Generating Synthetic Images from Text using RNN and CNN

POLAVARAPU VINEEL KUMAR,
PG scholar, Department of MCA, DNR college, Bhimavaram, Andhra Pradesh.
CH.JEEVAN BABU
(Assistant Professor), Master of Computer Applications, DNR college, Bhimavaram, Andhra Pradesh.

Abstract: A method called content-to-picture creation aims to generate lifelike images that match text descriptions. These visuals find use in tasks like photo editing. Advanced neural networks like GANs have shown promise in this field. Key considerations include making the images look real and ensuring they match the provided text accurately. Despite recent progress, achieving both realism and content consistency remains challenging. To tackle this, a new model called Bridge GAN is introduced, which creates a bridge between text and images. By combining Bridge GAN with a char CNN – RNN model, the system produces images with high content consistency, surpassing previous methods. In these paper we used we have used FLICKER TEXT and IMAGE dataset. Proposed model performs better than state of art techniques.

Keywords: GAN, CNN, RNN, BI-LSTM.

I. INTRODUCTION

This project aims to create synthetic images from textual descriptions using a combination of RNN and CNN. RNNs are used to process sequential data from text descriptions, capturing contextual information, while CNNs are employed to extract features from images. By integrating both RNN and CNN architectures, we aim to generate visually coherent images that correspond to the provided textual descriptions.

Generating synthetic images from textual descriptions has emerged as an intriguing area of research at the intersection of computer vision and NLP. This project delves into the exciting realm of creating visual content directly from textual input using advanced machine learning techniques. Leveraging the power of RNN and CNN we aim to develop a robust system capable of translating textual descriptions into corresponding realistic images.

The objective of this project is to develop a system capable of generating synthetic images from textual descriptions using a CNN and RNN,

- Implementing an architecture that combines RNN and CNN algorithms to process textual descriptions and extract image features effectively.
- Training the model to learn the mapping between text and corresponding image features, enabling it to generate synthetic images based on textual input.
- Ensuring that the generated images are visually coherent and semantically relevant to the provided textual descriptions.
- Evaluating the performance of the model through quantitative metrics and qualitative assessments to measure the accuracy and quality of the generated images.

Generating synthetic images from text using RNN and CNN involves leveraging both RNN and CNN architectures to translate textual descriptions into corresponding images. RNNs are adept at processing sequential data, making them suitable for handling text inputs, while CNNs excel at extracting features from images. By combining these two architectures, the model can effectively learn the relationship between textual descriptions and visual representations, ultimately producing synthetic images that align with the provided text.

In this project as per your instructions we have utilized CNN and Bi-LSTM algorithms to generate images from text. CNN layers utilized to extract features from images and then Bi-LSTM utilized to extract features from text and then both layers will get trained using sigmoid activation function. BI-LSTM will take text as input and then feed to CNN layer which is responsible to generate images as per text features.

Normally GAN algorithms consider best for text to image generation but we are using CNN and RNN based algorithm so its predicted image will be wrong for few questions.

The ability to generate synthetic images from text holds immense potential across various domains, including content creation, virtual environments, and visual storytelling. By harnessing the synergy between RNNs, designed to comprehend sequential data like textual descriptions, and CNNs, adept at extracting features from images, we strive to bridge the semantic gap between language and vision.

This project not only seeks to advance the state-of-the-art in text-to-image synthesis but also explores the broader implications of such technology in enhancing human-computer interaction and multimedia content generation. Through meticulous experimentation and innovative model architectures, we aim to push the boundaries of what is possible in generating synthetic images from textual descriptions, opening up new avenues for creativity and expression in the digital realm.

II. LITERATURE SURVEY

Generative adversarial networks have made it more and more common to create images from text descriptions in recent years. Even if visual realism and diversity have advanced significantly, there are still issues to be resolved, like producing high-resolution photos with numerous objects and developing reliable evaluation measures that are in line with human judgement. This paper presents a taxonomy based on supervision levels and reviews the state of the art in adversarial text-to-image synthesis models over the last five years. We also address current limitations in evaluation methods and suggest areas for future research, including dataset improvement and architectural enhancements, aiming to propel the field forward. [1]

