DreAmy

AYESHA ANWAAR

Spring 2024

Department of Computer Science
COMSATS University Islamabad
Attock Campus-Pakistan

COMSATS University Islamabad Attock Campus

Attock Pakistan

DreAmy

By

AYESHA ANWAAR CIIT/FA20-BCS-018/ATK

Supervisor

Dr. Fahad Khan

Bachelor of Science in Computer Science (2020-2024)

COMSATS University Islamabad Attock Campus

Attock Pakistan

DreAmy

A project presented to

COMSATS University, Islamabad

In partial fulfillment

Of the requirement for the degree of

Bachelors of Science in Computer Science (2020-2024)

By

AYESHA ANWAAR	**CIIT/FA20-BCS-018/ATK**

DECLARATION

We certify that this is my/our own work. The work has not, in whole or in part, been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged. If this statement is untrue, we acknowledge that we will have committed an assessment offence and shall be liable to punishable action under the Plagiarism rules of HEC.

AYESHA ANWAAR

FA20-BCS-018

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS) ―DreAmy was developed by AYESHA ANWAAR **(CIIT/FA20-BCS-018/ATK)** under the supervision of ―DR. Fahad Khan and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

1. External Examiner

DR. Javaid Iqbal
Assistant Professor
Dept. of Computer Science
University of Engineering and Technology,
Taxila

2. Supervisor

Dr. Khalid Iqbal
Associate Professor
Department of Computer Science
COMSATS University Islamabad
Attock Campus

3. Head of the Department

Dr. Zahoor ur Rehman
Associate Professor HOD(CS)
Department of Computer Science
COMSATS University Islamabad
Attock Campus

Executive Summary

The Image/Video to Animation Converter is a groundbreaking project that transforms static images and videos into engaging animations. This innovative system utilizes advanced algorithms and machine learning techniques to analyze and convert visual inputs into captivating animations With its user-friendly interface, users can upload images or videos, select customizable settings, and download animations with ease.

This project unlocks new creative possibilities for artists, designers, and content creators, saving time and effort in the animation process. The generated animations can be used in various applications, including social media, advertising, education, and art. By automating the animation process, this project enhances user experience and interaction, making it a valuable tool for various industries.

The Image/Video to Animation Converter has the potential to revolutionize the way we experience and interact with visual content. With its cutting-edge technology and user-friendly design, this project is poised to make a significant impact in the world of animation and visual storytelling. By converting images and videos into engaging animations, this project opens up new avenues for creativity, innovation, and expression.

Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledgeby virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor **Dr.Fahad Khan**. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. Weare grateful to them for their encouragement and continual help during this work. And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us with the values of honesty & hard work.

AYESHA ANWAAR

Abbreviations

SRS	Software Require Specification
PC	Personal Computer

Table of Contents

Table of figures

1 Introduction

1.1 Statement

Imagine a tool that turns everyday videos and photos into stunning cartoon-style art. That's what we're building—a top-notch solution that blends advanced computer vision tech with the flair of artistic expression. It’s all about making it super easy for anyone, whether they’re hobbyists or pros, to add a creative twist to their visuals.

We're using OpenCV because we want everyone to have a shot at this—no fancy skills required. Our goal? To mix innovation with user-friendliness so that our tool works for all sorts of people: content creators looking to spice up their work, educators who want more engaging materials, and even folks in entertainment seeking some fresh magic.

As trends evolve and new needs emerge, we plan on refining our product continuously. We see it becoming the go-to standard for turning plain images into eye-catching cartoons while building an enthusiastic community keen on exploring the limits of visual storytelling.

1.2 Related System Analysis/Literature Review

Table 1 Related System Analysis with proposed project solution

Application Name	Weakness	Proposed Project Solution
Toonify	Limited customization options for cartoonization parameters.	The proposed project will offer a comprehensive set of customization options, allowing users to fine-tune parameters such as line thickness, color palette, and stylization effects.

DeepArt	Relatively slow processing speed, especially for high-resolution images and videos.	By leveraging optimized algorithms and parallel processing techniques, the proposed project aims to significantly improve processing speed without compromising on output quality.
Cartoon Yourself	Limited support for batch processing of multiple images or videos.	The proposed project will include batch processing capabilities, enabling users to efficiently cartoonize large sets of visual content in a single operation.
Cartoonizer Pro	Lack of real-time preview functionality, requiring users to iterate through multiple trial-and-error runs.	The proposed project will incorporate real-time preview features, allowing users to visualize the cartoonization effect instantly and make adjustments in real-time for a more intuitive workflow.

1.3 Project Deliverables

1. Software Application: A functional system capable of transforming videos and images into cartoonized versions using OpenCV.

2. User Interface Design: A user-friendly interface design facilitating easy interaction with the application.

3. Documentation: Comprehensive documentation including installation instructions, user guide, and technical specifications.

4. Codebase: Well-organized and documented codebase suitable for maintenance and further development.

5. Final Report: A detailed report summarizing project objectives, methodologies, findings, and conclusions.

1.4 System Limitations/Constraints

LC-1: Processing Speed: The cartoonization process may be computationally intensive, leading to longer processing times for high-resolution videos and images.

LC-2: Hardware Requirements: The performance of the application may be limited by the hardware capabilities of the user's device, particularly in terms of CPU and GPU resources.

LC-3: Quality Variation: The cartoonization effect may vary depending on the input content and parameters chosen, leading to potential inconsistencies in output quality.

LC-4: Memory Usage: The application may require significant memory resources, especially when processing large videos or images, potentially limiting its usability on devices with limited RAM.

LC-5: Compatibility: The application may not be compatible with all operating systems or versions of OpenCV, restricting its accessibility to certain user groups.

1.5 Tools and Technologies

Table 2 Tools and Technologies for Proposed Project

	Tools	Version	Rationale
	Python	3.8.5	Main programming language for implementing image and video processing algorithms.

Tools And Technologies	OpenCV	4.5.3	Computer vision library used for image and video processing, including cartoonization.
	Flask	2.0.1	Lightweight web framework for building the backend of the web application.
	HTML/CSS/JavaScript	-	Frontend development for creating the user interface of the web application.

1.6 Relevance to Course Modules

Computer Vision: The project heavily relies on concepts and techniques from computer vision, such as edge detection, image processing, and feature extraction, which are fundamental topics covered in computer vision courses.

Software Engineering: The development of the software application involves principles of software engineering, including requirements analysis, design patterns, and software testing, aligning with the curriculum of software engineering courses.

Data Structures and Algorithms: Optimization of the cartoonization process requires efficient algorithms and data structures, which are core topics covered in data structures and algorithms courses.

Artificial Intelligence: The project intersects with artificial intelligence through the utilization of machine learning algorithms for image analysis and pattern recognition, topics commonly taught in AI courses.

Human-Computer Interaction (HCI): Designing an intuitive user interface involves understanding principles of HCI, such as user-centered design, usability testing, and interaction design, which are integral to HCI courses.

Operating Systems: Understanding the underlying mechanisms of operating systems is essential for optimizing performance and resource management, which are pertinent to the project's development and covered in operating systems courses.

2 Problem Definition

2.1 Problem Statement

Currently, individuals seeking to transform videos and images into cartoonized versions face several challenges. Existing solutions often lack customization options, resulting in cartoonized content that may not fully align with users' preferences. Furthermore, the process can be time-consuming and technically complex, requiring expertise in computer vision techniques and software development. As a result, many potential users, including hobbyists, educators, and professionals in the creative industry, may be deterred from utilizing cartoonization techniques in their projects. Our software system aims to address these challenges by providing a user-friendly platform that enables seamless transformation of visual content into captivating cartoons. By leveraging the power of OpenCV and implementing advanced computer vision algorithms, our system will offer a range of customization options, allowing users to adjust parameters such as line thickness, color palette, and stylization effects. Additionally, our system will prioritize processing speed and output quality, ensuring a satisfying user experience. Ultimately, our software system seeks to democratize the process of cartoonization, empowering users to unleash their creativity without the barriers of technical complexity or limited customization options.

2.2 Problem Solution

1. Efficiency: Streamline the process of cartoonization to save users time and effort compared to manual methods or complex software solutions.

2. Quality: Ensure high-quality output by optimizing processing algorithms and techniques to preserve the essence of the original content while achieving a visually appealing cartoonized effect.

3. Accessibility: Develop a user-friendly interface that is intuitive and easy to navigate, making the application accessible to users with varying levels of technical expertise.

4. Performance: Prioritize processing speed to provide users with real-time or near-real-time feedback on the cartoonization effect, enhancing the overall user experience.

5. Versatility: Enable users to apply cartoonization effects to both videos and images, supporting a wide range of creative projects and applications.

6. Feedback Loop: Establish mechanisms for gathering user feedback and incorporating it into future updates and improvements, ensuring that the application evolves to meet the changing needs and expectations of its user base.

2.3 Objectives of the Proposed System

1: Develop a user-friendly software application capable of transforming videos and images into cartoonized versions using OpenCV.

2: Optimize processing algorithms to prioritize speed and output quality, ensuring a seamless and satisfying user experience.

3: Design an intuitive user interface that is accessible to users with varying levels of technical expertise, facilitating ease of navigation and interaction.

4: Support both batch processing and real-time preview functionalities, enabling users to efficiently cartoonize multiple images or videos and visualize the effect instantly.

