International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Dynamic Visual Creation: Implementing Text to Image Generation

Priyanka S A
Electronics and Communication
Engineering, Global Academy of Technology
Bengaluru, India

Shivani R
Electronics and Communication
Engineering, Global Academy of Technology
Bengaluru, India

Sahana R
Electronics and Communication
Engineering, Global Academy of Technology
Bengaluru, India

Abstract—Generative Adversarial Network (GANs) has become one of the most interesting ideas in the last years in Machine Learning. Generative Adversarial Network is a very exciting area and that's why researchers are so excited about building generative models as they are set to vary what machines can do for humans. This paper proposes the generation of realistic images according to their semantics based on text description using a Knowledge Graph alongside Knowledge Guided Generative Adversarial Network (KG-GAN) that comes with the embeddings generated from the Knowledge Graph (KG) into GAN. This project focuses on the development of a dynamic text-to-image generation system that translates textual descriptions into corresponding visual representations. The system leverages Open-AI's language models for processing and understanding user-input text, generating descriptive prompts which are then passed to a Replipt-based model for image synthesis.

The back-end of the system is powered by Flask, which serves as a lightweight framework to handle user requests, manage communication between the language and image generation models, and deliver the resulting images. The integration of these technologies enables real-time creation of unique images based on textual input, facilitating a seamless user experience. This work demonstrates the potential of combining advanced AI techniques—natural language processing and image generation—into a cohesive platform capable of generating high-quality, contextually relevant visuals from textual descriptions.

The project emphasizes the need for efficient communication between diverse AI models and frameworks, ensuring scalability and adaptability. Additionally, the system allows for further exploration of model training, enhancing image fidelity and accuracy with iterative improvements.

Keywords*— *Knowledge Graph, GCN, GAN, NLP.

IV INTRODUCTION

Text to Image synthesis has become one of the foremost challenging tasks in machine learning. Text to Image synthesis is the method of generating high-quality realistic images from the given text descriptions. Synthesizing images from text descriptions has been a well-liked topic among natural language processing and computer vision. The core challenge is to get images that are visually realistic and semantically sensible high-quality images matching the text descriptions. Synthesis is the process of retrieving relevant information that supports a given input. The input is especially the text and therefore the relevant information to be generated are mainly the image features also referred to as global or local features. The global features represent the visual contents of a picture including their color, texture, and shape information whereas local features represent a picture pattern that differs from their immediate surroundings that are the global features.

Generative Adversarial Networks (GANs) are introduced to get visually realistic and semantically sensible high-quality images supported text descriptions. The recent Generative Adversarial The phrase "A picture is worth a thousand words" underscores the profound power of visual communication. In the context of artificial intelligence (AI), this adage has become a guiding principle for the development of systems capable of generating realistic and meaningful images from textual descriptions. The **text-to-image (T2I)** generation task, which involves creating high-quality images based on natural language prompts, has emerged as one of the most exciting challenges in the field of computer vision and natural language processing. Achieving this capability represents a crucial step toward developing artificial intelligence systems that not only understand language but also can generate corresponding visual representations, bringing us closer to human-like or even general-purpose AI. The task of T2I generation presents multiple challenges, the most significant of which is ensuring that the generated images are both **semantically accurate** and **visually coherent** with the provided text. This challenge requires the integration of **natural language understanding** and **image generation** technologies, which has led to the development of various deep learning models designed specifically for this task. Early work in this field primarily relied on **Generative Adversarial Networks (GANs)**, a framework that trains a generator to create images and a discriminator to evaluate their realism. While GANs showed promise in generating compelling images, they often struggled with training instability and limited control over image content [1]. These issues were particularly evident in early GAN-based models like **StackGAN**, which faced difficulties in scaling to higher resolution and producing images with fine-grained detail [2].

II. RELATED WORKS

Numerous methodologies are implemented for text-to-image synthesis based on Generative Adversarial Network in the literature so far. The generation of high-quality image pixels from the given words and sophisticated descriptions is one of the foremost difficult tasks within the field of machine learning. Generative Adversarial Networks are a recent innovation that has been popular in the field of text-to-image synthesis.

