

**A  
PROJECT REPORT  
ON**

# **Text to Video Generation**

**SUBMITTED TO  
SHIVAJI UNIVERSITY, KOLHAPUR  
IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF  
DEGREE BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND  
ENGINEERING**

**SUBMITTED BY**

<b>MISS. CHOTHE AISHWARYA MAHADEV</b>	<b>19UCS021</b>
<b>MISS. DHAVAL KALYANI RAJARAM</b>	<b>19UCS031</b>
<b>MISS. GOILKAR VAISHNAVI RAVINDRA</b>	<b>19UCS040</b>
<b>MISS. HIRAVE PRANOTI PRAMOD</b>	<b>19UCS044</b>
<b>MISS. JADHAV MANALI MALLIKARJUN</b>	<b>19UCS046</b>

**UNDER THE GUIDANCE OF**

**Dr. S. S. More**



Promoting Excellence in  
Teaching, Learning & Research

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
DKTE SOCIETY'S TEXTILE AND ENGINEERING INSTITUTE,  
ICHALKARANJI 2022-23**

**D.K.T.E. SOCIETY'S**  
**TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI**  
**(AN AUTONOMOUS INSTITUTE)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



Promoting Excellence in  
Teaching, Learning & Research

# **CERTIFICATE**

**This is to certify that, project work entitled**

## **Text to Video Generation**

**Is a bonafide record of project work carried out in this college by**

<b>MISS. CHOTHE AISHWARYA MAHADEV</b>	<b>19UCS021</b>
<b>MISS. DHAVAL KALYANI RAJARAM</b>	<b>19UCS031</b>
<b>MISS. GOILKAR VAISHNAVI RAVINDRA</b>	<b>19UCS040</b>
<b>MISS. HIRAVE PRANOTI PRAMOD</b>	<b>19UCS044</b>
<b>MISS. JADHAV MANALI MALLIKARJUN</b>	<b>19UCS046</b>

**is in the partial fulfillment of award of degree Bachelor's in technology in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2022-2023.**

**Dr. S. S. MORE**

**(PROJECT GUIDE)**

**PROF.(DR.) D.V. KODAVADE**  
**(HOD CSE DEPT.)**

**PROF.(DR.) L. S. ADMUTHE**  
**(DIRECTOR)**

**EXAMINER: \_\_\_\_\_**

# DECLARATION

We hereby declare that the project work report entitled “Text to video generation” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.Tech.(CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

MISS. Chothe Aishwarya Mahadev	19UCS021
MISS. Dhavale Kalyani Rajaram	19UCS031
MISS. Goilkar Vaishnavi Ravindra	19UCS040
MISS. Hirave Pranoti Pramod	19UCS044
MISS. Jadhav Manali Mallikarjun	19UCS046

# ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to Dr. S. S. More for his valuable guidance, support, and encouragement in the completion of this project report.

Also, we would like to take the opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.

Thank you,

MISS. Chothe Aishwarya Mahadev	19UCS021
MISS. Dhavale Kalyani Rajaram	19UCS031
MISS. Goilkar Vaishnavi Ravindra	19UCS040
MISS. Hirave Pranoti Pramod	19UCS044
MISS. Jadhav Manali Mallikarjun	19UCS046

# **ABSTRACT**

The project focuses on generating videos from textual descriptions. It utilizes a combination of natural language processing and computer vision techniques to convert textual input into corresponding video sequences. The objective is to bridge the gap between language and visual content, enabling the generation of dynamic visual representations based on textual input.

The algorithm incorporates a deep learning architecture known as the Pseudo-3D Residual Network (Pseudo3DResNet) to handle spatio-temporal data efficiently. By extending the popular ResNet framework, the Pseudo3DResNet integrates temporal information into a 2D convolutional neural network (CNN), allowing the model to effectively capture both spatial and temporal aspects of video data.

The implementation of the Pseudo3DResNet involves the use of residual blocks with shortcut connections to address the vanishing gradient problem during training. Additionally, modules such as sinusoidal positional embedding, continuous positional bias calculation, and spatio-temporal attention are incorporated to enhance the model's ability to learn complex spatio-temporal patterns and capture long-range dependencies within video sequences.

To run the code in Google Collaboratory, users need to set up the appropriate environment and ensure the availability of necessary libraries and datasets. The code provides detailed methods and implementation details, including steps such as data pre-processing, network architecture definition, loss function and optimizer selection, training loop implementation, and model evaluation on test data.

By following the outlined steps, users can successfully train a Pseudo3DResNet model on video datasets and generate dynamic videos based on textual descriptions. This program contributes to the advancement of multimedia content generation and paves the way for applications such as video synthesis, video summarization, and interactive storytelling.

# **INDEX**

<b>1. Introduction</b>	<b>1</b>
a. Problem definition	3
b. Aim and objective of the project	4
c. Scope and limitation of the project	6
d. Timeline of the project	8
e. Project Management Plan	9
f. Project Cost	10
<b>2. Background study and literature overview</b>	<b>11</b>
a. Literature overview	11
b. Critical appraisal of other people's work	12
c. Investigation of current project and related work	13
<b>3. Requirement analysis</b>	<b>15</b>
a. Requirement Gathering	15
b. Requirement Specification	17
c. Use case Diagram	18
<b>4. System design</b>	<b>19</b>
a. Architectural Design	19
b. User Interface Design	19
c. System Modeling	20
• Dataflow Diagram	20-21
• Sequence Diagram	22
• Activity Diagram	23
• Component Diagram	23
<b>5. Implementation</b>	<b>24</b>
a. Environmental Setting for Running the Project	24
b. Detailed Description of Methods	24
c. Implementation Details	25
<b>6. Integration and Testing</b>	<b>26</b>
a. Unit Testing	26
b. Integration Testing	27
<b>7. Performance Analysis</b>	<b>28</b>
<b>8. Future Scope</b>	<b>30</b>
<b>9. Applications</b>	<b>32</b>

<b>10.Installation Guide and User Manual</b>	<b>35</b>
<b>11.Plagiarism Report</b>	<b>39</b>
<b>12.Ethics</b>	<b>41</b>
<b>13.References</b>	<b>43</b>

# 1. Introduction

Text-to-video generation is an exciting and rapidly evolving field that combines natural language processing (NLP) and computer vision techniques to automatically generate video content from textual input. This innovative technology aims to bridge the gap between language and visual understanding, enabling computers to interpret and represent text in the form of dynamic and visually appealing videos.

In the text-to-video generation process, several steps are typically involved. First, the input text undergoes NLP analysis to extract relevant information, such as entities, actions, and context. This information is then translated into visual representations, including scene descriptions, object appearances, camera movements, and character animations. Advanced algorithms and models are used to generate video sequences that align with the textual input.

Text-to-video generation has a broad range of practical applications across various industries. In the entertainment industry, it can be used to automatically create video summaries or trailers based on scripts or book summaries, providing a glimpse into the content and enticing potential viewers. In e-learning platforms, it can transform textual content into engaging and interactive educational videos, enhancing the learning experience for students.

For marketing purposes, text-to-video generation enables businesses to automatically generate promotional videos from product descriptions or customer testimonials. This streamlines the content creation process and facilitates effective marketing campaigns. Additionally, text-to-video generation can contribute to improving accessibility by generating video descriptions for visually impaired individuals, making digital content more inclusive and accessible to a wider audience.

While text-to-video generation has made significant progress, several challenges remain. Ambiguous or metaphorical language can be difficult to interpret accurately, and maintaining visual consistency throughout the generated videos is a continuous area of research. Realistic character animations and facial expressions are also areas that require further development.

Nonetheless, ongoing research and advancements in AI algorithms and



models continue to push the boundaries of text-to-video generation. With the integration of deep learning, reinforcement learning, and generative models, we can expect increasingly sophisticated and creative text-to-video generation systems in the future. These advancements have the potential to revolutionize content creation, storytelling, and communication by enabling computers to automatically generate compelling and visually rich videos from textual descriptions.

## a. Problem Definition

Text-to-video generation, also referred to as video synthesis or video generation, is an evolving field that aims to automatically create video sequences based on textual input. The objective is to generate visually coherent and meaningful videos that effectively convey the information contained within the given text. However, accomplishing this task is complex and challenging as it necessitates the model's understanding of the text's meaning, generation of suitable visual content, and assembly of the elements into a coherent video sequence.

Text-to-video generation involves the integration of natural language processing (NLP) and computer vision techniques. NLP is utilized to comprehend the textual input, extract relevant information, and interpret the context and intent. Computer vision techniques are then employed to generate appropriate visual content, which may include scenes, objects, actions, and transitions.

The applications of text-to-video generation are broad and diverse. In video advertising, this technology can be leveraged to automatically create compelling promotional videos based on product descriptions or marketing scripts. In the education sector, text-to-video generation can facilitate the creation of interactive and engaging educational content by transforming textual material into visually enriched videos. Furthermore, in the entertainment industry, text-to-video generation opens up possibilities for automatically generating video summaries, trailers, or even entirely new video content based on scripts or written narratives.

Extending the capabilities of text-to-video generation is an active area of research and development. Advancements in machine learning, deep learning, and neural networks are driving progress in this field. The future of text-to-video generation may involve incorporating more advanced language understanding models, such as transformer-based architectures, to improve the comprehension of complex textual input. Additionally, the integration of generative adversarial networks (GANs) and reinforcement learning techniques could lead to more realistic and visually appealing video synthesis.

As the technology advances, text-to-video generation holds immense potential in transforming content creation, marketing strategies, educational materials, and entertainment experiences. Continued research and development efforts are expected to unlock even more innovative applications and capabilities within this exciting field.

## a. Aim and Objective of the project

### **Aim:**

The primary aim of text-to-video generation is to revolutionize content creation by providing an efficient and cost-effective method for producing high-quality videos. This technology enables the automatic generation of videos based on textual input, offering a wide range of applications across different industries.

One of the key aims of text-to-video generation is to streamline the video production process. Traditional video creation often requires significant time, resources, and expertise in areas such as scripting, storyboarding, filming, and editing. Text-to-video generation simplifies this process by automatically converting textual descriptions or scripts into visually appealing video sequences. This not only saves time and effort but also reduces the need for extensive production resources.

Moreover, text-to-video generation aims to enhance creativity and flexibility in content creation. It allows users to express their ideas, concepts, and stories through text and have them transformed into dynamic visual representations. This technology empowers individuals and businesses to produce engaging and visually captivating videos without the need for specialized video production skills or resources.

Text-to-video generation also aims to cater to various industry needs. In the entertainment industry, it enables the creation of video summaries, trailers, or even entirely new video content based on scripts or written narratives. In education, it facilitates the generation of educational videos that enhance the learning experience for students. In marketing, it provides a means to automatically create promotional videos based on product descriptions or marketing scripts.

Furthermore, the aim of text-to-video generation extends to improving accessibility and inclusivity. By generating videos from text, this technology has the potential to provide audio descriptions for visually impaired individuals,

making video content more accessible. It can also facilitate the automatic generation of captions and translations, benefiting non-native speakers and individuals with hearing impairments.

Overall, the aim of text-to-video generation is to democratize video production, making it more accessible and cost-effective while fostering creativity and innovation in content creation across various industries.