2.4 Scope

The Our project aims to develop a user-friendly software application utilizing OpenCV for transforming images and videos into cartoonized versions. By implementing computer vision techniques like edge detection and color quantization, we aim to emulate the visual characteristics of hand-drawn cartoons. Users can easily upload their visual content, apply the cartoonization effect, and preview the results in real-time. The software supports batch processing for efficiency and enables users to download the converted images or videos directly from the interface.

Continuous support and engagement channels ensure user collaboration and evolving needs are addressed, with secure protocols safeguarding user data and content integrity. Overall, the application provides a seamless solution for cartoonizing visual content while empowering users with convenient access and utilization options.

2.5 Modules

2.5.1 Module 1: Cartoonization

- Upload videos or images for cartoonization.

- Apply cartoonization effect to uploaded content.

- Provide real-time preview of cartoonization effect.

- Download cartoonized images or videos.

2.5.2 Module 2: User Interaction

- Design intuitive and user-friendly interface.

- Facilitate community engagement through forums and social media.

3 Requirement Analysis

3.1 Context Diagram:

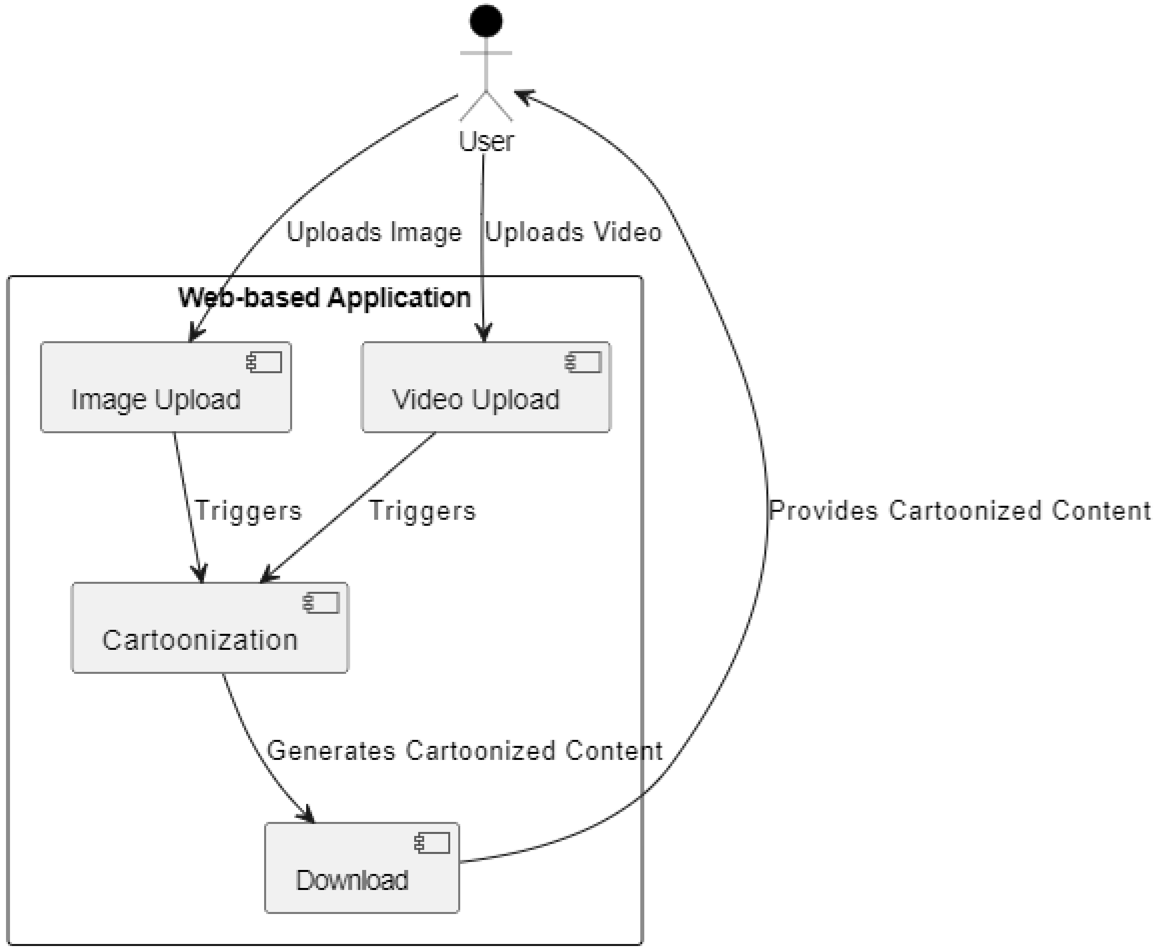


Figure 1 context diagram

3.1.1 User classes and characteristics

Table 3 User classes and characteristics

User class	**Description**
Content Creator	A Content Creator is an individual who produces videos or images for various purposes, including entertainment, education, or marketing. They may range from amateur enthusiasts to professional artists or filmmakers. Content Creators are likely to have varying levels of technical expertise in using software tools for image and video editing.
Graphic Designer	A Graphic Designer is a professional who specializes in visual communication and graphic design. They may use cartoonization techniques to enhance their design projects, create digital artwork, or develop branding materials. Graphic Designers are typically proficient in using design software and may require specific features for fine-tuning cartoonization effects.
Educator	An Educator is a teacher or instructor who utilizes visual content for educational purposes. They may use cartoonization techniques to create engaging and visually appealing educational materials, presentations, or tutorials. Educators may have limited technical expertise in image and video editing software but require user-friendly tools for creating educational content.
Marketing Professional	A Marketing Professional is an individual responsible for promoting products or services through various marketing channels. They may use cartoonization techniques for creating promotional videos, advertisements, or social media content. Marketing Professionals may require customizable features for aligning cartoonization effects with brand identity and messaging.
Hobbyist	A Hobbyist is an individual who engages in cartoonization as a leisure activity or creative outlet. They may use cartoonization techniques for personal projects, such as creating cartoons or animations for fun. Hobbyists may vary in their level of expertise, ranging from beginners to experienced enthusiasts, and require intuitive tools for exploring and experimenting with cartoonization effects.

3.2 Requirement Identifying Technique

3.2.1 Use Case:

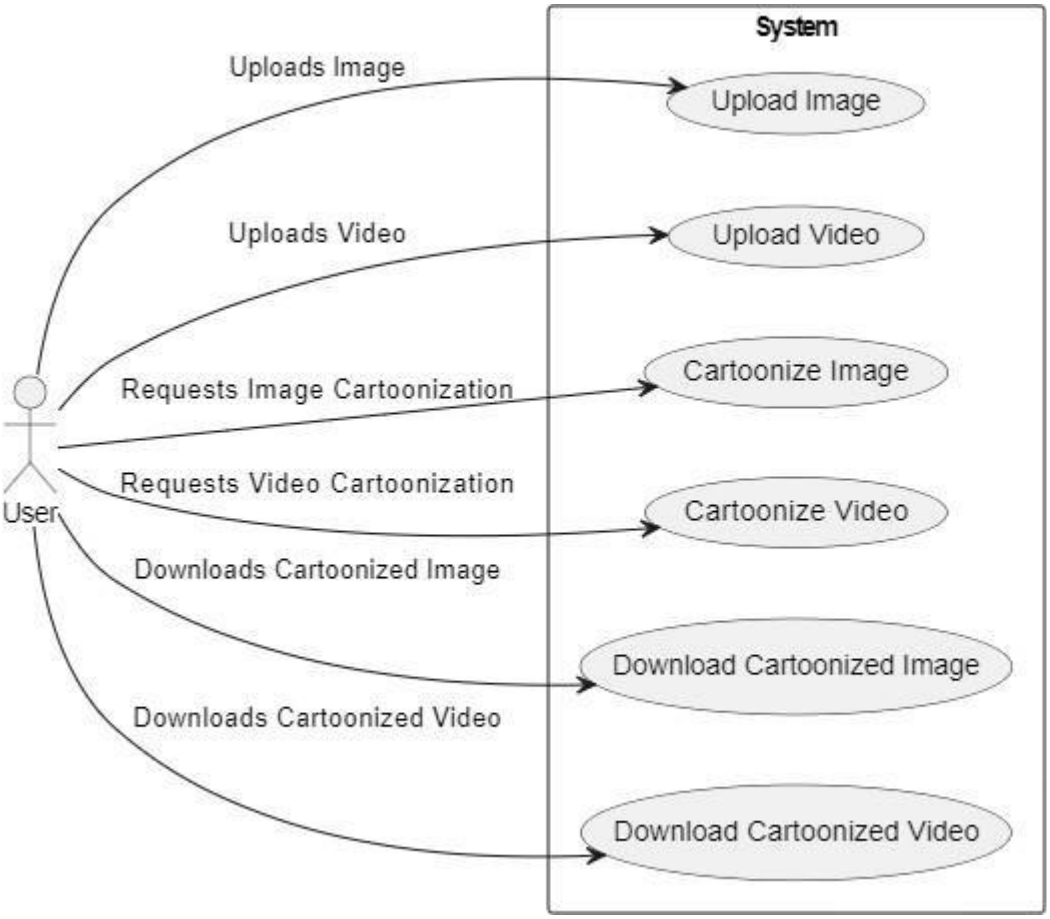

System
Uploads Image
Upload Image
Uploads Video
Upload Video
Requests Image Cartoonization
Cartoonize Image
Requests Video Cartoonization
Cartoonize Video
User
Downloads Cartoonized Image
Download Cartoonized Image
Downloads Cartoonized Video
Download Cartoonized Video

Figure 2 use-case diagram

3.2.2 Detail Use Case:

Table 4 Use-Case 1

Use Case ID	**UC-1**
Use Case Name	Upload Image
Actors	User
Description	The User accesses the system and initiates the process to upload an image file for cartoonization. Upon successful upload, the system processes the image and generates a cartoonized version of it. The cartoonized image is made available for download by the User.
Trigger	The User indicates the desire to upload an image for cartoonization.
Preconditions	The User is logged into the system.The User has access to an image file to upload.
Postconditions	The cartoonized image is successfully generated and made available for download.
Normal Flow	The User navigates to the upload image section of the system. The User selects the image file to upload. The system validates the uploaded file. If the file is valid, the system processes the image to generate a cartoonized version. The cartoonized image is displayed to the User for preview. The User confirms the cartoonized image. The system stores the cartoonized image and makes it available for download.
Alternative Flows	If the uploaded file is invalid or incompatible, the system displays an error message and prompts the User to upload a valid image file.

