Zero-Shot Video Generation

1st Amey Mahendra Thakur
dept. of electrical and computer engineering
University of Windsor
Windsor, Canada
0000-0001-5644-1575

2nd Jithin Gijo Varghese
dept. of electrical and computer engineering
University of Windsor
Windsor, Canada
gijovar@uwindsor.ca

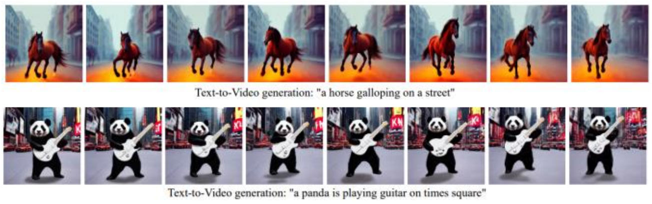
3rd Ritika Agarwal
dept. of electrical and computer engineering
University of Windsor
Windsor, Canada
agarwa73@uwindsor.ca

***Abstract*—The intersection of artificial intelligence and multimedia continues to evolve, breaking down barriers between different forms of media. In this project, the research titled "Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators" conducted by Picsart AI Research Lab represents a significant breakthrough. This study introduces a pioneering method that directly converts textual descriptions into videos, bridging the gap between natural language processing and computer vision. This development not only caters to the growing demand for dynamic visual content but also showcases the machine's capability to interpret and transform human language into a visual format. By addressing the challenge of text-to-video synthesis, this research sets a new standard for interdisciplinary studies in artificial intelligence.**

Keywords—artificial intelligence, text-to-video synthesis, computer vision, natural language processing, multimedia, zero-shot video generation, Picsart AI Research Lab.

I. Introduction

The field of artificial intelligence continually seeks to break barriers between different forms of media. At the forefront of this endeavor stands the research titled "Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators" conducted by Picsart AI Research Lab [1]. This study introduces a method that converts textual descriptions directly into videos, marking a significant advancement in the integration of natural language processing and computer vision. Such a development not only addresses the growing demand for dynamic visual content but also showcases the potential of machines to interpret and render human language in a visual format. By offering a solution to the challenge of text-to-video synthesis, the research sets a new benchmark for interdisciplinary studies in artificial intelligence.


Text-to-Video generation: "a horse galloping on a street"

Text-to-Video generation: "a panda is playing guitar on times square"

Figure 1: Text-to-Video generation [1]

II. Project Motivation

In an era where visual storytelling is paramount, the ability to convert textual narratives into dynamic videos holds transformative potential. As platforms and audiences increasingly favor visual content, the research from Picsart AI Research Lab, titled "Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators," emerges as a timely and innovative response to this demand [1]. The research by Picsart AI Research Lab is not just academically intriguing; it addresses a contemporary need, offering a solution that aligns with the evolving preferences of today's digital consumers. Delving into its significance reveals:

A. Blending Words with Vision

The initiative by Picsart AI Research Lab offers a novel approach, merging the realms of text interpretation and visual representation. This is not just about generating images; it's about crafting a coherent visual story based on textual cues.

B. A New Era of Content Creation

With the digital landscape being saturated with content, differentiation becomes key. A tool that can take textual descriptions and produce videos offers a unique edge, streamlining content creation and offering bespoke visual outputs.

C. Making Learning More Visual

In education, the value of a tool that can translate textual concepts into visual content is immeasurable. It offers a tangible way to represent abstract ideas, catering to a broader spectrum of learners.

D. Handling Big Data Challenges

The emphasis on utilizing large image datasets signifies the project's ambition to operate at scale, ensuring that vast amounts of data can be processed without compromising on the quality of the generated videos.

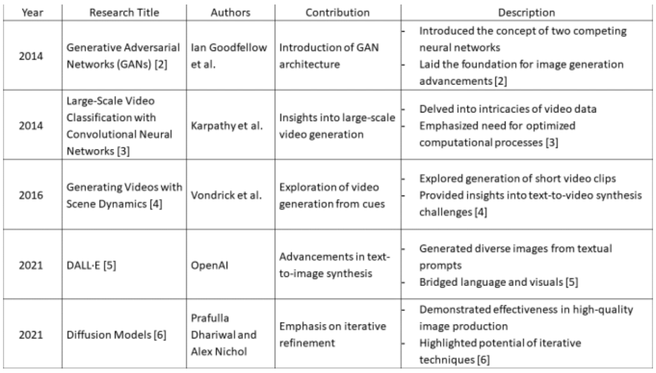
E. Prioritizing User Experience

By integrating a user interface, the project underscores its commitment to accessibility. It's a nod to the importance of ensuring that such groundbreaking technology is usable and beneficial to a wide audience.