Recent advances in text-to-image synthesis have been made possible by Generative Adversarial Networks (GAN). However, a significant amount of labour- and time-intensive human-labeled image-text data is typically needed for GAN model training. In this research, we present a novel method to train an unsupervised text-to-image synthesis model without requiring human-labeled input. We construct fake image-text pairs by bridging independent sets of words and images using visual concepts. In order to ensure that the generated images faithfully portray actual local visual concepts while suppressing noise concepts from the pseudo sentences, we offer a novel visual concept discrimination loss to train both the discriminator and the generator. Our experimental results show that, in comparison to certain existing models trained using supervised methods, our unsupervised training strategy effectively creates high-quality images corresponding to provided phrases [2]

Text-to-image synthesis, the process of generating images from text descriptions, poses a significant challenge in both NLP and Computer Vision fields. While humans effortlessly visualize images from textual descriptions, achieving the same with machines is complex. This technique holds promise in various domains such as medical imaging and fashion designing, where obtaining specific images is difficult. Generative Adversarial Networks (GANs) have revolutionized text-to-image synthesis, leading to significant advancements in the field. This paper provides an overview of recent models in text-to-image synthesis, discussing evaluation metrics, datasets, challenges, and future research directions. [3]

Text-to-image synthesis converts text descriptions into corresponding images, widely used in graphic design and image editing. Nevertheless, existing ones suffer from overconfidence and training instability problems, and they have trouble producing visually realistic images. This study suggests a self-supervised strategy with improvements such as feature matching, L1 distance loss, self-supervised learning, and one-sided label smoothing to address these issues. Self-supervised learning improves discriminator categorization by enhancing visual variability. The generator is encouraged to produce visually similar images to actual ones via feature matching and L1 distance. Correct discriminator classifications are penalised by one-sided label smoothing in order to reduce overconfidence and improve training stability. The suggested strategy outperforms current methods in terms of inception score and realism, producing images with better content diversity, semantic consistency, and realism when evaluated on the Oxford-102 and CUB datasets. [4]

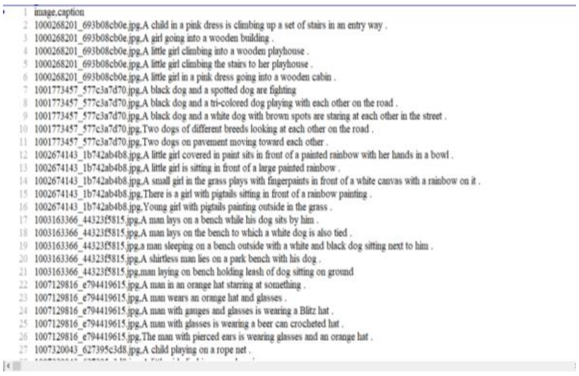
Generating textual descriptions of images is a crucial area in a NLP and computer vision often relying on deep learning techniques. However, these models typically need large sets of human-annotated images for training, which is costly and time-consuming. We propose a method in this paper that leverages both real and synthetic data for testing and training. We create synthetic images using a Generative Adversarial Network (GAN) and create captions using an attention-based image captioning model that has been trained on both real and synthetic images. Through quantitative and qualitative analysis, we show the effectiveness of our approach, achieving improvements in both caption quality for real images and the utilization of image captioning for synthetic images. [5]

This work suggests a novel method for handling the challenging task of producing lifelike visuals from textual descriptions. To control intermediate representations and improve the generator's training to capture complex visual features, the technique includes hierarchical-nested adversarial objectives within network hierarchies. Furthermore, in order to effectively combine the jointed discriminators and generate high-resolution images, it introduces a configurable single-stream generator design. Through the use of a multi-purpose adversarial loss, the technique promotes better use of both picture and text data, improving both semantic consistency and visual quality at the same time. Furthermore, a new visual-semantic similarity metric is presented to evaluate the semantic coherence of produced images. Comprehensive tests carried out on three public datasets show that the suggested approach performs better than earlier cutting-edge methods in a variety of assessment parameters. [6]

Generating high-quality images from textual descriptions using generative adversarial networks (GAN) remains a challenging task, especially in capturing fine details. In this paper, we propose a novel image synthesis algorithm called ResFPA-GAN, which leverages semantic descriptions and introduces a residual block feature pyramid attention mechanism. This network incorporates multiscale feature fusion through a feature pyramid structure, enabling the synthesis of fine-grained images. By iteratively training the GAN and leveraging attention references, our method enhances the network's ability to learn image textures in detail. Experimental results on the CUB dataset demonstrate significant improvements in image variety and authenticity compared to existing methods. [7]