Yali Cai et al. [1] proposed a Dual Attentional Generative Adversarial Network (DualAttn-GAN) to emphasize local details and global structures by attending to related features from the given input text description. The model comprises dual modules namely the textual and visual attention module. The textual attention module is used to explore the fine interaction between vision and language. On the other hand, the visual attention module is meant to enhance the standard of representations. The textual attention module uses the channel and spatial axes to capture the global structures. Channel Attention Module mine the relationship information between channel features of the images. The Spatial Attention Module enhances local feature representations by encoding rich contextual information. An attention embedding module is applied to merge multi-path features and to enhance the representation of the image features. The proposed method is experimented on two datasets mainly the CUB and Oxford - 102 datasets and showed a high performance during the evaluation.

Rintaro Yanagi et al. [2] proposed a retrieval framework, ''Query is GAN'', which is used to enhance the scene retrieval performance. The main idea behind the tactic is to make use of images generated from the GAN network and use it as queries for scene retrieval purposes. The framework consists of two phases mainly the query image generation and the estimation of relevant scenes. For the generation of query images, hierarchical AttnGAN is implemented. The scene retrieval is performed by using the generated query images from the primary phase query image generation. They experimented with the work on a true video dataset and proved efficient in scene retrieval

tasks but the image generated isn't visually pleasing.
Jiancheng Ni et al. [3] proposed an Instance Mask Embedding and Attribute-Adaptive Generative Adversarial Network (IMEAA-GAN) for Text-to-Image Synthesis for the generation of high-resolution images. The Instance Mask Embedding and Attribute Adaptive-Generative Adversarial Network (IMEAA-GAN) takes advantage of the multistage text-to-image generation strategy. The method experiments on two datasets, mainly the MS - COCO and Caltech UCSD Birds-200-2011and it has shown significant improvements in generating complex images while preserving its local features.

Annie Tian and Lu Lu [4] proposed an Attentional Generative Adversarial Networks (rdAttnGAN) with Representativeness and Diversity for Generating Text-to-realistic Image. The generative adversarial network generates images by training the multi-pair generators and discriminators. The proposed model consists of an Attentional Generative Adversarial Network and a Deep Attentional Multimodal Similarity Model (DAMSM). A Representation-Diversity Reward Model (RD) is used to optimize key areas of the image and thus promising the range of the generated images. The method is evaluated based on the COCO and CUB dataset and has performed well in maintaining the representativeness and the diversity of the image.
Han Zhang et al. [5] proposed StackGAN++ aims at generating high-resolution photo-realistic images. The Stage-I GAN model draws the primitive shape and color instance on the given input text, leading to a low-resolution image. Low-resolution images generated by Stage-I GAN are usually distorted and aren't visually good. The Stage-II GAN is used to generate a high-resolution image and is proposed for both conditional and unconditional generative tasks. The method is experimented on the datasets CUB, Oxford-102, and COCO for conditional image generation task and uses the datasets bedroom and church subsets of LSUN, and a dog-breed 2 and a cat-breed 3 subsets of ImageNet for unconditional image generation. The experiment has proved that the method is significant for both image generation tasks.

III. PROPOSED SYSTEM DESIGN

This section discusses various modules within the proposed model and outlines their basic process flow. The system architecture of the text to image generation model is depicted in figure 1.

The main objective of the proposed system is to get realistic images from the relevant text. The system architecture of the whole work is split mainly into two modules namely the Knowledge graph and Knowledge-Guided Generative Adversarial Network. The general input to the framework is the text description and therefore the output is the image matching the text description.

The primary part of the work is known as a knowledge graph; generated supported text description using Natural Processing Techniques. Input to the framework is the text description and therefore the Knowledge graph representation is that of the output of the section. The second part of the work is known as Knowledge-Guided Generative Adversarial Network. This takes the output of the primary section i.e. knowledge graph as its input and with the usage of GCN, it generates the embeddings of the Knowledge graph which are given to the GAN model that outputs realistic images that are rich in visual semantics. This part consists of training the networks supporting a given dataset. The model uses the dataset that mixes both images and their segmentation to perform the task during a learning-based fashion. This may help in differentiating whether the pictures generated are real or fake. The detailed design of every module is described within the coming sections.