Table 5 use-case 2

Use Case ID	**UC-2**
Use Case Name	Upload Video
Actors	User
Description	The User initiates the process to upload a video file for cartoonization. Upon successful upload, the system processes the video and generates a cartoonized version of it. The cartoonized video is made available for download by the User.
Trigger	The User indicates the desire to upload a video for cartoonization.
Preconditions	The User is logged into the system. The User has access to a video file to upload.
Postconditions	The cartoonized video is successfully generated and made available for download.
Normal Flow	The User navigates to the upload video section of the system. The User selects the video file to upload. The system validates the uploaded file. If the file is valid, the system processes the video to generate a cartoonized version. The cartoonized video is displayed to the User for preview. The User confirms the cartoonized video. The system stores the cartoonized video and makes it available for download.
Alternative Flows	If the uploaded file is invalid or incompatible, the system displays an error message and prompts the User to upload a valid video file.

3.3 Event Response:

Table 6 Event Response

Event	System State	Response
User uploads image file	System is idle	Begin processing the uploaded image for cartoonization
Image processing complete	Cartoonized image is generated	Display the cartoonized image to the user for preview
User confirms cartoonized image	Cartoonized image is displayed	Provide options for the user to download the cartoonized image
User selects download option	Cartoonized image is available for download	Initiate the download process for the user
User uploads video file	System is idle	Begin processing the uploaded video for cartoonization
Video processing complete	Cartoonized video is generated	Display the cartoonized video to the user for preview
User confirms cartoonized video	Cartoonized video is displayed	Provide options for the user to download the cartoonized video
User selects download option	Cartoonized video is available for download	Initiate the download process for the user

Video processing complete	Cartoonized video is generated	Display the cartoonized video to the user for preview

3.4 Functional Requirements

Feature: Image Cartoonization

The system shall allow users to upload images for cartoonization.

Users can upload images in formats such as jpg

The cartoonization process shall utilize computer vision algorithms to transform the images.

The system shall allow users to download the cartoonized images.

Feature: Video Cartoonization

The system shall allow users to upload videos for cartoonization.

Users can upload videos in common formats such as MP4.

Upon upload, the system shall verify the file format and size to ensure compatibility.

3.4.1 Functional Requirement X

Table 7 Description of FR

Identifier	**FR-1**
Title	Upload Videos/Images for Cartoonization

Requirement	The user shall be able to upload videos or images to the system for cartoonization.
Source	User request
Rationale	To provide users with the ability to input their visual content into the system for processing.
Priority	High

Identifier	**FR-2**
Title	Apply Cartoonization Effect
Requirement	The system shall apply the cartoonization effect to the uploaded videos or images.
Source	System functionality
Rationale	To process the uploaded content and transform it into a cartoonized version as requested by the user.
Priority	High

Identifier	**FR-3**
Title	Support Batch Processing
Requirement	The system shall support batch processing of multiple videos or images for cartoonization.
Source	User request
Rationale	To enable users to process multiple files simultaneously, saving time and effort.
Priority	Medium

Identifier	**FR-4**
Title	Download Cartoonized Output
Requirement	The user shall be able to download the cartoonized images or videos from the system.
Source	User request
Rationale	To provide users with the ability to save and use the cartoonized output as desired.
Priority	High

3.5 Non-Functional Requirements

3.5.1 Reliability

The software aims to achieve a mean time between failures (MTBF) of at least 1000 hours. A failure encompasses any event causing deviation from specified behavior, resulting in incorrect output or system downtime. Robust error handling mechanisms will be implemented to gracefully manage unexpected situations and prevent system crashes. Thorough testing, including unit, integration, and system tests, will be conducted for error detection before deployment. In the event of failure, a corrective strategy will involve identifying root causes, implementing necessary fixes or patches, and deploying updates promptly. Continuous monitoring and performance analysis will be conducted to detect failure patterns and implement preventive measures, enhancing system reliability over time.

3.5.2 Usability

The user interface is designed to be user-friendly and straightforward, ensuring users can easily grasp and navigate the system with minimal training. Clear instructions will accompany the interface, guiding users through tasks effectively. To reduce errors, the system will provide feedback on user actions and offer guidance for correct inputs. In the event of errors, informative messages will be displayed, explaining the issue and suggesting solutions for recovery. Interactions with the system will be efficient, featuring streamlined workflows and minimizing user input to accomplish tasks.

3.5.3 Performance

The system is designed to efficiently process image cartoonization requests, aiming for an average processing time of 5 seconds per image and 10 seconds per minute of video content. It supports concurrent processing of up to 100 image and 50 video cartoonization requests without compromising performance. User interactions, such as file uploads or settings adjustments, are expected to have a response time of less than 1 second. Additionally, the system maintains a minimum throughput of 100 image and 50 video cartoonization requests per hour during peak usage periods. To handle fluctuations in demand, the system automatically scales resources to ensure optimal performance and prevent slowdowns or service disruptions.

3.5.4 Security

The system prioritizes security by implementing comprehensive measures to safeguard against unauthorized access, data breaches, and malicious attacks. Access to administrative functions and sensitive data is restricted to authorized personnel through role-based access controls. Regular security assessments and penetration testing are conducted to proactively identify and mitigate potential vulnerabilities. Additionally, the system maintains thorough logs and monitors access attempts, unauthorized activities, and security events to promptly detect and respond to any security breaches.

3.6 External Interface Requirements

3.6.1 User Interfaces Requirements

The user interface of the system adheres to GUI standards and follows a consistent style guide to ensure a cohesive and intuitive user experience. Standard fonts, icons, button labels, and color schemes are used throughout the interface to maintain visual consistency. The layout of each screen is designed to accommodate various screen resolutions, ensuring optimal display on different devices. Standard buttons for common functions, such as uploading files, previewing content, and downloading processed files, are available on every screen for easy access. Additionally, navigation links are provided to facilitate seamless transition between different sections of the application, including the Home Screen, Privacy Policy, and About Us pages. Message display conventions are implemented to provide clear feedback to users regarding their actions or system status. The layout standards are designed with localization in mind, allowing for easy translation of text and adaptation to different languages. Accommodations for visually impaired users are included, such as alternative text for images and adherence to accessibility guidelines for screen reader compatibility. Specific dialog box layouts and screen mock-ups are documented in a separate user interface specification to provide detailed guidance for implementation while making it clear that the mock-ups are not final designs.

3.6.2 Software interfaces

SI-1: OpenCV Library

The system shall utilize the OpenCV library version 4.5.3 for implementing image and video processing algorithms to apply the cartoonization effect. The OpenCV library shall be integrated into the system's codebase to provide functionalities such as image filtering, edge detection, and color manipulation. Image and video processing operations, including cartoonization, shall be performed using functions and modules provided by the OpenCV library.

SI-2: Web Browser

Users shall access the system's web interface using modern web browsers such as Google Chrome, Mozilla Firefox, or Microsoft Edge. The system's web interface shall be compatible with the latest versions of web browsers to ensure optimal user experience and functionality.HTML, CSS, and JavaScript technologies shall be utilized to create dynamic and interactive user interfaces within web browsers.

SI-3: File System

The system shall interact with the underlying file system of the operating system to store, retrieve, and manage image and video files uploaded by users. Image and video files shall be stored in designated directories within the file system, organized based on user identifiers or timestamps. File I/O operations shall be performed using standard file handling libraries provided by the programming language and operating system.

SI-4: Operating System

The system shall be compatible with major operating systems such as Windows, macOS, and Linux distributions. Operating system-specific functionalities and system calls may be utilized for tasks such as file management, process scheduling, and memory allocation. The system's deployment environment and dependencies shall be verified to ensure compatibility with the targeted operating systems.

3.6.3 Hardware interfaces

The characteristics of each interface between the software components and hardware components of the system are as follows:

1. **Image Input Interface**:

The software component interacts with the hardware component to capture image data and transfer it to the system for processing. Supported Device Types: Webcams, digital cameras, smartphones, and other image-capturing devices Communication Protocols: USB, Wi-Fi, or Bluetooth protocols may be utilized for data transfer between the image-capturing devices and the system.

