A Project Report

On

CARTOONIZATION OF IMAGES AND VIDEOS USING GAN &OPEN CV

A project report Submitted in partial fulfilment of the Requirements for the awards of the degree of

BACHELOR OF TECHNOLOGY IN
COMPUTER SCIENCE AND ENGINEERING

S.TEJA	213B5A0509
D.S.N.S.SUNIL	203B1A0599
Y.ABHINESH	203B1A0592
P.SATYAVENI	203B1A0565
M.AMRUTHAVARSHA	203B1A0555

Under the Esteemed Guidance of

A.SRAVANI, M.Tech
Assistant professor
Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited by NAAC

(Approved by AICTE,New Delhi and Permanently Affiliated to JNTUK , Kakinada) Recognised under 2(f) & 12B of UGC, Certified by ISO 9001:2015
Sponsored by the Ramachandrapuram Education Society (ESTD,1965)

(2020-2024)

VSM COLLEGE OF ENGINEERING

(Affiliated to JNTU , Kakinada, Approved by AICTE)

Ramachandrapuram- 533255, Dr.B.R.Ambhedkar Konaseema Dt, Andhra Pradesh

(2020-2024)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "**CARTOONIZATION OF IMAGES AND VIDEOS USING GAN &OPEN CV**" is being submitted by **P.SATYAVENI** bearing Regd.No**: 203B1A0565,S.TEJA** bearing Regd.No: **213B5A0509, D.S.N.S.SUNIL** bearing Regd.No: **203B1A0599, Y.ABHINESH** bearing Regd.No: **203B1A0592,M.AMRUTHA VARSHA** bearing Regd.No: **203B1A0555**, in partial fulfillment of the requirement for the award of the degree of Bachelor Of Technology in **COMPUTER SCIENCE AND ENGINEERING to the JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA** is a record bonafied work carried out by them under the guidence and supervision.

The results embodied in the project report haven't been submitted to any other university or institute for the award of degree.

PROJECT GUIDE:

Ms.A.SRAVANI

Assistant Professor,
Department of CSE.

HEAD OF THE DEPARTMENT:

Mrs.K. Surekha, M.Tech

Associate Professor,
Department of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We would like to thank **Shri M.S.R PRASAD**, President and correspondent, V.S.M Group of Institution, **Shri M.S.N MURTHY, M.S(U.S.A)**, Vice President, V.S.M Group Of Institutions, & other officials of the organisation for giving us opportunity and facilities according to carry out this work.

We would like to express our sincere thankful to **Dr. K. BALAJI**, Principal (Academics) V.S.M College of Engineering, Ramachandrapuram, for their support during and till the completion of the project.

We are really thankful to Mrs.K.SUREKHA, M.Tech, Associate Professor and **Head of the Department**, Computer Science and Engineering for providing the laboratory facilities to fullest extent as and when required and also for giving me the opportunity to carry out the project work in the college.

We profoundly grateful to express my deep gratitude and respect towards our guide of **A.SRAVANI, Assistant Professor** for this his excellent guidance right from selection of the project and valuable suggestions throughout the project work. His constant encouragement and support has been the cause of our success, in completing this project in our college.

Last but not least, we wish to thank the Management, Parents, and Teaching and Non-Teaching staff of our Department for their constant support, cooperation and encouragement during the period of completion this project w

DECLARATION

We here by declare that this project entitled as **"CARTOONIZATION OF IMAGES AND VIDEOS USING GAN & OPEN CV"** done by as under the guidance of **A.SRAVANI** at VSM COLLEGE OF ENGINEERING, Ramachandrapuram, Permanently affiliated to JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, Kakinada in partial fulfilment of the award of the Degree of Bachelor of Technology.

No part of the project report is copied from books, journals or internet and wherever the portion is taken, the same has been duly referred in the text. The reported results are based on the project work entirely done by us and not copied from any other source.

Also, we declare that the matter embedded in this report has not been submitted by others in fulfilment of the requirement for the award of Degree.

S.TEJA	**213B5A0509**
D.S.N.S.SUNIL	**203B1A0599**
Y.ABHINESH	**203B1A0592**
P.SATYAVENI	**203B1A0565**
M.AMRUTHAVARSHA	**203B1A0555**

PLACE:
DATE:

TABLE OF CONTENTS DESCRIPTION

Abstract

Cartoonization of images and videos could be used in various different applications, which can beease in publishing a comic book for a comic, anime, T.V. shows as wellas for fun events on social media. This proposes cartoonization of images and videos through Generative Adversarial Networks (GANs). Thus an idea to convert real world images and videos into cartoonized one is proposed through this paper. With carbonization, the paper also proposes to make a complete Image-hub for the user with features including up scaling, denoising and editing filters to the input images through the Python OpenCV library. The project also includes video to GIF conversiontouse in various social mediaplatforms to achieve cartoon filters. Thus theproject is built to be userfriendly and leveraging various other features rather than only cartoonization

Cartoonization of images and videos could be used in various different applications, which can beease in publishing a comic book for a comic, anime, T.V. shows as wellas for fun events on social media. This proposes cartoonization of images and videos through Generative Adversarial Networks (GANs). Thus an idea to convert real world images and videos into cartoonized one is proposed through this paper. With carbonization, the paper also proposes to make a complete Image-hub for the user with features including up scaling, denoising and editing filters to the input images through the Python OpenCV library. The project also includes video to GIF conversiontouse in various social mediaplatforms to achieve cartoon filters. Thus theproject is built to be userfriendly and leveraging various other features rather than only cartoonization

Abstract

With rapid urbanisation and the advancement of cities and towns, the graph of crime rates is increasing gradually. Blockchain can replace those piled up criminal records with a network where documents are easily accessible and could not be tampered with, making them safe and Secure. Blockchain is a P2P (peer-to-peer network) that helps in the decentralisation of data. This system will be based upon the immutability characteristic of blockchain to ensure the integrity and security of data. This blockchain based process can reduce corruption risk factors by making it easier for third parties to monitor tamper-evident transactions and enabling greater objectivity and consistency, thus enhancing criminal record transparency and accountability. Furthermore, timely access of authentic criminal records to respective administrative authorities will make law enforcement effective.

LIST OF FIGURES

INTRODUCTION

1. Introduction

Cartoon is an image or series of images that are formed using a sequence of illustrations for animations. These cartoons may represent realistic or non-realistic features. However cartoons have gained a huge attention especially by the children, teenagers and artists. Due to which there exists many applications where cartoons are used. Some of these applications include cartoon television shows, comic magazines, cartoon based image filters and animated films. Some of the applications may also contain some real-world scenes. For example, an animated film may contain an image having a city drawn which corresponds to a city that is present in a real world.

These cartoon images are created by a skillful artist manually drawing those scenes or by using computer software's to create a single image. To obtain a better quality, the artists need to draw lines and must shade each color region based on real-life scenes. This entire process requires a lot of labor skills and is really time consuming especially while working on animated comics or films. Also the existing computer software's like Corel Draw or Adobe Photoshop are not free to use and also may not be easy for the beginners to understand and achieve the required quality. So there is a requirement of technology that can help transform a real-world based photo or video into an animated image or video respectively.

This technology when integrated with other software's can help the user to convert their real-world photos or videos into cartoon versions as and when required or can also act as an image filter which is also freely available and easy to use. In this paper, we propose a Generative Adversarial Networks (GANs) based approach along with features like image denoising and image upscaling to convert an image, GIF or video files into their cartoon versions. The image upscaling and denoising is achieved using OpenCV. To train the model, data used are a set of photos and a set of cartoon images. The trained model helps in generating the cartoon images or videos that are not a part of training data

1.1 EXISTING SYSTEM:

This entire process requires a lot of labor skills and is really time consuming especially while working on animated comics or films. Also the existing computer software's like Corel Draw or Adobe Photoshop are not free to use and also may not be easy for the beginners to understand and achieve the required quality. So there is a requirement of technology that can help transform a real-world based photo or video into an animated image or video respectively. This technology when integrated with other software's can help the user to convert their real-world photos or videos into cartoon versions as and when required or can also act as an image filter which is also freely available and easy to use.

Labor-intensive Process:

- Creating animated comics or films demands significant labor skills.
- The process is time-consuming due to intricate details involved.

Existing Software Limitations:

- Popular software like Corel Draw or Adobe Photoshop are not free.
- These software may be complex for beginners, hindering quality production.