III. Literature Review

The journey towards the synthesis of textual narratives into visual content has been paved by several groundbreaking works, each contributing a piece to the puzzle. The following literature chronicles this progression, leading up to the innovative "Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators" by Picsart AI Research Lab:

Table 1: Literature Review



Year	Research Title	Authors	Contribution	Description
2014	Generative Adversarial Networks (GANs) [2]	Ian Goodfellow et al.	Introduction of GAN architecture	- Introduced the concept of two competing neural networks - Laid the foundation for image generation advancements [2]
2014	Large-Scale Video Classification with Convolutional Neural Networks [3]	Karpathy et al.	Insights into large-scale video generation	- Delved into intricacies of video data - Emphasized need for optimized computational processes [3]
2016	Generating Videos with Scene Dynamics [4]	Vondrick et al.	Exploration of video generation from cues	- Explored generation of short video clips - Provided insights into text-to-video synthesis challenges [4]
2021	DALL·E [5]	OpenAI	Advancements in text-to-image synthesis	- Generated diverse images from textual prompts - Bridged language and visuals [5]
2021	Diffusion Models [6]	Prafulla Dhariwal and Alex Nichol	Emphasis on iterative refinement	- Demonstrated effectiveness in high-quality image production - Highlighted potential of iterative techniques [6]

The following table visually represents the sequence of foundational works providing a clearer understanding of the research progression.

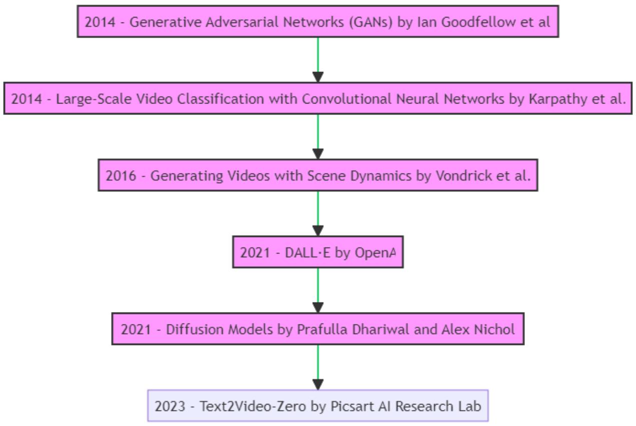

2014 - Generative Adversarial Networks (GANs) by Ian Goodfellow et al
2014 - Large-Scale Video Classification with Convolutional Neural Networks by Karpathy et al.
2016 - Generating Videos with Scene Dynamics by Vondrick et al.
2021 - DALL·E by OpenA
2021 - Diffusion Models by Prafulla Dhariwal and Alex Nichol
2023 - Text2Video-Zero by Picsart AI Research Lab

Figure 2: Research Progression

IV. Project Implementation

The project focuses on implementing a web-based application for zero-shot video generation using text prompts. It integrates text-to-image diffusion models with a user-friendly interface, allowing users to generate videos based on textual descriptions. The implementation is structured across three main Python files: **app.py**, **model.py**, and **app_text_to_video.py**.

A. System Architecture

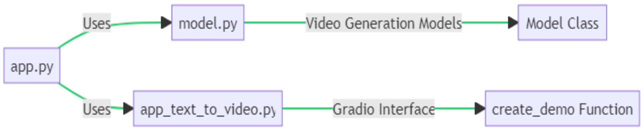


Figure 3: System Architecture

Table 2: System Architecture

Component	Description
app.py	- Entry point for the application. - Creates the web interface. - Manages the application's flow.
model.py	- Defines the Model class. - Implements various model pipelines for video generation.
app_text_to_video.py	- Contains the create_demo function. - Integrates the Gradio interface with the Model class.

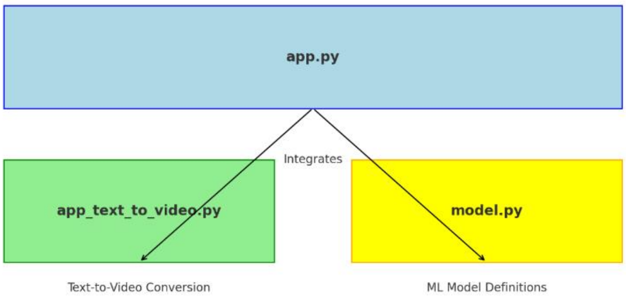


Figure 4: Project Structure

The diagram above illustrates the structure of the "Zero-Shot Video Generation" project:

1) **app.py (Main Application):** This is the central script that integrates various components of the application. It likely handles user interactions and coordinates the workflow.
2) **app_text_to_video.py (Text-to-Video Conversion):** This script is specifically focused on converting text inputs into video outputs. It is integrated into the main application and utilizes the machine learning models defined in **model.py**.
3) **model.py (ML Model Definitions):** This script contains the definitions of the machine learning models used in the project, particularly for the text-to-video conversion process.

The arrows indicate the integration of **app_text_to_video.py** and **model.py** into the main application (**app.py**). This setup allows for a modular and organized approach, separating the user interface, conversion logic, and model definitions.

B. Model Pipeline

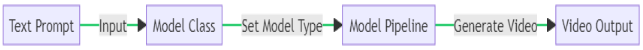


Figure 5: Model Pipeline

The model pipeline diagram showcases the process from text input to video output:

Table 3: Model Pipeline

Stage	**Description**
Text Prompt	- Users provide text prompts as input.
Model Class	- The Model class handles model selection and setup.
Model Pipeline	- Different model pipelines, e.g., Text2Video, process the input.
Video Output	- The final output is generated as a video.