2. **Video Input Interface:**

The image input interface, the software component interacts with the hardware component to capture video data and transfer it to the system for processing. Supported Device Types: Webcams, digital camcorders, smartphones, and other video-capturing devices. Communication Protocols: USB, Wi-Fi, or Bluetooth protocols may be utilized for data transfer between the video-capturing devices and the system.

3. **File Output Interface**:

The software component writes the processed image and video data to storage devices for saving or further distribution. Supported Device Types: Hard disk drives, solid-state drives, external storage devices. Communication Protocols: File I/O operations are performed using standard operating system interfaces and protocols for data storage and retrieval.

4. **User Interface (UI) Interface**:

The software component interacts with input devices (e.g., mouse, keyboard, touchscreen) to receive user commands and display information to users via output devices (e.g., monitor, screen). Supported Device Types: Desktop computers, laptops, tablets, smartphones. Communication Protocols: Web-based UIs may utilize HTTP or WebSocket protocols for communication between the client and server components.

3.6.4 Communications interfaces

The system shall utilize HTTP/HTTPS protocols for communication between the user's web browser and the server to facilitate user interactions with the web interface. Web browser compatibility shall be ensured for major browsers including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari, to provide consistent access and functionality across different platforms. The system shall employ secure communication protocols (e.g., SSL/TLS) to encrypt data transmitted between the client and server components, ensuring the confidentiality and integrity of user data and system communications.

4 Design and Architecture.

4.1 Architectural Design

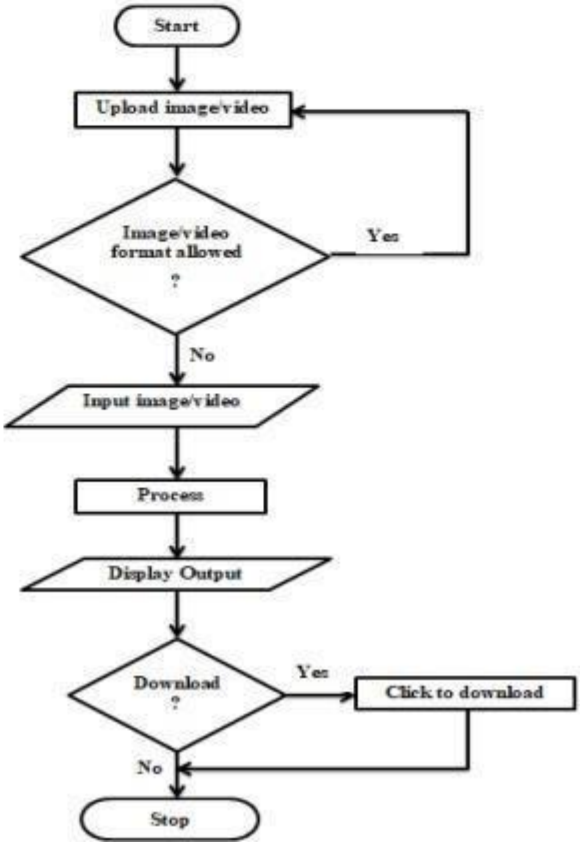


Figure 3 architectural design

4.2 Box-Line Diagram:

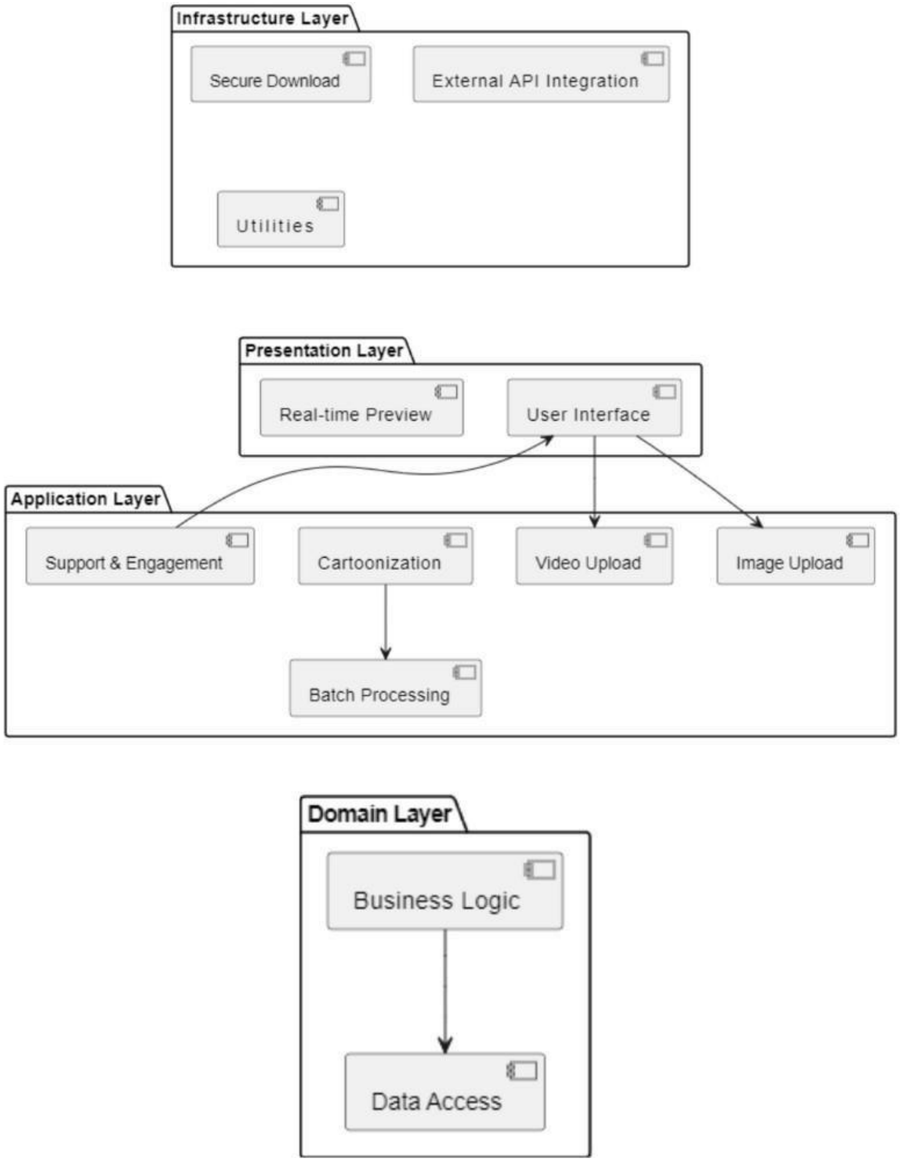


Figure 4 box-line diagram

4.3 Design Models

4.3.1 Activity Diagram:

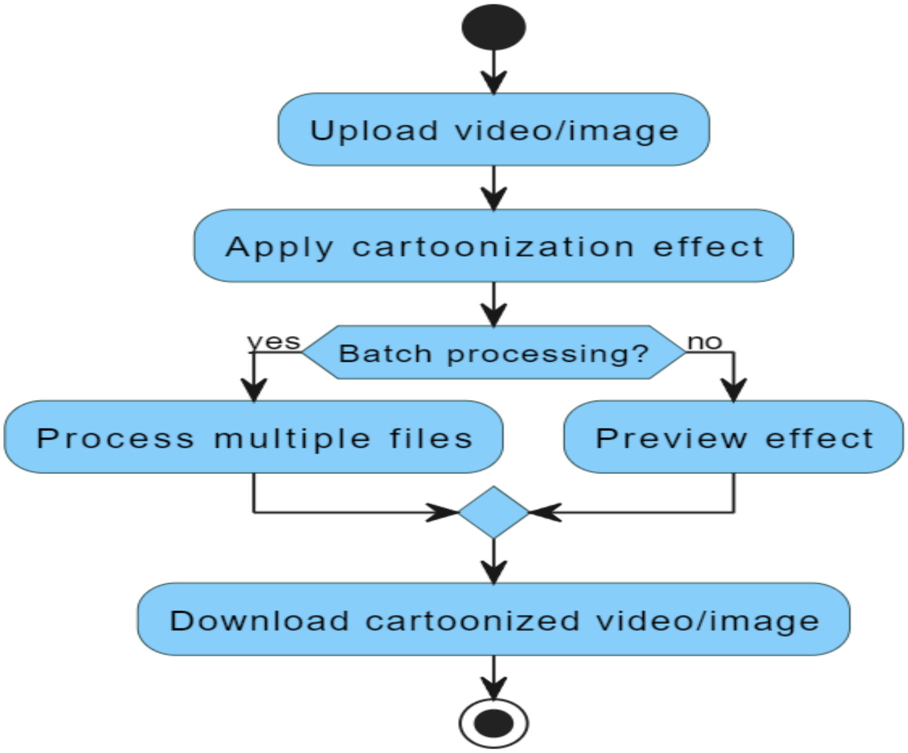


Figure 5 activity diagram

Table 8 activity diagram description

Activity	**Description**
Start	The starting point of the activity diagram, indicating the beginning of the process.
Upload video/image	Represents the user uploading a video or image file to the system.

Apply cartoonization effect	After the file is uploaded, the system applies the cartoonization effect to the uploaded video/image.
Batch processing?	Checks if the user wants to process multiple files simultaneously.
Process multiple files	If the user opts for batch processing, the system processes multiple files in one go.
Preview effect	If the user chooses not to batch process, they can preview the cartoonization effect applied.
Download cartoonized video/image	Allows the user to download the cartoonized version of the video/image from the system.
Stop	The endpoint of the activity diagram, indicating the completion of the process.