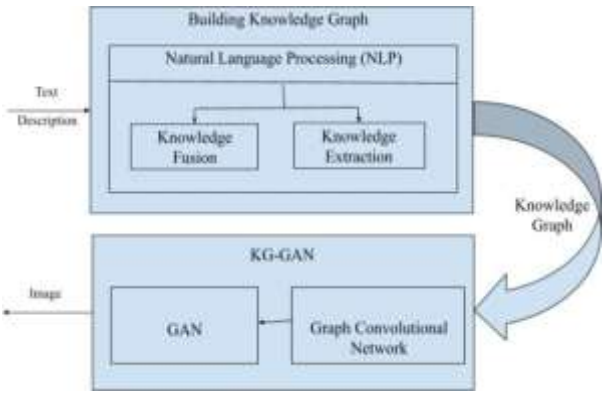


Fig 1 : Architecture of the proposed system

A. KNOWLEDGE GRAPH

Knowledge graph [9] is a powerful data science technique to mine information from text. The knowledge graph (KG) represents a set of data domains of entities like real-world objects and events, or abstract concepts. The knowledge graphs are represented by nodes and edges almost like the traditional graphs. The text analysis technology makes use of a knowledge graph because it provides a background of data, human concepts, and an awareness of the entity to form more accurate interpretations from text. The description has formal semantics that permits machines and humans to interpret efficiently. Facts generated from the text are often used to enrich the generated knowledge graph which makes it more valuable for visualization, reporting, and analysis.

This section mainly discusses the workflow for building a knowledge graph. The input given for the knowledge graph generation is especially the text description. To create a knowledge graph the machine should understand Natural Language Processing techniques. There are mainly two parts in knowledge graph construction and that is knowledge extraction and knowledge fusion [8]. Detailed functionalities of the generation of a knowledge graph are explained below and their detailed design is shown in figure 2 [13].

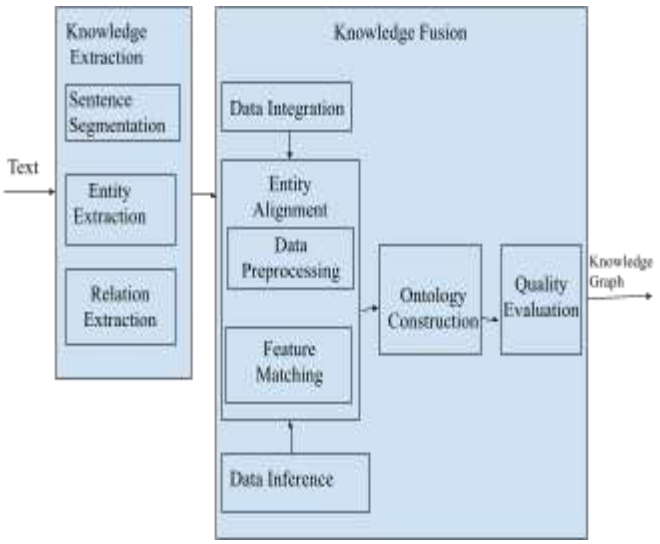


Fig 2 : Building Knowledge Graph from text description [13]

1) KNOWLEDGE EXTRACTION

The first module in building a knowledge graph is named a knowledge extraction module containing mainly three steps and they are sentence segmentation, entity extraction, and relation extraction.

Sentence segmentation is the initial step in building a knowledge graph. It splits the whole document or text article into sentences. The tactic makes use of another process called tokenization to spot the essential units within the sentence, mainly the tokens. The sentence segmentation method mainly splits the given text description into small sentences for understanding the text if the text contains multi-words or quite one sentence. Then it shortlists the sentences that contain just one subject and object.

The entity extraction is especially done to seek out the subject and object from the shortlisted sentence extracted during the sentence segmentation. The entities are mainly the nouns, pronouns, adjectives, verbs, adverbs, prepositions, interjection, conjunction, compound words. For identifying these entities it makes use of parts of speech tags, dependency parsing, entity recognition to extract subject and object along with its modifiers and it also extracts the punctuation marks among the input. The subject and object identified during the entity extraction are wont to represent as nodes of the knowledge graph.

