FYP Final Report

VideoCraftXtend: AI-Enhanced Text-to-Video Generation with Extended Length and Enhanced Motion Smoothness

by

LEE Hsin-ning, CHEN Yu-zhen

CQF2

Advised by

Prof. Qifeng CHEN

Submitted in partial fulfillment of the requirements for COMP 4981

in the Department of Computer Science

The Hong Kong University of Science and Technology

2023-2024

Date of submission: April 18, 2024

Abstract

Text-to-Video (T2V) technology has gained significant popularity in various domains such as advertising, education, entertainment, and more. While the capability to generate lengthy videos remains promising, it is not yet fully matured. This project aims to tackle challenges in T2V generation, specifically focusing on the production of long videos, enhancing motion smoothness quality and improving content diversity. To address these challenges, we propose a comprehensive framework that integrates a T2V diffusion model, utilizes the OpenAI GPT API, incorporates a Video Quality Assessment (VQA) model, and refines an Interpolation model. Through our open-sourced framework, users can generate and download videos using a user-friendly interface. By leveraging these components, we aim to advance the capabilities of T2V generation and provide a valuable tool for video creation.

Table of Content

1 Introduction

1.1 Overview

A developing area in artificial intelligence and computer vision is text-to-video (T2V), which is often referred to as video synthesis or video generation. Text-to-video technology aims to produce a corresponding video sequence from a given textual description. While this process appears to be similar to solving a sequence of text-to-image (T2I) tasks, it is far more complex. So far, diffusion-based architecture has been used to build an increasing number of well-known T2I models, including Imagen [9] and Midjourney [11].

Despite the huge success in T2I generation, these models are difficult to generalize to T2V tasks. Currently, T2V models encounter three main issues: content diversity, long video generation, and motion smoothness [1]. Generating longer videos, such as those spanning 10 seconds, presents a notable challenge primarily due to limited dataset availability and the substantial computational demands involved. Additionally, T2V models struggle with the issue of content diversity. In some cases, the generated videos tend to be visually monotonous, lacking variation and creativity. This limitation arises from relying solely on the semantic meaning of user input prompts, which may not sufficiently capture the breadth of possible visual content. Moreover, AI-generated videos often exhibit flaws in motion boundaries between frames, resulting in visible artifacts that disrupt the overall visual cohesiveness.

To tackle those issues, this project aims to enhance the performance of the T2V model by utilizing OpenAI GPT API [12], accompanied by a Video Quality Assessment (VQA) model [32] and a Video Frame Interpolation model [33]. With ChatGPT API, we leverage the ability to modify user prompts, thereby enhancing the content diversity of our output videos. The VQA model guides and evaluates the T2V model's output to ensure quality before presenting to users. Moreover, a crucial aspect of our work involves training a Video Frame Interpolation model with the objective of improving video smoothness and extending video sequences.

1.2 Objectives

This project aims to tackle challenges in T2V generation, specifically focusing on the production of long videos, enhancing motion smoothness quality, and improving content diversity. We have proposed a T2V pipeline by integrating a lightweight T2V diffusion model, OpenAI GPT API [12], a Video Quality Assessment (VQA) model [32], and a Video Interpolation model [33]. We mainly focused on the following objectives during the project development phase:

1. Utilize the OpenAI GPT API to modify the user prompts and incorporate them as inputs to the Text-to-Video (T2V) base model.
2. Train a CNN-based Interpolation model to extend video length up to 10 seconds and improve motion smoothness of generated videos with an average increase of 2-5% in motion smoothness scores, measured by VBench [36] VQA model.
3. Develop a user interface that allows users to input text prompts and define their desired output video length. The interface should also provide the functionality to download the resulting videos and suggest several sample prompts to assist users in forming a coherent storyline.

During the project's progression, we encountered various challenges, including the significant task of training the interpolation model. This challenge was compounded by two factors: the large size of the dataset and limited computational resources available. The dataset used for the training process was extensive, requiring substantial computational power and time investment. The total training time for the interpolation model alone amounted to approximately 15 hours, which was further extended by numerous trial runs and iterations aimed at achieving optimal results.

1.3 Literature Survey

1.3.1 Text-to-Video Generation

In the field of T2V Generation, diffusion models [1] have emerged as a powerful category of deep generative models, displaying exceptional performance in diverse domains like image synthesis and video generation. These models utilize a progressive deconstruction process by introducing noise, followed by learning to reconstruct the original data, enabling the generation of synthetic samples. The adoption of a diffusion model is well-suited for our cases that demand the capture of long-term dependencies. Notable recent advancements, including Stable Diffusion [14], Imagen Video [9], Make-a-Video [22], and VideoCrafter [5] [6] [30], can be classified as diffusion models, employing U-Net architectures to model videos in either pixel or latent space.

Despite significant progress in the field, several challenges persist. Firstly, current T2V models often encounter difficulties when generating long videos, primarily due to the limited availability of existing datasets consisting of short video clips. These datasets fail to capture the complexity and diversity required for generating high-quality long videos [1]. Secondly, the quality of output videos often falls short of human expectations due to limited content diversity and potential mismatches in the positioning of moving objects. To address these challenges, our work focuses on utilizing VideoCrafter2 [5] as our base model. By incorporating innovative techniques and refining the existing model, we aim to tackle these two key challenges and improve the overall performance of T2V generation.

1.3.2 Video Frame Interpolation

Video frame interpolation is a technique that synthesizes multiple frames between two adjacent frames to increase the video frame rate and improve the smoothness of motion [33]. A common approach for video frame interpolation is flow-based algorithms which consists of warping the input frames according to approximated optical flows and fusing the warped frames using Convolutional Neural Networks (CNN).

In addition to the conventional method of estimating optical flow, Jiang et al. [7] propose Super SloMo, which involves utilizing a linear combination of two bi-directional flows to approximate the intermediate flows. These intermediate flows are then further refined using the U-Net architecture. Bao et al. [24] proposes DAIN, using an approach that estimates the intermediate

flow by employing a weighted combination of bidirectional flow.

Real-Time Intermediate Flow Estimation (RIFE) [33] is a CNN-based model that leverages IFNet for estimating the intermediate optical flows, the architecture is shown in ***Figure 1***. IFNet employs a coarse-to-fine strategy, progressively increasing the resolution by iteratively updating the intermediate flows and soft fusion mask through successive IFBlocks. RIFE presents an effective framework for video frame interpolation, offering faster processing speed compared to other popular methods, including Super SloMo and DAIN. Our work focuses on modifying the architecture of IFBlocks to enhance the optical flow estimation process, ensuring a balance between time complexity and achieving our objectives of extending the video length and improving motion smoothness.