4.3.2 Class Diagram:

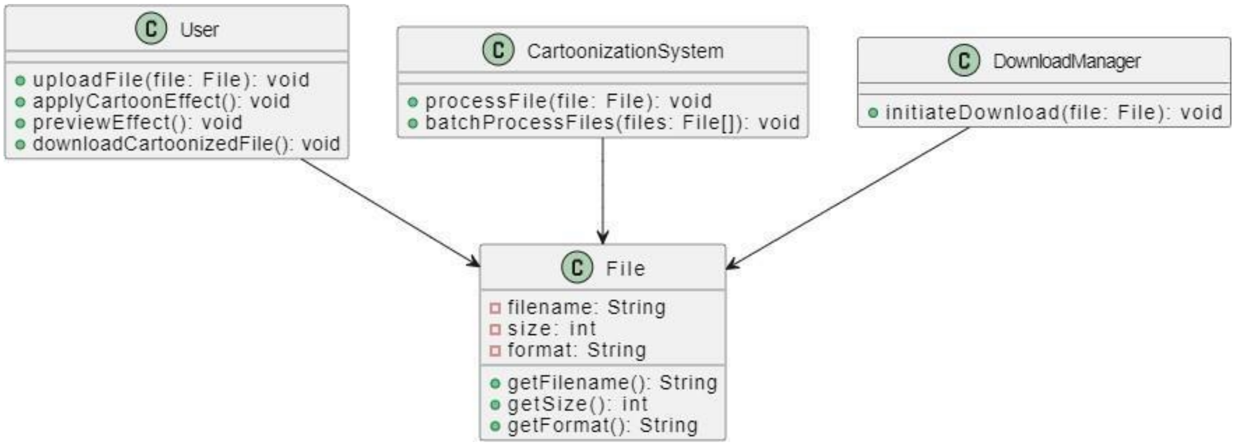

C User
uploadFile(file: File): void
applyCartoonEffect(): void
previewEffect(): void
downloadCartoonizedFile(): void
C CartoonizationSystem
processFile(file: File): void
batchProcessFiles(files: File[]): void
C DownloadManager
initiateDownload(file: File): void
C File
filename: String
size: int
format: String
getFilename(): String
getSize(): int
getFormat(): String

Figure 6 class diagram

Table 9 class diagram description

Class Name	**Attributes**	**Methods**	**Relationships**
User	username: String password: String	uploadFile(file: File): void applyCartoonEffect(): void previewEffect(): void downloadCartoonizedFile(): void	Uploads and interacts with File objects.
File	filename: String size:int format: String	getFilename(): String getSize(): int getFormat(): String	Represents files uploaded to the system.
Cartoonization System		processFile(file:File): void batchProcessFiles(files: File[]): void	Utilizes File objects for processing.
DownloadManager		initiateDownload(file:File): void	Facilitates downloading of File objects.

4.3.3 Sequence Diagram:

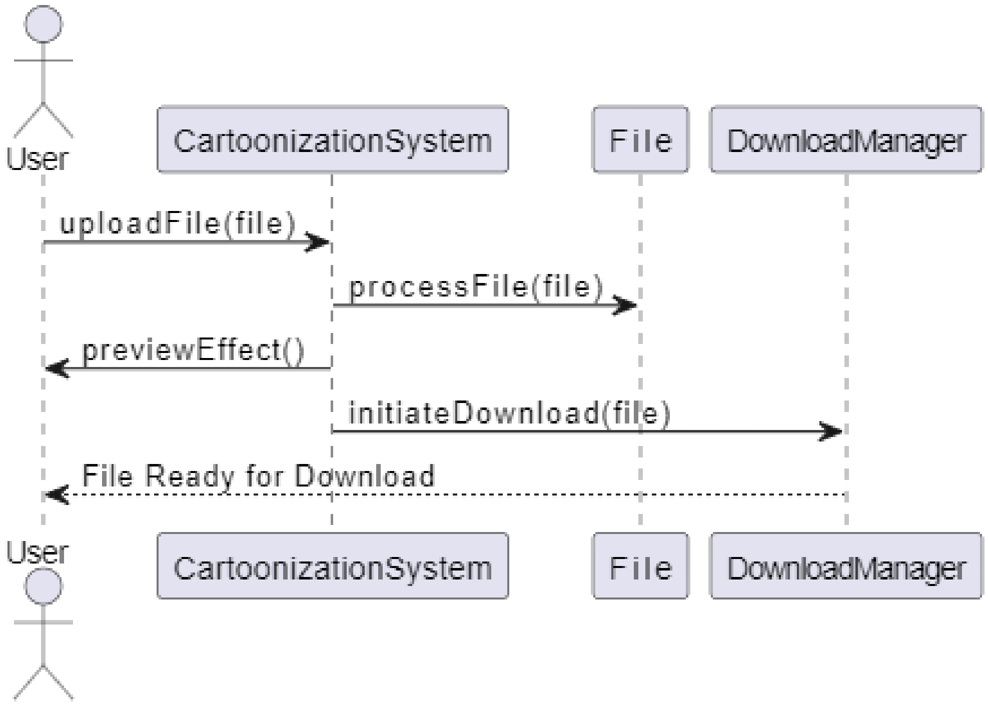


Figure 7 sequence diagram

Table 10 sequence diagram description

Step	**Actor**	**Actions**
1	User	Initiates the file upload process.
2		Sends the uploaded file to the system.
3		Requests to preview the cartoonization effect.
4	CartoonizationSystem	Processes the uploaded file.
5		Generates a preview of the cartoonized file.

6		Requests to download the cartoonized file.
7	DownloadManager	Initiates the download process for the file.
8		Prepares the file for download and notifies the user.

4.3.4 State Transition Diagram:

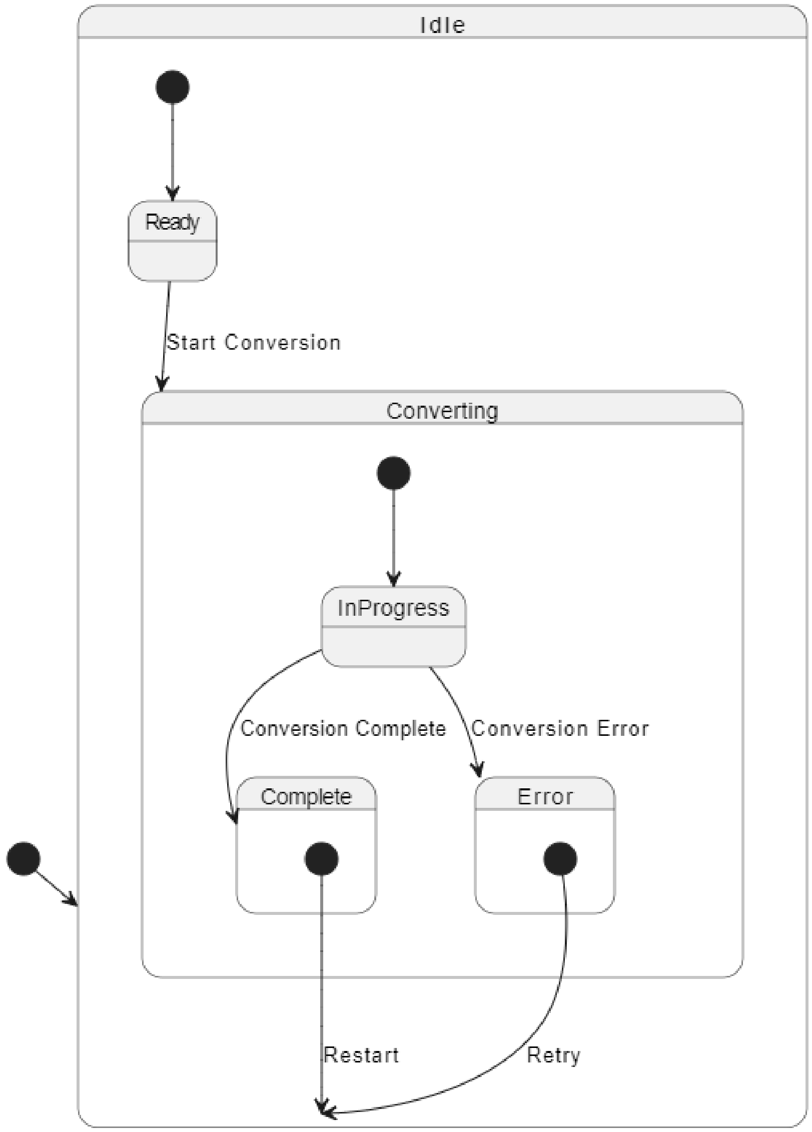


Figure 8 state transition diagram

Table 11 state diagram description

State	**Description**	**Transitions**
Idle	Initial state where the system is waiting for input or action	- Ready: Indicates that the system is ready to start conversion
Ready	System is ready to start the conversion process	- Converting: Transition to start the conversion process
Converting	Conversion process is in progress	- InProgress: Indicates that the conversion is ongoing
InProgress	Conversion process is currently ongoing	- Complete: Transition to indicate successful completion of conversion - Error: Transition to indicate an error during conversion
Complete	Conversion process has been successfully completed	- Idle: Transition to return to the idle state and await further instructions
Error	An error occurred during the conversion process	- Idle: Transition to return to the idle state and await further instructions

4.4 Data Design

The system operates within an information domain where user-provided images and videos are transformed into structured data for storage, processing, and management. These data, including original and cartoonized images and videos, are housed in a central repository. To facilitate storage, image and video files are represented as file objects or binary data structures. Utilizing specialized algorithms, the system processes uploaded files to apply the cartoonization effect, generating cartoonized versions. Both original and cartoonized media are stored within the repository, potentially organized based on user identifiers, timestamps, or metadata attributes. Efficient storage and retrieval are facilitated through relational databases or file storage systems, which accommodate various data items such as image and video repositories, error logs, user profiles, and system configurations.