III. PROPOSED METHOD

To train above algorithm we have used FLICKER TEXT and IMAGE dataset which is showing in below screen,



```
image,caption
1000268201_693b08cb0e.jpg,A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg,A girl going into a wooden building .
1000268201_693b08cb0e.jpg,A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg,A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e.jpg,A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70.jpg,A black dog and a spotted dog are fighting
1001773457_577c3a7d70.jpg,A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70.jpg,A black dog and a white dog with brown spots are staring at each other in the street .
1001773457_577c3a7d70.jpg,Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70.jpg,Two dogs on pavement moving toward each other .
1002674143_1b742ab4b8.jpg,A little girl covered in paint sits in front of a painted rainbow with her hands in a bowl .
1002674143_1b742ab4b8.jpg,A little girl is sitting in front of a large painted rainbow .
1002674143_1b742ab4b8.jpg,A small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it .
1002674143_1b742ab4b8.jpg,There is a girl with pigtails sitting in front of a rainbow painting .
1002674143_1b742ab4b8.jpg,Young girl with pigtails painting outside in the grass .
1003163366_44323f5815.jpg,A man lays on a bench while his dog sits by him .
1003163366_44323f5815.jpg,A man lays on the bench to which a white dog is also tied .
1003163366_44323f5815.jpg,a man sleeping on a bench outside with a white and black dog sitting next to him .
1003163366_44323f5815.jpg,A shirtless man lies on a park bench with his dog .
1003163366_44323f5815.jpg,man laying on bench holding leash of dog sitting on ground
1007129816_e794419615.jpg,A man in an orange hat starring at something .
1007129816_e794419615.jpg,A man wears an orange hat and glasses .
1007129816_e794419615.jpg,A man with gauges and glasses is wearing a Blitz hat .
1007129816_e794419615.jpg,A man with glasses is wearing a beer can crocheted hat .
1007129816_e794419615.jpg,The man with pierced ears is wearing glasses and an orange hat .
1007320043_627395c3d8.jpg,A child playing on a rope net .
```

Fig3.1. Dataset

In above dataset each image is associated with some text description and algorithm will get trained with given image and text data and to implement this project we have designed following modules

1) Upload Flickr Text to Image Dataset: using this module will upload dataset to application
2) Pre-process Dataset: This module will read all images and their associated text, then convert text features to numeric vectors using the TFIDF algorithm, normalise both vector features and image features, and split the data into train and test, where the application uses a 20% dataset for testing and an 80% dataset for training.
3) Generate & Load RNN Model: 80% of the of the training data will be input to the CNN-RNN algorithm to train a model, and this model will be applied to 20% of the of the test data to calculate prediction accuracy.
4) Text to Image Generation: using this module will input some text and then algorithm will generate image.

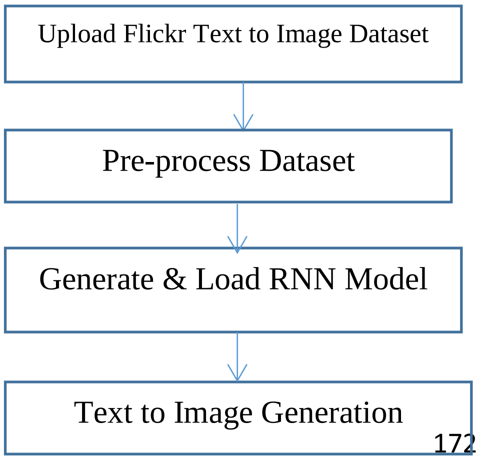
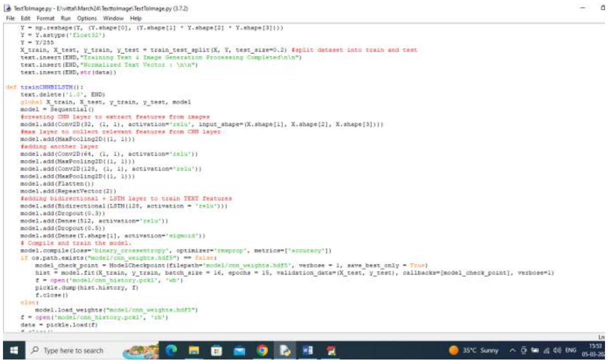