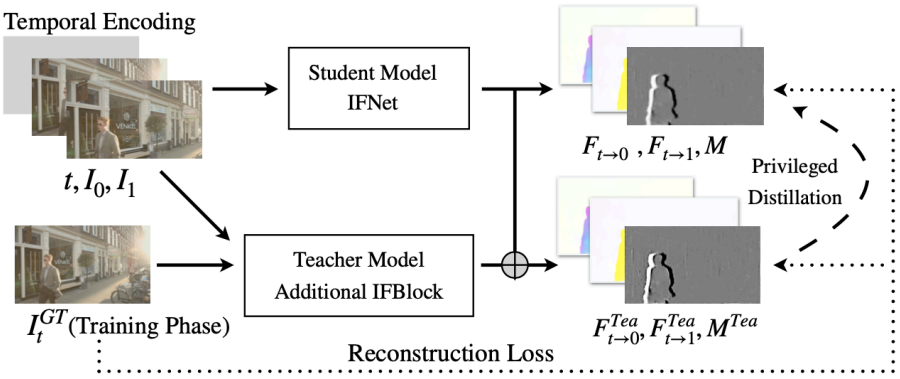

Figure 1: Overview of RIFE pipeline.

2 Methodology

2.1 Design

During the Design phase of the project, we thoroughly reviewed and refined our plans in response to the situations and challenges encountered. In the following sections, we discuss the major aspects of our design.

2.1.1 Pipeline Overview

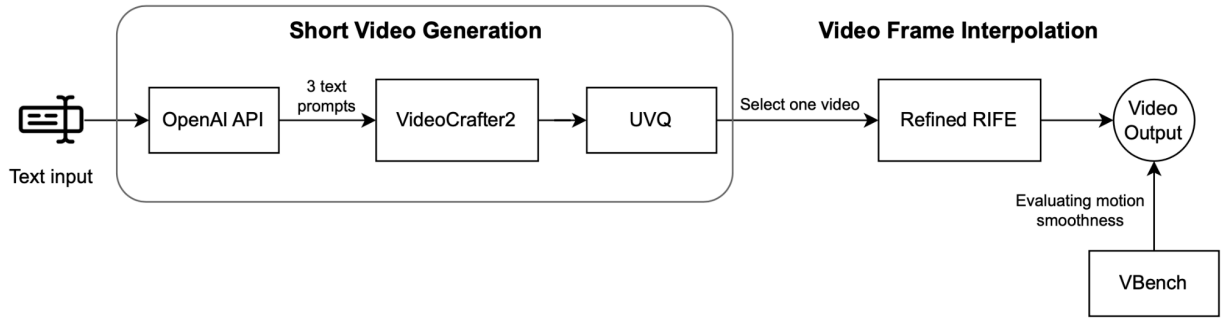

Figure 2: Pipeline Overview.

Our T2V pipeline includes 4 major modules:

1. Integration of OpenAI GPT API [12] to introduce two additional prompts by modifying the original prompt.
2. Utilization of a T2V base model for generating three short videos.
3. Inclusion of a Video Quality Assessment (VQA) model - UVQ model [32] to select the video with the highest Mean Opinion Score (MOS).
4. Incorporation of an interpolation model, an enhanced version of Real-Time Intermediate Flow Estimation (RIFE) [33], to generate additional frames based on the existing ones, thereby extending the video length.

2.1.1 Design Short Video Generation Framework

To address the issue of the T2V model producing similar videos for the same prompt, we introduced the OpenAI GPT API. This API allows us to modify user prompts, thereby increasing

semantic variability and fostering diverse video generation outcomes. We selected VideoCrafter2 [5], a T2V diffusion model, as the foundation for our project. VideoCrafter2 [5] is a further improvement of VideoCrafter1 [6], both sharing similar architectures ***Figure 3***. While many existing T2V models rely on large-scale and well-filtered datasets to attain satisfactory results, VideoCrafter2 endeavors to demonstrate that training the spatial and temporal modules comprehensively using both low-quality videos and high-quality images can also yield high-quality video outputs. This framework allows the generation of longer videos by generating future latent codes in an autoregressive manner based on prior ones.

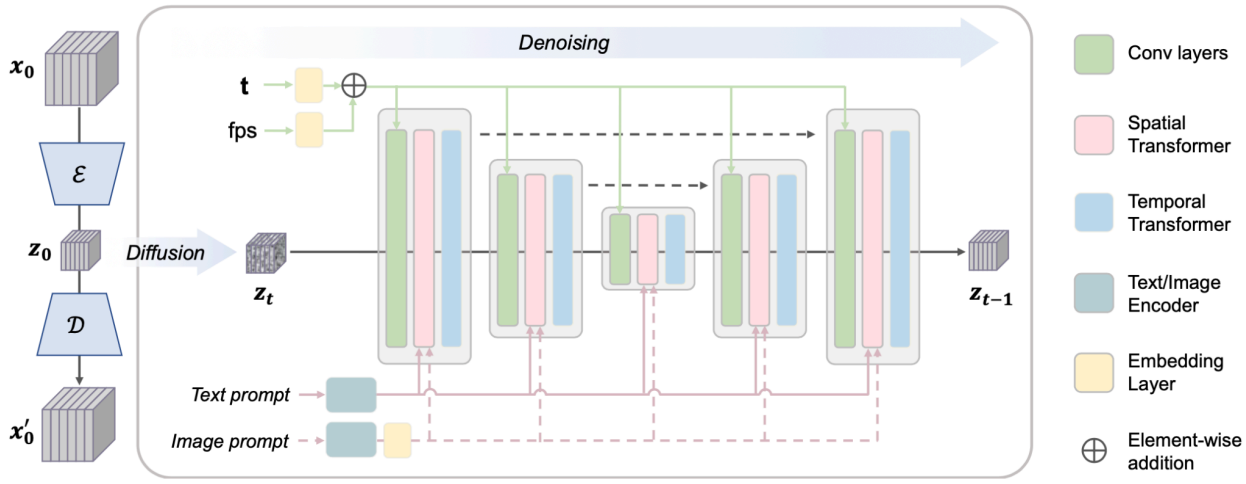

Figure 3: VideoCrafter T2V model is a Latent Video Diffusion Model (LVDM) consisting of two key components: a video Variational autoencoder (VAE) and a video latent diffusion model.

To enhance the decision-making process, we incorporated the UVQ model, developed by Google, as an additional Video Quality Assessment (VQA) model [32]. The UVQ model contains four main parts: ContentNet, DistortionNet, CompressionNet, and AggregationNet; the former three subnetworks each correspond to a factor that affects video quality perception. For example, DistortionNet detects various visual distortions, such as jitter and lens blur. With the help of its subnetworks, the model provides a single quality score called mean opinion score (MOS), where 1 represents the lowest quality and 5 represents the highest. ***Figure 4*** shows the Comprehensive Interpretation Network for Video Quality (CoINVQ) framework that is used in the UVQ model. While the T2V model generates multiple videos during the process, only one with the highest mean opinion score (MOS) will be chosen.