4.4.1 Data Dictionary

1. **Cartoonized Image**

Table 12 cartoonized image

Type	**Description**
File or Binary Data	The output image after applying the cartoonization effect to uploaded images.

2. **Cartoonized Video**

Table 13 Cartoonized Video

Type	**Description**
File or Binary Data	The output video after applying the cartoonization effect to uploaded videos.

4. Image File

Table 14 Image File

Type	**Description**
File or Binary Data	The original image uploaded by the user to be cartoonized.

5. User Input

Table 15 User Input

Type	**Description**
String, File, or Binary Data	Data provided by the user through the system's interface, including image and video files for cartoonization.

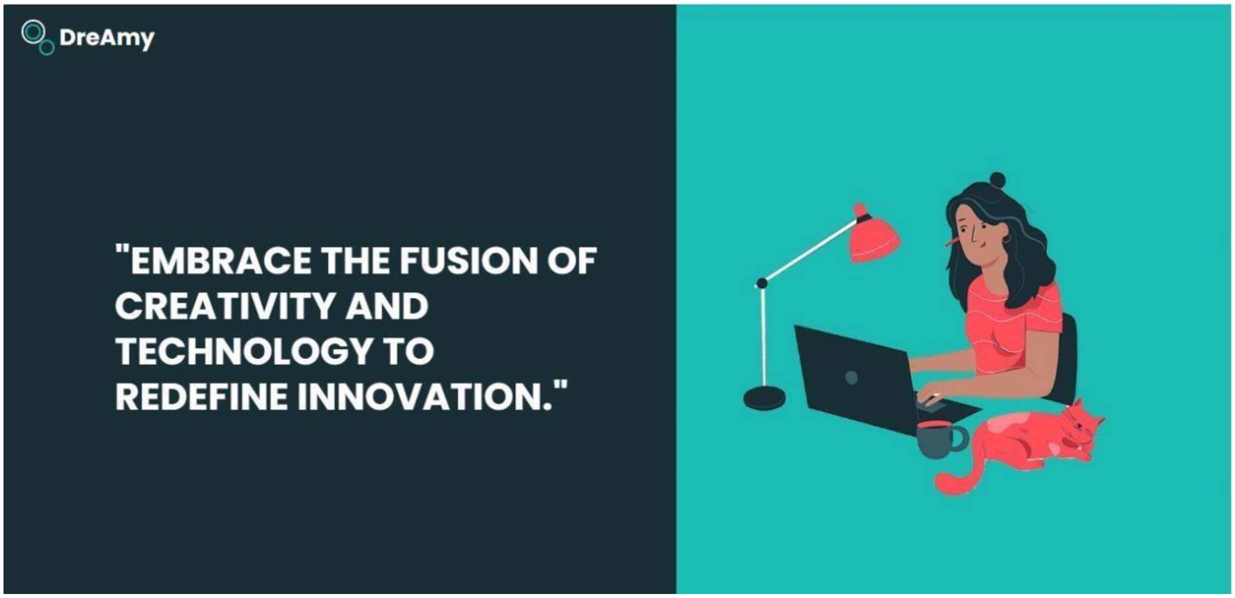
6. Video File

Table 16 Video File

Type	**Description**
File or Binary Data	The original video uploaded by the user to be cartoonized

4.5 Human Interface Design

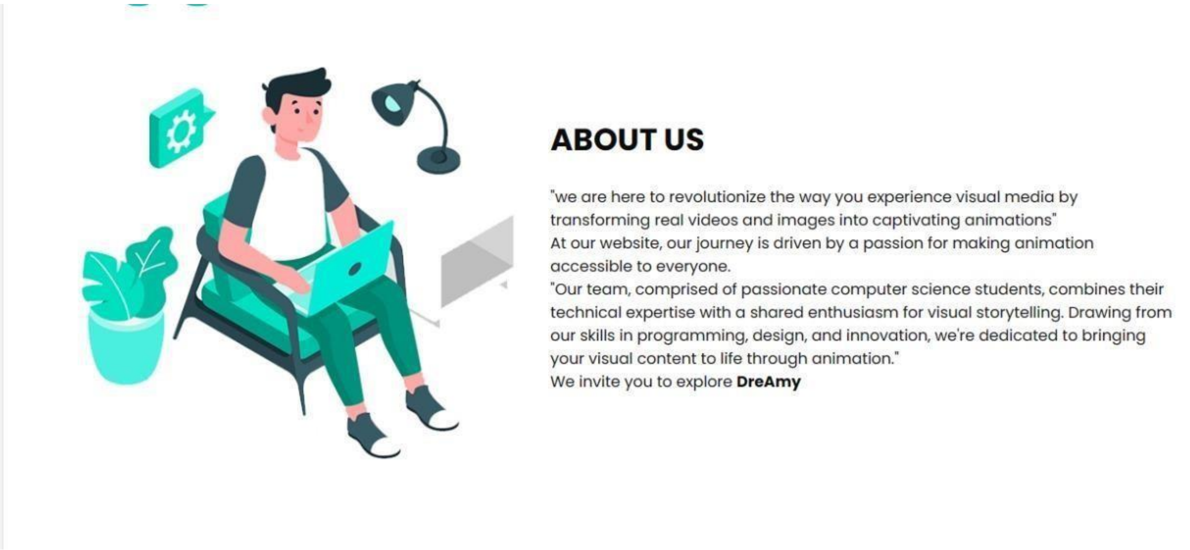
4.5.1 Screen Images



DreAmy

"EMBRACE THE FUSION OF CREATIVITY AND TECHNOLOGY TO REDEFINE INNOVATION."

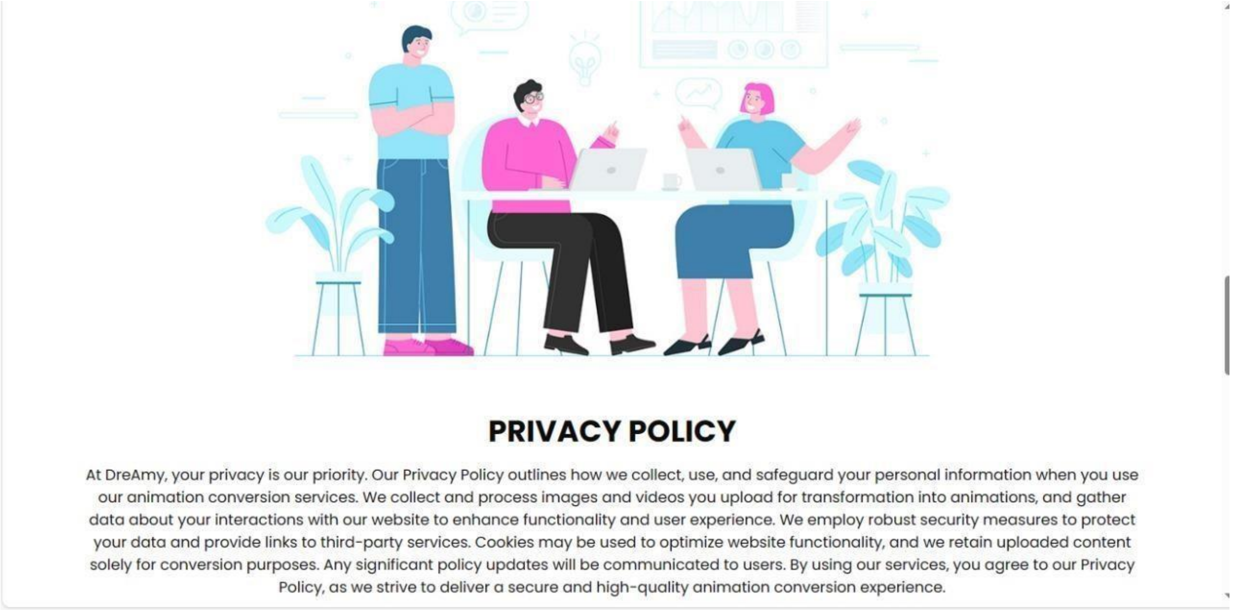
Figure 9 home



ABOUT US

"we are here to revolutionize the way you experience visual media by transforming real videos and images into captivating animations"
At our website, our journey is driven by a passion for making animation accessible to everyone.
"Our team, comprised of passionate computer science students, combines their technical expertise with a shared enthusiasm for visual storytelling. Drawing from our skills in programming, design, and innovation, we're dedicated to bringing your visual content to life through animation."
We invite you to explore **DreAmy**