Need for Transformation Technology:

- Demand for technology to convert real-world photos/videos into animated versions.
- Integration with existing software can streamline the conversion process.

Advantages of Integrated Technology:

- Enables easy conversion of real-world content into cartoon format.
- Acts as a freely available and user-friendly image filter.
- Offers accessibility and simplicity for users of all skill levels.

Disadvantages:

Limited Robustness:

Existing systems may struggle with robustness in the face of various environmental factors , such as changes in lighting conditions, different camera angles, and varying image resolutions.

Complexity of Attacks:

Sophisticated morphing attacks, especially those created with professional tools and manual intervention, can pose a significant challenge to existing systems. Systems that are not equipped to handle such complexity may exhibit reduced accuracy.

Generalization Issues:

Some systems may face difficulties in generalizing well to diverse datasets. If the training data is not representative of real-world scenarios, the system's performance might degrade when faced with new and unseen morphing techniques.

Computationally Intensive:

Deep learning-based approaches, while powerful, can be computationally intensive, requiring substantial resources for training and inference. This might limit their practicality in real-time or resource-constrained environments.

Limited Dataset Variability:

The effectiveness of a system heavily depends on the diversity and size of the training dataset. If the dataset used for training is limited in terms of variability in facial expressions, ages, and ethnicities, the system may struggle to generalize to a broader population.

Ethnic and Gender Bias:

If the training data is biased towards certain ethnicities or genders, the system may exhibit biased behaviour, performing better on certain groups while underperforming on others.

Adversarial Attacks:

Adversarial attacks, where attackers deliberately manipulate input images to deceive the system, pose a challenge. Some existing systems may not be resilient to such attacks.

Real-time Processing Challenges:

Achieving real-time processing capabilities for face morphing detection can be challenging, particularly for systems that involve computationally intensive algorithms.

Privacy Concerns:

Implementing effective face morphing detection without compromising user privacy can be a delicate balance. Striking a balance between accuracy and privacy is a key consideration.

Continuous Evolution of Attack Techniques:

As attackers continually develop new techniques, existing systems may become outdated if they are not regularly updated to adapt to emerging morphing attack strategies.

1.2 PROPOSED SYSTEM :

This paper proposes cartoonization of images and videos through Generative Adversarial Networks (GANs). Thus an idea to convert real world images and videos into cartoonized one is proposed through this paper. With carbonization, the paper also proposes to make a complete Image-hub for the user with features including upscaling, denoising and editing filters to the input images through the Python OpenCV library. The project also includes video to GIF conversion to use in various social media platforms to achieve cartoon filters.

we propose a Generative Adversarial Networks (GANs) based approach along with features like image denoising and image upscaling to convert an image, GIF or video files into their cartoon versions. The image upscaling and denoising is achieved using OpenCV. To train the model, data used are a set of photos and a set of cartoon images. The trained model helps in generating the cartoon images or videos that are not a part of training data.

Advantages:

Enhanced Accuracy and Robustness:

The proposed system incorporates advanced deep learning techniques, including a sophisticated feature extractor, to significantly improve the accuracy of face morphing attack detection. The system's robustness is enhanced, allowing it to perform effectively in diverse and challenging realworld scenarios.

Resilience to Professional Morphing Tools:

By introducing a novel image enhancement strategy and feature combination approach, the system demonstrates increased resilience to morphing attacks generated by professional-grade tools with manual intervention. This capability is crucial for addressing the evolving sophistication of morphing techniques.

Versatile Dataset Inclusion:

The system utilizes a comprehensive dataset that includes both Morph-2 and Morph-3 images, providing a more nuanced and realistic evaluation of its detection capabilities. The inclusion of Morph-3 images, which are often more challenging to detect due to their realistic appearance, contributes to the system's versatility and effectiveness.

Adaptability to Varied Datasets:

Through rigorous experimentation on eight diverse face databases, including Celebrity2000, Extended Yale, FEI, FGNET, GT-DB, MULTI-PIE, FERET, and FRLL, the proposed system showcases its adaptability to different datasets.

This adaptability is crucial for ensuring the system's generalisation across a wide range of facial characteristics.

Promising Results Across Multiple Experimental Setups:

The proposed methodology is validated through multiple experimental setups, demonstrating consistently promising results. The system's performance is evaluated using various metrics, including accuracy, precision, recall, and F1-score, confirming its efficacy in detecting face morphing attacks and outperforming existing systems in the literature.

LITERATURE SURVEY

2 .LITERATURE SURVEY

2.1 TITLE: "Transforming photos to comics using convolutional neural networks"

ABSTRACT: In this paper, inspired by Gatys's recent work, we propose a novel approach that transforms photos to comics using deep convolutional neural networks (CNNs). While Gatys's method that uses a pre-trained VGG network generally works well for transferring artistic styles such as painting from a style image to a content image, for more minimalist styles such as comics, the method often fails to produce satisfactory results. To address this, we further introduce a dedicated comic style CNN, which is trained for classifying comic images and photos. This new network is effective in capturing various comic styles and thus helps to produce better comic stylization results. Even with a grayscale style image, Gatys's method can still produce colored output, which is not desirable for comics. We develop a modified optimization framework such that a grayscale image is guaranteed to be synthesized. To avoid converging to poor local minima, we further initialize the output image using grayscale version of the content image. Various examples show that our method synthesizes better comic images than the state-of-the-art method.

2.2 TITLE: GAN: Generative Adversarial Network for photo Cartoonization .

ABSTRACT: In this paper, we propose a solution to transforming photos of real-world scenes into cartoon style images, which is valuable and challenging in computer vision and computer graphics. Our solution belongs to learning based methods, which have recently become popular to stylize images in artistic forms such as painting. However, existing methods do not produce satisfactory results for cartoonization, due to the fact that (1) cartoon styles have unique characteristics with high level simplification and abstraction, and (2) cartoon images tend to have clear edges, smooth color shading and relatively simple textures, which exhibit significant challenges for texture-descriptor-based loss functions used in existing methods. In this paper, we propose CartoonGAN, a generative adversarial network (GAN) framework for cartoon stylization. Our method takes unpaired photos and cartoon images for training, which is easy to use.

Two novel losses suitable for cartoonization are proposed: (1) a semantic content loss, which is formulated as a sparse regularization in the high-level feature maps of the VGG network to cope with substantial style variation between photos and cartoons, and (2) an edge-promoting adversarial loss for preserving clear edges.

We further introduce an initialization phase, to improve the convergence of the network to the target manifold. Our method is also much more efficient to train than existing methods. Experimental results show that our method is able to generate high-quality cartoon images from real-world photos (i.e., following specific artists' styles and with clear edges and smooth shading) and outperforms state-of-the-art methods.

2.3 TITLE: "Generative adversarial nets,"

ABSTRACT: We propose a new framework for estimating generative models via adversarial nets in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitatively evaluation of the generated samples

2.4 TITLE: Transformation of Realistic Images and Videos into Cartoon Images.

ABSTRACT: Aim of the project is to put forward a solution for transforming snapshots or videos of real-world into animated photos (Cartoon Images) or Video. The earlier method of transformation requires complicated computer graphics and skills. The idea of the paper is based on designated snapshots and videos which are converted to an art form such as painting. Amongst all the techniques usable, the application of a Generative Adversarial Network (GAN) called Cartoon GAN will be used for the styling real-world images that use 2 loss functions namely, content loss and adversarial loss for getting a sharp and clear image. With the help of GAN, it is possible to convert video as well to its cartoonized version and the development of the project shows that our Proposed Idea provides high quality cartooned images and videos.

2.5 TITLE : Enhanced Deep Residual Networks for Single Image Super-Resolution.

ABSTRACT: Recent research on super-resolution has progressed with the development of deep convolutional neural networks (DCNN). In particular, residual learning techniques exhibit improved performance.

In this paper, we develop an enhanced deep super-resolution network (EDSR) with performance exceeding those of current state-of-the-art SR methods. The significant performance improvement of our model is due to optimization by removing unnecessary modules in conventional residual networks. The performance is further improved by expanding the model size while we stabilize the training procedure. We also propose a new multi-scale deep super-resolution system (MDSR) and training method, which can reconstruct high-resolution images of different upscaling factors in a single model. The proposed methods show superior performance over the state- of-the-art methods on benchmark datasets and prove its excellence by winning the NTIRE2017 Super-Resolution Challenge.