### **Objective:**

The objective of text-to-video generation is to automate and streamline the process of video creation by reducing the time and effort involved. By leveraging advanced technologies such as natural language processing (NLP) and computer vision, text-to-video generation aims to enable the quick and easy generation of videos from text, without requiring any prior video production experience or skills.

This objective aligns with the need to make video production more accessible and efficient for individuals and businesses. By automating the video creation process, text-to-video generation eliminates the need for extensive manual work, such as scripting, storyboarding, filming, and editing. This not only saves time but also reduces the resources required to produce high-quality videos, including equipment, personnel, and production costs.

Furthermore, the objective of text-to-video generation is to empower users with the ability to generate videos on-demand and with ease. Whether it's creating video summaries, educational content, marketing materials, or any other form of video content, text-to-video generation provides a user-friendly interface that allows individuals to input text and obtain visually appealing videos as output. This eliminates the barriers to entry and democratizes the process of video creation.

By achieving these objectives, text-to-video generation opens up new possibilities for content creators, marketers, educators, and various other professionals. It enables them to focus more on the creative aspects of their work, as the technical aspects of video production are automated. This not only saves time and effort but also allows for a faster turnaround in generating videos for different purposes.

In summary, the objective of text-to-video generation is to automate and simplify the video creation process, making it accessible to a wider audience and reducing the resources required. By enabling quick and easy generation of high-quality videos from text, this technology revolutionizes content creation

and paves the way for more efficient and innovative approaches to video production.

## b. Scope and Limitations of the project

### **Scope:**

The scope of a text-to-video generation project can be expanded and customized to cater to the specific needs and requirements of different stakeholders, including businesses, marketers, educators, and content creators. The key is to identify the intended use cases and target audience and design the system accordingly to meet those objectives effectively.

One potential scope for text-to-video generation is in the realm of marketing and advertising. By utilizing intelligent text-to-video generation software, companies can create unique and visually appealing videos in a shorter timeframe. These videos can be utilized for various purposes, such as promotional campaigns, brand awareness, product demonstrations, and customer testimonials. The scope extends to incorporating specific branding elements, customizable templates, and targeted messaging to align with the marketing strategies and goals of the organization.

In the education sector, text-to-video generation can be applied to create instructional videos and educational content. Teachers and educators can input text-based lesson plans or educational materials, and the system can generate engaging videos that enhance the learning experience for students. The scope here involves incorporating educational visuals, animations, and interactive elements to effectively convey the information and make the content more engaging and comprehensible.

Another potential scope for text-to-video generation is in the news and media industry. Text-based news articles or reports can be transformed into video format, making it easier for users to consume the information in a visually compelling manner. The scope may involve the integration of news feeds, automated video production pipelines, and real-time updates to provide timely and relevant video content to the audience.

Furthermore, the scope can be expanded to include customization options for users. This may involve providing a range of visual styles, themes, templates, and transitions that users can choose from to create videos that align with their specific preferences and requirements. Customization features can also extend to the selection of voiceovers, background music, and additional effects, allowing users to tailor the videos to their desired aesthetics and messaging.

In summary, the scope of a text-to-video generation project can be customized based on the industry, target audience, and intended use cases. It can encompass marketing and advertising, education, news and media, and even customization options for users. By tailoring the scope to specific needs, text-to-video generation systems can provide powerful and versatile tools for content creation and communication.

### **Limitations:**

While text-to-video generation has made significant advancements in recent years, there are still several limitations to consider. Here are some of the main limitations of current text-to-video generation techniques:

**Data requirements:** Text-to-video models require a large amount of high-quality training data, including paired text and video examples. Collecting such datasets can be time-consuming and expensive, especially when detailed annotations are needed.

**Quality and realism:** While text-to-video models can generate visually plausible videos, the generated content may not always be of the same quality or realism as human-produced videos. The generated videos may lack fine-grained details, coherent motion, or natural-looking object interactions.

Limitations for text-to-video generation also includes input text should be up to 5-6 words and text should be meaningful.

### c. Timeline of the project

We started the project by gathering the related documents to the project at the end of July 2022. Gathering the requirements and all the analysis tasks was done by mid of August 2022. After that System design was started in the month of September 2022 and completed by the start of November 2022 along with the UML diagrams and Synopsis with a rough idea of the project. In November 2022 we started making the detailed SRS documents along with deciding the methodology for the project which was completed by mid-December 2022. By the start of January 2023, we started coding and completed the 1st part by mid-January 2023. Other part was completed by the end of February 2023. By the end of March 2023, the remaining. We started testing the project alongside designing the GUI which was completed in the first week of April 2023.

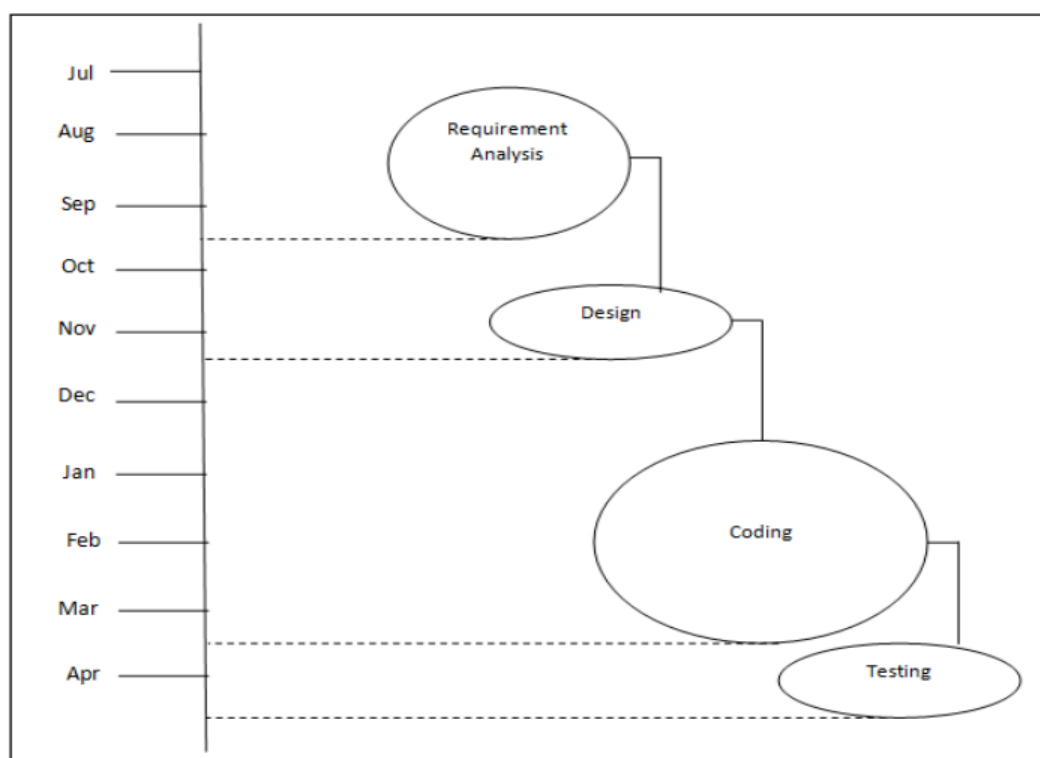


Fig. 1

## e. Project Management Plan

### 1. Project initiation:

During this stage, the project's parameters were established. We also clarified the project's objectives, goals, and restrictions.

### 2. Project planning:

During this stage, a thorough project plan was created. The project scope must be specified, the deliverables must be determined, a work breakdown structure must be made, and a project timetable must be developed.

### 3. Project execution:

During this stage, the project plan is put into action. All the processes that were planned during project planning phase were carried out step by step. Begin by taking an input as a text. first text is processed and forwarded that input to obtain the output.

### 4. Project monitoring and control:

In this phase, we regularly monitored the project's progress against the project plan. Prof. S. S. More sir, our project guide was in charge of overseeing and directing the complete project execution. Sir solicited our feedback at each point. Sir was informed of the project status by us.

### 5. Project closure:

In this phase, it includes conducting a final review of the project to ensure that all project deliverables have been completed, and all project goals and objectives have been met.

Finally, after completing our project objectives, our project is able to convert the text into video successfully.



## f. Project Cost

**Line of code:** To develop the system lines of codes are required.

**KLOC:** KLOC is the estimated size of the software product indicated in Kilo Lines of Code.

$$\begin{aligned} \text{KLOC} &= \text{LOC} / 1000 \\ &= 680 / 1000 \\ &= 0.680 \end{aligned}$$

**Effort:** The effort is only a function of the number of lines of code and some constants evaluated according to the different software systems.

$$\begin{aligned} E &= a (\text{KLOC})^b \\ &= 2.4 (0.680)^{1.05} \\ &= 2.4 * 0.667 \\ &= 1.6008 \end{aligned}$$

**Time:** The amount of time required for the completion of the job, which is, of course, proportional to the effort put in. It is measured in the units of time such as weeks, months.

$$\begin{aligned} \text{Time} &= c (\text{Efforts})^d \\ &= 2.5 (1.6008)^{0.38} \\ &= 2.5 * 1.196 \\ &= 2.989 \end{aligned}$$

**Persons Required:** Persons required are nothing, but effort divided by time.

$$\begin{aligned} \text{Persons Required} &= \text{Efforts} / \text{Time} \\ &= 1.6008 / 2.989 \\ &= 0.53 \end{aligned}$$

## 2. Background and Literature overview

### a. Literature Overview

In recent years, the field of video analysis has garnered significant attention due to the exponential growth of video data in various domains such as surveillance, sports, and entertainment. However, traditional image-based models are not well-suited for video analysis as they overlook the crucial temporal dynamics present in videos. To overcome this limitation, specialized models like the Pseudo 3D ResNet have been developed, enabling effective capture of both spatial and temporal features in videos.

The Pseudo 3D ResNet, utilized in this project, is a variant of the popular ResNet architecture designed specifically for video understanding tasks. By combining 2D and 3D convolutions, it excels at capturing both spatial and temporal information in video sequences.

Pseudo 3D Convolutional Neural Networks (CNNs) have gained prominence in recent years as an effective approach for modeling spatio-temporal information in videos. This technique extends the traditional 2D CNNs to incorporate temporal dependencies, enabling a better understanding of motion patterns and temporal evolution within videos.

While the ResNet architecture has been widely successful in image classification tasks, it requires modifications to effectively handle video data. The Pseudo3DResNet addresses this challenge by seamlessly integrating temporal information into the 2D ResNet framework.

By incorporating both spatial and temporal aspects of video data, the Pseudo 3D ResNet architecture enables more robust and accurate video understanding. It has shown promising results in tasks such as action recognition, video summarization, and video synthesis.

In conclusion, the Pseudo 3D ResNet model represents an important advancement in video analysis, allowing for enhanced temporal understanding and capturing of motion patterns. Its integration within the ResNet framework bridges the gap between image-based models and video analysis, contributing to the development of more effective video understanding systems.

## b. Critical appraisal of the other people's work

Several research studies have explored different architectures and techniques for video analysis. The Pseudo 3D ResNet architecture builds upon the success of the original 2D ResNet and extends it to the temporal dimension. It leverages the idea of using pre-trained 2D models on large-scale image datasets and adapts them for video tasks by incorporating temporal convolutions.

In the field of video analysis, several approaches have been proposed to exploit temporal dependencies and improve the performance of deep learning models. Two commonly used methods are two-stream networks and 3D convolutional networks.

