International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 06 | June - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Automated Visual Generation using GAN with Textual InformationFeeds

Sibi Mathew[1], Milan Mani Mathew[1], Nitheesh D[1], Sreedevi S[1]
[1]B.Tech Scholar, Amal Jyothi College of Engineering, Kottayam, Kerala, India, 686518

***Abstract*-Visualising textual content could be helpful to professionals as well as amateurs across several fields. However, training a text-to-image generator in the mainstream domain requires large amounts of paired text-image and data, which is too expensive to collect since labeling millions of images and videos can be tiresome. GANs like StackGAN and StyleGAN can be considered as solutions to generate images from text. But the images generated may be of low accuracy and resolution, and the entire processing can be highly time-consuming. Moreover, image generation is a notion that is still being researched. Hence, the process of developing a Video Generation model necessitates substantial research. Despite the need for such a model, modern technology has lagged behind the solutions to this problem. This proposal suggests combining two methods, Text modification for Action Definition (TexAD) and SeQuential Image Generation for Video Synthesis (SQIGen). The proposed solution synthesises a sequence of images from textual information feeds and combines these images to create a video. TexAD uses Natural Language Processing and Deep Learning techniques to process, classify and modify text data. SQIGen is an extension of the VQGAN+CLIP neural network architecture that generates a sequence of images from the modified text data.**

***Index Terms*—Visualization, Sequential Image Generation, GANs, Natural Language Processing and Deep Learning, TexAD, VQGAN+CLIP**

I. INTRODUCTION

There are many instances where we face difficulties explaining certain situations and concepts to others through words. It is easier to explain concepts with visual representations than merely by words. Hence, we propose this system to make it easier for people to convey their message to their counterparts in a graphical way to understand it easily.

The common methods used are:-
1. Physically animate each frame.
2. Manually capture images and videos using a camera.

These methods are still practiced but are tedious and time-consuming as humans do a significant part of the work. Text-to-video conversion is a new field emerging out of AI applications. The presence of AI has been felt in almost every new technology introduced. Text-to-video conversion is a mechanism in which a text input collected from the user is converted to the corresponding video. That is what is done using a text to a video generator.

The main aim of our model is to decrease the dependency on humans to generate videos by incorporating artificial intelligence. Methods that have been previously used are

1. Using XML: Create a browser, store the predefined dataset in a database, and display the result when the user types in the text. However, the user may not get the desired result because this method works only for a limited number of images that are already present in the database.
2. Using GANs: Video generation from a single statement provides videos with low accuracy and resolution.

All these disadvantages have created various hurdles in developing a video from a piece of textual information. VQGAN+CLIP[9] is a neural network architecture built upon the revolutionary CLIP architecture published by Open-AI in January 2021. It is a text-to-image model that generates images of variable size given a set of text prompts. There have been other text-to-image models before, but the VQGAN+CLIP architecture creates more crisp, coherent, and high-resolution quality images than AI art tools that have come ahead.

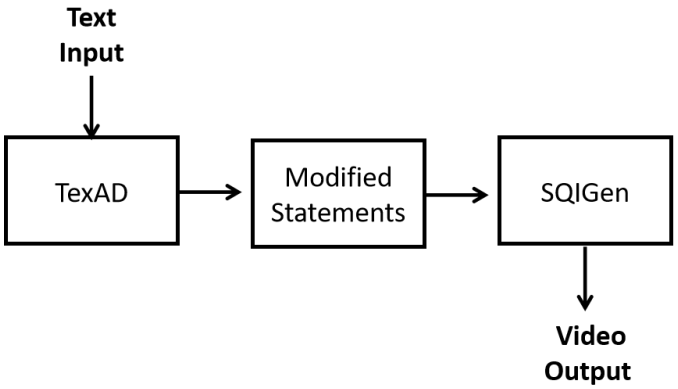

Fig. 1. General block diagram of proposed system

In this solution, we propose a new method of TexAD which stands for “Text modification for Action Definition”, in addition to VQGAN+CLIP, which is used to modify the given text. The model identifies locations, characters, and actions and creates revised statements. The modified text generates numerous images that can be played frame by frame. Here we use our model SQIGen to generate a series of images. This can provide us with numerous actions and increase the performance of the video.
There are several applications for our solution. For example,

a forensic investigator who needs to send a detailed report to the higher authorities can create a video of what happened by briefly describing the case. In the case of the medical field, the surgeon can write all the symptoms and help patients visualize their condition.