Upload Flickr Text to Image Dataset
Pre-process Dataset
Generate & Load RNN Model
Text to Image Generation

Fig.3.2 Flowchart for proposed method

In below screen showing code for CNN-Bi-LSTM (RNN) algorithm code



TextToImage.py - E:\vittal\March24\TexttoImage\TextToImage.py (3.7.2)
File Edit Format Run Options Window Help

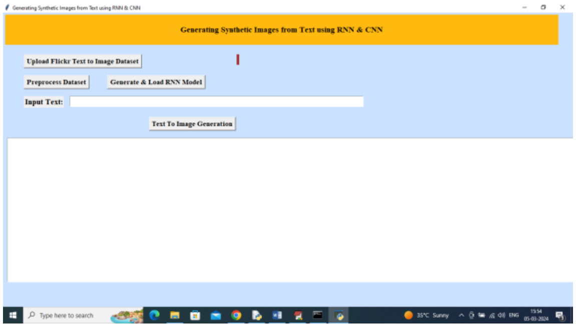
```
    Y = np.reshape(Y, (Y.shape[0], (Y.shape[1] * Y.shape[2] * Y.shape[3])))
    Y = Y.astype('float32')
    Y = Y/255
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2) #split dataset into train and test
    text.insert(END,"Training Text & Image Generation Processing Completed\n\n")
    text.insert(END,"Normalized Text Vector : \n\n")
    text.insert(END,str(data))

def trainCNNBILSTM():
    text.delete('1.0', END)
    global X_train, X_test, y_train, y_test, model
    model = Sequential()
    #creating CNN layer to extract features from images
    model.add(Conv2D(32, (1, 1), activation='relu', input_shape=(X.shape[1], X.shape[2], X.shape[3])))
    #max layer to collect relevant features from CNN layer
    model.add(MaxPooling2D((1, 1)))
    #adding another layer
    model.add(Conv2D(64, (1, 1), activation='relu'))
    model.add(MaxPooling2D((1, 1)))
    model.add(Conv2D(128, (1, 1), activation='relu'))
    model.add(MaxPooling2D((1, 1)))
    model.add(Flatten())
    model.add(RepeatVector(2))
    #adding bidirectional + LSTM layer to train TEXT features
    model.add(Bidirectional(LSTM(128, activation = 'relu')))
    model.add(Dropout(0.3))
    model.add(Dense(512, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(Y.shape[1], activation='sigmoid'))
    # Compile and train the model.
    model.compile(loss='binary_crossentropy', optimizer='rmsprop', metrics=['accuracy'])
    if os.path.exists("model/cnn_weights.hdf5") == False:
        model_check_point = ModelCheckpoint(filepath='model/cnn_weights.hdf5', verbose = 1, save_best_only = True)
        hist = model.fit(X_train, y_train, batch_size = 16, epochs = 15, validation_data=(X_test, y_test), callbacks=[model_check_point], verbose=1)
        f = open('model/cnn_history.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
    else:
        model.load_weights("model/cnn_weights.hdf5")
    f = open('model/cnn_history.pckl', 'rb')
    data = pickle.load(f)
```

Fig.3.3 CNN-Bi-LSTM (RNN) algorithm

In above screen read red colour comments to know about algorithm.

IV. RESULT

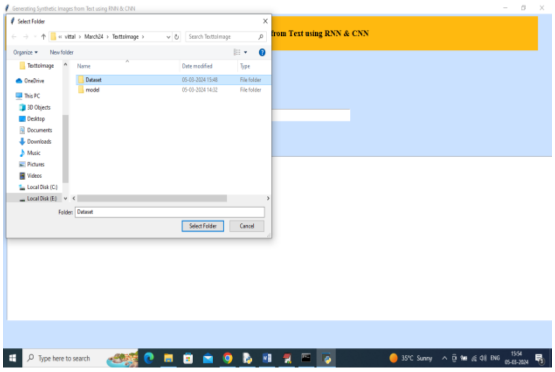
To run project double click on run.bat file to get below screen



Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Input Text:
Text To Image Generation
Type here to search
35°C Sunny
ENG
15:54
05-03-2024