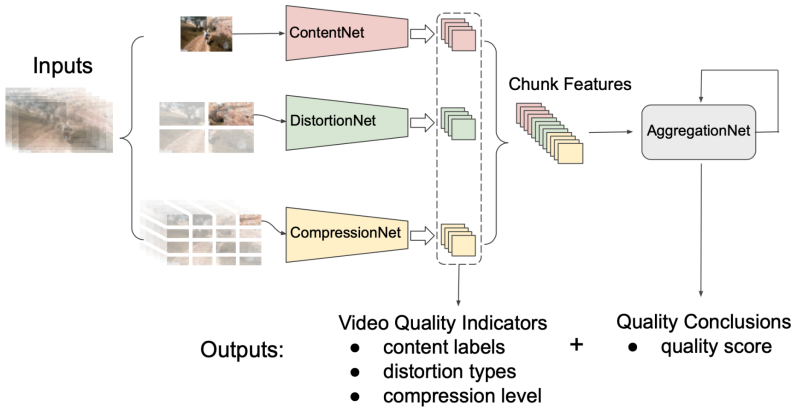

Figure 4: Comprehensive Interpretation Network for Video Quality (CoINVQ) framework. The output of three subnetworks is aggregated to the final model for quality score.

2.1.2 Design Video Interpolation Process

We incorporated an interpolation model to further improve the smoothness of the generated content. The UVQ model will select the desired short video, which will then undergo the interpolation process to generate additional frames and extend the duration of the video. Our choice for this purpose is Real-Time Intermediate Flow Estimation (RIFE) [33]. In the context of flow-based methods, given the input frames I_0 and I_1, the goal is to approximate the intermediate flows F_0 and F_1 from the perspective of the frame I_t, which is to be synthesized, illustrated in ***Eq(1)***, ***Eq(2)*** and ***Figure 5***. In the following formulations, w is the image backward warping and and M is the fusion map ($0 \leq M \leq 1$).

$$\widehat{I}_t = M \odot \widehat{I}_0 + (1 - M) \odot \widehat{I}_1, \qquad \textbf{Eq (1)}$$

$$\widehat{I}_0 = w(I_0, F_0), \widehat{I}_1 = w(I_1, F_1), \qquad \textbf{Eq (2)}$$

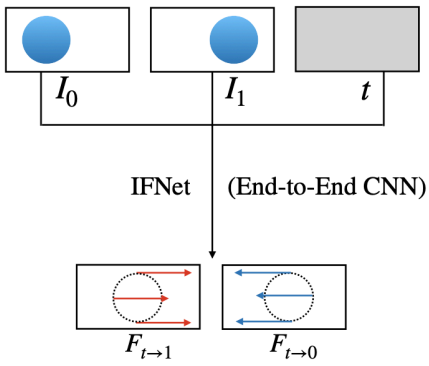

Figure 5: Using IFNet, CNNs can learn intermediate flow estimates end-to-end.

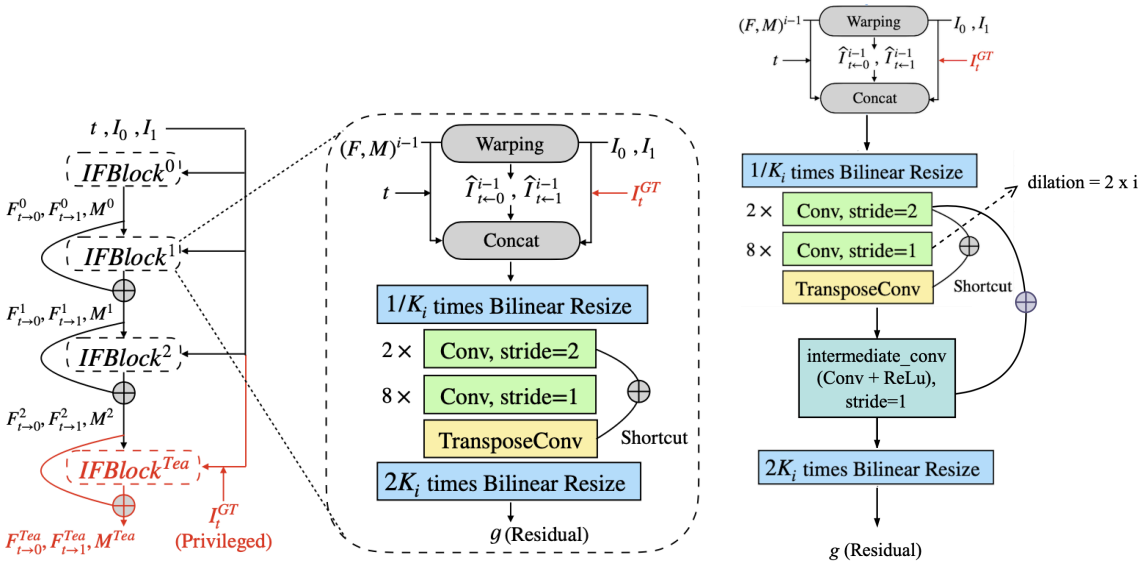

Figure 6: The IFNet consists of 3 IFBlocks and each with different resolution parameters (K0, K1, K2) = (4,2,1). During training, a privileged Teacher provides direct supervision by refining the Student results using ground truth data. Compare IFBlock architecture: original (left) and modified (right).

Building upon the achievements of RIFE, we have made some improvements to the IFBlocks module. In the modified version, we introduced several enhancements, as illustrated in ***Figure 6***. Firstly, we incorporated dilated convolutions by adding a dilation parameter to the Conv layers. This inclusion allows the IFBlock module to better capture long-range dependencies and extract global contextual information, potentially leading to improved performance. Additionally, we introduced an intermediate convolutional layer, followed by a ReLU activation function. By

applying the intermediate convolutional layer, we augment the model's capacity to learn intricate patterns and enhance representation learning. Furthermore, we implemented an additional residual connection, combining the original intermediate feature maps with the refined ones output by the intermediate convolutional layer. This enables the module to facilitate gradient flow during training and promotes the modeling of complex transformations.

These modifications collectively contribute to the enhanced performance and effectiveness of the IFBlocks module. By using the refined version of RIFE, we can effectively estimate optical flows and predict intermediate frames, enabling us to achieve 4 times interpolation while maintaining fast processing speed.

2.1.3 Design Evaluation Methods

Once the T2V generation pipeline is established, it becomes crucial to establish a method for evaluating the output. Given our objective of enhancing motion smoothness, we have chosen to utilize VBench [36], an evaluation tool specifically designed for video generative models. VBench offers a hierarchical Evaluation Dimension Suite encompassing 16 dimensions, including motion smoothness among others. This tool aligns perfectly with our requirement to accurately measure the improvement achieved in motion smoothness, thus providing the ideal evaluation framework for our project.

The evaluation method can be divided into two primary aspects: user feedback and VBench evaluation. Despite the evaluation model, recognizing that humans are the ultimate users of our model, we believe that the evaluation should include feedback from human observers in addition to VQA tools. Categorizing and visualizing the results based on user feedback can provide further validation of the improvements attained.