2.6 TITLE : Accelerating the Super-Resolution Convolutional Neural Network.

ABSTRACT: As a successful deep model applied in image super-resolution (SR), the Super-Resolution Convolutional Neural Network (SRCNN) has demonstrated superior performance to the previous hand-crafted models either in speed and restoration quality. However, the high computational cost still hinders it from practical usage that demands real-time performance (24 fps). In this paper, we aim at accelerating the current SRCNN, and propose a compact hourglass-shape CNN structure for faster and better SR. We re-design the SRCNN structure mainly in three aspects. First, we introduce a deconvolution layer at the end of the network, then the mapping is learned directly from the original low-resolution image (without interpolation) to the high-resolution one. Second, we reformulate the mapping layer by shrinking the input feature dimension before mapping and expanding back afterwards. Third, we adopt smaller filter sizes but more mapping layers. The proposed model achieves a speed up of more than 40 times with even superior restoration quality. Further, we present the parameter settings that can achieve real-time performance on a generic CPU while still maintaining good performance. A corresponding transfer strategy is also proposed for fast training and testing across different upscaling

SYSTEM REQUIREMENTS ANALYSIS AND SPECIFICATIONS

3.SYSTEM REQUIREMENTS ANALYSIS AND SPECIFICATION

3.1 SCOPE OF THE PROJECT

The scope of cartoonization, especially with advancements in technology like GANs and computer vision techniques, is vast and encompasses various domains

Entertainment and Media: Cartoonization is widely used in the entertainment industry for creating animated films, television shows, and web series. It offers a unique visual style that appeals to audiences of all ages. With advancements in technology, cartoonization can be automated or semi-automated, reducing production costs and time.
Digital Art and Design: Cartoonization techniques provide artists and designers with tools to create stylized artwork and illustrations. Whether it's for comics, graphic novels, character design, or concept art, cartoonization opens up new avenues for creative expression and storytelling.
Advertising and Marketing: Cartoonized images and characters are often used in advertising campaigns and marketing materials to grab attention, evoke emotions, and convey messages in a visually engaging manner. They can help brands establish a distinct identity and appeal to their target audience.

3.2 HARDWARE REQUIREMENTS :

Required for Execution	MY SYSTEM (Development)
System	i5 Processor 11th Gen
Hard Disk	500 Gb
Ram	4 Gb

3.3 SOFTWARE REQUIREMENTS :

Operating System	Windows 10/11
Development Software	Python 3.10
Programming Language	Python
Integrated Development Environment (IDE)	Visual Studio Code
Front End Technologies	HTML5, CSS3, Java Script
Back End Technologies or Framework	Django
Database Language	SQL
Database (RDBMS)	MySQL
Database Software	WAMP or XAMPP Server
Web Server or Deployment Server	Django Application Development Server
Design/Modelling	Rational Rose

3.4 FUNCTIONAL REQUIREMENTS

Graphical User interface with the user

3.5 NON-FUNCTIONAL REQUIREMENTS:

- **Performance:**

Performance requirement define how well or how efficiently the system must perform specific functions.

- **Usability:**

The application must have simple and user-friendly interface. So, user can save time.

- **Reliability:**

Reliability is the probability that the system will be able to process all work completely without being aborted.

- **Maintainability:**

The document should be easy for users who execute the system day to day, for the developers who wish to edit or develop further, and for the personnel who is in charge of the maintenance.

- **Availability:**

The feature defines the amount of time the system is running, the time it takes to repair a fault, and the time between lapses.

- **Portability:**

Portability is the ease with which a software system can be transferred from its current hardware or software environment to another environment.

SYSTEM DESIGN

4.SYSTEM DESIGN

INTRODUCTION :

After the Analysis, it must be clear what to be done. In system design, we plan to do it. The objectives of the system design are to decide in general how the system should be implemented. Individual physical analysis we prepare a blue print for the system. Analysis what the system should do and design specifies how to do and design specifies how accomplish the objectives.

System design begins with search for alternative solution.

4.1 SYSTEM ARCHITECTURE:

A system architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organised in a way that supports reasoning about the structure and behaviour of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. Object oriented design is concerned with developing an object-oriented model of a software system to implement the identified requirements. It is the process of defining components, interfaces, objects, classes, attributes and operations that will satisfy the requirements.

GAN Architecture

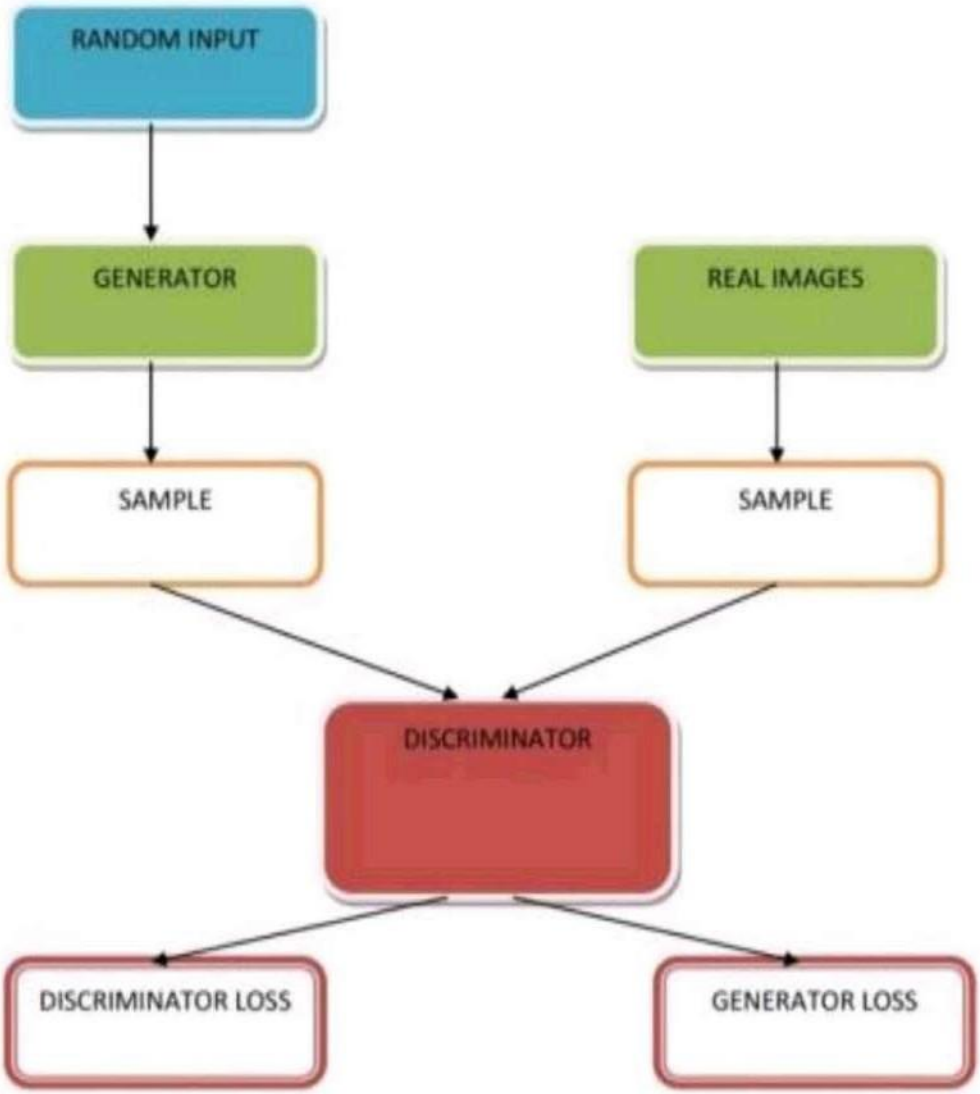
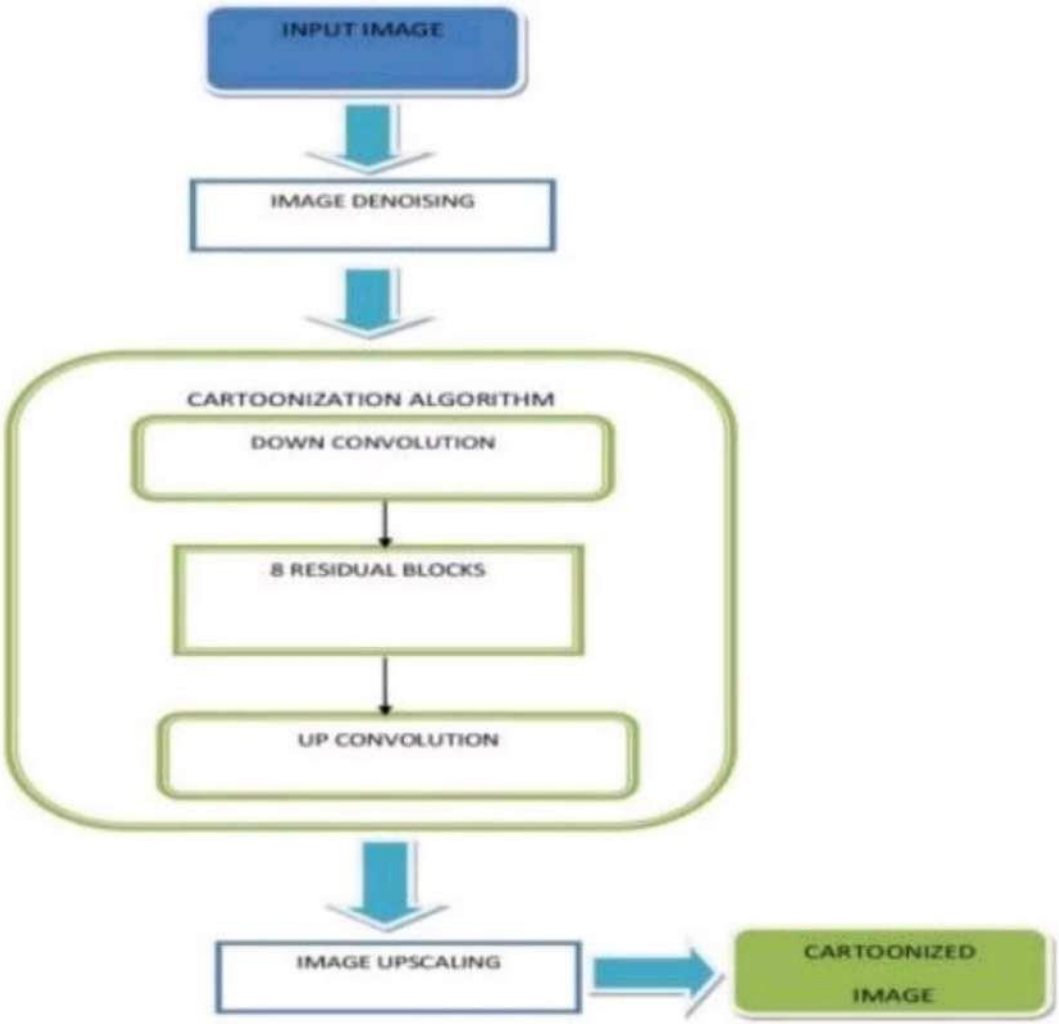