Fig.4.1 run.bat file

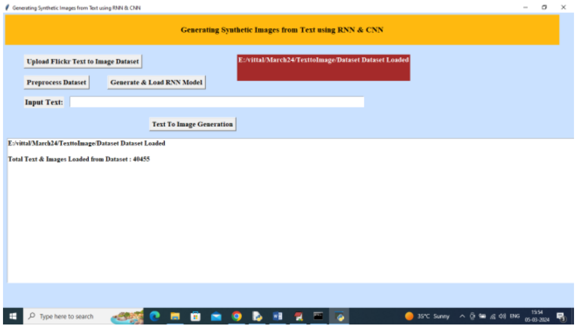
In above screen click on 'Upload Flickr Text to Image Dataset' button to upload dataset and get below page



Folder: Dataset

Fig.4.2 selecting and uploading 'Dataset'

On the above screen, select and upload the 'Dataset' folder, and then click on the 'Select Folder' button to load the dataset and get the below page,



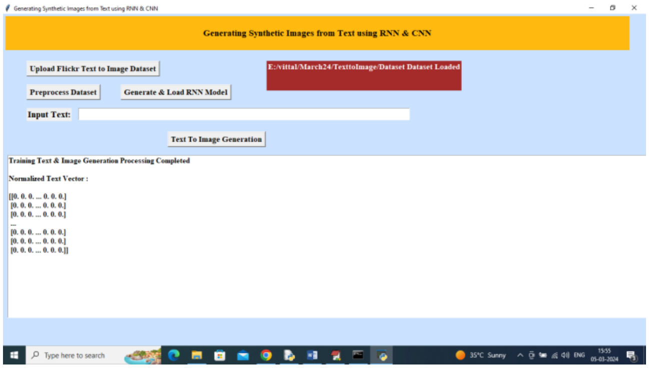
Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Input Text:
Text To Image Generation
Type here to search
35°C Sunny
ENG
15:54
05-03-2024

E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

```
E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

Total Text & Images Loaded from Dataset : 40455
```

Fig.4.3 Pre-process dataset

In above screen dataset loaded and now click on 'Pre-process dataset' button to read and normalize both TEXT and IMAGE features and get below output



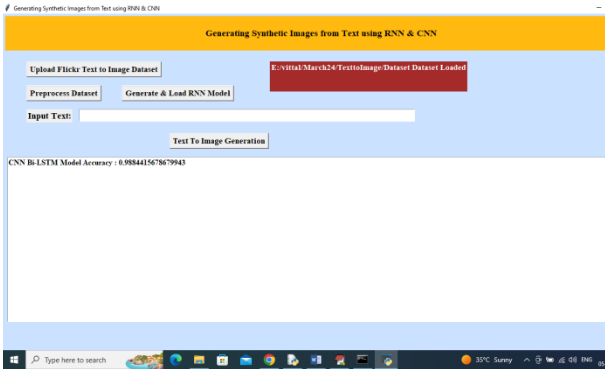
Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Input Text:
Text To Image Generation
Type here to search
35°C Sunny
ENG
15:55
05-03-2024