2.1.4 Design User Interface

The final stage of the project is developing a user interface that not only showcases the performance of our model but also provides an intuitive experience for our users. The initial design of our interface is shown below ***Figure 7***.

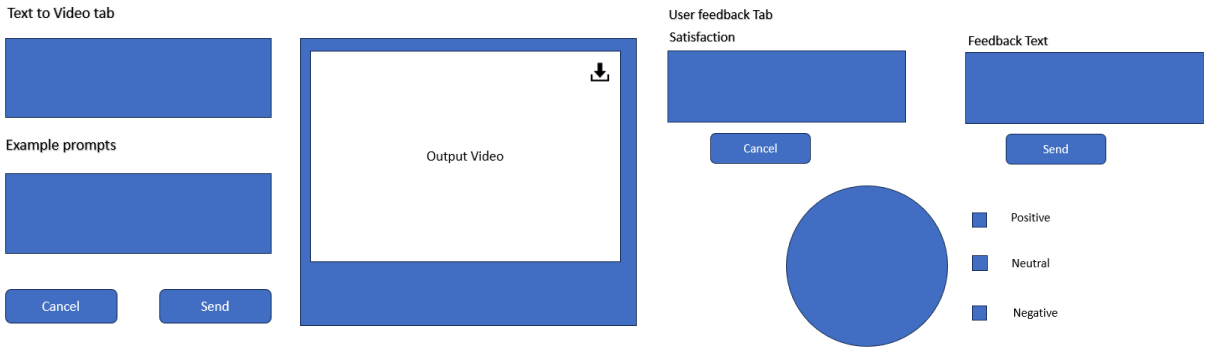

Figure 7: The design of the interface. The interface is composed of a "Text-to-video" tab (left) and a "User feedback" tab (right).

The interface comprises two distinct tabs, each catering to different functionalities: T2V Generation and User Feedback. Within the T2V Generation tab, users are provided with the capability to input text prompts, adjust the desired video length, and access suggested prompts for constructing a storyline. The resulting video is displayed on the right side of the interface, and users have the option to download the generated video.

The User Feedback tab is designed to gather valuable input from users. It includes three rating categories that allow users to provide an overall assessment of their experience. Additionally, a text box is available for users to provide specific feedback or additional comments. The ratings for each category are visualized in an interactive pie chart.

2.2 Implementation

The Implementation section provides a comprehensive explanation of our components, including the underlying mechanisms and algorithms employed in our framework.

2.2.1 GPT-based User Input Modification

The OpenAI GPT API serves as a simple interface, enabling the utilization of the GPT-3.5 model. To access the OpenAI API, we begin by creating an API key through the OpenAI account page. In Google Colab, a POST request is made to the designated OpenAI GPT-3 API endpoint. This request includes headers that define the content type and authorize access using the API key. Additionally, the API offers parameters such as *temperature* and *max_tokens*, which provide control over the response generated.

The output produced by the API undergoes preprocessing to eliminate any extraneous wording. These preprocessed results are then appended to the original text prompts file. Considering our default configuration, a total of three prompts, including the original and two GPT-modified prompts, are stored in the text prompts file, ready to be utilized as input for the T2V model.

2.2.2 Integration of T2V and VQA models

The base model of VideoCrafter2, LVDM [30], encompasses a video Variational Autoencoder (VAE) and a video latent diffusion model. The pretrained VAE from the Stable Diffusion model [14] is employed, where the video data undergoes compression into lower-dimensional representations through an Encoder then reconstructed back using a Decoder. The video latent diffusion model utilizes a 3D U-Net architecture, comprising multiple spatial-temporal blocks with skip connections. This architecture enables effective information flow and captures essential temporal dependencies.

By inputting the text prompt file into this model, we generate three videos that share similar content. These videos are then passed through the Universal Video Quality (UVQ) Model, which assesses their quality based on three key aspects: content, distortion, and compression. The UVQ model outputs Mean Opinion Score (MOS) in the form of a CSV file for further processing. The UVQ model was trained using self-supervised learning on large-scale samples, without relying on ground-truth quality scores. By utilizing the UVQ model, we select the short video with the highest MOS as the optimal output for further processing in the VideoCraftXtend system.

2.2.3 Refining the Interpolation Model

Training Dataset

To train the refined RIFE [33], we used a triplet dataset extracted from the Vimeo-90k Dataset [28]. This dataset consists of 51,312 three-frame sequences for training and validation, each maintaining a fixed resolution of 448 x 256. The dataset is organized in a folder structure, with clips, sequences, and images contained within the 'sequences' directory of the Vimeo-90K-triplet Dataset. The complete dataset file structure is shown in ***Figure 8***. As part of the data preprocessing steps, the frames undergo an initial cropping to dimensions of (224, 224). To augment the data, we applied random transformations such as horizontal and vertical flipping, reversing the temporal order, and rotating the frames by 90 degrees.

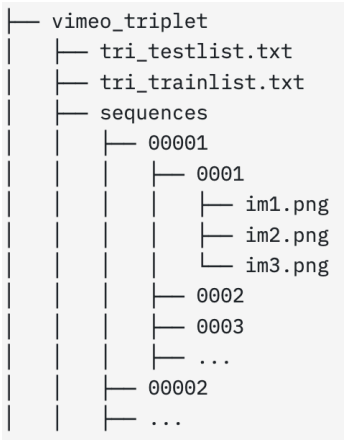

Figure 8: Vimeo-90K-triplet Dataset structure.

Training Details

We trained the refined RIFE on Google Colab Pro using A100 Nvidia GPU with System RAM 51GB, GPU RAM 16GB, and Disk 201 GB. RIFE is optimized by AdamW [2] with learning rate = 10^{-6} and weight decay = 10^{-4}. We set batch size = 64 and the number of epochs = 300.

Once the short video has been generated and a single video is selected, we pass the chosen video through the interpolation model for 4-times interpolation, meaning this process extends the video from its initial length of 20 frames to 80 frames in our settings. By adjusting the fps parameter, we can control the resulting video duration based on these 80 frames. The outcome is a more seamless and visually pleasing viewing experience for our users. The interpolated frames fill in the gaps between keyframes, resulting in smoother motion and improved transitions throughout the video.

2.2.4 Develop the User Interface

To provide users with direct access to the model's results through an interface, we developed a demo utilizing Gradio [26]. Gradio is a Python library that enables the creation of custom UIs for machine learning models, allowing them to be integrated into Jupyter notebooks or presented as web pages. The user interface consists of two distinct sections: the video generation tab and the feedback tab, as depicted in ***Figure 9*** and ***Figure 10***. For collecting user feedback, we leveraged Firebase as our backend storage solution. Within the database, there exists a collection named "user_feedbacks", which stores two variables: "experience" and "feedback_text".