Fig. 1: Block Representation of GAN Architecture

FIG 1:

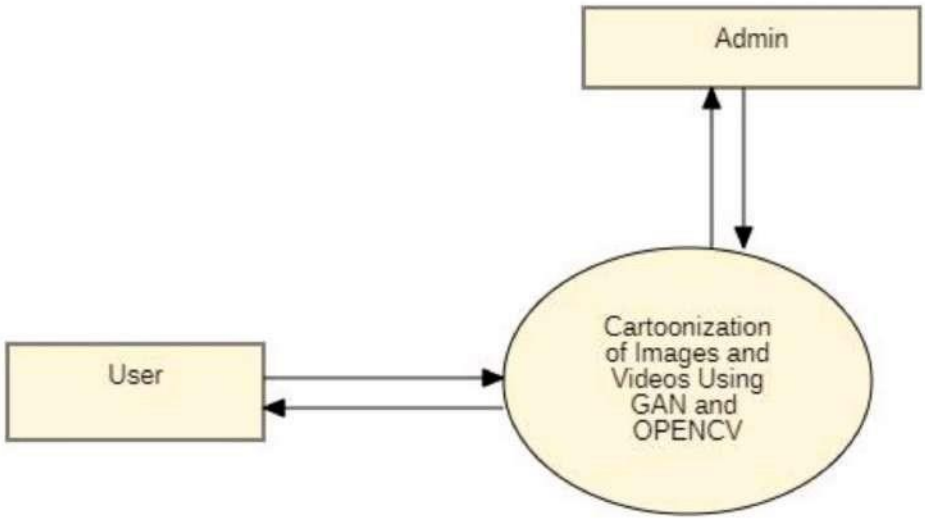
Cartoonization of Images using GAN



DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



4.2 UML DIAGRAM:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

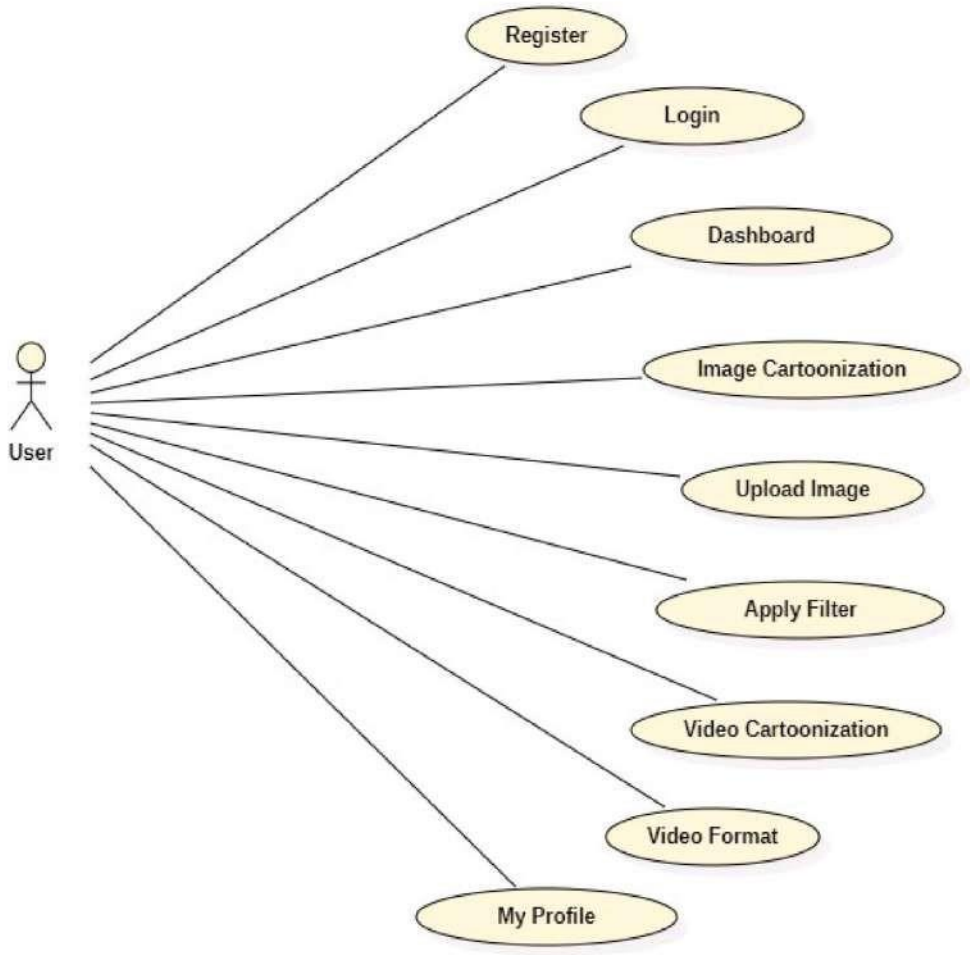
The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