C. Project Flowchart

The flowchart illustrates the sequential steps involved in the project:

1) ***Data Collection & Preprocessing:*** The initial stage involves sourcing and preparing the dataset.
2) ***Feature Extraction:*** This step focuses on extracting relevant features from both textual and visual data.
3) ***Model Implementation & Training:*** Here, the "Text2Video-Zero" model is adapted and trained.
4) ***Evaluation:*** The model's performance and the quality of generated videos are assessed.
5) ***User Interface Development:*** The project concludes with the creation of a user-friendly interface.

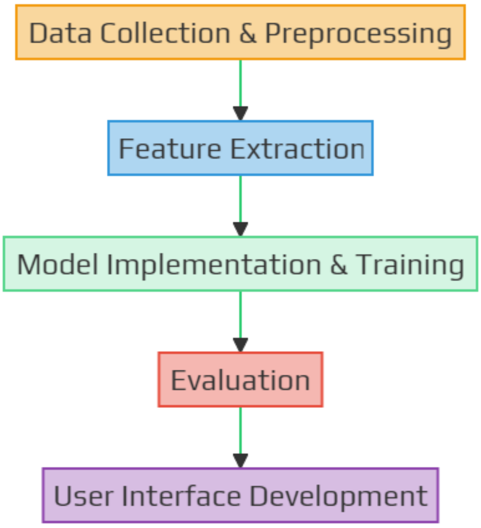

Data Collection & Preprocessing
Feature Extraction
Model Implementation & Training
Evaluation
User Interface Development

Figure 6: Flowchart

V. Project Progress

A. Overview

The data collection and feature extraction phases have been successfully completed within the allocated timeframe. The model implementation is also finalized, and the user interface design has been completed, marking a commendable achievement in our project's progress.

Table 4: Project Progress

Task	Status	Completion Date
Data Collection	Completed	Oct 15
Feature Extraction	Completed	Oct 23
Model Implementation	Completed	Oct 31
Evaluation	Completed	Nov 5
User Interface Development	Completed	Nov 10

B. Task Status

Table 5: Task Status

Task	Status	Deadline
Data Collection	Completed	Oct 15
Gathered diverse datasets from reputable sources, focusing on varied content categories. Implemented data preprocessing techniques for cleaning and standardization.		
Feature Extraction	Completed	Oct 23
Utilized advanced feature extraction methods, including deep learning techniques and natural language processing algorithms. Extracted rich features from textual and visual data, enabling comprehensive analysis and model training. Conducted exploratory data analysis to identify key features relevant to the project scope.		
Model Implementation	Completed	Oct 31
Developed machine learning models using Gradio and PyTorch frameworks. Experimented with various architectures, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to optimize performance. Iteratively refined the models through training and validation phases. Implemented transfer learning techniques for leveraging pre-trained models.		
Evaluation	Completed	Nov 5
Defined comprehensive evaluation metrics, including accuracy, precision, recall, and F1-score. Evaluated model performance against benchmark datasets and real-world scenarios. Analyzed evaluation results to identify strengths and areas for improvement.		
User Interface Development	Completed	Nov 10
Designed user-friendly interfaces for seamless interaction. Created wireframes and mockups to visualize the user journey. Incorporated intuitive navigation and interactive elements.		

C. Challenges and Solutions

Table 6: Challenges and Solutions

Challenge	Solution
Resource Intensity	. Collaborated with cloud service providers to optimize TPU utilization. . Implemented cloud-based solutions and refined algorithms to reduce resource requirements.
Limited Knowledge and Resources	. Invested in extensive research and learning through online courses and workshops to bridge the knowledge gap.
Collaboration and Communication	. Conducted regular team meetings and agile project management tools for effective communication. . Used GitHub for immediate issue resolution and code sharing.

D. Future Milestones and Goals

Table 7: Future Milestones and Goals

Milestones	Description
Implement Advanced ML Models	Apply advanced algorithms for improved accuracy and performance.
Optimize Model Efficiency	Optimize models for resource efficiency and faster processing.
Enhance User Interface	Refine UI based on user feedback and conduct usability testing.
Integration and Testing	Integrate components, conduct tests, and address identified issues.
Documentation and Finalization	Prepare comprehensive documentation and review for completion.

VI. Methodology

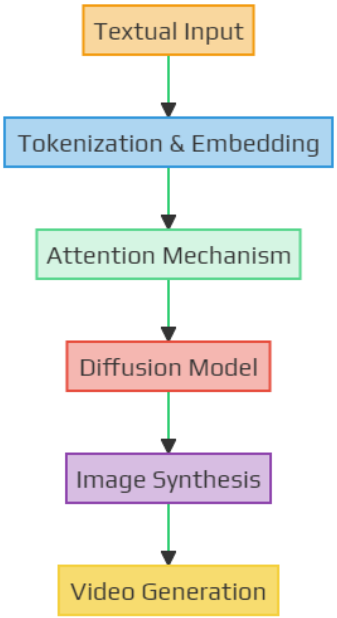
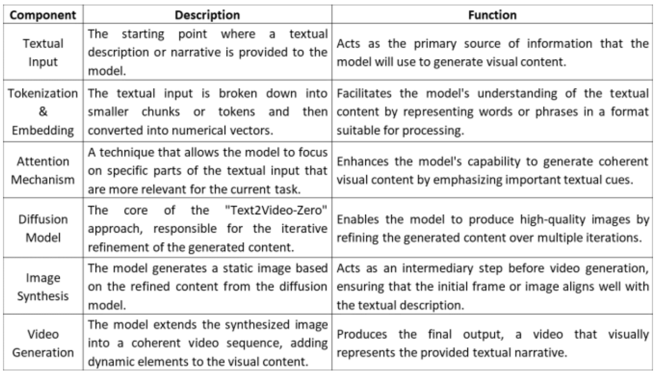