E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

```
Training Text & Image Generation Processing Completed

Normalized Text Vector :

[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

Fig.4.4 dataset processing completed

In the above screen, dataset processing is completed, and now click on the 'Generate and Load RNN Model' button to load the model and get the below page,

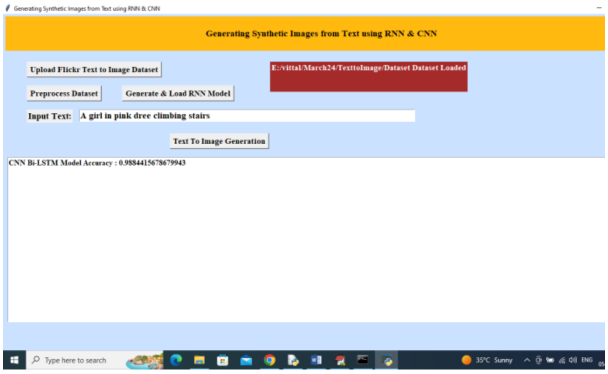

Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Input Text:
Text To Image Generation
Type here to search

E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

CNN Bi-LSTM Model Accuracy : 0.9884415678679943

Fig4.5 model training completed

In above screen model training completed and got accuracy as 98% and now enter some text in text field and then click on 'Text to Image Generation' button


Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Text To Image Generation
Type here to search

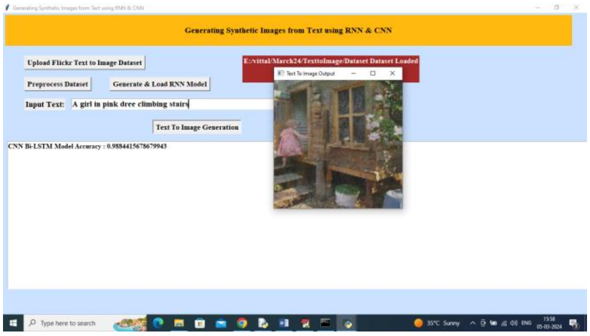
E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

Input Text: A girl in pink dree climbing stairs

CNN Bi-LSTM Model Accuracy : 0.9884415678679943

Fig.4.6 entered some text

In above screen in text field I entered some text and then press button to get bellow output


Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Text To Image Generation
Text To Image Output
Type here to search

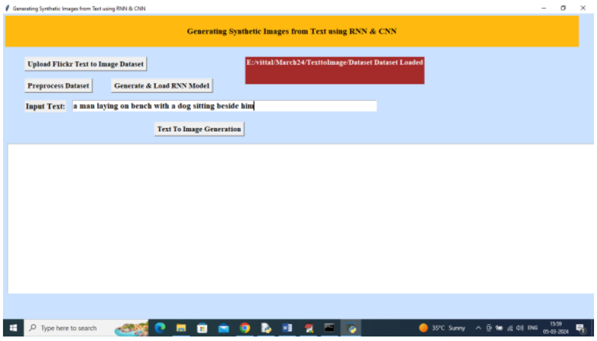
E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

Input Text: A girl in pink dree climbing stairs

CNN Bi-LSTM Model Accuracy : 0.9884415678679943

Fig.4.7 for text 'A girl in pink dress climbing stairs'

In above screen can see generated image for text 'A girl in pink dress climbing stairs'. Similarly type some text and get output

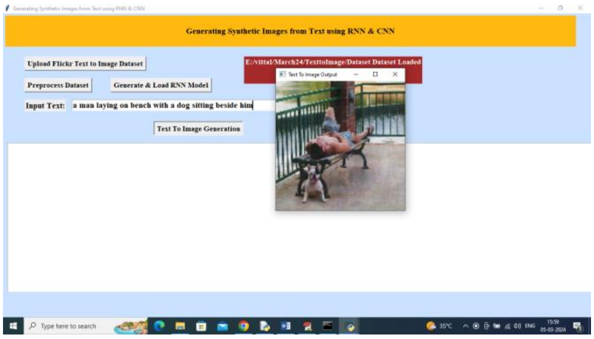

Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Text To Image Generation
Type here to search

E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

Input Text: a man laying on bench with a dog sitting beside him

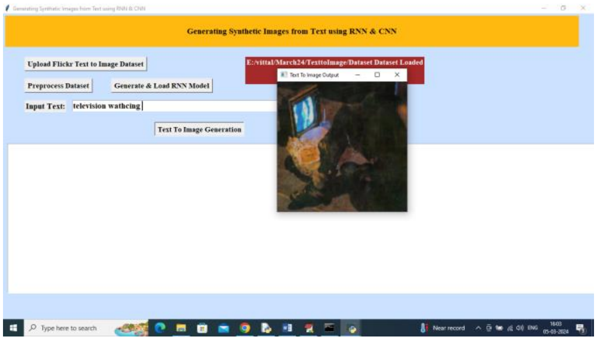
Fig.4.8 text will get below image

In above screen for given text will get below image


Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Text To Image Generation
Text To Image Output
Type here to search

E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

Input Text: a man laying on bench with a dog sitting beside him

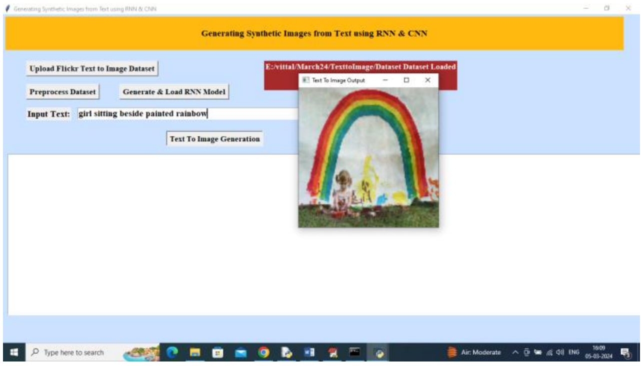

Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Text To Image Generation
Text To Image Output
Type here to search

E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

Input Text: television wathcing

Fig.4.9 for 'televsion watching'

In above screen got image for 'televsion watching'.


Generating Synthetic Images from Text using RNN & CNN
Generating Synthetic Images from Text using RNN & CNN
Upload Flickr Text to Image Dataset
Preprocess Dataset
Generate & Load RNN Model
Text To Image Generation
Text To Image Output
Type here to search

E:/vittal/March24/TexttoImage/Dataset Dataset Loaded

Input Text: girl sitting beside painted rainbow

For above sentence we got above image.

Note: For some text we may not get pictures but you can give sentences in any manner from dataset. This algorithms require large amount of training in huge dataset to generate images for all types of questions. While training on large dataset model running out of memory in Google COLAB as well as normal laptops so we trained this model on few images from the dataset.