There are numerous advantages to using this proposed system. Firstly, by using this system, we can reduce the dependency on humans. We need to type what is required. Secondly, our proposed approach is much faster and requires less time to generate the result. Thirdly, it is much easier to develop as the VQGAN+CLIP model is open-sourced.The code is available at Code

II. Related Works

According to Masaki Hayashi [1],T2V(Text-To-Vision) is an innovative method that creates TV-program-like Computer Graphics animation generated automatically from a script and can be considered as one of the first of its kind. T2V Player is created using this technology. The software allows users to create animated videos by typing text. The T2V Player employs a framework that converts text to animation. Using this technology, we can map text to animated images. T2V Player provides a limited number of characters and commands for creating an animated video. The adaption of artificial intelligence and modern technology can let the application expand its functionalities.

In Han Zhanget's[5] proposed machine Stacked Generative Adversarial Network (StackGAN) is used to create 256x256 photos primarily based totally on textual content descriptions. The textual content-to-photographic technology is split into two ways in terms of usage of StackGAN: Stage-I GAN and Stage-II GAN. Stage-I GAN creates a low-decision photograph, and Stage-II GAN corrects defects within the low-decision photograph created via way of means of Stage-I and creates a high-decision photograph. But this method requires improvement in the way that StackGAN calls for a big quantity of GPU. Kambhampati's [8] proposed model uses Cyclic GAN to generate images of 128 by 128 pixels that correspond to the content of the information. This model is based on the Oxford-102 blooms data set. This system works by executing a Deep Convolutional Generative Adversarial Network (DCGAN), This method employs a version of Cyclic GAN that is designed to create text images. This method includes 102 blossom classes and 102 classification data sets.Sergey's[3] MoCoGAN framework creates a video clip by utilizing an image generator to sequentially generate images. To improve the quality of created videos, the framework can make use of developments in picture production in the GAN framework.

Masaki's [4] Temporal Generative Adversarial Net (TGAN) can create an entirely new video by learning representation from an unlabeled video data set. The generator in this model is formed by two sub-networks: a temporal generator and an image generator. The temporal generator, in particular, initially produces a series of latent variables, each of which correlates to a latent variable for the image generator. The image generator then converts these latent variables into a video with the same frame count as the variables. MoCoGAN and TGAN have higher Frechet Inception Distance (FID), which indicates that the generated images will be less similar to the training data. However, notwithstanding these examples, the quantity of research on textual-to-video stays small. Therefore, we present a new solution that is compatible with methodology conditioned on textual content.MoCoGAN creates a video clip by generating video frames in a specified order. An image-generative network maps a random vector into an image at each time step. The random vector is composed of two components, the first of which is drawn from a content subspace and the second from a motion subspace. We model the content space using a Gaussian distribution and use the same realization to generate each frame in a video clip because the content in a short video clip is usually the same. A recurrent neural network is used to sample from the motion space, and the network parameters are learned during training.

VGAN is a generative adversarial network for video with a Spatiotemporal convolutional architecture that untangles the scene's subject from the background, according to Carl [6]. This model can produce small films at a full-frame rate for up to a second. This model uses data to train an autoencoder, and the decoder uses a two-stream generator network. After that, it feeds instances through the encoder and fits a 256-component Gaussian Mixture Model (GMM) over the 100-dimensional hidden space. This model takes a sample from the GMM and feeds it through the decoder to create a new video. The created scenarios in films generated with this model are mostly very crisp, and the movement patterns are often realistic for the scenario. The lack of object resolution is a key flaw in this approach. Yogesh's [7] Text-Filter Condition Generative Adversarial Network (TFGAN), is a conditional GAN model with a novel multi-scale text-conditioning scheme that improves text-video associations. TFGAN creates high-quality films from the text on complex real-world video data sets by integrating this conditioning strategy with a deep GAN architecture. TFGAN can also create videos of different concepts that haven't been seen previously during training.