Figure 7: High-level architecture of the "Text2Video-Zero" model

Table 8: Components of high-level architecture of the "Text2Video-Zero" model



Component	Description	Function
Textual Input	The starting point where a textual description or narrative is provided to the model.	Acts as the primary source of information that the model will use to generate visual content.
Tokenization & Embedding	The textual input is broken down into smaller chunks or tokens and then converted into numerical vectors.	Facilitates the model's understanding of the textual content by representing words or phrases in a format suitable for processing.
Attention Mechanism	A technique that allows the model to focus on specific parts of the textual input that are more relevant for the current task.	Enhances the model's capability to generate coherent visual content by emphasizing important textual cues.
Diffusion Model	The core of the "Text2Video-Zero" approach, responsible for the iterative refinement of the generated content.	Enables the model to produce high-quality images by refining the generated content over multiple iterations.
Image Synthesis	The model generates a static image based on the refined content from the diffusion model.	Acts as an intermediary step before video generation, ensuring that the initial frame or image aligns well with the textual description.
Video Generation	The model extends the synthesized image into a coherent video sequence, adding dynamic elements to the visual content.	Produces the final output, a video that visually represents the provided textual narrative.

VII. Dataset

A. Source

While there are several datasets available for image generation tasks, for this project, datasets like COCO (Common Objects in Context) or ImageNet could be considered due to their vastness and diversity. Additionally, datasets specifically designed for video tasks, like UCF101 or Kinetics, might be explored to understand temporal dynamics.

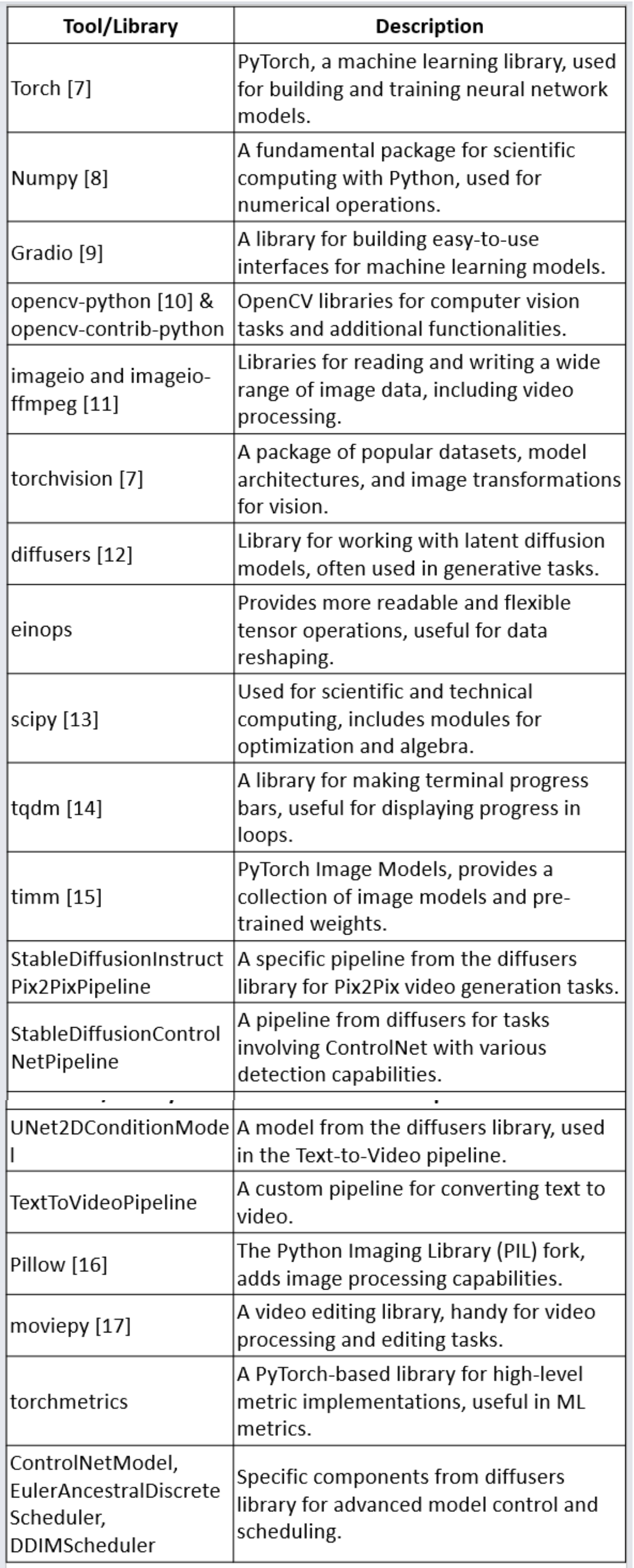
B. Key Features of the Dataset

1) ***Diversity***: The dataset will include images from various categories, ensuring that the model can handle a wide range of textual prompts.
2) ***High-Resolution:*** To generate quality videos, the dataset will prioritize high-resolution images.
3) ***Annotated Data:*** Each image in the dataset will be paired with textual descriptions, aiding in supervised training.
4) ***Temporal Consistency:*** For video generation, the dataset will also include sequences of images that showcase movement or change over time.