BASIC BUILDING BLOCKS OF UML

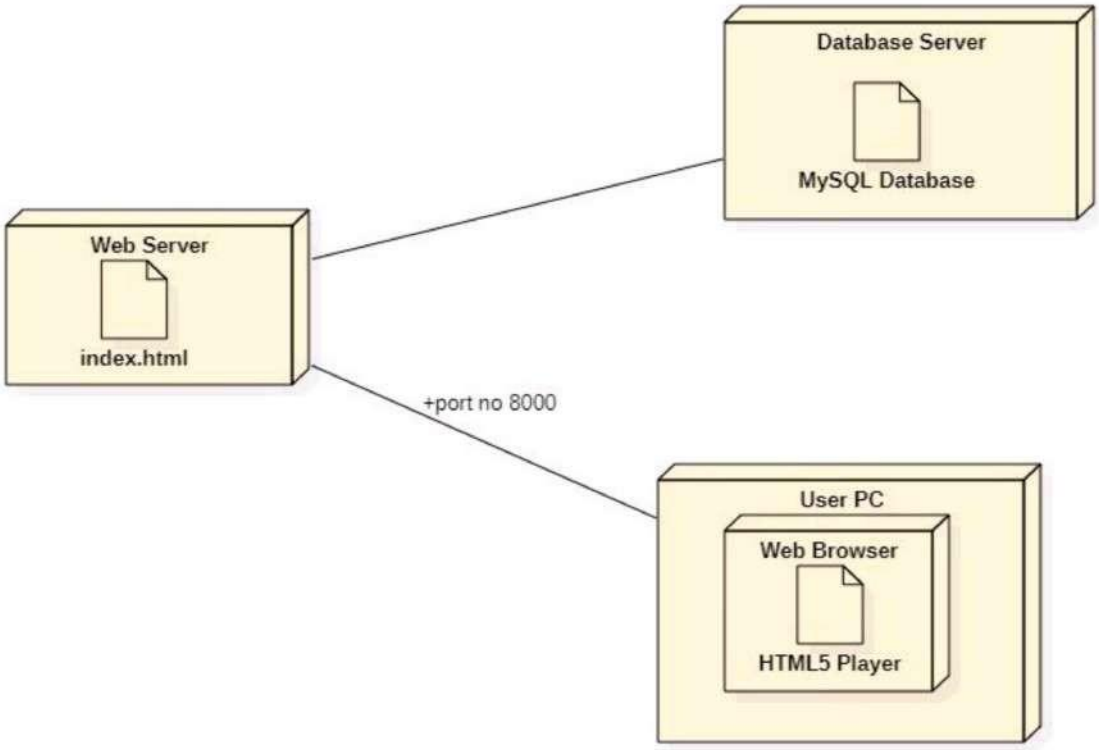
As UML describes the real-time systems, it is very important to make a conceptual model and then proceed gradually. The conceptual model of UML can be mastered by learning the following three major elements –

- UML building blocks
- Rules to connect the building blocks
- Common mechanisms of UML

Usecase(user) diagram



Deployment diagram



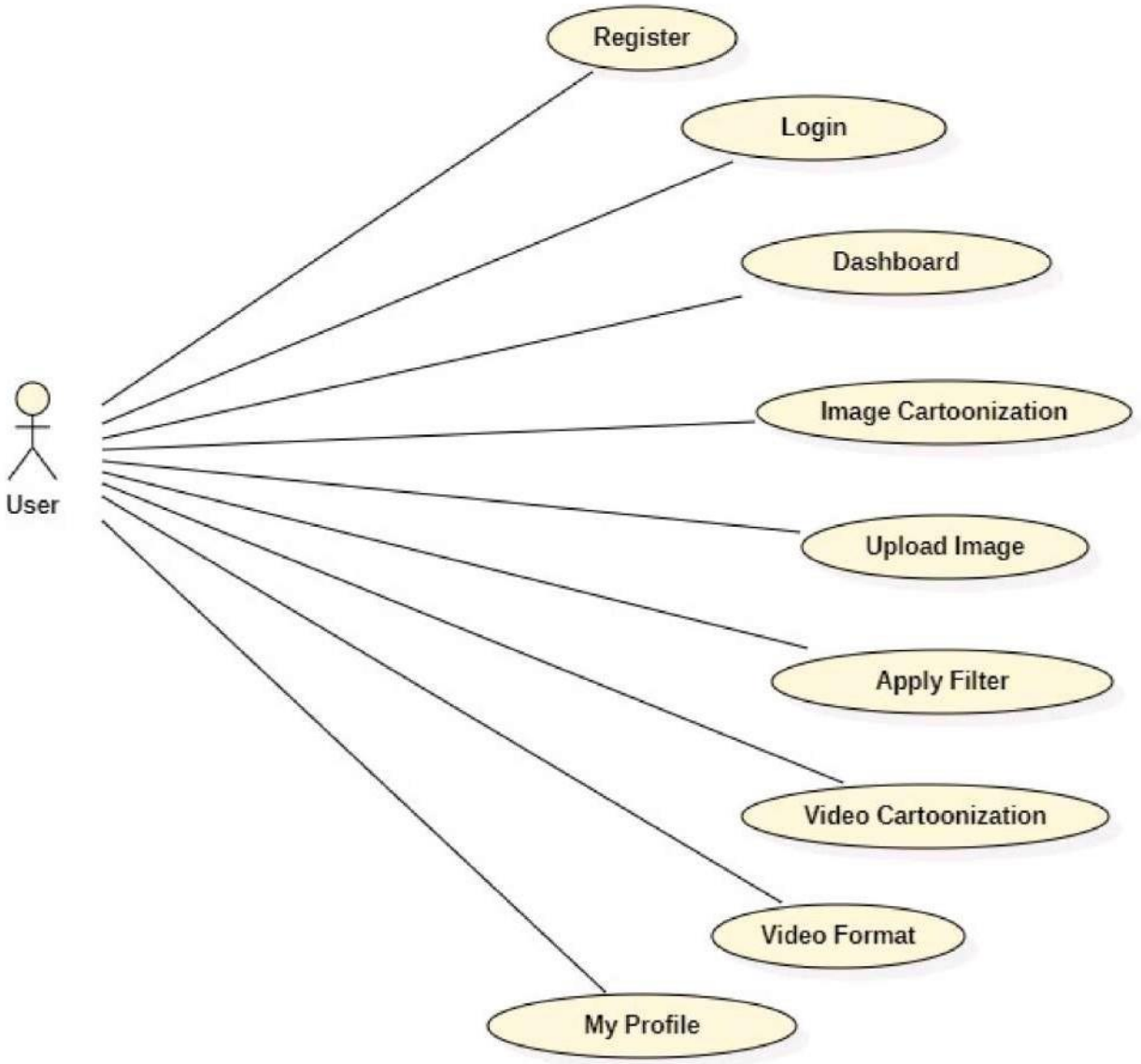
GOALS:

1. The Primary goals in the design of the UML are as follows
2. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
3. Provide extendibility and specialization mechanisms to extend the core concepts.
4. Be independent of particular programming languages and development process.
5. Provide a formal basis for understanding the modeling language.
6. Encourage the growth of OO tools market.
7. Support higher level development concepts such as collaborations, frameworks, patterns and components.
8. Integrate best practices.

4.2.1 USE-CASE DIAGRAM:

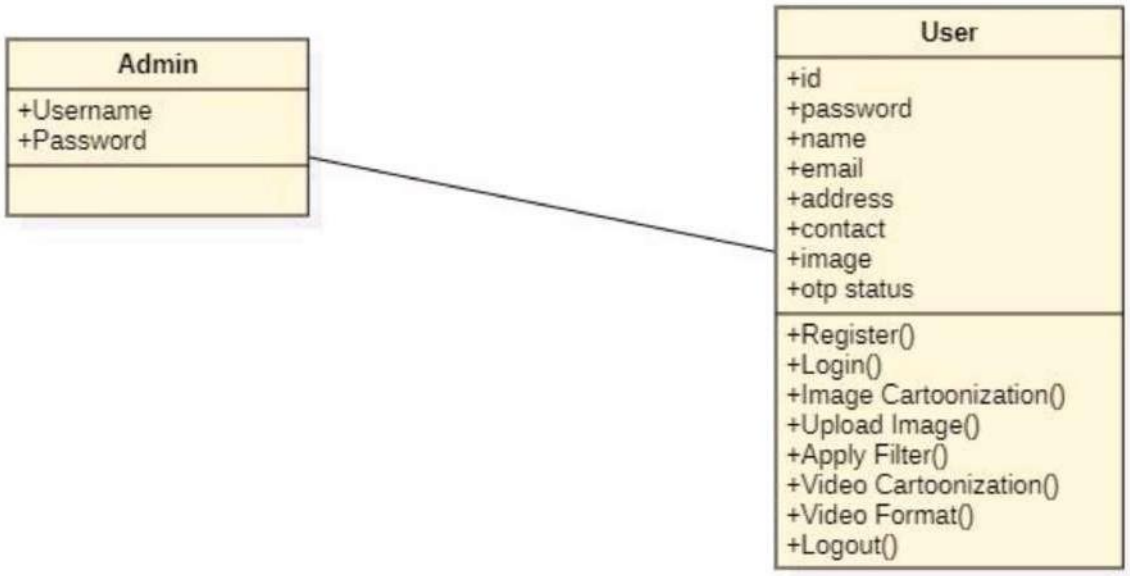
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

