Introduction

- Text-to-image generation has made great strides, with models like Stable Diffusion able to create realistic images from text descriptions.

- However, extending this image generation capability to videos remains a challenge. Current methods for text-to-video require extensive training on large datasets of paired text and video data, which is computationally expensive.

- In this project, we tackle the novel problem of generating videos directly from text prompts, without any training on video data.

- Our key idea is that we can modify existing pre-trained text-to-image models like Stable Diffusion to generate videos, with minimal extra work.

- We introduce two main techniques: First, we modify the internal representations (latent codes) of the images to include motion information, ensuring consistency in the background and overall scene motions across video frames.

Motivation

- Text-to-image models like Stable Diffusion have enabled high-fidelity image generation from text descriptions.
- However, extending these models to synthesize videos has remained challenging, with most methods requiring large-scale training on paired text-video datasets.
- This costly training hinders wider accessibility and limits text-to-video applications.
- To address this, we introduce the new problem of text-to-video synthesis , where the goal is to generate videos from text without any training.
- Our key ideas are simple yet effective modifications to leverage pre-trained image models for video synthesis.
- By making text-to-video generation ”training-free” our work provides an efficient and practical way to make video synthesis and editing more accessible to general users.

Objectives

- The primary objective of this work is to enable high-quality text-to-video generation without requiring any training or optimization, formulated as the novel problem of text-to-video synthesis.

- To achieve this goal, we propose techniques to modify pre-trained text-to-image models like Stable Diffusion to make them suitable for temporally coherent video generation out-of-the-box.

- Our first key contribution is a method to enrich the latent codes with motion dynamics.

- This is done by warping the latent code of the first frame using motion flow fields to generate subsequent frames.
- The motion dynamics induce global background consistency while providing flexibility for object motions.

Problem Statement

- Text-to-image synthesis has seen great progress, with models like Stable Diffusion able to generate photorealistic images from text prompts.
- However, extending these capabilities to video generation remains an open challenge.
- Current text-to-video methods require extensive training on large paired text-video datasets , which is computationally expensive and restricts wider access.
- For example, recent state-of-the-art approaches leverage datasets with millions of text-video pairs for training video generation models.
- This costly training requirement motivates the need for more efficient text-to-video generation paradigms.
- To address this limitation, we introduce the novel problem setting of text-to-video synthesis .

Literature review

- Text-to-video generation has gained significant attention in recent years, with the advent of powerful diffusion models and advancements in generative modeling.
- This literature survey explores the current state-of-the-art techniques and challenges in this domain, focusing on the specific project about generating animated GIFs representing a space rocket launching into space from the desert using the Diffusers library and pre-trained models.
- Diffusion models have demonstrated remarkable success in image synthesis tasks, outperforming traditional generative adversarial networks (GANs).
- These models leverage score-based generative modeling through stochastic differential equations [9], allowing for high-fidelity image generation from textual prompts.
- However, extending these capabilities to the video domain poses unique challenges, as temporal consistency and coherence across frames become crucial factors