VIII. ML Libraries

The successful implementation of the "Text2Video-Zero" model requires a combination of specialized tools and libraries, each tailored to handle specific tasks within the project.

Table 9: Library/Tool Used



Tool/Library	Description
Torch [7]	PyTorch, a machine learning library, used for building and training neural network models.
Numpy [8]	A fundamental package for scientific computing with Python, used for numerical operations.
Gradio [9]	A library for building easy-to-use interfaces for machine learning models.
opencv-python [10] & opencv-contrib-python	OpenCV libraries for computer vision tasks and additional functionalities.
imageio and imageio-ffmpeg [11]	Libraries for reading and writing a wide range of image data, including video processing.
torchvision [7]	A package of popular datasets, model architectures, and image transformations for vision.
diffusers [12]	Library for working with latent diffusion models, often used in generative tasks.
einops	Provides more readable and flexible tensor operations, useful for data reshaping.
scipy [13]	Used for scientific and technical computing, includes modules for optimization and algebra.
tqdm [14]	A library for making terminal progress bars, useful for displaying progress in loops.
timm [15]	PyTorch Image Models, provides a collection of image models and pre-trained weights.
StableDiffusionInstructPix2PixPipeline	A specific pipeline from the diffusers library for Pix2Pix video generation tasks.
StableDiffusionControlNetPipeline	A pipeline from diffusers for tasks involving ControlNet with various detection capabilities.
UNet2DConditionModel	A model from the diffusers library, used in the Text-to-Video pipeline.
TextToVideoPipeline	A custom pipeline for converting text to video.
Pillow [16]	The Python Imaging Library (PIL) fork, adds image processing capabilities.
moviepy [17]	A video editing library, handy for video processing and editing tasks.
torchmetrics	A PyTorch-based library for high-level metric implementations, useful in ML metrics.
ControlNetModel, EulerAncestralDiscreteScheduler, DDIMScheduler	Specific components from diffusers library for advanced model control and scheduling.

IX. Timeline

A. *Timeline*

Table 10: Project Timeline

Date	Task
Oct 1	Project Description/Proposal Submission
Oct 2 - Oct 6	Conference with Instructor
Oct 7 - Oct 15	Data Preprocessing
Oct 16 - Oct 23	Model Development
Oct 24 - Oct 31	Training Phase I & Progress Report
Nov 1 - Nov 10	Training Phase II & UI Development
Nov 11 - Nov 15	Testing & Debugging
Nov 16 - Nov 19	Finalization, Documentation & Submissions
Nov 19	PPT File Submission

B. *Gantt Chart*

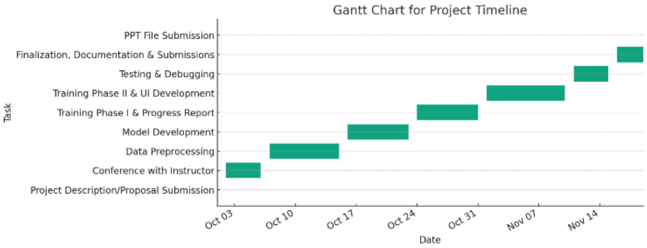


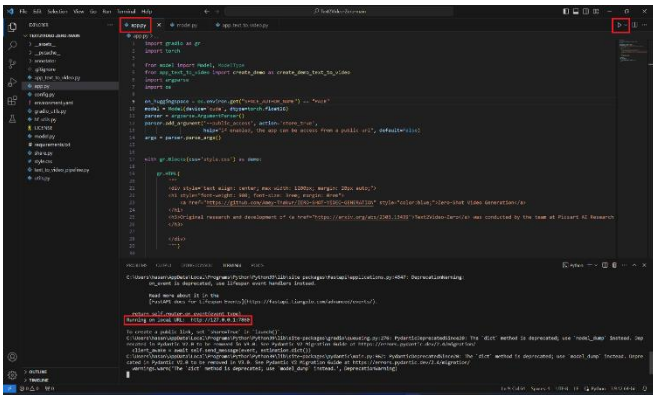
Figure 8: Gantt Chart

X. Launching Zero-Shot Video Generation ML Project

The local deployment of the Text2Video ML project provides a direct and interactive method for converting text into video content. The process prioritizes user engagement and efficiency, allowing for rapid access to and assessment of the generated videos.