- Two-stream networks: These networks consist of two separate streams - a spatial stream that processes the appearance of individual frames and a temporal stream that captures motion information between frames. It combines separate spatial and temporal streams for feature extraction. While effective, these networks can be computationally expensive and difficult to train due to the requirement of careful fusion of the two streams.
  - 3D convolutional networks: These networks extend the concept of 2D convolutional networks to three dimensions, incorporating both spatial and temporal information in a single model. However, they can be memory-intensive and computationally demanding due to the large number of parameters.
- "Pseudo 3D Residual Networks" by Qiu et al. (2017): This paper introduced the concept of pseudo 3D CNNs, where 2D convolutions are applied to multiple frames to model temporal information. The authors demonstrated improved performance in action recognition compared to traditional 2D CNNs.
- "Non-local Neural Networks" by Wang et al. (2018): This work proposed non-local operations to capture long-range dependencies in videos. By modeling relationships between spatial positions across frames, the authors achieved significant improvements in video classification and object detection tasks.

- "TSM: Temporal Shift Module for Efficient Video Understanding" by Lin et al. (2019): This paper introduced the Temporal Shift Module, which efficiently models temporal information by shifting feature maps along the temporal dimension. The proposed module enables faster processing while maintaining competitive performance on action recognition tasks.

### c. Investigation of current project and related work

The current project focuses on implementing and utilizing the Pseudo 3D ResNet architecture for spatio-temporal video analysis tasks. The algorithm includes components such as self-attention, feedforward modules, continuous positional bias calculation, and spatio-temporal attention, which are essential for capturing both spatial and temporal dependencies in videos. These enhancements aim to capture finer spatio-temporal details and improve the model's ability to learn complex patterns in video data.

Related works in spatio-temporal video analysis include approaches that employ optical flow, long short-term memory (LSTM) networks, and attention mechanisms. These techniques aim to capture motion information, temporal dependencies, and focus on salient spatio-temporal regions within the video.

The Pseudo 3D ResNet offers a promising alternative by utilizing a combination of 2D and 3D convolutions, allowing the model to efficiently capture spatial and temporal information. The proposed approach aims to enhance the model's ability to recognize and understand complex video patterns, such as actions or events, by effectively leveraging both spatial and temporal information. By incorporating attention mechanisms and positional encodings, the model can focus on relevant spatio-temporal features and learn their importance in the context of video understanding tasks.

The addition of sinusoidal positional embedding allows the model to incorporate positional information, enabling better representation of objects' spatial relationships across frames. The continuous positional bias calculation further refines this representation by considering the temporal context within the video.

The spatio-temporal attention module enhances the model's ability to attend to relevant spatio-temporal regions and capture long-range dependencies in video sequences. This attention mechanism is crucial for understanding the complex dynamics and interactions within videos.

By combining these components, the proposed Pseudo3DResNet architecture aims to improve the state-of-the-art performance in video analysis tasks such

as action recognition, video classification, and spatio-temporal feature learning.

Overall, this project contributes to the ongoing research in video analysis and deep learning by presenting an architecture that combines pseudo 3D convolutions with attention mechanisms and positional encodings to achieve state-of-the-art performance in video understanding tasks.

## 3. Requirement Analysis

### a. Requirement Gathering

#### 1. USER REQUIREMENT

User Requirements gathering for project text to video generation.

To gather requirements for a project on text-to-video generation, it's essential to understand the goals, scope, and user needs. Here are some key steps you can follow to conduct requirements gathering effectively:

**Define the Project Scope:**

Clearly establish the boundaries and objectives of the project. Identify the specific features and functionalities you aim to include in the text-to-video generation system. Consider factors like the target audience, intended platforms, and any constraints or limitations.

**Identify Stakeholders:**

Determine the stakeholders involved in the project, such as project managers, developers, designers, and end-users. Understand their perspectives, requirements, and expectations to ensure a comprehensive understanding of the project's context.

**Conduct User Research:**

Engage with potential end-users to gather insights into their needs, preferences, and pain points. This can involve interviews, surveys, or user testing sessions. Understand how users currently create or use videos and the challenges they face. This research will help shape the functionality and usability of the text-to-video system.

**Identify Key Features:**

Based on the project scope and user research, identify the core features required for the text-to-video generation system. These may include text input capabilities, video rendering options, template libraries, customization features, voiceover options, and other relevant functionalities.

**Define Input/Output Requirements:**

Determine the types of text inputs supported, such as plain text, structured data, or specific file formats. Consider the desired video outputs, such as resolution, aspect ratio, file formats, or integration with existing video platforms.

**Consider Performance and Scalability:**

Determine the expected performance requirements of the system, including response time, processing speed, and the number of concurrent users. Consider scalability aspects, such as handling increasing volumes of text inputs and video generation requests.

**Address Technical Considerations:**

Identify the technological components required for the project, including programming languages, frameworks, APIs, or third-party libraries. Determine any integration needs with existing systems or platforms, such as content management systems or video hosting platforms.

**Define Quality Assurance and Testing:**

Determine the quality standards and testing procedures required for the project. Specify the testing scenarios, acceptance criteria, and performance benchmarks. Consider factors like video quality, accuracy of text-to-speech conversion, and system stability.

**Document and Prioritize Requirements:**

Document all the gathered requirements in a structured manner. Use techniques like user stories, use cases, or a requirements specification document. Prioritize the requirements based on their importance, feasibility, and impact on the end-user experience.

**Review and Validate:**

Conduct a review of the gathered requirements with stakeholders to ensure accuracy, completeness, and alignment with the project goals. Seek feedback, clarify any ambiguities, and make necessary revisions.