The functions and steps of using the interface are outlined as follows:

1. Users have the option to either select an example prompt or input their own prompt.
2. Upon clicking the "Send" button, the video generation process starts.
3. After approximately 15 minutes, the generated video appears in the result column. Users can choose to download this video.
4. The examples provided in the interface adapt to each video generation, culminating in a coherent storyline.
5. Users can navigate to the feedback tab to share their experience and provide feedback in textual form.
6. The pie chart dynamically updates and responds to each submission, ensuring real-time interaction and interface updates.

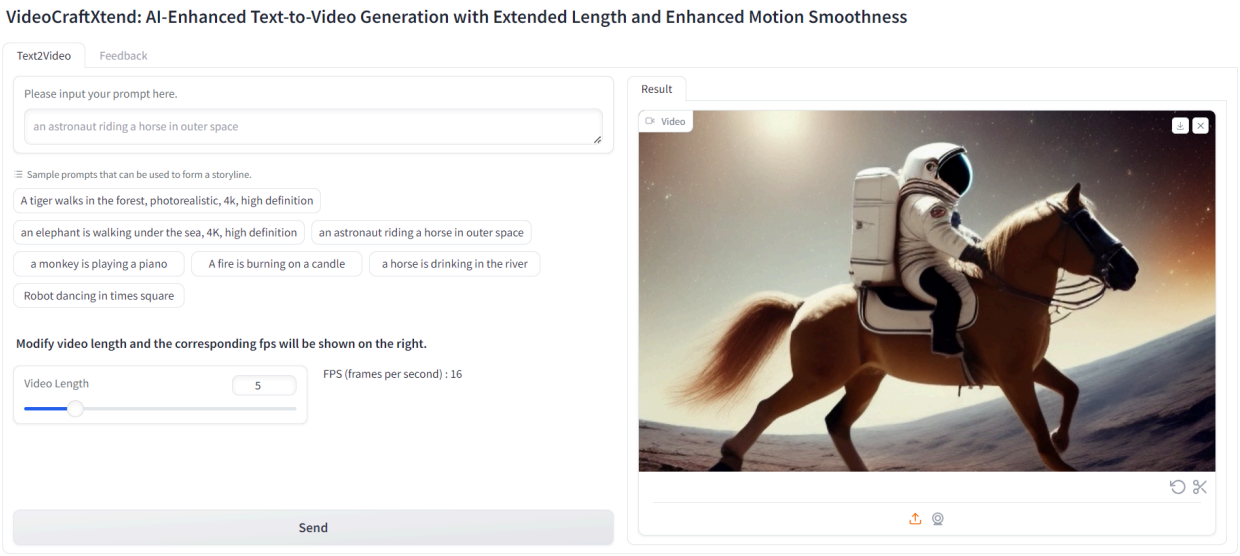

Figure 9: T2V Tab. In this tab, users can input their prompts, and the model will generate and display the corresponding output video. With each submission, the sample prompts will dynamically update to align with the given prompt, forming a coherent storyline.

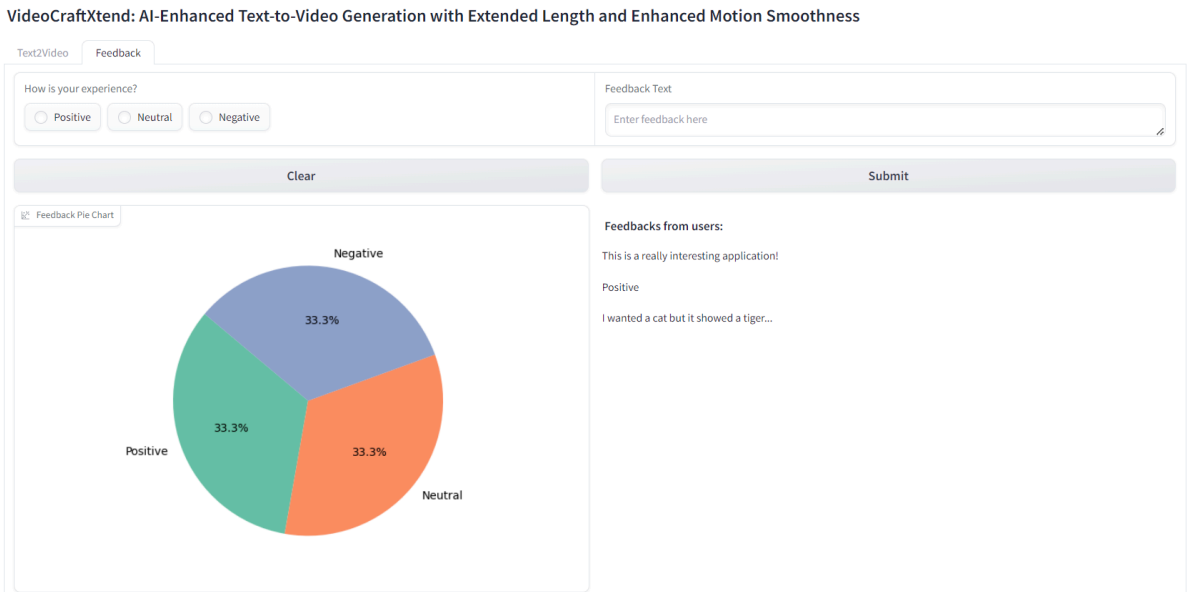

Figure 10: User Feedback tab. In this tab, users can rate their experience and provide feedback.

2.3 Testing

Throughout the development process, we conducted unit testing to ensure the accurate construction of all modules. Our testing efforts were categorized into the following sections, and we also plan to undertake further comprehensive testing listed below.

2.3.1 Testing the Short Video Framework

Testing different instructions to GPT API

Testing the ChatGPT API involved a comprehensive assessment of its functionality and resilience. To enhance the model's robustness, we conducted tests by modifying key parameters, including *temperature*, *max tokens*, and instruction prompts. We explored 5 different prompt wordings when interacting with the GPT API and examined the effects of two temperature settings: 0.4 and 0.5. Additionally, we set the *max_tokens* parameter to 200 to avoid premature truncation of responses. The results are presented in ***Table 1.*** This testing process allowed us to refine and optimize the performance of the GPT API, ensuring its reliability and effectiveness in generating coherent and contextually relevant responses. ***Figure 11*** illustrates the impact of prompt modifications on enhancing the diversity of video content.

input prompt	output result	temperature	max tokens
Generate 2 similar prompts and add some reasonable words to the given prompt and not change the meaning, each within 30 words: an elephant is walking under the sea, 4K, high definition	['An elephant is majestically strolling beneath the sea, showcasing its colossal size and grace, all captured in stunning 4K high definition quality.', 'Witness the breathtaking sight of an elephant leisurely traversing the underwater world, beautifully filmed in mesmerizing 4K high definition for an unparalleled viewing experience.']	0.5	200
Generate 2 similar prompts similar to the given prompt: an elephant is walking under the sea, 4K, high definition	['A polar bear is swimming in a volcano, 8K, ultra high definition', 'A giraffe is flying through outer space, 6K, crystal clear resolution']	0.4	200
Generate 2 prompts similar to the given prompt: an elephant is walking under the sea, 4K, high definition	['A whale is soaring through the clouds, captured in ultra-high definition.', 'A mermaid is dancing in a crystal-clear underwater cave, filmed in 4K resolution.']	0.5	200
Generate 2 prompts according to the given prompt: an elephant is walking under the sea, 4K, high definition	['Imagine an elephant gracefully strolling through the depths of the ocean, captured in stunning 4K high definition, as it explores the underwater world with majestic elegance.', 'Picture the incredible sight of an elephant moving effortlessly beneath the sea, its every detail crystal clear in 4K high definition, creating a mesmerizing and surreal visual experience.']	0.4	200

Table 1: Trials using different parameters and input prompts (instructions) to ChatGPT API.