The relation extraction method is employed to spot the connection between the subject and objects of a shortlisted sentence from the sentence segmentation phase. For this, it makes use of dependency parsing which is the process of extracting the grammatical structure of the sentence. The relation is represented as an edge between the subject and object pairs.

Thus the output from the knowledge extraction module is especially the subject and object pairs alongside their relationship. This extracted information is passed to subsequent sections of the knowledge graph building which is the knowledge fusion.

2) KNOWLEDGE FUSION

The output from the knowledge extraction module, mainly the subject and object pairs along with their relationship is input to the subsequent module i.e. the knowledge fusion module containing mainly the given steps: data integration, data inference, entity alignment, ontology construction, and quality evaluation.

Data integration is employed to represent the extracted information supported by a standard described by the Resource Description framework [7]. Data integration involves combining data from different sources and providing a unified view of the data.

Data inference is especially wont to describe the data and to form predictions from the given data to enable a more accurate interpretation of the text. Data inference improves the worth of knowledge over time which ultimately leads to more accurate data. This information is fed to the entity alignment section.

The entity alignment is the process of identifying entities from two knowledge graphs that represent the real-world entities. The entity alignment section contains mainly the preprocessing and therefore the feature matching stage. The preprocessing is completed to avoid any noise or redundant information from the extracted data. The feature matching is employed to spot the single objects instead of the multiple objects. It's mainly used to integrate multiple information derived from the textual descriptions.

Ontologies are frameworks used to represent knowledge across a domain. Its ability to explain interconnection and relation makes it the base for representing high quality, informational and coherent data. It represents the extracted data within the form of subject-predicate-object (SPO) triples. It also identifies the most verb within the sentence i.e. the root. After identifying the root, it then adds the already extracted subject-predicate-object triples to the root node. These subjects and objects extracted are represented as nodes and therefore the relationship between them is represented as edges. This representation of nodes and edges is termed a knowledge graph.

Quality evaluation is the process of evaluating the accuracy, quality, or standards of data. Here, the quality evaluation mainly focuses on evaluating the standard of the graph representation from ontology constructions supported text extractions and text descriptions. It checks whether the knowledge graph produced satisfies the standard and fits the input description. The output of the knowledge fusion module is the knowledge graph for the given text description. The generated knowledge graphs are often visualized by using the visualizing techniques and functionalities.

B. KNOWLEDGE-GUIDED GENERATIVE ADVERSARIAL NETWORK

Knowledge-Guided Generative Adversarial Network (KG-GAN) [6] is a machine learning framework with Variational Autoencoders. But in this proposed model a Graph Convolutional Network (GCN) is employed rather than variational autoencoders for more accuracy. Therefore the proposed model contains mainly two parts, that's GCN and GAN. The detailed workflow and functionalities of the section and its detailed framework are shown in figure 3.

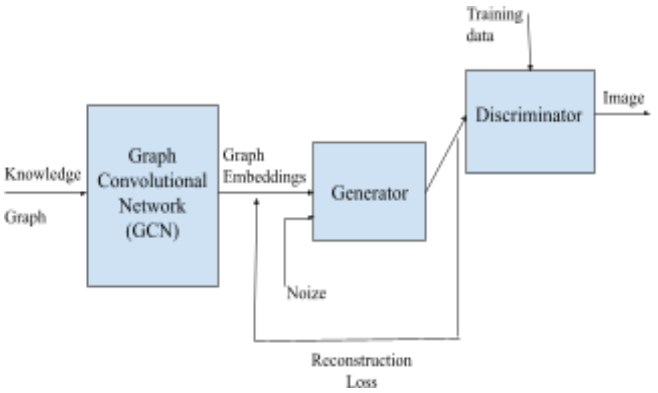


Fig 3 : Knowledge-Guided Generative Adversarial Network

1) GRAPH CONVOLUTIONAL NETWORKS (GCN)

The Knowledge Graph which is generated is fed into a Knowledge Guided Generative Adversarial Network for converting the knowledge graph into a sensible image of a given text description. For this, it makes use of GCN to get the vector representation or embeddings of the knowledge graph.

