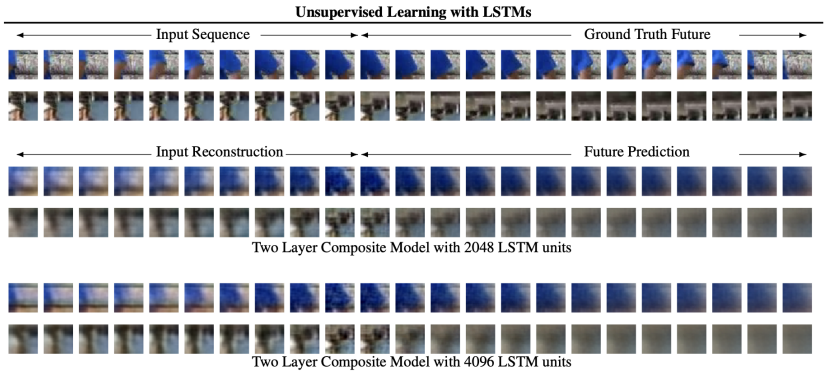

Fig. 10. LSTM+GAN architecture

Table 2. Performance Metrics of Different Models

Model Name	Inception Score	Frechet Inception Distance
CogView	32.2	23.6
Discrete Variational Autoencoder	23.6	30
AttentionGAN	4.58	19
GAN	17	23
LSTM + GAN	16	21
VQ-VAE	18.2	23.6
DF-GAN	5.10	19.32
MirrorGAN	4.54	20
StackGAN + BERT	4.44	37.7
CycleGAN + BERT	6	28

4 Comparative Analysis

This section contains the analysis and findings from our investigation.
The performance of the models explained above are highlighted in Table 2. One of the challenges encountered in the CogView framework is the slow generation process inherent to autoregressive models, as images are being generated token-bytoken. Additionally, blurriness is introduced as a substantial restriction by the use of VQVAE. When discretizing continuous data for use with discrete variational auto-encoders (dVAEs), there are disadvantages including limited expressiveness and possible information loss. Due to a lack of textimage pairs for each category and the inclusion of more abstract captions in the dataset, such as COCO, the multistage Attention-GAN model is constrained.
The image quality of VQ-SEG, a modified VQVAE, could be better, however the improvements also cause losses in perceptual knowledge and awareness of a specific region. Further studies are required to construct complex loss functions and efficiently create images from text with little data for StackGAN+fine-tuned BERT text encoding models.
Finally, Conditional Adversarial Networks (cGAN) still have difficulties in producing visually and semantically cohesive video sequences from textual descriptions.

5 Performance Metrics

5.1 Inception Score(IS)

A statistic called the Inception Score is employed to evaluate the calibre and variety of images produced by GANs. It indicates both image quality and diversity by measuring the difference between the average class probabilities across all generated images and the individual class probabilities.

5.2 Fréchet Inception Distance(FID)

To evaluate efficacy, use the Frenchet Inception Distance, a variant of the Inception Distance measure designed for image production models. This metric mea-

sures the distribution similarity of feature representations from a pre-trained Inceptionv3 model for real and produced images by combining Inception Distance and Frechet Distance. Higher quality image production is indicated by a smaller Frenchet Inception Distance.

5.3 Mean Opinion Score(MOS)

Utilize the Mean Opinion Score (MOS), which is derived from human participants rating images according to perceived fidelity or quality, to assess the quality of generated photos. The MOS is obtained by averaging these assessments; higher numbers denote more visually appealing or realistic images, while lower values denote inferior quality or fidelity.

6 Future Research Directions

Text to image generation models have come a long way, but there are still a number of areas that need more research and development. This section highlights potential future research directions based on the current state of models and identified areas for improvement

Improved Semantic Understanding: Subsequent studies on text-to-image generation ought to improve semantic comprehension by utilizing sophisticated natural language processing methods such as knowledge graphs or pre-trained language models. By using knowledge graphs created from text embeddings, models may be able to produce visuals that are more closely matched with the intended meaning of the input text, potentially improving positional and contextual understanding..

Increased Resolution and Realism: Though current models have come a long way in producing high-quality photographs, resolution and photo-realism may still use some work. Future research could focus on developing techniques to generate images at higher resolutions, allowing for more detailed and visually appealing results. Additionally, exploring advanced loss functions or perceptual similarity metrics could further enhance the realism of generated images, making them indistinguishable from real photographs.

Fine-grained Control and Manipulation: Current text-to-image models often lack fine-grained control over generated images. Future research could investigate methods to enable precise control and manipulation of image attributes, such as object positions, colors, and styles, based on textual input. This could involve exploring novel conditioning techniques or incorporating additional information during the generation process to produce images that align with specific user requirements.