A. *Step 1: Starting the Project*

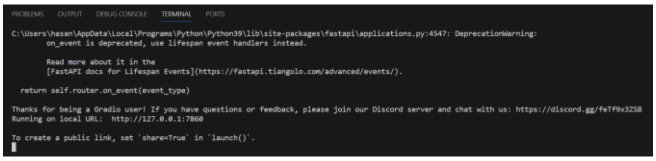
Navigate to the project directory in the terminal. Run the Machine Learning project with the command python app.py. This action initiates the server and readies the Text2Video model for local deployment.



```
import gradio as gr
import torch

from model import Model, ModelType
from app_text_to_video import create_demo as create_demo_text_to_video
import argparse
import os

on_huggingspace = os.environ.get("SPACE_AUTHOR_NAME") == "PAIR"
model = Model(device='cuda', dtype=torch.float16)
parser = argparse.ArgumentParser()
parser.add_argument('--public_access', action='store_true',
                    help="if enabled, the app can be access from a public url", default=False)
args = parser.parse_args()


with gr.Blocks(css='style.css') as demo:

    gr.HTML(
        """
        <div style="text-align: center; max-width: 1200px; margin: 20px auto;">
        <h1 style="font-weight: 900; font-size: 3rem; margin: 0rem">
            <a href="https://github.com/Amey-Thakur/ZERO-SHOT-VIDEO-GENERATION" style="color:blue;">Zero-Shot Video Generation</a>
        </h1>
        <h3>Original research and development of <a href="https://arxiv.org/abs/2303.13439">Text2Video-Zero</a> was conducted by the team at Picsart AI Research
        </h3>

        </div>
        """)
```

```
C:\Users\hasan\AppData\Local\Programs\Python\Python39\lib\site-packages\fastapi\applications.py:4547: DeprecationWarning:
        on_event is deprecated, use lifespan event handlers instead.

        Read more about it in the
        [FastAPI docs for Lifespan Events](https://fastapi.tiangolo.com/advanced/events/).

  return self.router.on_event(event_type)
Running on local URL:  http://127.0.0.1:7860

To create a public link, set `share=True` in `launch()`.
C:\Users\hasan\AppData\Local\Programs\Python\Python39\lib\site-packages\gradio\queueing.py:276: PydanticDeprecatedSince20: The `dict` method is deprecated; use `model_dump` instead. Dep
recated in Pydantic V2.0 to be removed in V3.0. See Pydantic V2 Migration Guide at https://errors.pydantic.dev/2.4/migration/
  client_awake = await self.send_message(event, estimation.dict())
C:\Users\hasan\AppData\Local\Programs\Python\Python39\lib\site-packages\pydantic\main.py:962: PydanticDeprecatedSince20: The `dict` method is deprecated; use `model_dump` instead. Depre
cated in Pydantic V2.0 to be removed in V3.0. See Pydantic V2 Migration Guide at https://errors.pydantic.dev/2.4/migration/
  warnings.warn('The `dict` method is deprecated; use `model_dump` instead.', DeprecationWarning)
```

Figure 9: Starting the Project

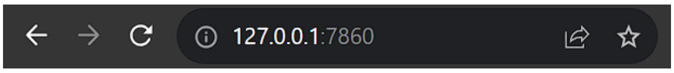


```
C:\Users\hasan\AppData\Local\Programs\Python\Python39\lib\site-packages\fastapi\applications.py:4547: DeprecationWarning:
        on_event is deprecated, use lifespan event handlers instead.

        Read more about it in the
        [FastAPI docs for Lifespan Events](https://fastapi.tiangolo.com/advanced/events/).

  return self.router.on_event(event_type)

Thanks for being a Gradio user! If you have questions or feedback, please join our Discord server and chat with us: https://discord.gg/feTf9x3ZSB
Running on local URL:  http://127.0.0.1:7860

To create a public link, set `share=True` in `launch()`.
```

Figure 10: Terminal

B. *Step 2: Accessing the Local Server*

Following the command execution, a localhost link appears in the terminal. This URL serves as the gateway to the project's user interface. Copy the provided localhost link, which generally appears as **http://127.0.0.1:7860**, and paste it into a web browser's address bar.

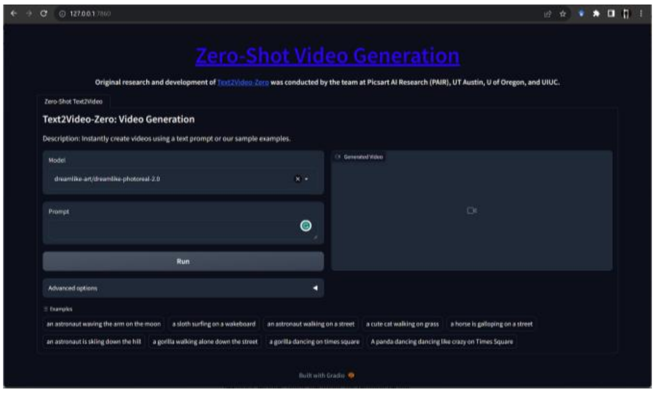


127.0.0.1:7860

Figure 11: Accessing the Local Server

C. *Step 3: Interacting with the Text2Video Model Identify the Headings*

On the local server page, the Text2Video model's interface welcomes visitors. Enter text into the model's interface to start the video generation process. The design ensures a seamless interaction with the ML model.



127.0.0.1:7860

Zero-Shot Video Generation

Original research and development of Text2Video-Zero was conducted by the team at Picsart AI Research (PAIR), UT Austin, U of Oregon, and UIUC.