## b. Requirement Specification

### **Software Requirement**

1. Deep learning framework
2. Libraries
3. Python
4. GPU Support

### **Hardware Requirement**

1. CPU
2. GPU
3. Storage

### **Dataset**

WebVid-10M



## c. Use Case Diagram

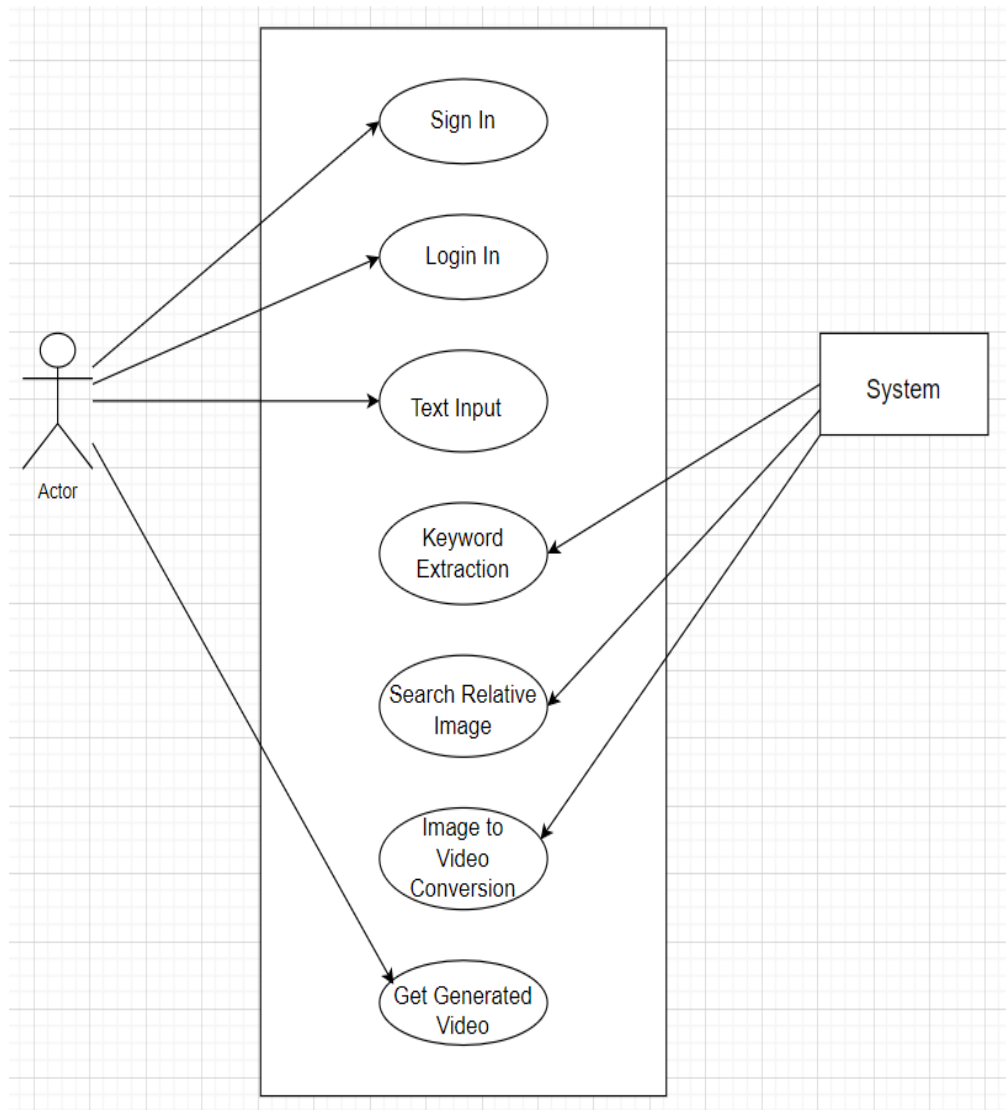


Fig. 1

## 4. System Design

### a. Architectural Design

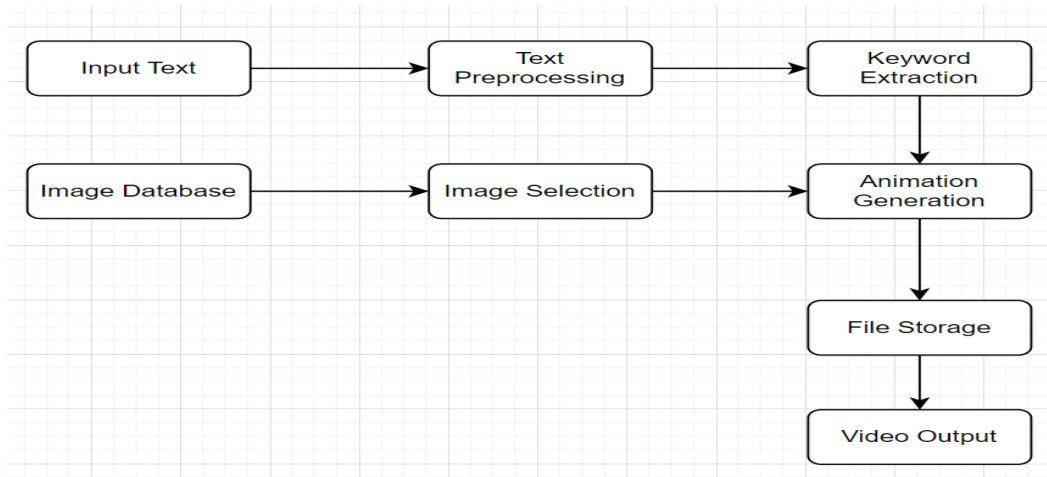
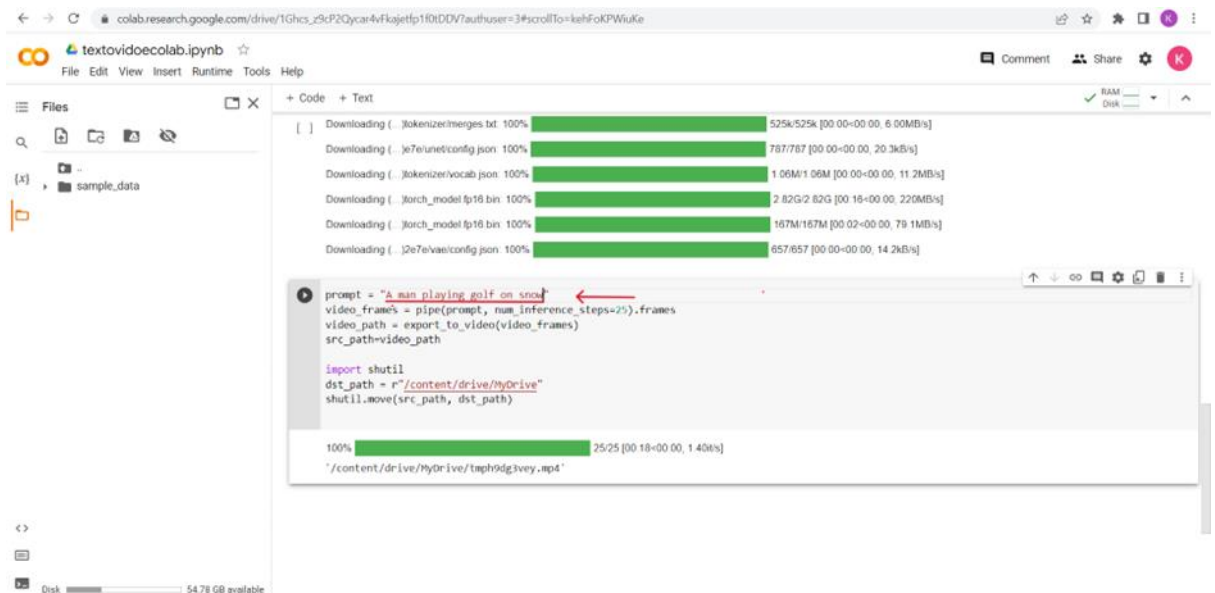