Scope

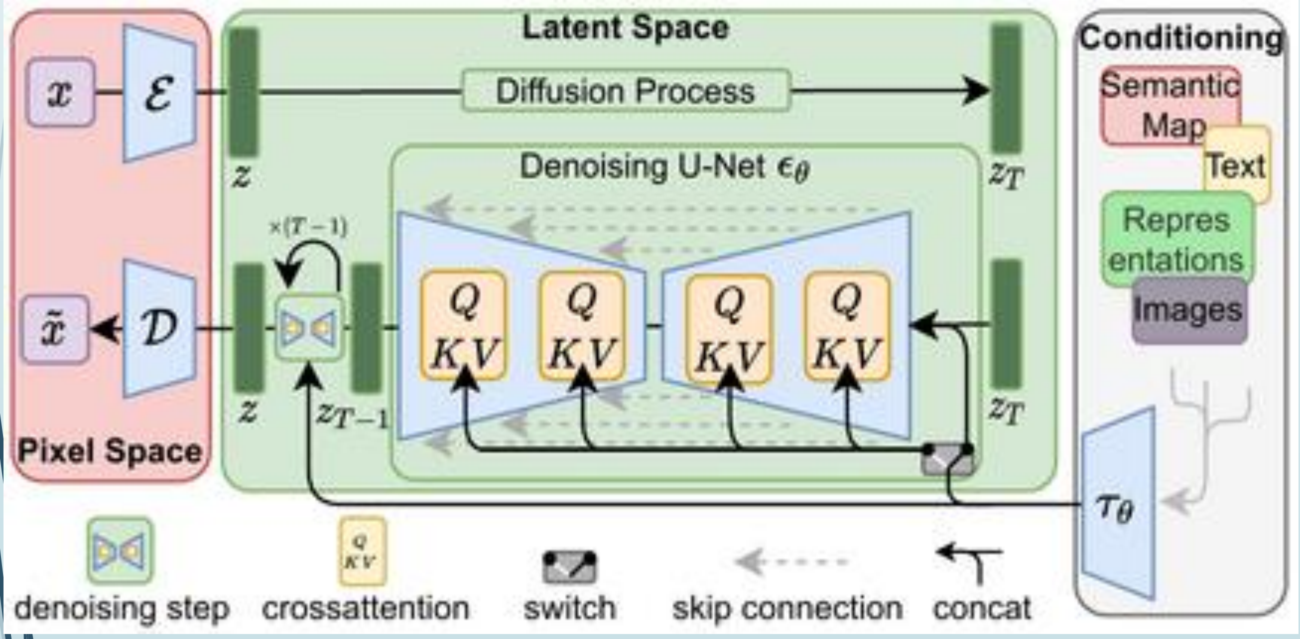
- The scope of a text-to-video generator is broad and it can be applied in various domains like-

- **Content creation**
- **Educational materials**
- **Creative Art and Design**
- **Advertising and Marketing**
- **Entertainment Industry**

Proposed System

- This project proposes a novel approach for text-to-video generation that can synthesize consistent video clips from only text prompts, without requiring any model training.

- It builds on pre-trained text-to-image diffusion models like Stable Diffusion. The key idea is to modify such models to enforce temporal consistency across generated frames.



Proposed System

- The core of the architecture is a pre-trained text-to-image diffusion model like Stable Diffusion (SD).
- SD contains an encoder-decoder model and a diffusion process trained to generate the original image from the noisy training data using text prompts.

Step 1 : Importing all the dependencies.

```
import os
import torch
from datasets import load_dataset
import soundfile as sf
from transformers import pipeline
from diffusers import DiffusionPipeline
import moviepy.editor as mp
```

Proposed System

Step 2 : This line initializes a text-to-speech synthesizer using the Microsoft SpeechT5 TTS model through the Hugging Face pipeline function.

```
# Initialize the text-to-speech synthesizer
synthesiser = pipeline("text-to-speech", "microsoft/speecht5_tts")
```

Step 3 : This line loads a dataset named "cmu-arctic-xvectors" from the Hugging Face datasets library for speaker embeddings.


```
embeddings_dataset = load_dataset("Matthijs/cmu-arctic-xvectors", split="validat
```

Proposed System

Step 4 : This section initializes a Diffusion Pipeline for generating images from text using a pre-trained model. The model used here is "**playground-v2-1024px-aesthetic**".

```
# Initialize the Diffusion Pipeline
pipe = DiffusionPipeline.from_pretrained(
    "playgroundai/playground-v2-1024px-aesthetic",
    torch_dtype=torch.float16,
    variant="fp16"
).to("cuda")
```

Step 5 : These lines create directories to store the generated images, audio files, short video clips, and the final video. The os.makedirs function is used with the exist_ok=True argument to ensure that the directories are created if they do not exist.

```
# Create directories to store generated files
image_dir = os.path.join(base_dir, "images")
audio_dir = os.path.join(base_dir, "audio")
short_video_dir = os.path.join(base_dir, "short_videos")
final_video_dir = os.path.join(base_dir, "final_video")
os.makedirs(image_dir, exist_ok=True)
os.makedirs(audio_dir, exist_ok=True)
os.makedirs(short_video_dir, exist_ok=True)
os.makedirs(final_video_dir, exist_ok=True)
```