For the generation of the knowledge graph embeddings, the GCN makes use of three layers: The input layer, the hidden layer, and an output layer. There's some information that has got to be retrieved from the generated knowledge graph before feeding it to the GCN. This information is especially extracting the dependency and representations of the knowledge graph. The step-by-step extraction of dependencies is illustrated as follows: First, generate the adjacency matrix and the identity matrix of the generated knowledge graph. Then represent each node as the sum of its neighbor's features. After this, build a self-loop for every node within the graph by adding the adjacency and identity matrix together, then normalize the obtained representation. Initialize the weights randomly for the graph. Here we use a Rectified function (ReLU) as an activation function.

After obtaining the above-specified representations define the GCN function and its layers. The primary layer i.e. the input layer takes the self-loop setup, normalized features, representation in categorical form, and randomly initialized weights. The output from this layer is fed into the subsequent layer which takes an equivalent input of the primary layer but with a small variation that rather than giving a categorical representation of the node we input the output from the last layer. The output from the last layer is the lower vector or embedding representation of the generated knowledge graph within the form of a matrix. Then transform the matrix into the categorical variable which

describes the feature vector of the knowledge graph. The output of the GCN output layer is the embeddings that are often said as the low dimensional vector representations of the generated knowledge graph. Since the GCN isn't trained using any dataset or real files; whenever the GCN is run it shows different visualization when plotted through the python visualization utility.

GENERATIVE ADVERSARIAL NETWORK (GAN)

The generated vector representation from the Graph Convolutional Networks is given input to the GAN model [10] [11]. The GAN model contains mainly two networks; the generator and therefore the discriminator. The generator network as the name describes is especially wont to generate images supporting the given input and therefore the discriminator is employed to gauge the generator-generated output. The main aim behind the generator is to fool the discriminator and to mark the generator-generated image as real.

In this proposed model, the generator contains mainly five layers: Input layer, Dense layer consisting of three hidden layers, and an output layer. The generator network takes the input, the embedding of the graph, and also a random noize. The noize is generated randomly for every iteration supporting the discriminator's evaluation. The noize is additionally inputted into the generator within the type of a vector representation. The generator takes these both inputs together through the input layer and it concatenates the two inputs using an embedding layer to get a one-hot vector. The concatenation is completed before the dense layer and serves as further input to the generator model. Within the dense layer, the stacking of hidden layers occurs consisting of Batch normalization, Activation function ReLU, and a Conv2D transpose is employed to form changes in weight, reshaping, and normalizing the input to get fake images. These functionalities happening in hidden layers of the dense layer leads to a resized image. The output from the dense layer is given as input to the output layer. The output layer contains the activation function and here the activation is going to be a sigmoid for the GAN model because the sigmoid converges easily. The images generated from the output layer are going to support the given inputs and mostly it might be fake to fool the generator.

The working of the discriminator is more like the generator model with slight variations. It also has five layers: The input layer, dense layer with three hidden layers, and an output layer. The discriminator takes input the vector representation of the graph and also the dataset containing real images and segmentations. The discriminator also takes the generated images of the generator as input to the discriminator. The discriminator takes all the input together within the input layer and concatenates it to get one hot matrix within the embedding layer. The concatenation is completed before the dense layer as in the GAN generator model. Within the dense layer, a stacking of hidden layers consisting of the Activation function ReLU and a Conv2D transpose and flattening is employed to differentiate between the real and therefore the fake images. The image resizing and also the reshaping of the image also occur in these layers. As within the generator, the discriminator doesn't use batch normalization because it doesn't converge fastly. The output from the dense layer is given as input to the output layer and it's the output layer containing the sigmoid function which identifies whether the generated image from the generator is real or fake. This evaluation of the image is fake or real is completed by training the GAN model.