III. Methodology

To generate an image alone from a textual feed, using VQGAN+CLIP architecture produces satisfactory results. Un-like image generation, the concept of video generation has to consider numerous use cases that vary according to characters and locations. The primary issue regarding this concept is that the meaning of these statements should remain the same in the generated video.
This is why Automated Visual Generation is performed in two phases
1. Text Classification and Modification
2. Visual Generation

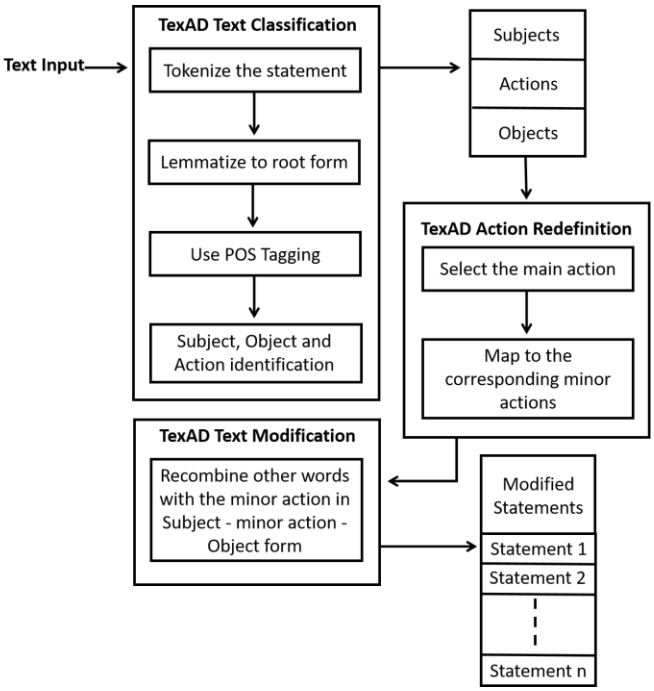

Fig. 2. TexAD Block Diagram

A. Text Classification and Modification

The first phase includes collecting text input from the user and performing classification and modification. These operations are executed using the TexAD model. This phase is done in three stages: TexAD Text Classification, TexAD Action Definition and TexAD Text Modification.

1) Stage 1: TexAD Text Classification

The text input is collected from the user. The input statement must be in subject-verb-object form and in active speech. The text data collected from the user is passed to the TexAD Text Classification model, which uses POS Tagging within the Natural Language ToolKit module to identify the parts of speech for each word in the sentence. The classification process assumes that all nouns and pronouns that occur before the main action word or verb are the subjects, and all nouns and pronouns that occur after the verb are the objects. If the sentence contains two or more action verbs and is joined using conjunction, the sentence will be split. The subject from the original sentence will be concatenated with each verb individually and then with the remaining words. Thus, the new number of sentences will be equal to the number of actions. Each sentence in this new set of sentences is processed discretely in the remaining stages.

2) Stage 2: TexAD Action Definition

The classified data from the previous stage is passed to the TexAD Action Definition model. The verb or action in the classified data is defined using TexAD Action Definition. TexAD focuses primarily on human actions. It has a collection of the most common human activities and a data set explaining the steps or sequence of actions that take part in the execution of these activities. Each action word or verb is explained using a set of minor actions that are sequentially performed to execute the main action from the user's input. The set of minor actions is a collection of phrases or words with at least one verb in each element. Using the minor actions, we can create a sequence of meaningful statements that, when combined, sequentially retains the idea expressed in the original statement. To perform this task, TexAD initially has to perform lemmatization on each verb to convert it into its root form. Then we map the main action verb to its equivalent set of minor actions to perform the action definition.

3) Stage 3: TexAD Text Modification

After action definition, we recombine these words by replacing the original verb with each element in the set of minor actions. All the other words in the sentence remain the same. Grammar correction packages can be used too. This set of meaningful sentences is considered as the sequence of modified statements.