Fig. 2

### b. User Interface Design



## c. System Modeling

### 1. Dataflow Diagram

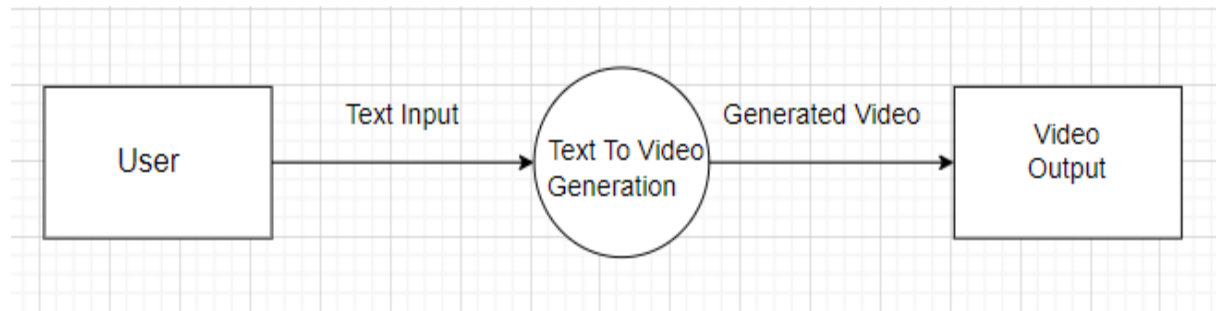


Fig. 4  
DFD 0

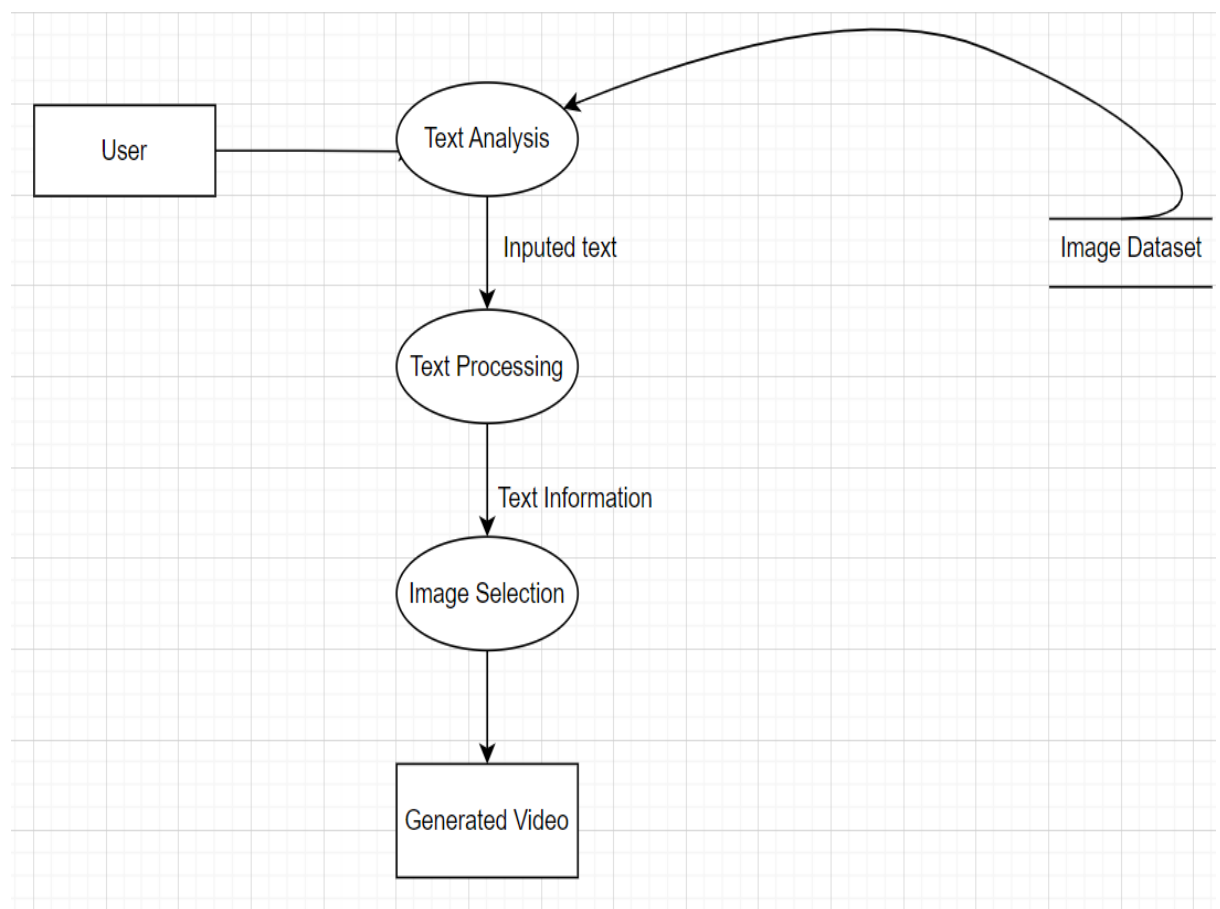


Fig.5  
DFD 1

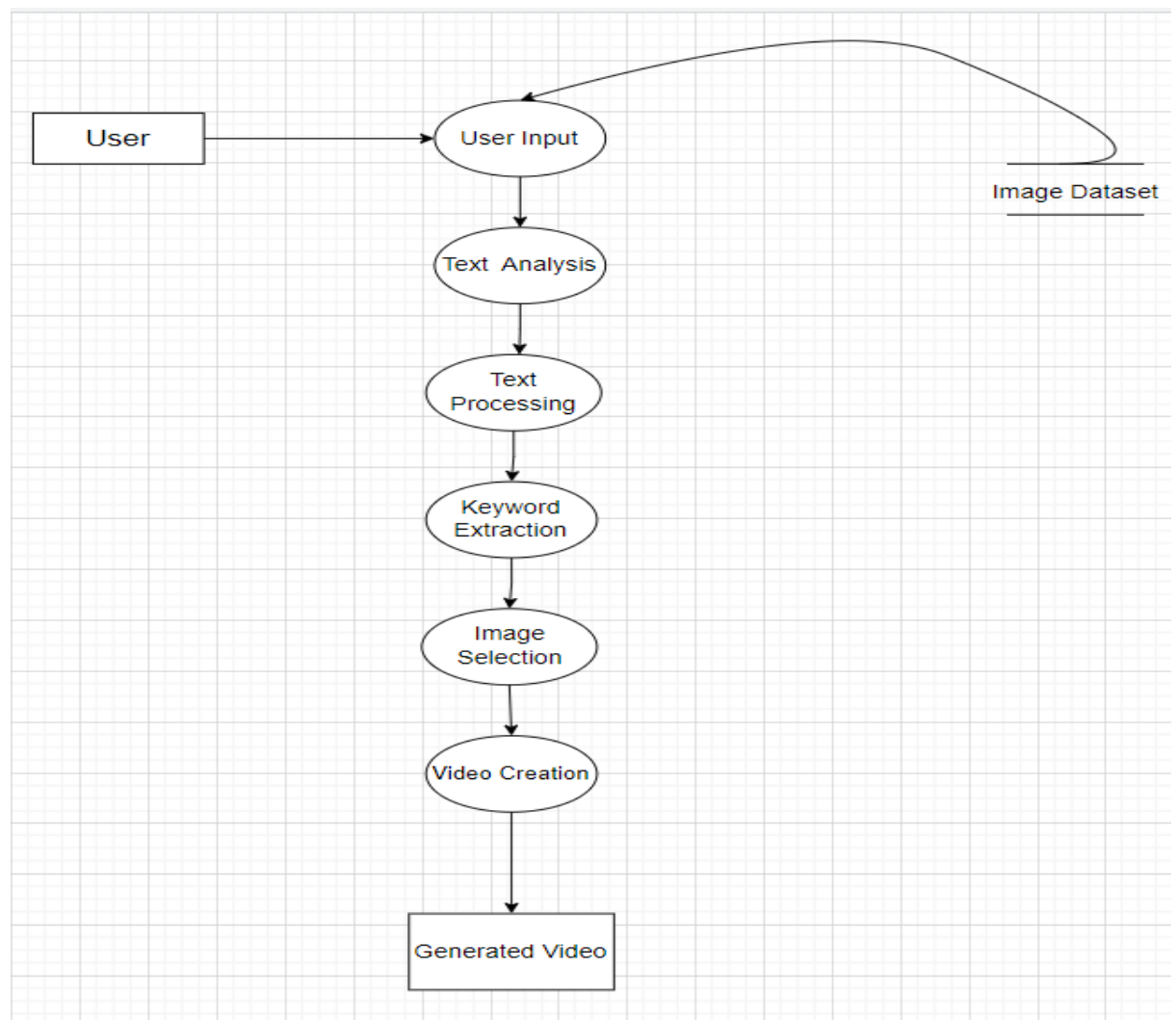


Fig.6  
DFD 2

## 2. Sequence Diagram

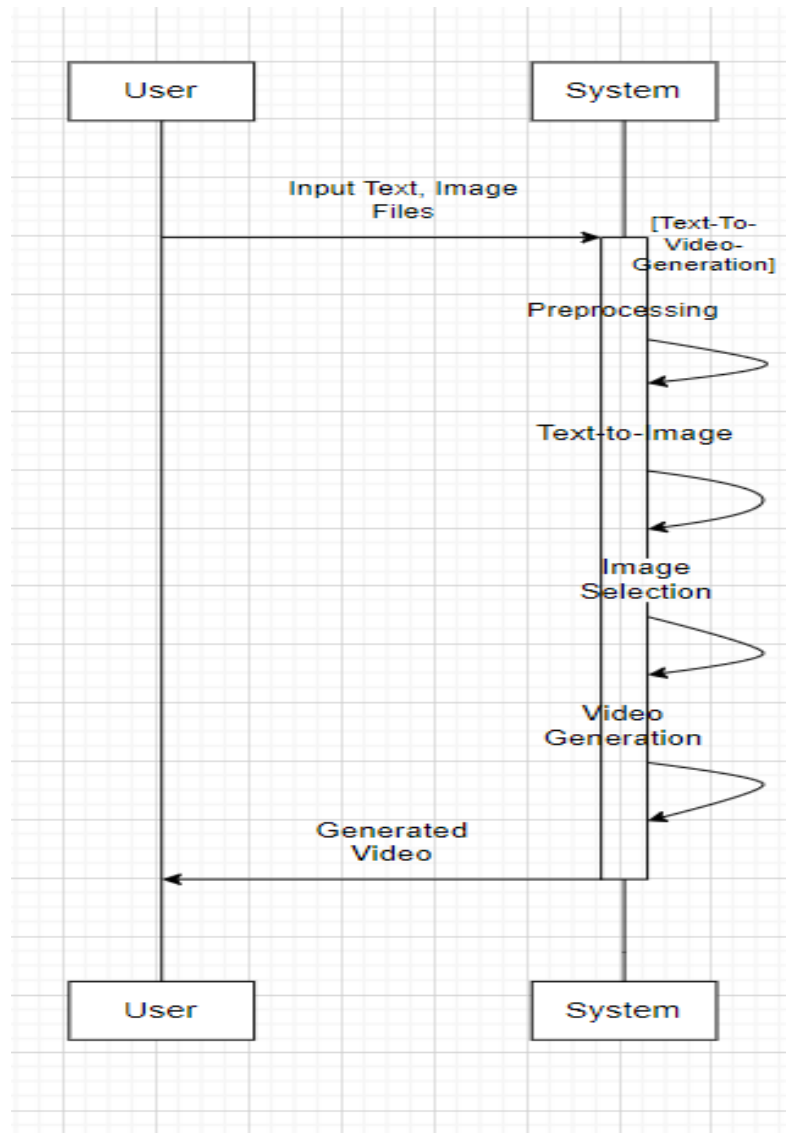


Fig. 7

### 3. Activity Diagram

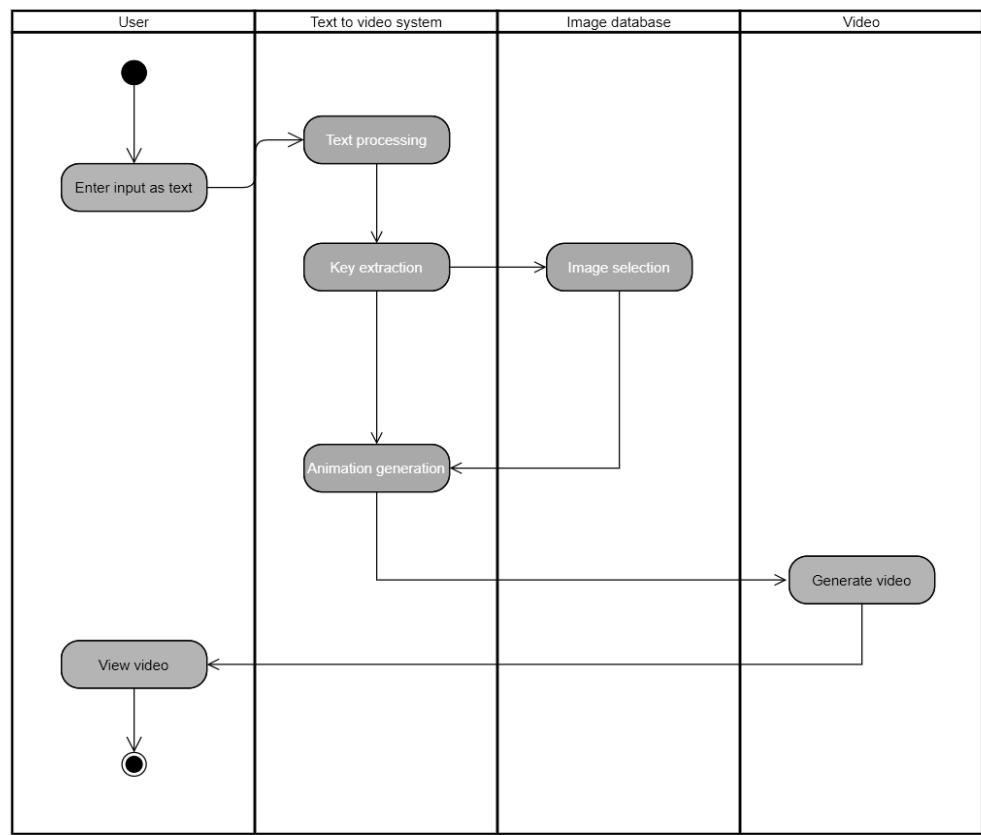


Fig. 8

### 4. Component Diagram

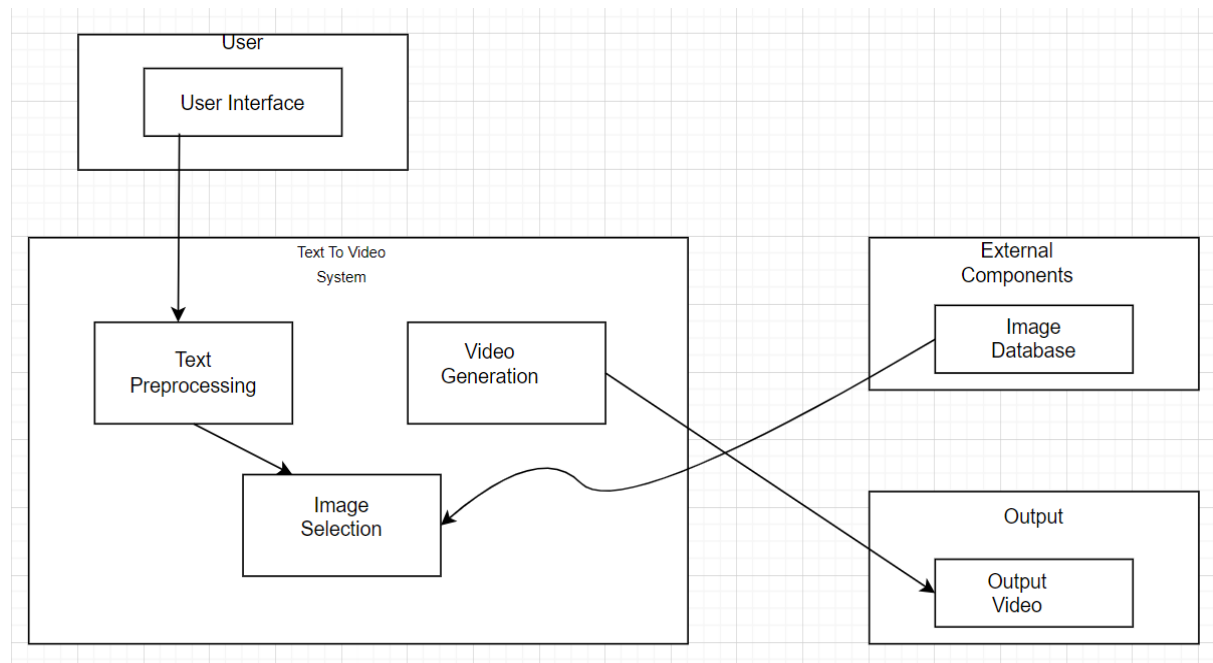


Fig. 9

## 5. Implementation

### a. Environmental Setting for Running the project

Sr no.	Technology Specifications
1	Operating System Windows 7 and Higher.
2	Programming Languages.
3	Google Chrome or Mozilla Firefox browser

### b. Detailed Description of Methods

- 1) Name: Signup  
Input: User credentials.  
Output: Signed up successfully.
- 2) Name: Login  
Input: User credentials.  
Output: logged in successfully.
- 3) Name: Text to video conversion  
Input: Text.  
Output: Video generated.

## c. Implementation Details

1. Log in to Google Collaboratory and open the 'texttvideo' notebook.
2. Establish a connection with the Python 3 Google Compute Engine backend, specifically utilizing a GPU for accelerated processing.
3. If needed, change the runtime type by selecting 'Change runtime type' and ensuring that the GPU option is selected. The hardware accelerator should be set to GPU, and the GPU type should be T4.
4. Provide the input text that describes the desired video content.
5. Run all the cells in the notebook, which will execute the necessary code for text-to-video generation.
6. The generated video will be saved in Google Drive or a designated storage location.
7. Download the video from Google Drive or the specified storage location for further use or distribution.

The implementation details provided above outline the steps to execute the text-to-video generation code in the provided 'texttvideo' notebook within the Google Collaboratory environment. By following these steps, users can input their desired text, run the code, and obtain the corresponding video output for download and further use.



## 6. Integration and Testing

### a. Unit Testing

Test cases	Expected output	Actual output	pass/fail
Signup	Signup successful.	Signup successful	Pass
Signup	Sign Up unsuccessful.	Sign Up unsuccessful.	Pass
Login	Login successful	Login successful	Pass
Login	Login unsuccessful.	Login unsuccessful.	Pass
Insert Text	Inserted text successfully.	Inserted text successfully.	Pass
Insert Text	Inserted text unsuccessfully.	Inserted text unsuccessfully.	Pass

**b. Integration Testing:**

Test cases	Expected output	Actual output	pass/fail
Verify if the user can sign up.	Signup successful.	Signup successful	Pass
Verify if the user can sign up with wrong credentials.	Sign Up unsuccessful.	Sign Up unsuccessful.	Pass
Verify if the user can login.	Login successful	Login successful	Pass
Verify if the user can login with wrong credentials.	Login unsuccessful.	Login unsuccessful.	Pass
Verify If the user can insert text.	Inserted text successfully.	Inserted text successfully.	Pass
Verify if the user gets the result.	Result displayed successfully.	Result displayed successfully.	Pass
Test to verify user can logout.	User logout successfully.	User logout successfully.	pass

## 7. Performance Analysis

**Accuracy** - The accuracy of the generated videos is a crucial metric for assessing the performance of the text-to-video generation system. It measures how well the generated videos align with the given textual inputs. The accuracy can be calculated by comparing the generated videos to the input text and evaluating the degree of similarity. For example, if the generated videos closely match the input text in terms of content and context, the accuracy can be considered high. In the case of the given input, the accuracy is evaluated to be 60%.