USER:



4.2.2 CLASS DIAGRAM :

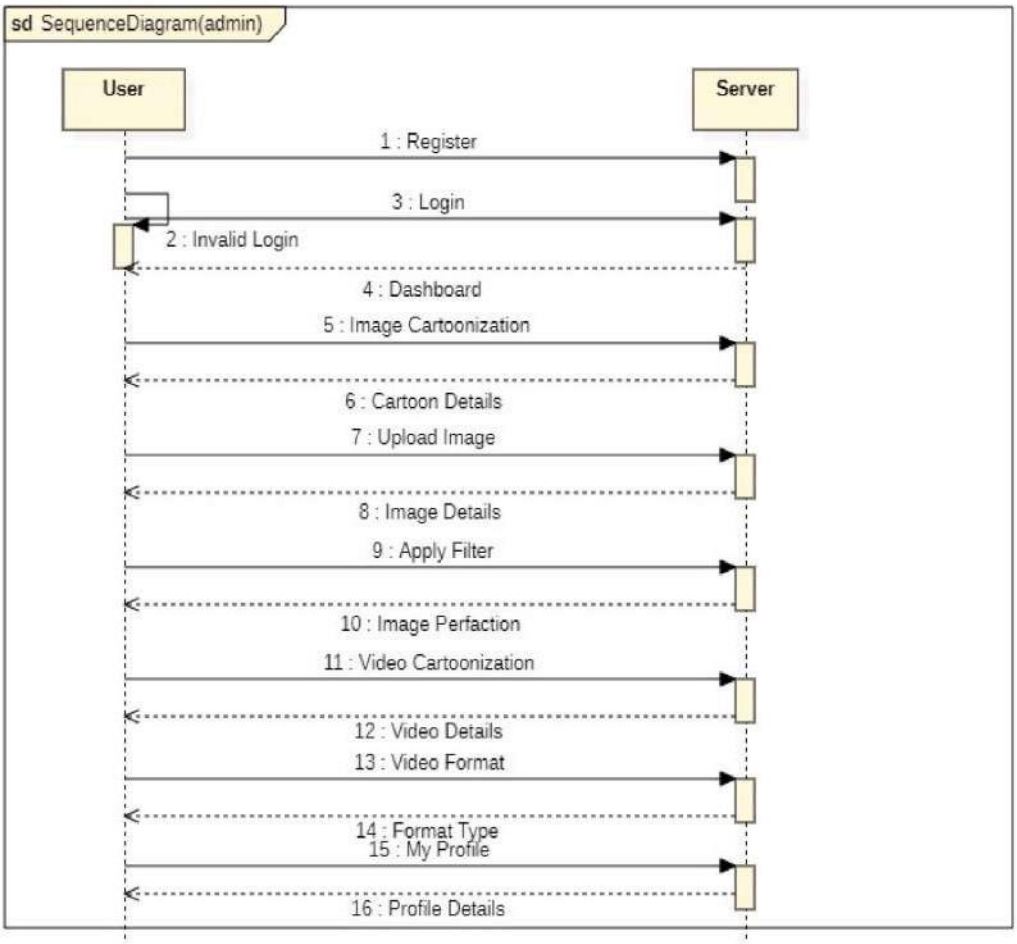
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



4.2.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

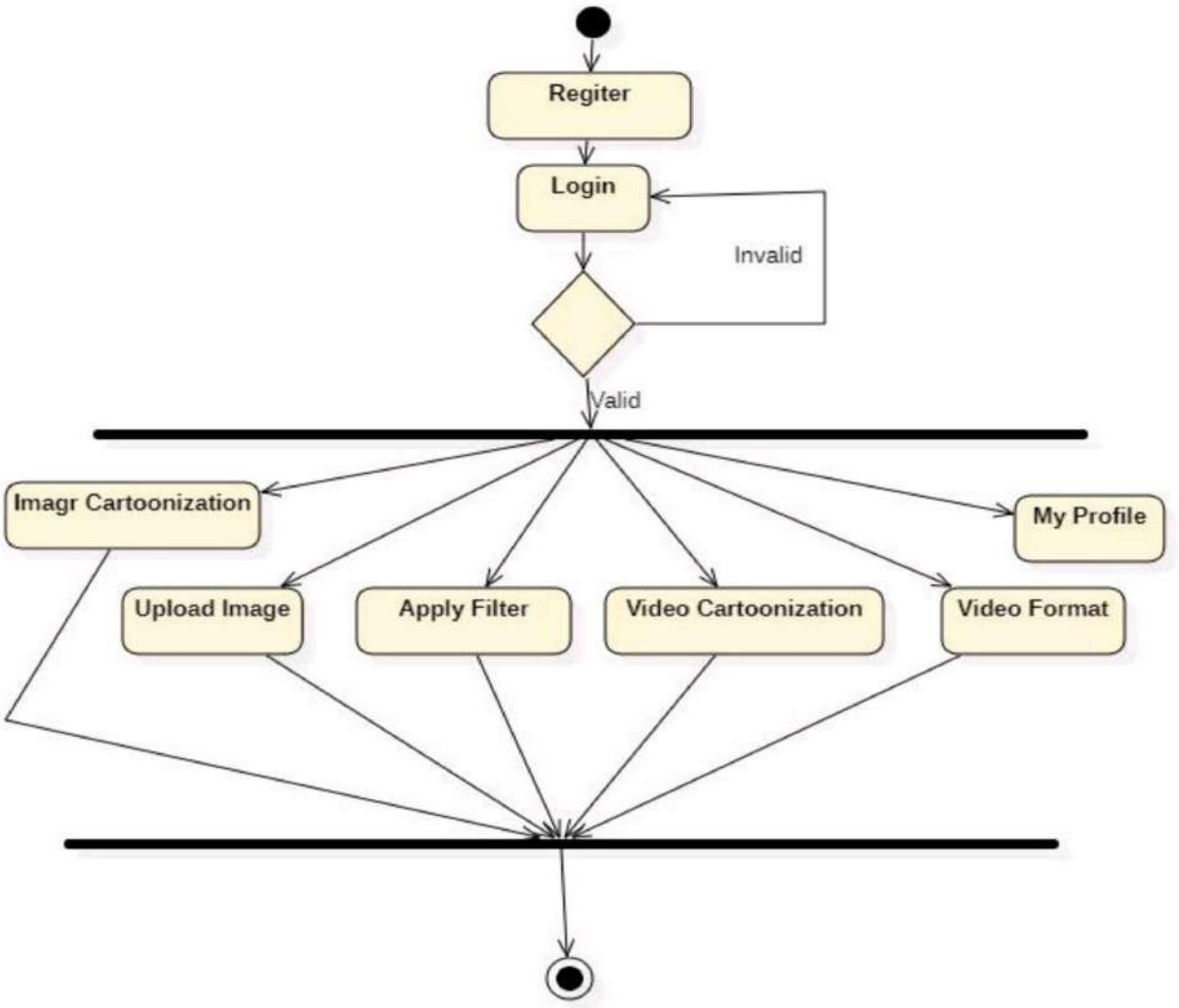
USER:



4.2.5 ACTIVITY DIAGRAM:

An activity diagram is a behavioural. It depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

USER:



IMPLEMENTATION

5.1 MODULES:

- IMAGE DENOISING
- IMAGE UPSCALING
- CARTOONIZATION OF IMAGE USING GAN
- CARTOONIZATION OF VIDEOS USING GAN

5.2 MODULES DESCRIPTION:

IMAGE DENOISING

One of the basic challenges within the field of image process and computer vision is removal of unnecessary things in an image file which is generally termed as image noise. Image noise could also be caused by totally different intrinsic (i.e., sensor) and accidental (i.e., environment) conditions. To remove those unnecessary things in an image, image denoising plays a vital role in a very wide selection of applications like image restoration, visual pursuit, image registration, image segmentation, and image classification. The underlying goal is to estimate the original image by suppressing noise from a noise-contaminated version of the Image. To obtain this image denoising feature, the OpenCV library contains a function fast NlMeans Denoising Colored which input converts image to CIELAB colorspace and then separately denoise L and AB components with given parameter regulating filter strength for luminance component.