Zero-Shot Text2Video

Text2Video-Zero: Video Generation

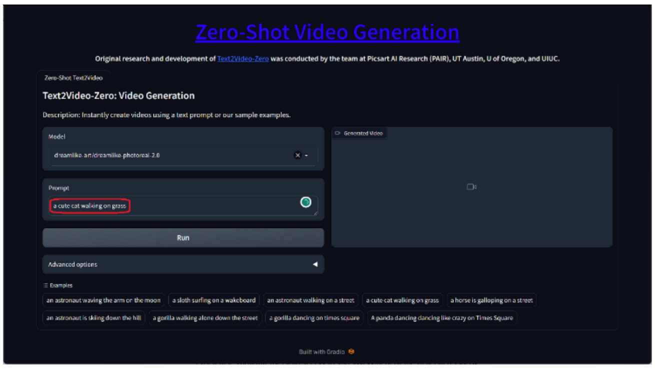
Description: Instantly create videos using a text prompt or our sample examples.

Model: dreamlike-art/dreamlike-photoreal-2.0

Prompt:

Generated Video
Run
Advanced options
Examples
an astronaut waving the arm on the moon
a sloth surfing on a wakeboard
an astronaut walking on a street
a cute cat walking on grass
a horse is galloping on a street
an astronaut is skiing down the hill
a gorilla walking alone down the street
a gorilla dancing on times square
A panda dancing dancing like crazy on Times Square
Built with Gradio

Figure 12: User Interface



Zero-Shot Video Generation

Original research and development of Text2Video-Zero was conducted by the team at Picsart AI Research (PAIR), UT Austin, U of Oregon, and UIUC.

Zero-Shot Text2Video

Text2Video-Zero: Video Generation

Description: Instantly create videos using a text prompt or our sample examples.

Model: dreamlike-art/dreamlike-photoreal-2.0

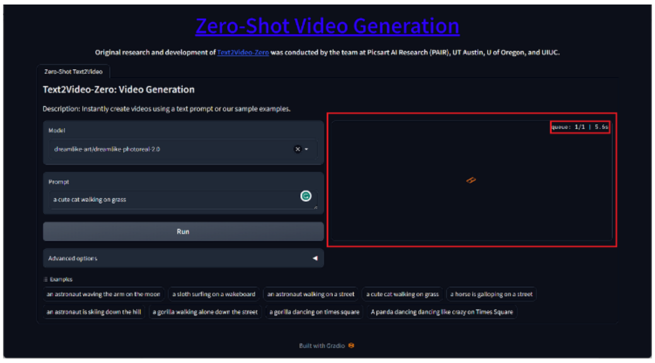
Prompt: a cute cat walking on grass

Generated Video
Run
Advanced options
Examples
an astronaut waving the arm on the moon
a sloth surfing on a wakeboard
an astronaut walking on a street
a cute cat walking on grass
a horse is galloping on a street
an astronaut is skiing down the hill
a gorilla walking alone down the street
a gorilla dancing on times square
A panda dancing dancing like crazy on Times Square
Built with Gradio

Figure 13: Interacting with the Text2Video Model

D. *Step 4: Using the Model*

Type the desired text into the model's input field. This text acts as the input for the Text2Video model. Trigger the model to begin transforming the text into a video. The model processes the text and generates a corresponding video.



Zero-Shot Video Generation

Original research and development of Text2Video-Zero was conducted by the team at Picsart AI Research (PAIR), UT Austin, U of Oregon, and UIUC.

Text2Video-Zero: Video Generation

Description: Instantly create videos using a text prompt or our sample examples.

Model: dreamlike-art/dreamlike-photoreal-2.0

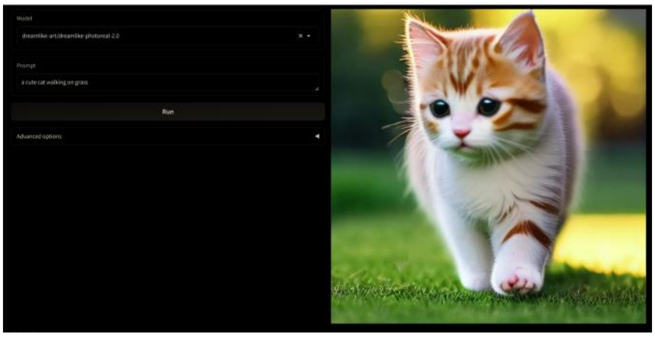
Prompt: a cute cat walking on grass

Zero-Shot Text2Video
Run
Advanced options
Examples
an astronaut waving the arm on the moon
a sloth surfing on a wakeboard
an astronaut walking on a street
a cute cat walking on grass
a horse is galloping on a street
an astronaut is skiing down the hill
a gorilla walking alone down the street
a gorilla dancing on times square
A panda dancing dancing like crazy on Times Square
queue: 1/1 | 5.6s
Built with Gradio

Figure 14: Using the Model

E. *Step 5: Viewing the Results*

Once the model processes the input text, the generated video displays on the webpage. Review and assess the output directly in the browser. This feedback enables quick iterations for refining results.