Handling Ambiguity and Multi-modal Outputs: In order to achieve a variety of outputs that might capture many meanings, future research in text-to-image generation should investigate strategies for handling ambiguity in textual descriptions. One could use strategies like adversarial learning, variational techniques, and uncertainty estimation. To potentially reduce ambiguity, one way is to train the image generator using the prompt and the parsed scene graph output.

Incorporating User Feedback and Interactive Generation: Interactive text-to-image generation systems that incorporate user feedback and preferences hold great potential for enhancing user satisfaction and enabling personalized image generation. Future research could focus on developing models that can adapt and refine their generation process based on user interactions, allowing users to provide feedback and guide the image synthesis process in real-time.

7 Conclusion

In conclusion, our comparative study of 11 text-to-image generation models highlighted StackGAN as the top performer.
StackGAN achieved a remarkable inception score of 4.44, indicating its ability to generate visually diverse and high-quality images. Additionally, StackGAN outperformed other models with a FID score of 37.7, demonstrating its superior ability to capture image fidelity and similarity to real images. While other models, such as Cogview and dVAE, showcased strengths in specific areas, they fell short in terms of overall performance compared to StackGAN. The models based on GAN architecture, including Multi-Stage AttnGAN, LSTM+GAN, and CycleGAN+BERT, exhibited promising results in capturing global and local image details, but StackGAN surpassed them in terms of both inception and FID scores.
Our study also emphasized the impact of dataset selection on model performance. The MS-COCO dataset provided a diverse range of images, contributing to the evaluation and comparison of the models. The outcomes demonstrated that StackGAN could make good use of the dataset, producing better image creation results. These results offer insightful information to the text-to-image generating sector and help practitioners and researchers select models that are suitable for their particular requirements.
Future studies can concentrate on developing StackGAN even more and investigating its possible uses in a range of fields, including as virtual reality, multimedia content generation, and computer vision.

References

1. Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K.-H., Poland, D., Borth, D., Li, L.-J.: YFCC100M: The New Data in Multimedia Research. Available at https://dl.acm.org/doi/10.1145/2812802

2. Lin, Microsoft COCO: Common Objects in Context. Available at https://arxiv.org/abs/1405.0312
3. Wah, The Caltech-UCSD Birds-200-2011 Dataset. Available at http://www.vision.caltech.edu/visipedia/CUB-200-2011.html
4. Nilsback, Oxford 102 Dataset; Automated flower classification over a large number of classes. Available at http://www.robots.ox.ac.uk/ vgg/data/flowers/102/
5. Schuldt, KTH Action Recognition; Recognizing human actions: a local SVM approach. Available at http://www.nada.kth.se/cvap/actions/
6. Rodriguez, UCF Sports; Action mach a: a new representation for human action recognition. Available at https://www.crcv.ucf.edu/data/ucf_sports_actions/
7. Ding, M.: CogView: Mastering Text-to-Image Generation via Transformers. arXiv preprint arXiv:2106.13700, 2021
8. Biswas, Biswajit, Ghosh, Swarup Kr, Ghosh, Anupam", "DVAE: Deep Variational Auto-Encoders for Denoising Retinal Fundus Image",2020
9. Xu, Tao and Zhang, Pengchuan and Huang, Qiuyuan and Zhang, Han and Gan, Zhe and Huang, Xiaolei and He, Xiaodong, AttnGAN: Fine-Grained Text to Image Generation With Attentional Generative Adversarial Networks, 2018
10. Tsue, T., Sen, S., Li, J.: Cycle Text-To-Image GAN with BERT. arXiv preprint arXiv:2003.12137 (2020)
11. Qiao, T., Zhang, J., Xu, D., Tao, D.: MirrorGAN: Learning Text-to-image Generation by Redescription. arXiv preprint arXiv:1903.05854 (2019)
12. Tao, M., Tang, H., Wu, F., Jing, X., Bao, B., Xu, C.: DF-GAN: A Simple and Effective Baseline for Text-to-Image Synthesis. arXiv preprint arXiv:2008.05865 (2022)
13. Srivastava, Nitish and Mansimov, Elman and Salakhudinov, Ruslan: Unsupervised Learning of Video Representations using LSTMs, Proceedings of the 32nd International Conference on Machine Learning, 2015
14. Zhang, Han and Xu, Tao and Li, Hongsheng and Zhang, Shaoting and Wang, Xiaogang and Huang, Xiaolei and Metaxas, Dimitris N., StackGAN: Text to Photo-Realistic Image Synthesis With Stacked Generative Adversarial Networks, 2017
15. Ramesh, A., Chu, C., Dhariwal, P., Nichol, A., Chen, M.: Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv preprint arXiv:2204.06125, 2022
16. Make-A-Scene: Scene-Based Text-to-Image Generation with Human Priors, Oran Gafni and Adam Polyak and Oron Ashual and Shelly Sheynin and Devi Parikh and Yaniv Taigman, 2022,