Proposed System

Step 6 : This loop iterates over each sentence in the input text and performs the following tasks:Generates an image from the sentence using the initialized Diffusion Pipeline and saves it as a PNG file in the image_dir.

```
# Generate images and corresponding audio for each sentence
for i, sentence in enumerate(prompt_sentences):
    # Generate image
    image = pipe(sentence).images[0]
    image_path = os.path.join(image_dir, f"image_{i}.png")
    image.save(image_path)

    # Convert text to speech and save audio
    speech = synthesiser(sentence, forward_params={"speaker_embeddings": speaker
    audio_path = os.path.join(audio_dir, f"audio_{i}.wav")
    sf.write(audio_path, speech["audio"], samplerate=speech["sampling_rate"])
```

Step 7 : In this section, each image generated from the sentences is combined with its corresponding audio to create a short video clip.

```
# Convert each image to a short video clip
video_clips = []
for i in range(len(prompt_sentences)):
    image_path = os.path.join(image_dir, f"image_{i}.png")
    audio_path = os.path.join(audio_dir, f"audio_{i}.wav")
    video_clip_path = os.path.join(short_video_dir, f"video_{i}.mp4")

    # Calculate duration based on audio length
    audio_duration = len(sf.SoundFile(audio_path)) / speech["sampling_rate"]

    # Create a short video clip from the image with corresponding audio
    clip = mp.ImageClip(image_path)
    audio_clip = mp.AudioFileClip(audio_path)
    clip = clip.set_duration(audio_duration)
    clip = clip.set_audio(audio_clip)
    clip.write_videofile(video_clip_path, codec="libx264", fps=24)

    video_clips.append(clip)
```

Proposed System

Step 7: Finally, the short video clips generated from each sentence are combined into a single final video.

```
# Combine video clips into a final video
final_video_path = os.path.join(final_video_dir, "final_video.mp4")
final_video = mp.concatenate_videoclips(video_clips, method="compose")
final_video.write_videofile(final_video_path, codec="libx264", fps=24)

print(f"Saved final video to {final_video_path}")
```

Observations

Testing plays a crucial role in ensuring the quality, reliability, and performance of the text-to-video synthesis system. A comprehensive testing strategy was employed to validate the system’s functionality, identify potential issues, and evaluate its effectiveness in generating coherent and high-quality videos from text prompts.

Parameters	runwayml/stable-diffusion-v1-5	stabilityai/stable-diffusion-2-1	CompVis/stable-diffusion-v1-4	stabilityai/stable-diffusion-2-base
Source	RunwayML	StabilityAI	Anthropic	StabilityAI
Version	1.5	2.1	1.4	2
CPU	YES	NO	NO	YES
GPU	NO	YES	YES	NO
Time Taken	13 min	45 min	30 min	7 min
Key Features	Improved image quality and coherence Better face restoration and detail generation	Significantly improved image quality Reduced artifacting and better fine details Improved face generation	Base model with good overall image generation Less prone to distortions or artifacts	Foundation model for Stability v2 series Good image quality and coherence
Image				

		
"A space war against aliens." Model- Runaway ML	"A space war against aliens." Model-playground-v2	"A space war against aliens." Model-Linaqruf/anything-v3
Size- **5gb** Time - **30 min**	Size- **7 to 8gb** Time - **30 sec** on colab and **can't run** offline	Size- **5gb** Time - **10 sec** on colab and **7 to 8 min** offline

Conclusion

The testing process yielded valuable insights and results, guiding the iterative development and refinement of the text-to-video synthesis system. Some key observations and findings include:

1. **High-Quality Video Generation:** The system demonstrated the ability to generate compelling and coherent videos from a wide range of text prompts, accurately capturing the described scenes, objects, and actions.
2. **Temporal Consistency:** The integration of motion encoding and cross-frame attention techniques effectively maintained temporal consistency and object identities across video frames, resulting in smooth and coherent video outputs.
3. **Scalability and Performance:** The system exhibited good scalability, capable of handling varying input text lengths and complexities, albeit with some performance trade-offs for extremely long or intricate video generations.