In the training phase, the discriminator is trained with the real inputs which are the dataset getting used. The real and fake values are being determined by a cost function. The cost function used is the Binary Cross Entropy which predicts the similarity between the generated image and the given input. The optimizer utilized in training the model is the Adam optimizer. After training the discriminator model with the real data, it becomes efficient to label if the generated image is real or not based on the cost function and thus the discriminator starts evaluating the generated image from the real images. After evaluating the generated image the discriminator back propagates the value of the image generated from the cost function to both the generator and the discriminator. It represents real images with a value of 1 and fake images with a value of 0. When it's said that the real images are represented by the value of 1; it truly means it'll assign a value between 0 and 1 rather than assigning 1 to train the GAN better and supply a more efficient result. The value being real is fed towards the generator which helps it to get new images with more accuracy. The value of the image being fake is given back to the discriminator to differentiate whether the image generated is real or fake. This process or iteration continues until the generator can generate a picture that's relevant to the input given i.e. the vector representation or can also be said to be almost like the text description thus making the discriminator label it as a true image of the given input. This training with backpropagation is completed to enhance both the generator and the discriminator together which suggests that none of the networks should remain superior over the other.

Thus, the proposed model generates the image that seems to be realistic and visually semantic matching the text description. The model can extract all the global and local features of an image making it more visually pleasing.

C. DATASET

The dataset collected is the Caltech-UCSD Birds 200-2011 (CUB-200-2011) [12] dataset. The dataset contains images and their segmentation of the birds used to classify based on the given features. It is one of the foremost used datasets for fine-grained visual classification tasks. The dataset contains 200 classifications of birds with 11,788 images. The dataset also includes 15 part locations, 312 binary attributes, and 1 bounding box as annotations per image. It contains ten single image descriptions for every image within the dataset. To collect these natural language descriptions the Amazon Mechanical Turk (AMT) platform is used. The dataset is split into training and testing sets for evaluating the precision. The training set of the dataset contains 5,994 images and the testing set contains 5,794 images.

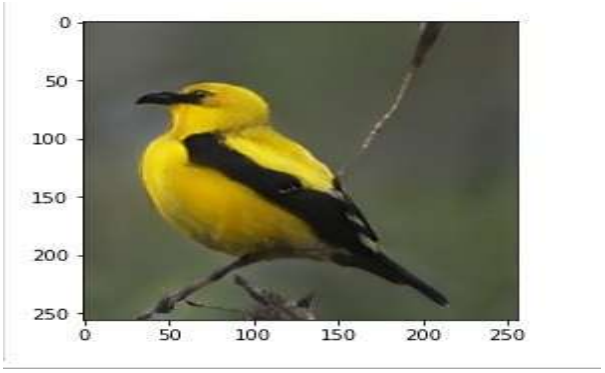
IV RESULTS

The experimental study of the proposed model on the Caltech-UCSD Birds 200-2011 dataset performs well with more accuracy than the other Text to Image GAN model. The generation of the image from text based on a knowledge graph and its embedding using the Knowledge - Guided generative Adversarial Network has proven to be quite a better and efficient way of image generation.

As the model utilizes the knowledge graph generation, it's ready to take text without having any word limit or line limit. Thus helping the extraction of the relevant features and taking away the unwanted to get the image with all its given requirements. The test results of the given proposed system are given below in figure 4 representing the generated knowledge graph and image for the given input descriptions. The feature representations of the knowledge graph referred to as the embeddings or the low dimensional vectors are shown in figure 5. The proposed model is evaluated based on the Inception score and is compared along with the other established models generated in the recent years. The evaluation of the model based on the Inception score has proven that the method is efficient in terms of the other conventional model. The table 1 depicts the performance comparison of the proposed model with the other conventional model. Bird is small with yellow in color. The Bird has a black crown. It has a short black pointed beak.

(a)

(b)



(c)

Fig 4 : Result of the image generated by the proposed model. (a) Text description as input to the model. (b) Knowledge Graph generated from the text description. (c) Image Generated

Model	***Inception Score***
Dual-Attn GAN [1]	4.59±0.07
IMEAA-GAN [3]	4.75±0.07
rdAttn GAN [4]	4.39±0.03
StackGAN++ [5]	4.04± 0. 05
Our Proposed Model	4.84±0.04