Example:
User Input:The man is jumping on the bed
After processing with TexAD model
Subject: Man
Object: Bed
Action: Jumping
Set of minor actions : ['stand', 'squat', 'raise hands', 'jumps','squat' ,'lowers hands','stand']
Original Statement : The man is jumping on his bed
Modified Statements:
1. Statement 1 : Man stands on bed
2. Statement 2: Man squats on bed
3. Statement 3: Man raise hands on bed
4. Statement 4: Man jumps on bed
5. Statement 5: Man squats on bed
6. Statement 6: Man stands on bed

In previously attempted Video Generation GANs, a single statement is used to generate numerous images to synthesize a video. The advantage of using the TexAD model over a GAN alone is that each minor action can contribute one or more frames that are finally appended to create the video. Besides that, modifying the text to generate new images is easier than altering the image using different data sets to create separate frames for the video. As a result, the speed of video generation and the accuracy of the video is improved.

B. Visual Generation

The second phase of Automated visual generation involves generating a sequence of images with the sequence of modified data from the first phase and combining them to create a video. This phase uses the SQIGen architecture.
This phase is also in two stages: Image generation and Video generation.

1) Image Generation

The image generation process is performed using VQGAN+CLIP[2]. VQGAN stands for Vector Quantized

Generative Adversarial Network and CLIP for Contrastive Image-Language Pretraining. Vector quantization- Process of dividing vectors into groups that have approximately the same number of points closest to the. Vector quantization is a classic signal processing technique that finds the representative centroids for each cluster. VQGAN employs the two-stage structure by learning an intermediary representation, and then the output is passed to the transformer. It uses a codebook to represent visual parts. The codebook serves as the bridge for two-stage approach and it is created using vector quantization. CLIP is a multimodal model that blends English-language concept knowledge with image semantic information. CLIP is capable of Zero-Shot learning. VQGAN generates the images while CLIP judges how well an image matches our statement. With modifiers, VQGAN + CLIP can deliver better and more focused images. Simply said, modifiers are keywords that have been shown to have a significant impact on how the AI reads your query. Usually, adding one or more modifiers to your request will have a significant positive impact on the final image.When modifiers like ”Oil Painting” or ”Unreal Engine” are included, CLIP is aware that the image should have a specific appearance.

VQGAN+CLIP is trained on an enormous data set, and there is a possibility that for each statement, the images generated can turn out to be entirely different from one another. The contrast in images generated for each statement will lead to an inconsistency in the video. The issue is rectified by using a reference image. A reference image assists VQGAN+CLIP in generating images for all statements other than the first statement among the sequence of modified statements. VQGAN+CLIP can create an image corresponding to the input statement by altering some aspects of the reference image. Using a reference image brings consistency to the sequence of images generated and escalates the speed of image generation.

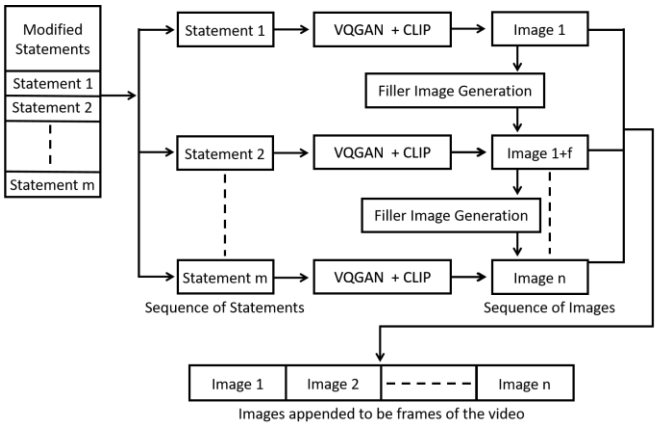

Fig. 3. SQIGen Architecture Block Diagram

The first statement among the sequence of modified statements is fed as input to generate the initial image. This initial image generated is used as the reference image to create the second image with the second statement. The image generated here will be the reference image for the following statement. This process is repeated until all the modified statements are utilized to create corresponding images. Before moving onto the video generation stage, certain conditions must be satisfied.

The minimum frame rate has been set to 20fps to produce a decent-quality video. Therefore, to generate a video of 5 seconds in length, there must be at least 100 images. But the number of modified statements is usually much lesser than the number of frames required. A filler image generation algorithm can be used to overcome this constraint.