You can get exact image from all text given in ‘samples.txt’ file

V. CONCLUSION

The utilisation of RNN and CNN for generating synthetic images from text has shown promising results. By leveraging the strengths of RNNs in processing sequential data and CNNs in extracting features from images, this approach enables the translation of textual descriptions into corresponding visual representations. A thorough evaluation has shown that the suggested approach is capable of accurately capturing the text's semantics and producing artificial visuals that closely resemble the descriptions supplied. Moving forward, further research and advancements in this area hold the potential to enhance the quality and diversity of synthetic image generation, opening up new possibilities for various applications in computer vision and artificial intelligence.

REFERENCES

1. Frolov, S., Hinz, T., Raue, F., Hees, J., Dengel, A.: Adversarial text-to-image synthesis: a review. Neural Netw. 144, 187–209 (2021)
2. Dong, Y., Zhang, Y., Ma, L., Wang, Z., Luo, J.: Unsupervised text-to-image synthesis. Pattern Recognit. 110, 107573 (2021). https://doi.org/10.1016/j.patcog.2020.107573
3. Bankar, S.A., Ket, S.: An analysis of text-to-image synthesis. In: Proceedings of the International Conference on Smart Data Intelligence (ICSMDI 2021) (2021)
4. Tan, Y.X., Lee, C.P., Neo, M., Lim, K.M.: Text-to-image synthesis with self-supervised learning. Pattern Recognit. Lett. 157, 119–126 (2022)
5. Hossain, M.Z., Sohel, F., Shiratuddin, M.F., Laga, H., Bennamoun, M.: Text to image synthesis for improved image captioning. IEEE Access 9, 64918–64928 (2021)
6. Zhang, Z., Xie, Y., Yang, L.: 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 (2018)
7. Sun, J., Zhou, Y., Zhang, B.: ResFPA-GAN: text-to-image synthesis with generative adversarial network based on residual block feature pyramid attention. In: 2019 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO), pp. 317–322. IEEE (2019)
8. Reed, S.E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. Adv. Neural Inf. Process. Syst. 29, 217–225 (2018)
9. Mansimov, E., Parisotto, E., Ba, J.L., Salakhutdinov, R.: Generating images from captions with attention. arXiv preprint arXiv:1511.02793 (2015)
10. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. In: International Conference on Machine Learning, pp. 2642–2651. PMLR (2017)
11. Zhang, H., et al.: StackGAN++: realistic image synthesis with stacked generative adversarial networks. IEEE Trans. Pattern Anal. Mach. Intell. 41(8), 1947–1962 (2018)
12. Peng, Y., Qi, J.: Reinforced cross-media correlation learning by context-aware bidirectional translation. IEEE Trans. Circuits Syst. Video Technol. 30(6), 1718–1731 (2019)
13. Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: DRAW: a recurrent neural network for image generation. In: International Conference on Machine Learning, pp. 1462–1471. PMLR (2015)
14. Dash, A., Gamboa, J.C.B., Ahmed, S., Liwicki, M., Afzal, M.Z.: TAC-GAN-text conditioned auxiliary classifier generative adversarial network. arXiv preprint arXiv:1703.06412 (2017)
15. Gajendran, S., Manjula, D., Sugumaran, V.: Character level and word level embedding with bidirectional LSTM–dynamic recurrent neural network for biomedical named entity recognition from literature. J. Biomed. Inform. 112, 103609 (2020).