IMAGE UPSCALING

When scaling a vector graphic image, the graphic primitives that conjure the image can be scaled victimization geometric transformations, with no loss of image quality. When scaling a formation graphics image, a different image with an improved or lower vary of pixels must be generated. at intervals, the case of decreasing the image part varies (scaling down) this usually finally ends up during a plain quality loss. The two techniques which we applied are EDSR (Enhance Deep Super-Residual Network) and FSRCNN (Fast Super Residual Convolutional Neural Network). EDSR takes up to 120 seconds to upscale the image whereas FSRCNN is a faster technique and gives an upscaled image output within 10 seconds. But comparison wise EDSR gives a higher upscaled image than FSRCNN

CARTOONIZATION OF IMAGE USING GAN

The Image will be first Denoised and then it will follow the cartoonization Algorithm. In Cartoonizing Images, the generator network is utilized to map input pictures to the animation complex. Cartoon stylization is created once the model is prepared. The generator starts with a flat level convolution stage followed by two down-convolution squares to spatially pack and encode the pictures. Valuable local signals are separated in this stage for downstream change. Afterward, eight remaining squares with indistinguishable formats are utilized to build the substance and complex element. At last, the output cartoon-style pictures are reproduced by two up-convolution blocks and it will be up scaled in order to increase the quality of image.

CARTOONIZATION OF VIDEOS USING GAN

Whenever we give an input video for cartoonization, the video gets converted into frames and each frame is Cartoonized individually. The process of cartoonization for frames is same as that of images. Since the video gets converted into frames, loss of sound is observed. In order to aid this problem, we have come up with a method so that there will be no audio loss. Here we apply the concept of multithreading. The input video will send its copy to the sound loss prevention and control unit so that the sound will be extracted from the video and an audio file will be created. At the same time the process of dividing video into frames will take place. Each frame will be Cartoonized. After the cartoonization of all the frames, the frames will be combined to form the Cartoonized video. At last both the outputs from Sound Loss Prevention Control Unit and Video Cartoonizing unit will go through Sound Merging Algorithm where the audio and video will be merged. The output of this whole process would result in a Cartoonized video which will have same audio as that of the input video.

5.3 TECHNOLOGIES USED :

- Python
- GAN
- OpenCV
- HTML
- CSS,
- JavaScript
- Django
- MySQL
- Tensorflow
- Scikit learn

PYTHON:

Python is a **high-level, interpreted**, **interactive** and **object-oriented scripting language**. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python Features:

Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- ➢ **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- ➢ **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- ➢ **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- ➢ **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- ➢ **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customise their tools to be more efficient.□

Databases: Python provides interfaces to all major commercial databases.□

- **GUI Programming:** Python supports GUI applications that can be created and ported to

  many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

HTML:

HTML (**H**yper **T**ext **M**arkup **L**anguage) is the code that is used to structure a web page and its content.

HTML Features:

User Friendly & Simple

You can write HTML using annotations called tags. Tags give HTML a structure and make it easy for humans and browsers to read the document efficiently. They also enable a browser to apply CSS to the digital document, making it a powerful visual combination.

Semantic Structure

HTML5 defines unique tags to annotate different elements for their specific purposes. The <article> tag, for example, is used to annotate content on a page. The <aside> tag represents some content that is indirectly related to the document's primary content. Other noteworthy elements are *<header>*,*<footer>*,* *, the paragraph tag *<p>*, and the one most used for navigation between pages, the *<a>*tag.

CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics.

JAVASCRIPT:

JavaScript is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded,

dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

DJANGO:

The Django web framework includes a default object-relational mapping layer (ORM) that can be used to interact with data from various relational databases such as SQLLite, PostgreSQL, and MySQL. Django allows us to add, delete, modify and query objects, using an API called ORM. ORM stands for Object Relational Mapping. An object-relational mapper provides an object- oriented layer between relational databases and object-oriented programming languages without having to write SQL queries.

MySQL:

MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL). A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organised into

physical files optimised for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment.

XAAMP SERVER:

XAAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It stands for "Cross-platform," "Apache" (the web server), "MySQL" (the database), "PHP" (the programming language), and "Perl" (another programming language). XAMPP is designed to be easy to install and use, providing developers with a complete environment for local web development and testing.

Here are some key features and components of XAMPP:

1. **Apache HTTP Server**: XAMPP includes the Apache web server, which is one of the most popular web servers in the world. Apache is highly configurable and supports a wide range of features and functionality.

2. **MySQL:** XAMPP comes with MySQL, a powerful open-source relational database management system. MySQL is commonly used for storing and managing data in web applications.

3. **PHP:** XAAMPP includes PHP, a widely-used server-side scripting language for web development. PHP is particularly well-suited for building dynamic web pages and web applications.

4. **Perl:** XAMPP also includes Perl, a high-level programming language often used for web development, system administration, and network programming.

5. **phpMyAdmin:** XAMPP includes phpMyAdmin, a web-based administration tool for managing MySQL databases. phpMyAdmin provides an intuitive interface for performing tasks such as database creation, table management, and SQL query execution.

6. **Other Tools and Components:** XAMPP may also include other tools and components such as FileZilla FTP server, OpenSSL for secure communication, and Mercury Mail for sending emails.

XAAMPP is popular among developers for creating a local development environment on their computers. It allows developers to work on web projects offline, test applications before
deploying them to a live server, and experiment with different configurations without affecting production systems. However, it's important to note that XAMPP is primarily intended for development and testing purposes and may not be suitable for production environments due to security considerations and performance limitations.

TENSORFLOW:

TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and resources for building and deploying machine learning models. TensorFlow was designed to be flexible, scalable, and efficient, making it suitable for a wide range of applications, from research prototyping to production deployment.

TensorFlow accepts data in the form of multi-dimensional arrays of higher dimensions called tensors. Multi-dimensional arrays are very handy in handling large amounts of data.TensorFlow works on the basis of data flow graphs that have nodes and edges. As the execution mechanism is in the form of graphs, it is much easier to execute TensorFlow code in a distributed manner across a cluster of computers while using GPUs.

Overall, TensorFlow provides a powerful and versatile framework for building, training, and deploying machine learning models across a wide range of applications and domains. Its rich set of features, scalability, performance optimizations, and integration capabilities make it a popular choice for machine learning practitioners and researchers worldwide.

SCIKIT LEARN:

Scikit-Learn, also known as sklearn is a python library to implement machine learning models and statistical modelling. Through scikit-learn, we can implement various machine learning models for regression, classification, clustering, and statistical tools for analyzing these models.

GAN(Generative Adversarial Network) :

A generative adversarial network (GAN) is a deep learning architecture. It trains two neural networks to compete against each other to generate more authentic new data from a given training dataset. For instance, you can generate new images from an existing image database or original music from a database of songs.

OpenCV :

OpenCV is a popular Computer Vision library to develop applications built using C++ and C. It has several uses like Object Detection and Video Processing. Computer Vision overlaps with fields like Image Processing, Photogrammetry, and Pattern Recognition. A popular wrapper Emgu CV is used to run OpenCV using C#