Original text prompt: `'A tiger walks in the forest, photorealistic, 4k, high definition'`

Original text prompt: `'a monkey is playing a piano'`

Figure 11: The first column is the T2V output generated by original text prompts. The subsequent two columns display the output from modified prompts.

After evaluation, any prompts fed to ChatGPT starting with "`Generate 2 similar sentences…`" would result in the output being unpredictable. Hence, we have selected the

following prompt: "`Generate 2 similar prompts and add some reasonable words to the given prompt and not change the meaning, each within 30 words: {Original Prompt}`", where {Gpt generated prompt 1, Gpt generated prompt2} will be generated. Therefore, we use {Gpt generated prompt 1, Gpt generated prompt2, Original Prompt} for short video generation.

Testing the UVQ model using VideoCrafter2 output videos

In order to understand the quality of the video output generated by VideoCrafter2, we have tested 10 random prompts and their corresponding MOS using the UVQ model. The results are shown in ***Table 2***.

video num	MOS
0001	3.345633554458618
0002	3.431928873062134
0003	3.411888027191162
0004	3.8738601207733154
0005	3.4962512016296388
0006	3.5615002155303954
0007	3.835473966598511
0008	3.8283190727233887
0009	3.4710131645202638
00010	3.4710131645202638
Average	3.572688136100769

Table 2: MOS across 10 prompts

Comparing VBench and UVQ model

To gain a deeper understanding of the evaluation dimensions of each model, we performed tests to compare the evaluation scores produced by VBench and the UVQ model. ***Table 3*** presents the scores obtained for a sample prompt. Through multiple experiment iterations, we noticed a discrepancy between the evaluation scores generated by VBench and the UVQ model. Upon closer examination, we discovered that the UVQ score did not align entirely with the human perspective, meaning that a better video, from a human viewpoint, might receive a lower Mean Opinion Score (MOS). This observation led us to conclude that the introduction of VBench is necessary, as the MOS alone does not effectively measure motion smoothness quality independently.

Prompt	VBench scores	MOS
Gpt generated prompt 1	0.9691780528038761	3.6391905307769776
Gpt generated prompt 2	0.9866513613366353	3.440376377105713
Original prompt	0.970816610498366	3.656974744796753

Table 3: Evaluation Results using a sample prompt

Testing the Effectiveness of the Interpolation Model

To assess the effectiveness of the refined interpolation model, we conducted an assessment by averaging the scores of 10 videos both before and after interpolation. Utilizing the VBench score, we observed a notable 5% increase in the average score, as shown in Table 4.

	VBench score	Increased Percentage
Before Interpolation	0.95	-
After Interpolation	0.998	5%

Table 4: Average VBench score over 10 videos

2.3.2 Testing the refined Interpolation Model using VBench

Our interpolation model plays a crucial role in improving the motion smoothness throughout the video. As shown in ***Figure 12***, when provided with a prompt such as `'a monkey is playing a piano'`, the interpolation model generates multiple frames between the original frames (a) and (b). These additional frames effectively mitigate motion artifacts by slowing down the transitions between movements. To illustrate the impact of this improvement, ***Figure 13*** displays the differences between frames by adjusting the transparency and overlapping the images. The motion smoothness enhancement becomes even more apparent when the results are demonstrated in the form of a video.

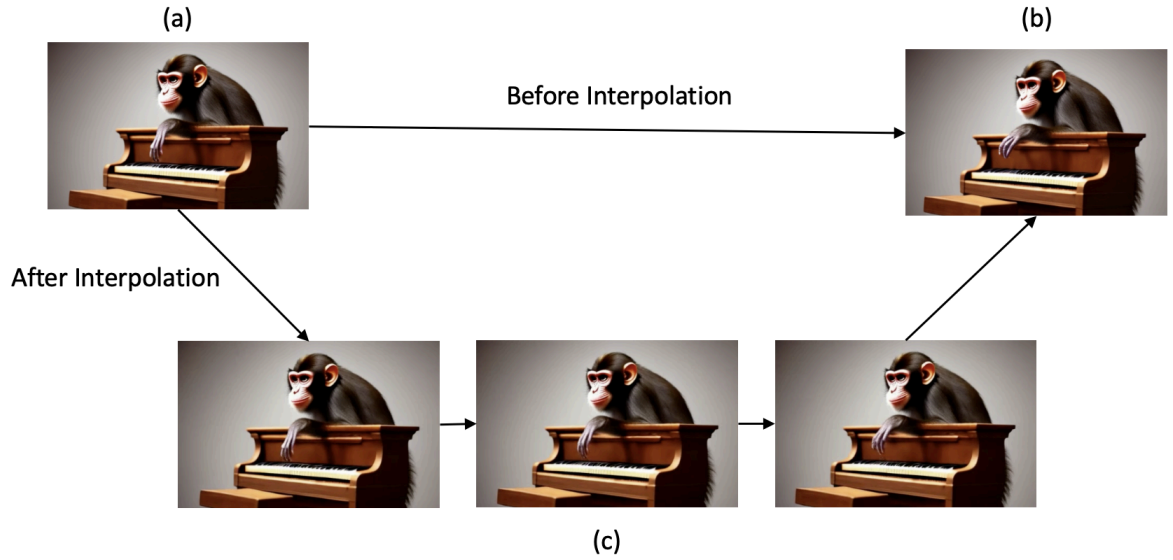

Figure 12: The interpolation model inserts multiple frames between two given frames, slowing down the transitions between movements.