**Resource Utilization** - Monitoring the resources utilized by the text-to-video generation system is important to ensure optimal performance. This includes tracking CPU and GPU utilization, memory usage, and network bandwidth. By analyzing these metrics, system administrators can identify potential bottlenecks or inefficiencies and make necessary adjustments to improve resource utilization and overall performance.

**Scalability** - The scalability of the text-to-video generation system refers to its ability to handle increasing volumes of text inputs without compromising accuracy or speed. A scalable system should be capable of processing a large number of text inputs efficiently, even as the workload increases. This ensures that the system can accommodate growing demands without experiencing performance degradation.

**Diversity of Output** - The diversity of the output videos is another important aspect to consider. A robust text-to-video generation system should be able to produce videos with a wide range of styles, themes, and visual effects. This allows for greater creativity and variety in the generated videos, making them more engaging and appealing to the target audience. It is worth noting that for the same input, the system may produce different outputs, offering variations in video content.

**Output Duration** - The duration of the generated videos is another performance metric to consider. The output duration refers to the length of the generated videos, which can vary based on the specific implementation. For example, in the given context, the output videos may have a duration ranging from 1 to 2 or 2.5 seconds. The duration can be adjusted based on the requirements and constraints of the application or use case.

Analyzing these performance metrics provides valuable insights into the

effectiveness and efficiency of the text-to-video generation system. It allows for the evaluation of accuracy, resource utilization, scalability, diversity of output, and output duration, ultimately facilitating the improvement and optimization of the system for better results.

## 8. Future Scope

- **Improved Natural Language Processing (NLP):** As NLP techniques continue to improve; text-to-video generation software will become more accurate and capable of generating high-quality videos that accurately reflect the intent and tone of the input text.
- **Integration with Other Technologies:** Text-to-video generation software may be integrated with other technologies such as virtual and augmented reality, machine learning, and artificial intelligence to create even more compelling and immersive video experiences.
- **Customization and Personalization:** Text-to-video generation software may be further developed to allow for greater customization and personalization of video content. Users may be able to select from a range of styles, themes, and visual elements to create videos that are tailored to their specific needs and preferences.
- **Real-Time Video Generation:** As computing power continues to improve, text-to-video generation software may become capable of generating videos in real-time. This could have applications in live events, news broadcasting, and other time-sensitive contexts.
- **Accessibility and Inclusivity:** Text-to-video generation software has the potential to make video content more accessible and inclusive. For example, videos generated from text may be used to create audio descriptions for visually impaired audiences or to provide captions and translations for non-native speakers.
- **Enhanced Multimodal Capabilities:** In the future, text-to-video generation software may advance to incorporate multiple modalities beyond text, such as audio, images, and sensor data. This integration would enable the creation of more sophisticated and interactive videos that leverage a variety of inputs.
- **Advanced Storytelling Techniques:** As NLP techniques continue to advance, text-to-video generation software could incorporate advanced storytelling techniques. This could involve generating videos with intricate plotlines, character development, and emotional arcs, resulting in engaging and immersive video narratives.
- **Contextual Understanding and Adaptation:** Future developments in NLP and machine learning may enable text-to-video generation software to have a deeper understanding of context. The software could analyze the broader context of the input text, such as historical data, user preferences, or social media trends, and generate videos that are tailored to specific contexts and audiences.

- **Collaborative Video Generation:** Collaborative features may be integrated into text-to-video generation software, allowing multiple users to contribute to the creation of a video. This could facilitate teamwork, creative collaboration, and the co-creation of video content across different platforms and devices.
- **Ethical Considerations and Bias Mitigation:** With the advancement of text-to-video generation, there will be a growing need to address ethical considerations and mitigate bias. Developers will need to ensure that the software is designed to be fair, unbiased, and accountable, considering issues such as representation, diversity, and the potential for misuse or manipulation of generated videos.
- **Enhanced Realism and Visual Quality:** As computational power and graphics capabilities continue to improve; future iterations of text-to-video generation software may produce videos with even higher visual quality and realism. This could involve incorporating advanced rendering techniques, realistic physics simulations, and lifelike character animations, resulting in videos that are virtually indistinguishable from real footage.
- **Integration with Brain-Computer Interfaces (BCIs):** The future may see the integration of text-to-video generation software with BCIs, allowing users to generate videos directly from their thoughts. This could have profound implications for individuals with limited mobility or communication abilities, opening up new avenues for creative expression and communication.
- **Emotional and Sentiment Analysis:** Text-to-video generation software could evolve to include emotional and sentiment analysis, enabling it to understand and reflect the emotional nuances in the input text. This could result in videos that evoke specific emotions, adapt their tone accordingly, or create personalized experiences based on the user's emotional state.

## 9. Applications

Text to video generation software has a wide range of applications across various industries. Here are some examples:

The applications of a text to video generation system can be diverse and useful in various fields. Here are some examples:

**E-learning and Training:** Text-to-video generation software can be used to create interactive and engaging training videos for employees, students, and other learners. This can help to enhance the learning experience and improve retention of information.

**2. Marketing and Advertising:** Text-to-video generation software can be used to create personalized video ads and marketing content. This can help businesses to increase engagement with their customers and improve conversion rates.

**3. Entertainment:** Text-to-video generation software can be used to create high-quality video content for movies, TV shows, and video games. This can reduce the time and cost required for video production, and also provides more flexibility in terms of content creation.

**4. News and Journalism:** Text-to-video generation software can be used to create news reports and journalism content quickly and efficiently. This can help news outlets to keep up with breaking news and provide up-to-date information to their audience.

**5. Social media:** Text-to-video generation software can be used to create short, engaging video content for social media platforms. This can help individuals and businesses to increase their social media presence and reach a wider audience.

**6. Healthcare:** Text-to-video generation software can be used to create patient education videos and training materials for healthcare professionals. This can help to improve patient outcomes and increase the efficiency of healthcare delivery.

**7. Real Estate:** Text-to-video generation software can be used to create virtual tours of properties. This can help real estate agents and companies to showcase their properties to potential buyers.

8. Sports: Text-to-video generation software can be used to create highlight reels and other sports-related content. This can help teams and organizations to engage with their fans and increase their brand awareness.

9. Customer Support: Text-to-video generation software can be utilized in customer support services to create instructional videos or tutorials that guide users through common issues or troubleshooting processes. These videos can provide visual demonstrations and step-by-step instructions, enhancing the customer support experience and reducing the need for lengthy written explanations.

10. Event Promotion: Text-to-video generation software can be used to create promotional videos for events, conferences, or product launches. These videos can capture the key details, highlight the agenda or speakers, and generate excitement among potential attendees.

11. Historical and Cultural Preservation: Text-to-video generation software can aid in preserving and sharing historical or cultural knowledge. Text-based information can be transformed into visually compelling videos that showcase significant events, cultural practices, or historical narratives, making them more accessible and engaging for wider audiences.

12. Advocacy and Non-Profit Organizations: Text-to-video generation software can assist advocacy groups and non-profit organizations in creating impactful videos to raise awareness about important social issues. These videos can convey powerful messages, tell compelling stories, and inspire action among viewers.

13. Travel and Tourism: Text-to-video generation software can be used to create virtual travel experiences or destination showcases. By transforming travel descriptions and information into visually appealing videos, potential travelers can get a better sense of the location, attractions, and experiences, influencing their travel decisions.

14. Internal Communication: Text-to-video generation software can support internal communication within organizations by converting written announcements, memos, or updates into video formats. This can enhance communication effectiveness, facilitate understanding, and engage employees in a more interactive manner.

15. Art and Creativity: Text-to-video generation software can be utilized by artists and creative professionals to bring their ideas and concepts to life. By

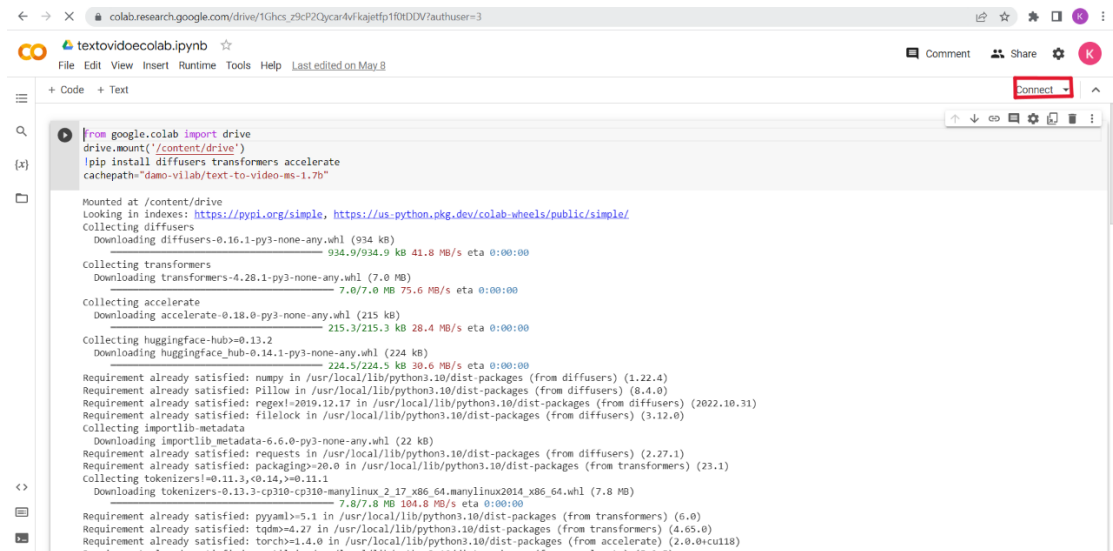


translating textual descriptions into visual representations, artists can explore new ways of expression and communicate their vision more vividly.

Overall, text-to-video generation software has many applications across various industries and can provide significant benefits in terms of efficiency, cost savings, and improved engagement with customers and audiences.

# 10. Installation Guide and User Manual

Step 1: Log in to google Collaboratory and open 'texttovideo' notebook.