Model: dreamlike-art/dreamlike-photoreal-2.0

Prompt: a cute cat walking on grass

Figure 15: Viewing the Results [18]

XI. Scope Limitations of the Text2Video-Zero Implementation

Several factors have shaped the scope of this project despite its successful implementation. The quality and diversity of the extensive image dataset have significantly influenced the model's performance, potentially resulting in suboptimal outcomes due to variations or biases in the data[1]. Available resources have also played a role, affecting the model's resource demands and potentially extending training durations or limiting model complexity.

The model's ability to generalize may vary depending on the complexity and ambiguity of the input text. While we designed the user interface for seamless interaction, it may encounter latency in real-time video generation due to the high computational requirements and resource demands, resulting in delays.

The project's dependence on specific machine learning libraries means that unforeseen updates or changes in these libraries could present challenges or deviations in the intended functionality, necessitating ongoing monitoring and adaptation.

XII. Conclusion

Implementing the "Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators" model marks a pivotal advancement in artificial intelligence, blending natural language processing with computer vision. This project transcends technical achievement, showcasing AI's evolving ability to interpret human language visually.

A. *Revolutionizing Content Creation*

This technology's capability to convert text into dynamic videos opens new horizons in various fields, from entertainment to education. It promises to change the way we create and interact with media, offering personalized and immersive experiences.

B. *Overcoming Technical Challenges*

The project tackled significant challenges in training AI models for accurate text interpretation and visualization. Employing diffusion models and attention mechanisms, the team demonstrated innovative solutions to these complex problems, contributing valuable insights to AI research.

C. *Enhancing User Accessibility*

Central to this project is making powerful technology accessible. Developing a user-friendly interface ensures that individuals with diverse technical backgrounds can utilize AI for video generation, democratizing access to advanced technology.

D. *Setting New Benchmarks and Exploring Future Possibilities*

The "Text2Video-Zero" project not only sets new standards in text-to-video synthesis but also opens doors to further research and applications. It prompts important discussions about AI's role in content creation and the ethical considerations it entails.

The project represents a significant step towards a future where AI seamlessly integrates language and visuals, forging a new path in digital storytelling. Committed to pushing AI boundaries, enhancing user experience, and unlocking the full potential of text-to-video synthesis, this project paves the way for future innovations in the field.

References

[1] L. Khachatryan et al., “Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators,” arXiv:2303.13439, March 23, 2023, https://arxiv.org/abs/2303.13439 [Accessed: October 1, 2023].

[2] I. J. Goodfellow et al., “Generative Adversarial Networks,” arXiv:1406.2661, 2014, https://arxiv.org/abs/1406.2661 [Accessed: October 1, 2023].

[3] A. Karpathy et al., “Large-Scale Video Classification with Convolutional Neural Networks,” 2014 IEEE Conference on Computer Vision and Pattern Recognition, June 2014, https://doi.org/10.1109/cvpr.2014.223 [Accessed: October 1, 2023].

[4] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating Videos with Scene Dynamics,” arXiv:1609.02612, October 26, 2016, https://arxiv.org/abs/1609.02612 [Accessed: October 1, 2023].

[5] OpenAI Research Lab, “DALL·E: Creating images from text,” https://openai.com/research/dall-e [Accessed: October 1, 2023].

[6] P. Dhariwal and A. Nichol, “Diffusion Models Beat GANs on Image Synthesis,” arXiv:2105.05233, June 2021, https://arxiv.org/abs/2105.05233 [Accessed: October 1, 2023].

[7] A. Paszke, S. Gross, and G.S. Chintala, "Pytorch: An Open Source Machine Learning Framework," PyTorch Documentation, 2016, https://pytorch.org [Accessed: October 1, 2023].

[8] Numpy, “NumPy,” Numpy Documentation, 2009, https://numpy.org/ [Accessed: October 1, 2023].

[9] Gradio Documentation, “Gradio Docs,” https://www.gradio.app/docs/interface [Accessed: October 1, 2023].

[10] OpenCV Documentation, “OpenCV: OpenCV-Python Tutorials,” https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html [Accessed: October 1, 2023].

[11] Imageio Documentation, “Welcome to imageio’s documentation! — imageio 2.13.3 documentation,” https://imageio.readthedocs.io/en/stable/ [Accessed: October 1, 2023].

[12] HuggingFace, “Installation,” https://huggingface.co/docs/diffusers/installation [Accessed: October 1, 2023].

[13] SciPy documentation, “SciPy v1.8.1 Manual,” https://docs.scipy.org/doc/scipy/ [Accessed: October 1, 2023].

[14] C. da Costa-Luis, “tqdm documentation,” tqdm.github.io. https://tqdm.github.io/ [Accessed: October 1, 2023].

[15] Timmdocs, “Pytorch Image Models (timm),” https://timm.fast.ai/ [Accessed: October 1, 2023].

[16] Pillow Documentation, “Pillow (PIL Fork) 6.2.1 documentation,” 2011. https://pillow.readthedocs.io/en/stable/ [Accessed: October 1, 2023].

[17] “User Guide — MoviePy 1.0.2 documentation,” zulko.github.io. https://zulko.github.io/moviepy/ [Accessed: October 1, 2023].

[18] T. Wolf et al., “HuggingFace’s Transformers: State-of-the-art Natural Language Processing,” arXiv:1910.03771, February 2020, https://arxiv.org/abs/1910.03771 [Accessed: October 1, 2023].