Table 1 : Performance comparison

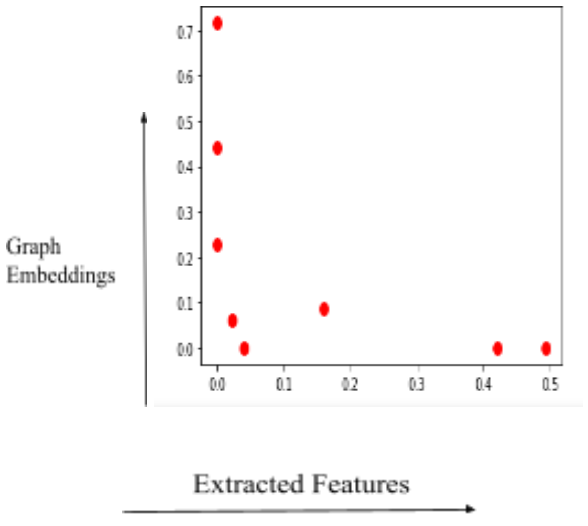


Fig 5: Scatter plot representing the embeddings of the knowledge graph

V. CONCLUSIONS

Text to Image synthesis as said is one of the most recent innovations in computer vision. It describes the method of generating images based on the relevant words and text descriptions. Knowledge-Guided Generative Adversarial Text to Image synthesis as said is one among the foremost recent innovations in computer vision. It describes the tactic of generating images from the relevant words and text descriptions. Knowledge-Guided Generative Adversarial Network is employed in Text-to-Image synthesis to get high-quality images that are visually realistic and semantically sensible matching the given text descriptions. The proposed method also uses Natural Language processing techniques to create a knowledge graph representation and Graph Convolutional Networks to get the embedding representations of the knowledge graph. This is then fed into the GAN model to get realistic images for the given input, mainly the text description. The work has experimented on a bird dataset showing high accuracy in comparison to the traditional models. As a future work, generating full-length video from images generated from knowledge graphs and text description are later proposed.

REFERENCES

[1]. Yali Cai, Xiaoru Wang, Zhihong Yu, Fu Li, Peirong Xu, Yueli Li, and Lixian Li, "*Dualattn-GAN : Text to Image Synthesis with Dual Attentional Generative Adversarial Network*", DOI 10.1109/Access.2019.2958864.

[2]. Rintaro Yanagi, Ren Togo, Takahiro Ogawa, and Miki Haseyama, "*Query is GAN : Scene Retrieval With Attentional Text-to-Image Generative Adversarial Network*",DOI 10.1109/Access.2019.2947409.

[3]. Jiancheng Ni, Susu Zhang, Zili Zhou, Jie Hou, and Deng Gap. "*Instance Mask Embedding and Attribute-Adaptive Generative Adversarial Network for Text-to-Image Synthesis*", DOI 10.1109/Access.2020.2975841.

[4]. Annie Tian and Lu Lu. " *Attentional Generative Adversarial Networks with Representativeness and Diversity for Generating Text-to-realistic Image*", DOI 10.1109/Access.2020.2964946.

[5]. Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris N Metaxas. "*StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks*", DOI 10.1109/TPAMI.2018.2856256.

[6]. Yuxia Geng, Jiaoyan Chen, Zhuo Chen, Zhiquan Ye, Zonggang Yuan, Yantao Jia, Huajun Chen. "*Generative Adversarial Zero-shot Learning via Knowledge Graphs*", 2020.

[7]. E. Miller, "*An introduction to the resource description framework*", Bull. Amer. Soc. Inf. Sci. Technol., vol. 25, no. 1, pp. 15-19, 1998.

[8] Q. Wang, Z. Mao, B. Wang and L. Guo, "*Knowledge graph embedding: A survey of approaches and applications*", IEEE Trans. Knowl. Data Eng., vol. 29, no. 12, pp. 2724-2743, Dec. 2017.

[9] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, "*Knowledge graphs*", arXiv:2003.02320, 2020.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "*Generative Adversarial Nets*", arXiv:1406.2661, 2014.

[11] Mehdi Mirza, Simon Osindero, "*Conditional Generative Adversarial Nets*", arXiv:1411.1784, 2014.

[12] Wah C., Branson S., Welinder P., Perona P., Belongie S, "*The Caltech-UCSD Birds-200-2011 Dataset.*" Computation & Neural Systems Technical Report, CNS- TR-2011-001.

[13] Zhanfang Zhao, Sung-Kook Han, In-Mi So, "*Architecture of Knowledge Graph Construction Techniques*", International Journal of Pure and Applied Mathematics, Volume 118 No. 19 2018.