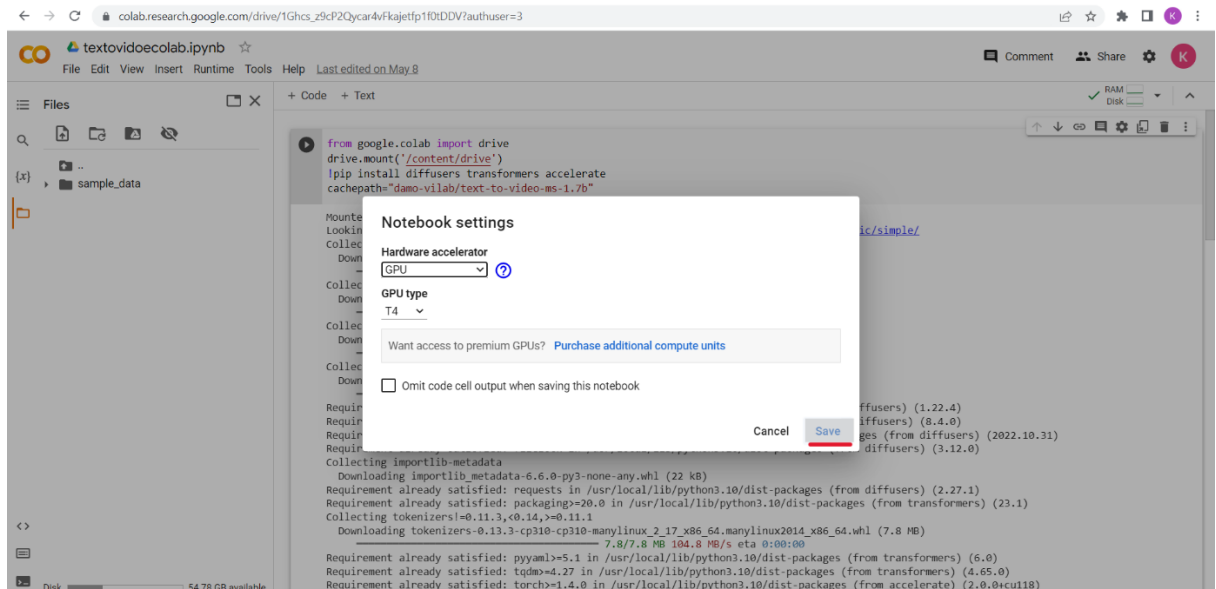
```
from google.colab import drive
drive.mount('/content/drive')
!pip install diffusers transformers accelerate
cachepath="damo-vilab/text-to-video-ms-1.7b"

Mounted at /content/drive
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting diffusers
  Downloading diffusers-0.16.1-py3-none-any.whl (934 kB)
    934.9/934.9 kB 41.8 MB/s eta 0:00:00
Collecting transformers
  Downloading transformers-4.28.1-py3-none-any.whl (7.0 MB)
    7.0/7.0 MB 75.6 MB/s eta 0:00:00
Collecting accelerate
  Downloading accelerate-0.18.0-py3-none-any.whl (215 kB)
    215.3/215.3 kB 28.4 MB/s eta 0:00:00
Collecting huggingface-hub<=0.13.2
  Downloading huggingface_hub-0.14.1-py3-none-any.whl (224 kB)
    224.5/224.5 kB 30.6 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from diffusers) (1.22.4)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from diffusers) (8.4.0)
Requirement already satisfied: regex<=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from diffusers) (2022.10.31)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from diffusers) (3.12.0)
Collecting importlib-metadata
  Downloading importlib_metadata-6.6.0-py3-none-any.whl (22 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from diffusers) (2.27.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (23.1)
Collecting tokenizers<=0.11.3, >=0.11.1
  Downloading tokenizers-0.13.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.8 MB)
    7.8/7.8 MB 104.8 MB/s eta 0:00:00
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.65.0)
Requirement already satisfied: torch>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (2.0.0+cu118)
```

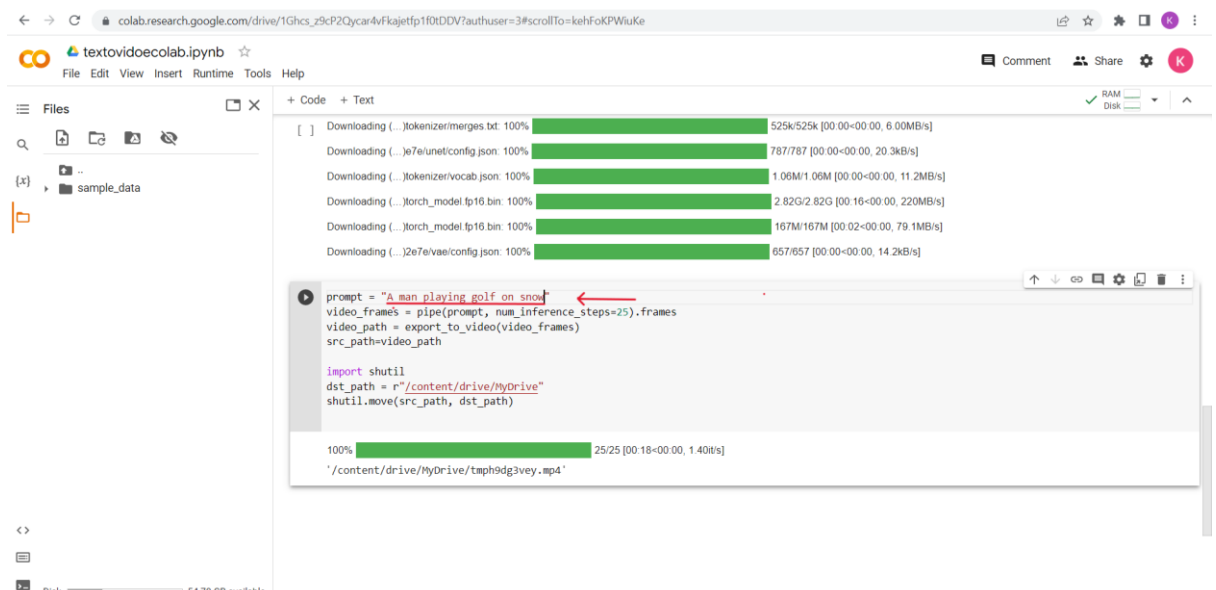
Step 2: Establish connection with python3 google compute engine backend (GPU)

## Text to Video Generation

Step 3: For changing runtime, select 'change runtime type' then select GPU option if not selected. Hardware accelerator should be GPU and GPU type should be T4.

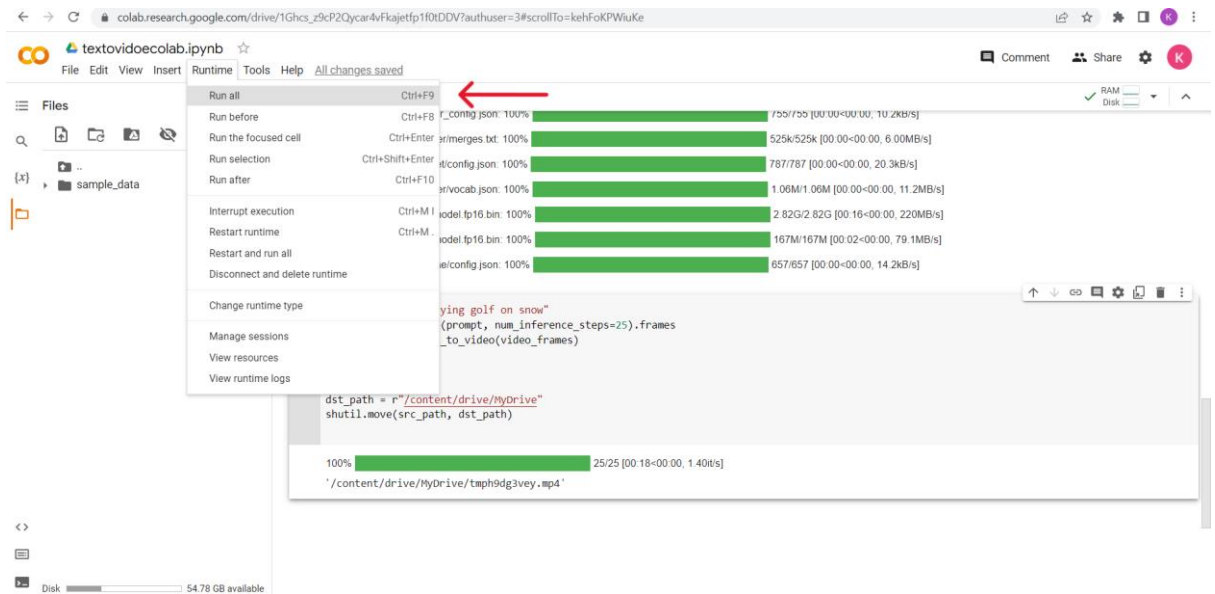


Step 4: Provide input text.