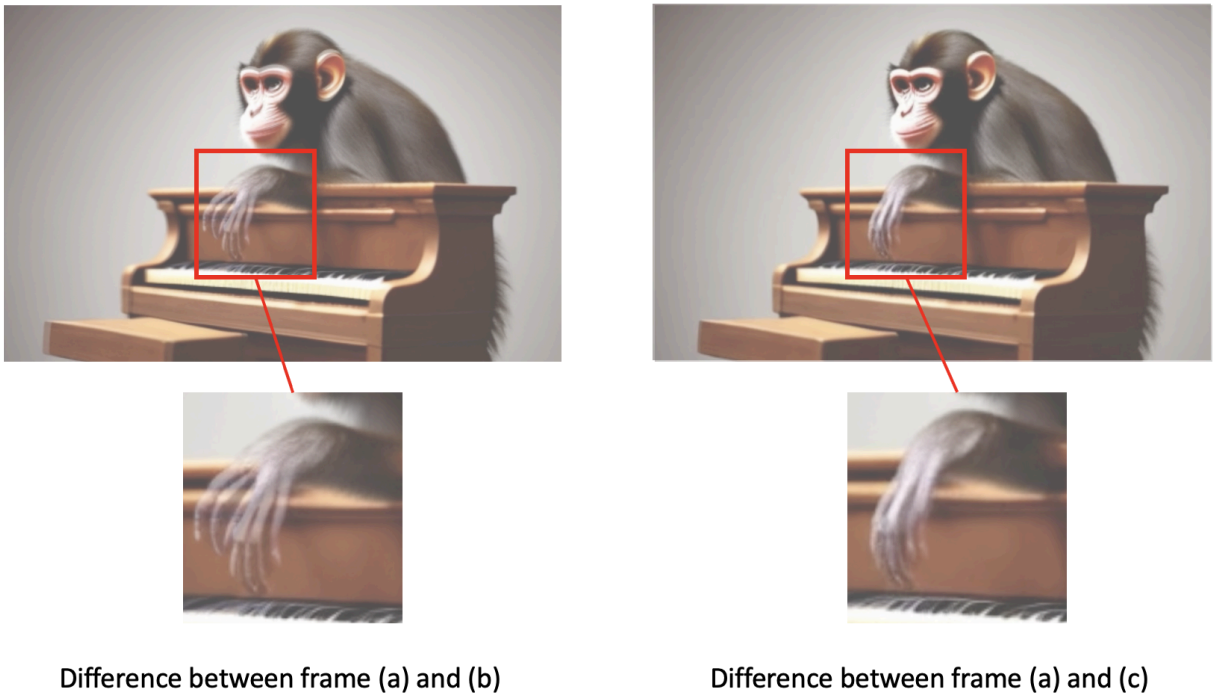

Figure 13: Overlapping different frames to visualize the differences in hand motion.

2.4 Evaluation

2.4.1 Using GPT API for Improving Content Diversity

Our primary objective was to leverage the OpenAI GPT API to enhance video content diversity by augmenting prompts with additional words. This accomplishment is visibly evident upon visual inspection in ***Figure 11***. In the testing phase, we encountered various challenges that we successfully resolved. Initially, we faced a challenge where we observed that GPT occasionally generated sentences unrelated to the prompts or included numerical indexes. To overcome this, we conducted multiple trials by refining the instructions provided to GPT and implemented a stripping mechanism to remove leading numerical indexes. Furthermore, we noticed that GPT often extended the original prompt excessively, resulting in sentences that resembled paragraphs. To address this concern, we implemented a solution by imposing a constraint that each modified sentence should not exceed 30 words in length.

While the improvement in content diversity has been achieved, it is important to acknowledge some limitations. One such limitation is the cost associated with the GPT API usage. Depending on the pricing structure and individual usage requirements, users may find the associated costs to be prohibitive, potentially impeding widespread adoption. To address this concern, future work could involve the development of a Language Model to mitigate cost-related challenges.

2.4.2 Refine Interpolation Model

Our second objective of training an interpolation model to extend video length and enhance motion smoothness has been successfully accomplished. By refining the IFBlock within RIFE and employing 4 times interpolation, we were able to extend the video duration up to 10 seconds. Furthermore, when evaluated by VBench, the output videos exhibited an average score increase of 5%, indicating improved motion smoothness. This improvement is also evident from a human perspective. The interpolation model effectively fills in multiple frames between two frames, resulting in a significant reduction in motion differences, shown in ***Figure 12*** and ***Figure 13***. As a result, the final output exhibits fewer motion artifacts, enhancing the overall visual quality of the video.

2.4.3 Building User Interface

One of our objectives was to create a user-friendly interface that provides convenient access to our model. We successfully achieved this by utilizing Gradio. Through this interface, users have the flexibility to customize the desired video length according to their preferences and easily download the generated video directly. Moreover, we integrated a GPT suggested prompts section. After users submit their initial prompt, they are presented with a selection of additional prompts within the interface, enabling them to expand the story based on the current prompt and explore new creative possibilities. As an additional feature, we have created a dedicated feedback page to gather user feedback. Users have the opportunity to rate their experience and provide comments.

One existing limitation of the interface is its current restriction to specifying only the desired video length. As a result, when the video length increases, there is a trade-off that needs to be considered in terms of frames per second (fps). This limitation is rooted in the underlying settings of our T2V model, as we have imposed constraints on the number of inference steps. To overcome this limitation, future work can focus on balancing the time complexity and computational constraints of the T2V model.

3 Discussion

3.1 Video Generation Time and Quality

In the development of our system, our primary focus revolved around improving the motion smoothness and extending the video length. Initially, we also considered reducing the generation time as an important objective. However, as our project progressed and we introduced additional features such as the GPT API, VQA, and Video Frame Interpolation, our priorities shifted.

The inclusion of these features brought about a change in our perspective. Currently, our framework takes approximately 15 minutes to generate a video. While we recognize the importance of efficient generation time, we believe this duration falls within an acceptable range considering the visually pleasing and extended videos we produce. The extended time required for generation allows us to focus on achieving enhanced motion smoothness and the ability to create longer videos, factors that significantly contribute to an immersive and satisfactory user experience.

3.2 T2V Model Constraints

While we prioritized the selection of a lightweight T2V base model, VideoCrafter2 remains the limiting factor in terms of generation time and computational capacity. To prevent GPU memory issues, we imposed a constraint on the number of inference steps. However, there is still potential for enhancements in the autoregressive techniques utilized for frame generation. If the base T2V model can already produce videos of sufficient length and desired frame rate, additional processing involving VQA and the Interpolation model can further enhance the overall result.

4 Conclusion

4.1 Achievements

In conclusion, our project has addressed some of the key challenges in Text-to-Video (T2V) generation. We have successfully achieved milestones in three key areas: extending video length, enhancing motion smoothness quality, and improving video content diversity. This accomplishment was made possible through the integration and refinement of a T2V diffusion model, OpenAI GPT API, Video Quality Assessment (VQA) model, and Video Interpolation model.

Our T2V pipeline showcases its capability to generate videos with a maximum duration of 10 seconds, while also achieving an average increase of 2-5% in motion smoothness scores as measured by VBench. These substantial improvements in motion smoothness serve as a validation of the effectiveness of our approach and contribute to creating visually appealing videos. Furthermore, the development of a user-friendly interface using Gradio has facilitated seamless interaction, empowering users to input prompts, define video output parameters, and download the resulting videos. Additionally, the interface offers several sample prompts to assist users in constructing coherent storylines, further enhancing the creative process and providing valuable guidance.