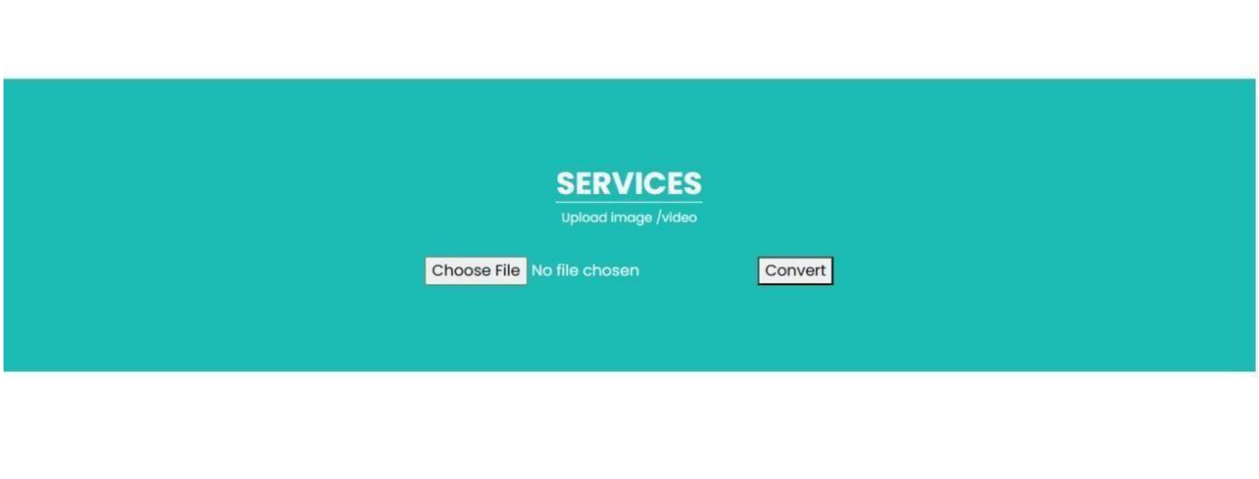
Figure 10 about-us



PRIVACY POLICY

At DreAmy, your privacy is our priority. Our Privacy Policy outlines how we collect, use, and safeguard your personal information when you use our animation conversion services. We collect and process images and videos you upload for transformation into animations, and gather data about your interactions with our website to enhance functionality and user experience. We employ robust security measures to protect your data and provide links to third-party services. Cookies may be used to optimize website functionality, and we retain uploaded content solely for conversion purposes. Any significant policy updates will be communicated to users. By using our services, you agree to our Privacy Policy, as we strive to deliver a secure and high-quality animation conversion experience.

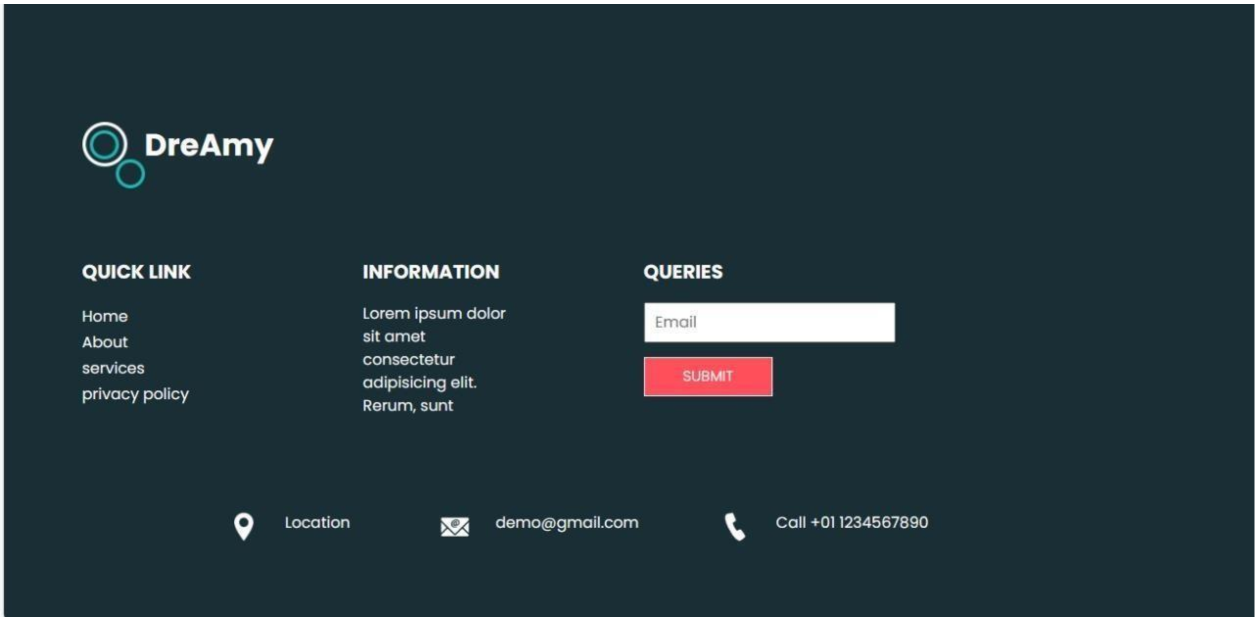
Figure 11 privacy policy



SERVICES

Upload image /video

Figure 12 services



Lorem ipsum dolor sit amet consectetur adipisicing elit. Rerum, sunt

Figure 13 contact us

4.5.2 Screen Objects and Actions

The project interface comprises three main screens: the Home Screen, Privacy Policy, and About Us. where users can upload their files using the designated button. Once uploaded, the system processes the content, whether it's an image or a video, utilizing the specified algorithms. Users can then view the processed content in the preview area and choose to download it using the download button. Additionally, users can navigate between screens using the navigation menu, which offers access to the Privacy Policy and About Us sections. The Privacy Policy screen provides detailed information about data privacy and security measures, while the About Us section offers insights into the project's objectives and team members. This interface design aims to provide users with a straightforward experience, allowing them to interact seamlessly with the system while accessing essential project-related information.

5 Implementation

5.1 Algorithms

Algorithm no 1.

Table 17 algorithm 1

Algorithm: process photo	
Input: image_path	Output: processed_image
1. Read the image using OpenCV.	1. Convert the image to RGB format.
2. Convert the image to RGB format.	2. Convert the image to grayscale.
3. Convert the image to grayscale.	3. Apply median blur to the image.
4. Apply median blur to the image.	4. Apply edge detection to the image.
5. Apply edge detection to the image.	5. Apply bilateral filter to remove noise.
6. Apply bilateral filter to remove noise.	6. Erode and dilate the image.
7. Erode and dilate the image.	7. Stylize the image.
8. Stylize the image.	8. Return the processed image.

Algorithm no 2.

Table 18 algorithm 2

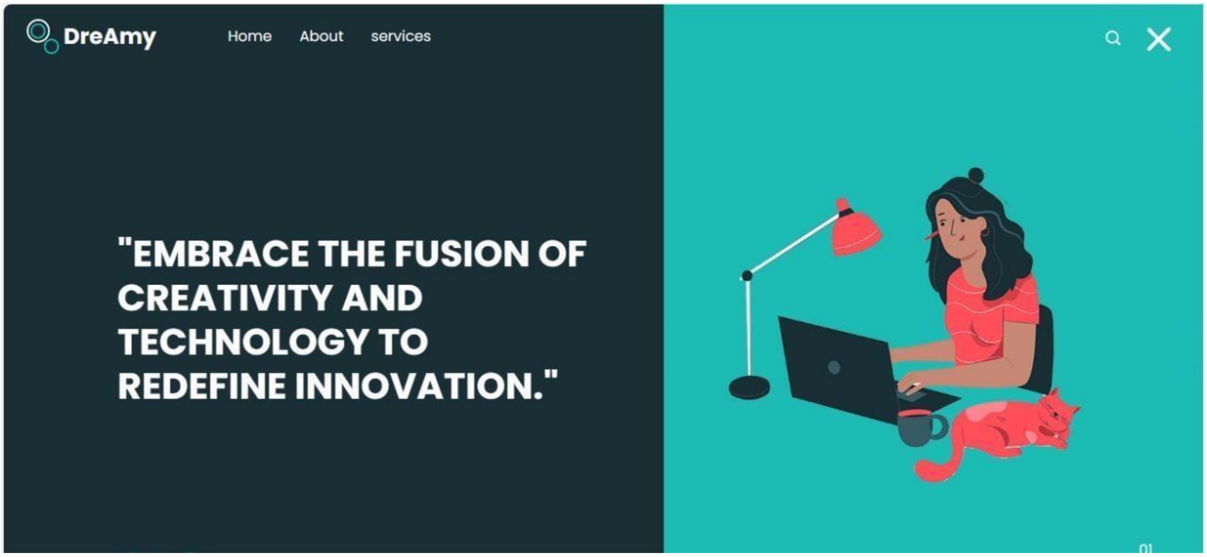
Algorithm: process video	
Input: video path	Output: processed frames
1. Open the video file using OpenCV.	1. Initialize an empty list to store frames.
2. Initialize an empty list to store frames.	2. Loop over each frame in the video.
3. Loop over each frame in the video.	3. Read the frame using OpenCV.
4. Read the frame using OpenCV.	4. Process the frame using process frame.
5. Process the frame using process frame.	5. Append the processed frame to frames list.
6. Append the processed frame to frames list.	6. Increment count.
7. Increment count.	7. Check if count reaches 120 frames.
8. Check if count reaches 120 frames.	8. Release the video capture.
9. Release the video capture.	9. Return the processed frames.