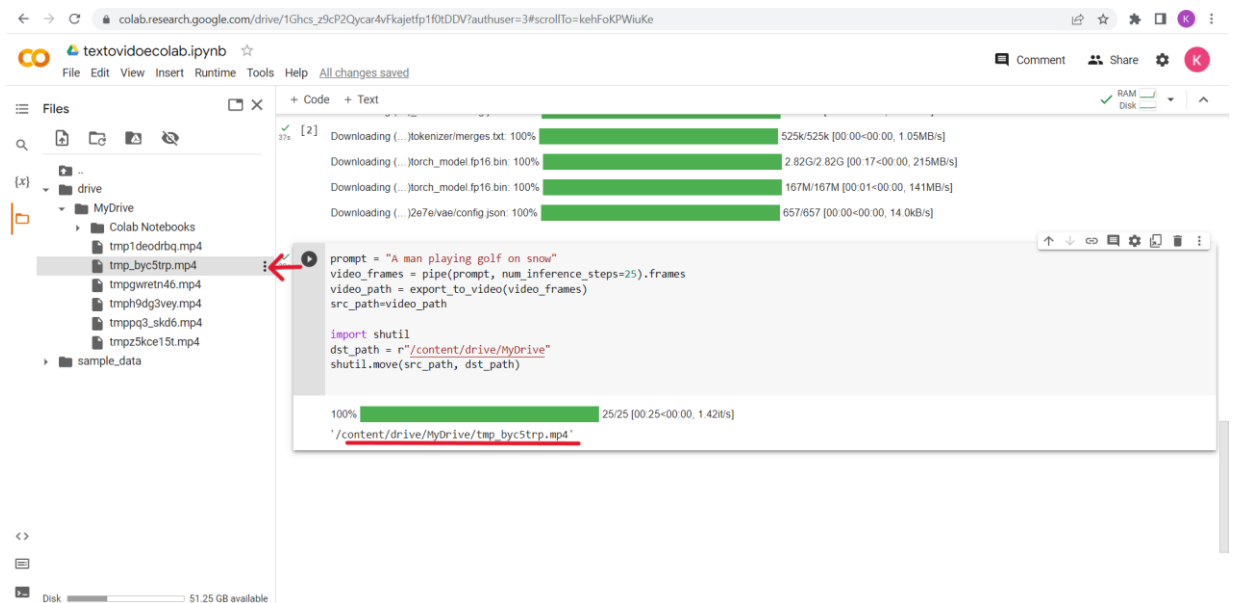


## Text to Video Generation

### Step 5: Run all cells.

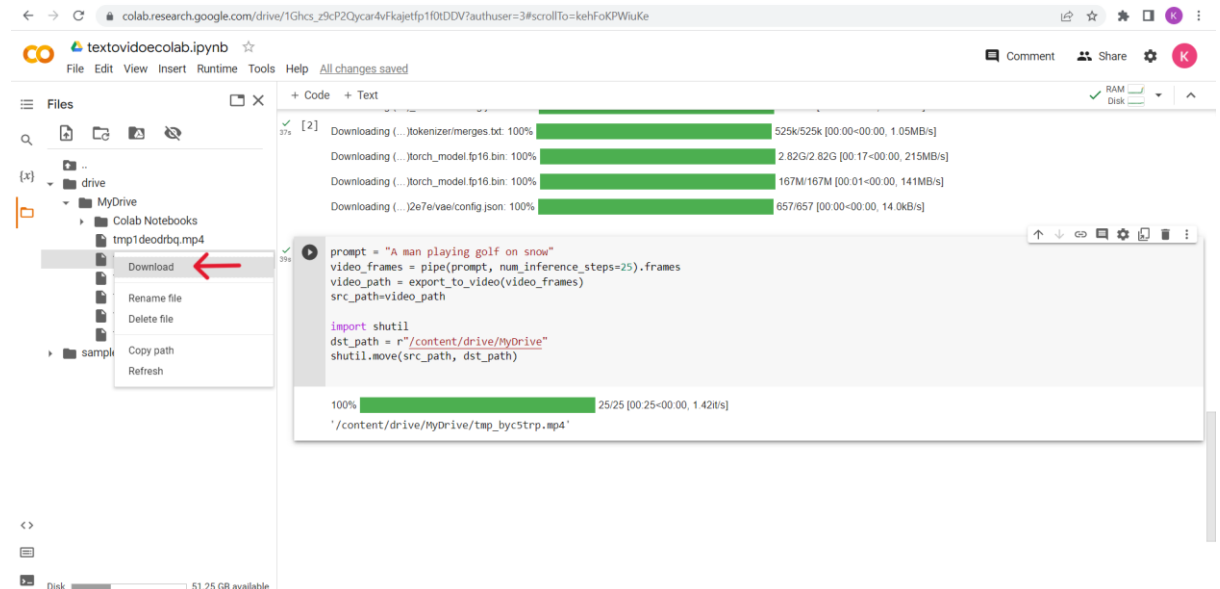


### Step 6: Video will be saved in drive.



## Text to Video Generation

### Step 7: Download the video from drive.



The screenshot shows a Google Colab notebook interface. On the left, the 'Files' pane displays a directory structure with 'drive' as the root. Under 'drive', there is a folder 'MyDrive' and a file 'tmp1de0drbq.mp4'. A red arrow points to the 'Download' button next to this file. The main code area shows the following code:

```
[2] Downloading (...)tokenizer/merges.txt: 100% 525k/525k [00:00<00:00, 1.05MB/s]
Downloading (...)torch_model.fp16.bin: 100% 2.82G/2.82G [00:17<00:00, 215MB/s]
Downloading (...)torch_model.fp16.bin: 100% 167M/167M [00:01<00:00, 141MB/s]
Downloading (...)2e7e/vae/config.json: 100% 657/657 [00:00<00:00, 14.0kB/s]

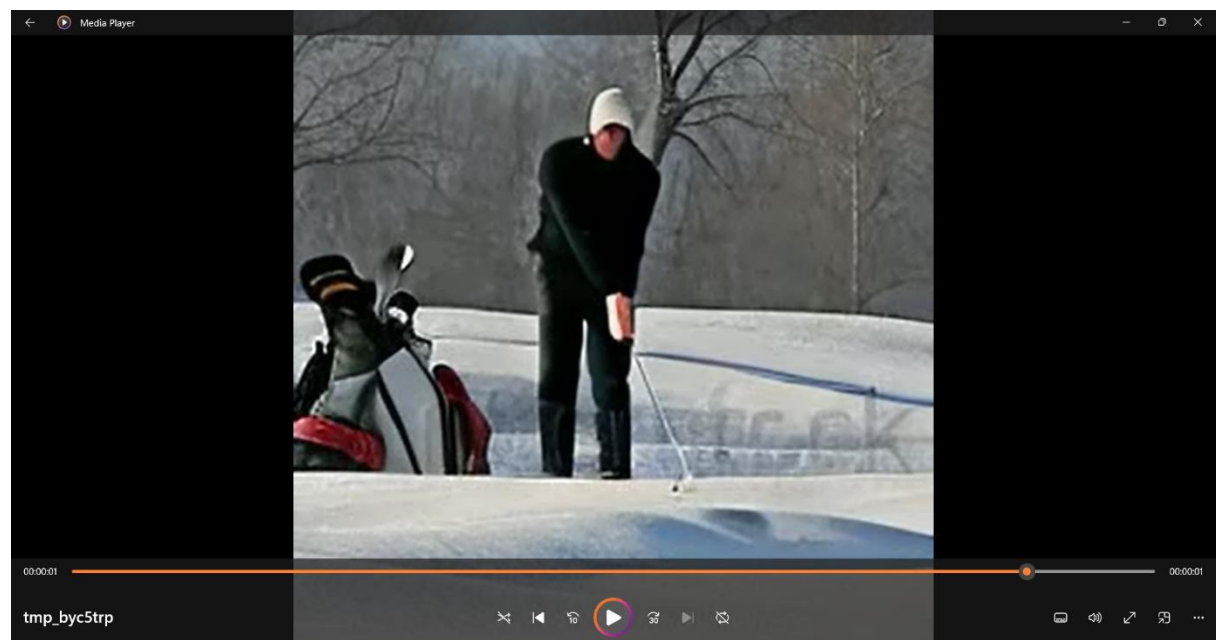
prompt = "A man playing golf on snow"
video_frames = pipe(prompt, num_inference_steps=25).frames
video_path = export_to_video(video_frames)
src_path=video_path

import shutil
dst_path = r"/content/drive/MyDrive"
shutil.move(src_path, dst_path)

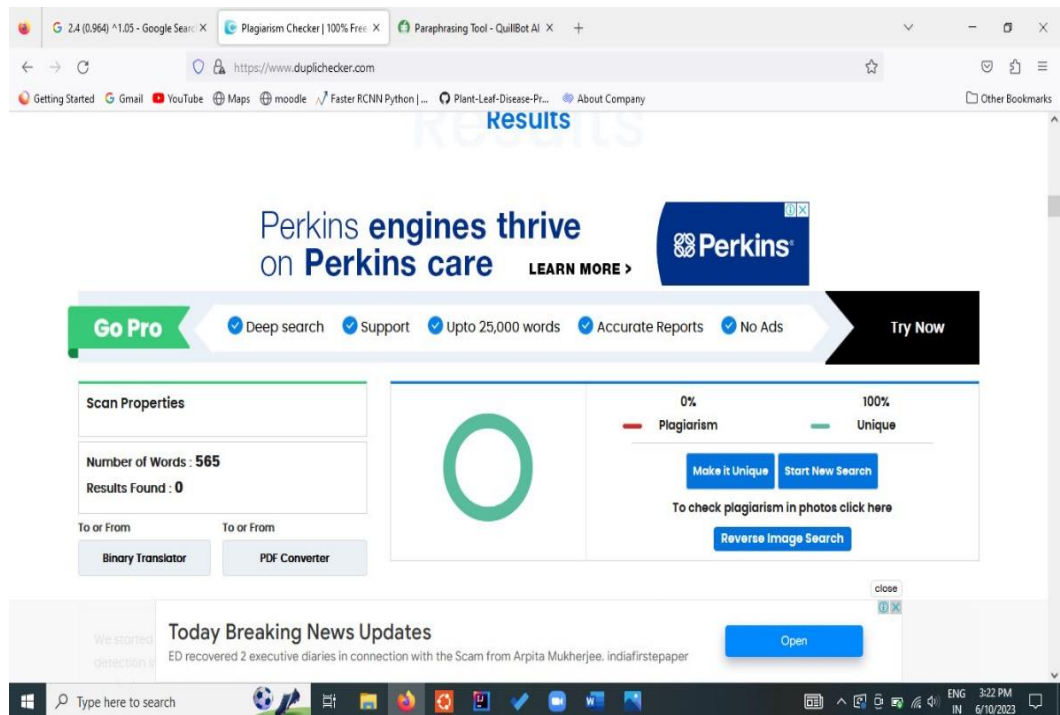
100% 25/25 [00:25<00:00, 1.42it/s]
'/content/drive/MyDrive/tmp_byc5trp.mp4'
```

The bottom status bar indicates 'Disk' usage and '51.25 GB available'.

### Step 8: Output.



# 11. Plagiarism Report



## Text to Video Generation

The screenshot shows the DupliChecker website interface. At the top, there's a navigation bar with links like 'Getting Started', 'Gmail', 'YouTube', 'Maps', 'moodle', 'Faster RCNN Python', 'Plant-Leaf-Disease-Pr...', and 'About Company'. Below the navigation bar, there's a 'Go Pro' button and a list of features: 'Deep search', 'Support', 'Upto 25,000 words', 'Accurate Reports', and 'No Ads'. The main content area displays 'Scan Properties' with 'Number of Words : 744' and 'Results Found : 1'. A donut chart shows '3% Plagiarism' and '97% Unique'. There are buttons for 'Make it Unique', 'Start New Search', and 'Reverse Image Search'. A 'Timed Out' message is visible on the left. The bottom of the page features a Windows taskbar with various application icons and a system clock showing 3:14 PM on 6/10/2023.

The screenshot shows the DupliChecker website interface. At the top, there's a navigation bar with links like 'Getting Started', 'Gmail', 'YouTube', 'Maps', 'moodle', 'Faster RCNN Python', 'Plant-Leaf-Disease-Pr...', and 'About Company'. Below the navigation bar, there's a 'Go Pro' button and a list of features: 'Deep search', 'Support', 'Upto 25,000 words', 'Accurate Reports', and 'No Ads'. The main content area displays 'Scan Properties' with 'Number of Words : 937' and 'Results Found : 2'. A donut chart shows '4% Plagiarism' and '96% Unique'. There are buttons for 'Make it Unique', 'Start New Search', and 'Reverse Image Search'. A 'Perkins engines thrive on Perkins care' advertisement is visible at the top. The bottom of the page features a Windows taskbar with various application icons and a system clock showing 3:03 PM on 6/10/2023.

## 12. Ethics

### **Creative Commons License**

#### **Attribution-Share Alike 4.0 International (CC BY SA 4.0)**

This is human-readable summary of (and not a substitute for) the license.

You are free to:

#### **Share:**

Copy and redistribute the material any medium or format.

#### **Adapt:**

Remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms.

#### **Using the following terms:**

**Attribution** you must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any responsible manner, but not in any way that suggests the licensor endorses you or your use.

#### **Share Alike:**

You remix, transform, and build upon the material for any purpose, you must distribute your contributions under the same license as the original.

No additional restrictions:

You must apply legal terms or technological measures that legally restrict others from doing anything the license permits.

#### **Notices:**

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an application exception or limitation. No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.



### **Ethical Practices for CSE Students:**

As Computer Sc. & Engineering student, I believe it is unethical to:

- Take credit for someone else's work.
- Hire someone to write an assignment.
- Purchase or submit a research or term paper from the internet to a class as one's own work.
- Cheat on a graded assignment.
- Cheat on an exam.
- Plagiarize other people's work without citing or referencing the work.
- Add the name of a non-contributing person as an author in project/research study.
- Copy and paste material found on the Internet for an assignment without acknowledging the authors of the Material.
- Deliberately provide inaccurate references for a project or research study

## 13. References

- [1] "Pseudo 3D Residual Networks" by Qiu et al. (2017)
- [2] "Spatiotemporal residual networks for video action recognition." by C. Feichtenhofer, A. Pinz, and R. Wildes. (2016)
- [3] "Non-local Neural Networks" by Wang et al. (2018)
- [4] "Temporal Shift Module for Efficient Video Understanding" by Lin et al. (2019).
- [5] "Large-scale video classification with convolutional neural networks" by A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei. (2014)
- [6] "GANs in Action: Deep learning with Generative Adversarial Networks" Written by Jakub Langr and Vladimir Bok, published in 2019.
- [7] "ImageNet classification with deep convolutional neural networks" by A. Krizhevsky, I. Sutskever, and G. E. Hinton. (2012)
- [8] "Encoding convolutional activations with deep generative model" by Z. Qiu, T. Yao, and T. Mei. Deep quantization (2017)