4.2 Future Work

Looking ahead, several avenues for future exploration and refinement could further advance our T2V framework:

Optimizing VideoCrafter2 Model

With sufficient computational resources, retraining the VideoCrafter2 model could lead to improved performance and expanded capabilities. One of the reasons we selected VideoCrafter2 is its ability to be trained or fine-tuned on low-quality datasets. By optimizing training processes and model architectures, we can streamline the T2V generation pipeline, making it more efficient. This would involve fine-tuning model parameters, exploring larger datasets, and incorporating advanced techniques to enhance video generation quality and diversity.

Incorporating Interpolation into the T2V Model

Integrating interpolation techniques directly into the T2V diffusion model itself offers potential for reducing overall time complexity. By optimizing the generation process and seamlessly integrating motion enhancement methods, we can further enhance the efficiency and effectiveness of our T2V pipeline.

Improving Interface Functions

Expanding the functionality of the user interface to include features such as video concatenation of the entire story. With the interface able to concatenate videos per each generation, users can easily compile cohesive narratives from individual video segments, enhancing storytelling capabilities and enlightening user experiences.

5 References

[1] A. Dirik, “A dive into text-to-video models,” A Dive into Text-to-Video Models, https://huggingface.co/blog/text-to-video.

[2] “AdamW”, Keras, https://keras.io/api/optimizers/adamw/.

[3] “A dive into text-to-video models,” A Dive into Text-to-Video Models, https://huggingface.co/blog/text-to-video.

[4] Gradio-App, “Gradio-app/gradio,” GitHub, https://github.com/gradio-app/gradio.

[5] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, Ying Shan. “VideoCrafter2: Overcoming Data Limitations for High-Quality Video Diffusion Models”, arXiv.org, https://arxiv.org/pdf/2401.09047.pdf.

[6] H. Chen et al., “Videocrafter1: Open diffusion models for high-quality video generation,” arXiv.org, https://arxiv.org/abs/2310.19512.

[7] H. Jian, D. Sun, V. Jampani, M. Yang, Erik Learned-Miller, Jan Kautz, “Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation”, arXiv.org, https://arxiv.org/pdf/1712.00080v2.pdf.

[8] I. J. Goodfellow et al., “Generative Adversarial Networks,” arXiv.org, https://arxiv.org/abs/1406.2661.

[9] J. Ho et al., “Imagen Video: High Definition Video Generation with Diffusion Models”. arXiv.org, https://arxiv.org/abs/2210.02303

[10] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, “Frozen in time: A joint video and image encoder for end-to-end retrieval,” arXiv.org, https://arxiv.org/abs/2104.00650.

[11] “Midjourney”. GitHub, https://github.com/midjourney

[12] OpenAI GPT API reference. https://platform.openai.com/docs/api-reference

[13] “Papers with code - tai-chi-HD dataset,” Dataset | Papers With Code, https://paperswithcode.com/dataset/tai-chi-hd.

[14] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with Latent Diffusion Models,” arXiv.org, https://arxiv.org/abs/2112.10752.

[15] S. Ge et al., “Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer,” European Computer Vision Association,

https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136770103.pdf.

[16] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. SorkineHornung. Phase-based frame interpolation for video. In CVPR, 2015

[17] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In ICCV, 2017.

[18] T. Xue, B. Chen, J.Wu, D. Wei, W. T. Freeman, MIT CSAIL , Harvard University, Google Research. “Video Enhancement with Task-Oriented Flow”. IJCV 2019. http://toflow.csail.mit.edu/.

[19] T. Höppe, A. Mehrjou, S. Bauer, D. Nielsen, and A. Dittadi, “Diffusion models for video prediction and infilling,” arXiv.org, https://arxiv.org/abs/2206.07696.

[20] T. Unterthiner et al., “FVD: A new metric for Video generation - openreview,” Open Review, https://openreview.net/pdf?id=rylgEULtdN.

[21] “UCF101 - Action Recognition Data Set,” CRCV, https://www.crcv.ucf.edu/data/UCF101.php.

[22] U. Singer et al., “Make-A-video: Text-to-video generation without text-video data,” arXiv.org, https://arxiv.org/abs/2209.14792.

[23] V. Voleti, A. Jolicoeur-Martineau, and C. Pal, “MCVD: Masked conditional video diffusion for prediction, generation, and interpolation,” arXiv.org, https://arxiv.org/abs/2205.09853.

[24] W. Bao et al., “Depth-aware video frame interpolation,” arXiv.org, https://arxiv.org/abs/1904.00830.

[25] W. Harvey, S. Naderiparizi, V. Masrani, C. Weilbach, and F. Wood, “Flexible diffusion modeling of long videos,” arXiv.org, https://arxiv.org/abs/2205.11495.

[26] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang, “CogVideo: Large-scale pretraining for text-to-video generation via Transformers,” arXiv.org, https://arxiv.org/abs/2205.15868.

[27] W. Xiong, W. Luo, L. Ma, W. Liu, and J. Luo, “Learning to generate time-lapse videos using multi-stage dynamic generative Adversarial Networks,” arXiv.org, https://arxiv.org/abs/1709.07592.

[28] X.-L. Yin et al., “Reducing the X-ray radiation exposure frequency in cardio-angiography via deep-learning based video interpolation,” arXiv.org, https://arxiv.org/abs/2006.00781.

[29] Xue et al.”Video Enhancement with Task-Oriented Flow”. Dataset| Papers with Code. https://paperswithcode.com/dataset/vimeo90k-1

[30] Y. He, T. Yang, Y. Zhang, Y. Shan, and Q. Chen, “Latent video diffusion models for high-fidelity long video generation,” arXiv.org, https://arxiv.org/abs/2211.13221.

[31] Y. Wang et al., “Rich features for Perceptual Quality Assessment of UGC Videos,” CVF Open Access, https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Rich_Features_for_Perceptual_Quality_Assessment_of_UGC_Videos_CVPR_2021_paper.html.

[32] Y. Wang and F. Yang, “UVQ: Measuring YouTube’s Perceptual Video Quality,” Google Research Blog, https://blog.research.google/2022/08/uvq-measuring-youtubes-perceptual-video.html.

[33] Z. Huang, T. Zhang, W. Heng, B. Shi, S. Zhou. “Real-Time Intermediate Flow Estimation for Video Frame Interpolation”. arXiv.org, https://arxiv.org/abs/2011.06294.

[34] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In ICCV, 2017.

[35] Z. Tu, Y. Wang, N. Birkbeck, B. Adsumilli, and A. C. Bovik, “UGC-VQA: Benchmarking Blind Video Quality Assessment for user generated content,” arXiv.org, https://arxiv.org/abs/2005.14354.

[36] Z. Huang et al., “VBench: Comprehensive Benchmark Suite for Video Generative Models”, arXiv.org, https://arxiv.org/abs/2311.17982.