Algorithm

Image Generation: This function uses the VQGAN+CLIP model to generate images. It takes two inputs, an input statement, and a reference image, and returns an image that is generated with this data.

```
generate_image(
        input_statement,reference_image)
return image
```

Filler Image Generation: This function knows the video length, frames per second, and a number of statements, and calculates the number of frames required and the number of filler images.

```
number_of_frames = video_length * fps
if number_of_statements < number_of_frames
        number_of_filler_images = number_of
          _frames/(number_of_statements-1)
image[0]=generate_image(statement[0],None)
i=1,list_index=1
while  i<number_of_statements
j=0
while j<number_of_filler_images
image[list_index++]=
    generate_image(statement[i],
                    image[list_index-1])
j=j+1
i=i+1
```

Filler image is generated for all statements other than the initial statement. Therefore, the image generated from the first statement is fed to the filler image generation algorithm, to create filler images between the first and second statements. This process is repeated until the generation of filler images between the second last and last statements.

2) Video Generation

All the images generated will be stored in the proper order as png files. Then, the images are combined and considered as frames of the subsequent video to be developed. FFmpeg and image2pipe tools can be used after all the images are generated, to pipe png data into FFmpeg to render it into a video.

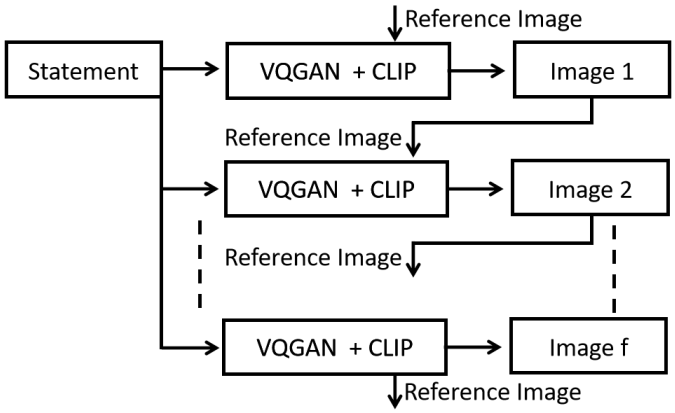

Fig. 4. Filler Image Generation Block Diagram

IV. Conclusion

This paper proposed a methodological approach to converting text to video. A textual feed from the user is processed by the TexAD model, which creates a new set of statements from it. The SQIGen model utilizes these statements to generate a sequence of images, which, when combined, gives a video. We are sure that our model would work under any given circumstances. Video generation from text is still a less explored field and has enough utility if there is progress in this area of research. Our future work includes creating a system that can convert a given paragraph into a multi-scene video with higher resolution. Currently, our proposed solution can work better only with high-end graphics processors. Thus, we feel that there would be a considerable advance in this field of research and that our effort would be a valuable contribution.

References

[1] Hayashi, Masaki, et al. ”T2v: New technology of converting text to cg animation.” ITE Transactions on Media Technology and Applications 2.1 (2014): 74-81.
[2] Esser, Patrick, Robin Rombach, and Bjorn Ommer. ”Taming transformers for high-resolution image synthesis.” Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.
[3] Tulyakov, Sergey, et al. ”Mocogan: Decomposing motion and content for video generation.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[4] Saito, Masaki, Eiichi Matsumoto, and Shunta Saito. ”Temporal generative adversarial nets with singular value clipping.” Proceedings of the IEEE international conference on computer vision. 2017.
[5] Zhang, Han, et al. ”Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks.” Proceedings of the IEEE international conference on computer vision. 2017.
[6] CarlVondrick, Carl, Hamed Pirsiavash, and Antonio Torralba. ”Generating videos with scene dynamics.” Advances in neural information processing systems 29 (2016).
[7] Balaji, Yogesh, et al. ”Conditional GAN with Discriminative Filter Generation for Text-to-Video Synthesis.” IJCAI. Vol. 1. No. 2019. 2019.
[8] Gorti, Satya Krishna, and Jeremy Ma. ”Text-to-image-to-text translation using cycle consistent adversarial networks.” arXiv preprint arXiv:1808.04538 (2018).
[9] Crowson, Katherine, et al. ”Vqgan-clip: Open domain image generation and editing with natural language guidance.” Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII. Cham: Springer Nature Switzerland, 2022.