5.2 External APIs/SDKs

In my project, I worked alone and didn't use any external tools or services. I mainly used the OpenCV library and some other tools that came with the programming language I chose. This way, I had full control over my project and could customize it exactly how I wanted without depending on other companies or people.

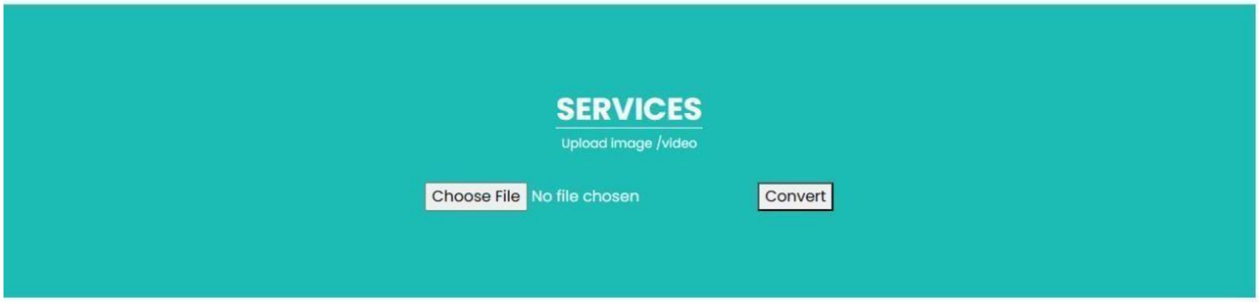
5.3 User Interface:

5.3.1 Home Screen:

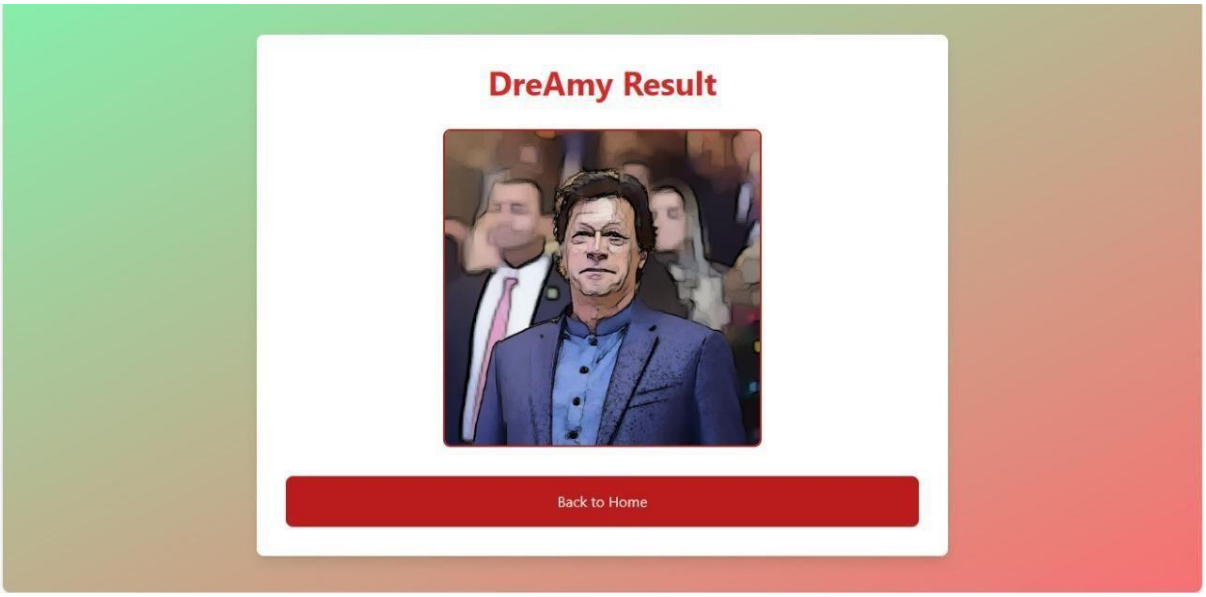


"EMBRACE THE FUSION OF CREATIVITY AND TECHNOLOGY TO REDEFINE INNOVATION."

5.3.2 Service screen:



5.3.3 Result Screen:



5.4 Deployment

1. **Hosting/Cloud Service**: AWS (Amazon Web Services)

AWS provides scalable cloud computing services, offering computing power, storage, and other functionalities to help businesses scale and grow.

2. **Set of Available Software and Their Versions**:

Programming Language: Python 3.8.10

Python Framework: flask 3.2.5

Computer Vision Library: OpenCV 4.5.3

3. **Deployment Environments**

Used for local development and testing.

Run on developer machines using tools like PyCharm, VSCode, or terminal.

Testing and Evaluation

5.5 Unit Testing

Table 19 unit testing

Test Case	**Test Scenario**	**Input**	**Expected Output**	**Result**
Convert Image to Cartoon	Verify image conversion to cartoon functionality.	Input image file	Cartoonized output image	Pass

Convert Video to Cartoon	Validate video conversion to cartoon functionality.	Input video file	Video with cartoonized frames	Pass
Error Handling - invalid Input File	Ensure proper error handling for invalid input files.	Invalid input file format	Error message indicating unsupported format	Pass
Performance Testing - Processing Time	Assess the performance of the conversion process.	Large image or video file	Processing time within acceptable limits	Pass

5.6 Functional Testing

Table 20 functional testing

Test Case	**Test Scenario**	**Steps**	**Expected Result**	**Result**	**Test Case**
User Interface - Image Conversion	Verify the functionality of image conversion feature.	Navigate to the image conversion section.	Image conversion interface is displayed.	Pass	User Interface Image Conversion
User Interface - Video Conversion	Verify the functionality of video conversion feature.	Navigate to the video conversion section.	Video conversion interface is displayed.	Pass	User Interface Video Conversion

Upload Image File	Test uploading an image file for conversion.	Click on the upload button.	File selection dialog opens.	Pass	Upload Image File
Upload Video File	Test uploading a video file for conversion.	Click on the upload button.	File selection dialog opens.	Pass	Upload Video File
Start Conversion	Test starting the conversion process.	Select a file.	Conversion process begins.	Pass	Start Conversion
Conversion Completion	Test completion of the conversion process.	Wait for the conversion to finish.	Conversion completes successfully.	Pass	Conversion Completion
Error Handling - Unsupported Format	Test error handling for unsupported file formats.	Upload a file with an unsupported format.	Error message indicating unsupported format.	Pass	Error Handling - Unsupported Format

5.7 Business Rules Testing

Decision table 1

Table 21 Access to video and image upload

Condition	**Rule 1: User Accessing Upload?**
Access Image/Video	Yes
Upload	Yes

Decision table 2

Table 22 Access to video and image download

Condition	**Rule 1: User Accessing Download?**
Access Image/Video	Yes
Download	Yes

Decision table 3

Table 23 Access to video and image conversion

Condition	**Rule 1: User Accessing Conversion?**
Access Image/Video	Yes
Conversion	Yes

5.8 Integration Testing

Testing Objective: To ensure the interaction between the modules for uploading and downloading media works correctly.

Table 24 interaction between the modules

No.	**Test case/Test script**	**Attribute and Value**	**Expected Result**	**Actual Result**	**Result**
1.	Upload Image	Image file	Image is uploaded successfully	Image uploaded successfully.	Pass

			and stored in the system.		
2.	Download Image	Image ID	Image is downloaded successfully from the system.	Image downloaded successfully.	Pass
3.	Upload Video	Video file	Video is uploaded successfully and stored in the system.	Video uploaded successfully.	Pass
4.	Download Video	Video ID	Video is downloaded successfully from the system.	Video downloaded successfully.	Pass

Testing objective: To ensure the conversion of images and videos functions properly and integrates seamlessly with the upload and download modules

Table 25 conversion of images and videos

No.	**Test case/Test script**	**Attribute and Value**	**Expected Result**	**Actual Result**	**Result**
1.	Convert Image to Cartoon	Image file	Image is successfully converted into a cartoon version.	Image converted successfully.	Pass

2.	Convert Video to Cartoon	Video file	Video is successfully converted into a cartoon version.	Video converted successfully.	Pass

6 Conclusion and Future Work

6.1 Conclusion

In conclusion, as the sole developer of this project, I have successfully created a user-friendly software application dedicated to transforming images and videos into captivating cartoons using OpenCV. Through the implementation of various computer vision techniques, the application offers seamless cartoonization effects. Key features such as batch processing, real-time previews, and direct download capabilities have been integrated to enhance user experience. Moving forward, I plan to introduce subscription plans for premium features and expand editing tools to provide users with greater customization options. Overall, this project reflects my dedication to delivering a comprehensive solution for cartoonizing visual content while setting the stage for future enhancements.

6.2 Future Work

Incorporating subscription-based premium features and expanding editing capabilities for images and videos are excellent ideas for future enhancements to your project

1. Subscription Plans:

- Users can pay for special features.

- Offer different plans with different benefits.

- They can pay securely online.

- Users can change or cancel their plans anytime.

2. **Better Editing Tools**:

- Allow users to change colors, brightness, etc.

- Add filters and effects to images and videos.

- Let user crop, resize, and rotate their media.

- Make sure it works with different types of files.

8. References

- https://www.analyticsvidhya.com/blog/2022/06/cartoonify-image-using-opencv-and-python/
- https://www.geeksforgeeks.org/cartooning-an-image-using-opencv-python/