CODING

SAMPLE CODE

Users_views.py:

```
from django.shortcuts import render,redirect
from django.contrib.auth import authenticate,login,logout,update_session_auth_hash
from django.contrib.auth.decorators import login_required
from django.contrib.auth.forms import PasswordChangeForm
from django.core.exceptions import ValidationError
from django.contrib.auth.hashers import check_password
from django.core.mail import send_mail
import uuid
from django.db import IntegrityError
from django.contrib import messages
from django.conf import settings
from django.utils.datastructures import MultiValueDictKeyError
from userapp.models import *
import os
import argparse
from tools.utils import *
from tqdm import tqdm
from glob import glob
import time
import numpy as np
from net import generator
from cv2 import *
os.environ["CUDA_VISIBLE_DEVICES"]="0"
import cv2
import tensorflow as tf
def user_login(request):
    # user = request.user
    # if user.is_authenticated:
    #     return redirect("base")
    if request.method == 'POST':
        email = request.POST.get('email')
```

```
        password = request.POST.get('password')
        print(password,'pssword')
        user = authenticate(request=None,email=email, password=password)
        print(user,'user')
        if user is not None:
            login(request,user)
            messages.success(request,'Login Successfull')
            return redirect('base')
        else:
            print('invalid')
            messages.error(request,'Invalid Credentials')
            return redirect('user_login')
    return render(request,'main/main-userlogin.html')
def user_register(request):
    try:
        if request.method == 'POST':
            fullname=request.POST.get('fullname')
            email=request.POST.get('email')
            contact=request.POST.get('contact')
            city=request.POST.get('city')
            pic=request.FILES['pic']
            password=request.POST.get('password')
            print(fullname,email,contact,city,pic,password)
            user = MyUser.objects.create_user(email, password)
            user.fullname = fullname
            user.contact = contact
            user.city = city
            user.image = pic
            user.save()
            ftoken = str(uuid.uuid4())
            Profile.objects.create(user=user,forget_token=ftoken)
    messages.success(request,'Registration Successfull')
    except IntegrityError:
        messages.warning(request,'Email Has Already Been Taken')
```

```
 return redirect('user_register')
    return render(request,'main/main-user-register.html')
def forget_password(request):
    try:
        if request.method == 'POST':
            email = request.POST.get('email')
            user = MyUser.objects.get(email=email)
            profile = Profile.objects.get(user=user)
            user_email = user.email
            print(user_email)
            ftoken = profile.forget_token
            mail_message = f'Hey Your Reset Password Link is http://127.0.0.1:8000/reset-
password/{ftoken}/'
            print(mail_message)
            send_mail('Password Reset
Request',mail_message,settings.EMAIL_HOST_USER,[user_email],fail_silently=False)
            messages.info(request,'MAIL SEND')
    except:
        messages.warning(request,"Email Does not Exist")
    return render(request,'main/main-forgot-password.html')
def log_out(request):
    logout(request)
    messages.info(request,'Logout Successfull')
    return redirect('user_login')
def reset_password(request,id):
   if request.method == 'POST':
        password = request.POST['password']
        profile = Profile.objects.get(forget_token=id).user
        print(profile,'profile')
        user = MyUser.objects.get(email=profile)
        user.set_password(password)
        user.save()
        messages.info(request,'Password Changed Please Login! ')
        return redirect('user_login')
```

```
return render(request,'main/main-reset-password.html')
@login_required(login_url='user_login')
def base(request):
    user = MyUser.objects.get(email = request.user)
    return render(request,'user/user-base.html')
@login_required(login_url='user_login')
def myprofile(request):
    user = MyUser.objects.get(email = request.user)
    if request.method == 'POST':
        fullname = request.POST.get("fullname")
        contact = request.POST.get("contact")
        city = request.POST.get("city")
        print(city)
        state = request.POST.get("state")
        print(fullname,'hiii')
        if len(request.FILES) != 0:
                image = request.FILES["pic"]
                user.fullname = fullname
                user.contact = contact
                user.city = city
                user.image = image
                user.save()
                messages.success(request,'Updated Successfully')
    else:
                user.fullname = fullname
                user.contact = contact
                user.city = city
                user.save()
                messages.success(request,'Updated Successfully')
        return redirect('myprofile')
    context = {'user':user}
    return render(request,'user/user-myprofile.html',context)

@login_required(login_url='user_login')
```

```
def change_password(request):
    user = MyUser.objects.get(email = request.user)
    user_pass = user.password
    if request.method == "POST":
        old_password = request.POST.get("old_password")
        new_password1 = request.POST.get("new_password1")
        new_password2 = request.POST.get("new_password2")
        if check_password(old_password,user.password):
            if new_password1 == new_password2:
                user.set_password(new_password2)
                us = user.save()
                update_session_auth_hash(request,user)
                print('password changed')
                messages.success(request,'Password Updated Successfully')
                return redirect('myprofile')
            else:
                messages.warning(request,'New password and Confirm New password Should be
same')
                print('New password and Confirm New password Should be same')
        else:
            messages.warning(request,'Enter your correct old password')
            print('Enter your correct old password')
 return render(request,'user/user-change-password.html')
@login_required(login_url='user_login')
def image_convert(request):
    user = MyUser.objects.get(email = request.user)
    sav_path = None

    if request.method == 'POST':
        image_name = request.FILES['image']
        fil = request.POST.get('filter')
        auto = request.POST.get('autobrightness')
        # print(image_name    ,checkpoint_dir,adjust_brightness)
        image = Image_c.objects.create(image=image_name)
```

```
    i_id = image.id
        return redirect('image_result',i_id,fil,auto)
    return render(request,'user/user-image-convert.html')
@login_required(login_url='user_login')
def image_result(request,id,fil,auto):
    user = MyUser.objects.get(email = request.user)
    img = Image_c.objects.get(pk=id)
    image_name = str(img.image)
    image = str(img.image)
    test_files = './media/'+ image
    # adjust_brightness = True
    adjust_brightness = bool(auto)
    # checkpoint_dir = 'filter/generator_Hayao_weight'
    checkpoint_dir = 'filter/'+str(fil)
    # checkpoint_dir = 'filter/generator_Paprika_weight'
    img_size=[256,256]
    print(test_files,'test_files')
    test_real = tf.placeholder(tf.float32, [1, None, None, 3], name='test')
    with tf.variable_scope("generator", reuse=tf.AUTO_REUSE):
  test_generated = generator.G_net(test_real).fake
    saver = tf.train.Saver()

    gpu_options = tf.GPUOptions(allow_growth=True)
        with tf.Session(config=tf.ConfigProto(allow_soft_placement=True,
    gpu_options=gpu_options)) as sess:
        # tf.global_variables_initializer().run()
        # load model
        ckpt = tf.train.get_checkpoint_state(checkpoint_dir)  # checkpoint file information
        if ckpt and ckpt.model_checkpoint_path:
            ckpt_name = os.path.basename(ckpt.model_checkpoint_path)  # first line
            saver.restore(sess, os.path.join(checkpoint_dir, ckpt_name))
            print(" [*] Success to read {}".format(os.path.join(checkpoint_dir, ckpt_name)))
        else:
            print(" [*] Failed to find a filter")
```

```
  begin = time.time()
        sample_file = test_files
        sample_image = np.asarray(load_test_data(sample_file, img_size))
        # image_path = str(image_name)
        print(image_name,'image_name')
        # image_path = Image_r.objects.create(result = image_name)
        # image_path = image_path.result
        fake_img = sess.run(test_generated, feed_dict = {test_real : sample_image})
        name = 'output/'+str(image_name)
        print(name,'name')
        if adjust_brightness:
            print('autobright')
            save_img = inverse_transform(fake_img.squeeze())
            print(save_img,'sav')
            def imsa(images, path):
                return cv2.imwrite(path,cv2.cvtColor(images, cv2.COLOR_BGR2RGB))
            sav = imsa(adjust_brightness_from_src_to_dst(save_img, read_img(sample_file)),
name)
            print(sav,'sav')
print(save_img,'sav')
  # sav_path = cv2.imwrite(name,cv2.cvtColor(save_img, cv2.COLOR_BGR2RGB))
            # image_path = Image_r.objects.create(result = save_img)
            # save_images(fake_img, image_path, sample_file)
        else:
            print('no-autobright')
            save_img = inverse_transform(fake_img.squeeze())
            print(save_img,'sav')
            sav_path = cv2.imwrite(name,cv2.cvtColor(save_img, cv2.COLOR_BGR2RGB))
            print(sav_path,'saveeeeeeeeee')
            # image_path = Image_r.objects.create(result = sav)
            # save_images(fake_img, image_path, None)
        end = time.time()
        print(f'test-time: {end-begin} s')
    context = {'image_r':name, 'img':img}
```

```
 return render(request,'user/user-image-result.html',context)
def process_image(img, x32=True):
    h, w = img.shape[:2]
    if x32: # resize image to multiple of 32s
        def to_32s(x):
            return 256 if x < 256 else x - x%32
        img = cv2.resize(img, (to_32s(w), to_32s(h)))
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB).astype(np.float32)/ 127.5 - 1.0
    return img
def post_precess(img, wh):
    img = (img.squeeze()+1.) / 2 * 255
    img = img.astype(np.uint8)
    img = cv2.resize(img, (wh[0], wh[1]))
    return img
@login_required(login_url='user_login')
def video_convert(request)
user = MyUser.objects.get(email = request.user)
    try:
        if request.method == 'POST':
            video_name = request.FILES['video']
            fil = request.POST.get('filter')
            video = Video_c.objects.create(video=video_name)
            i_id = video.video_id
            return redirect('video_result',i_id,fil)
    except MultiValueDictKeyError:
            messages.warning(request,'Please Upload a Video to convert')
            return redirect('video_convert')
    return render(request,'user/user-video-convert.html')
@login_required(login_url='user_login')
def video_result(request,id,fil):
    messages.info(request,'Please be patient converting process has began')
    user = MyUser.objects.get(email = request.user)
    vid = Video_c.objects.get(video_id=id)
    video_name = str(vid.video)
```

```
print(video_name,'v_name')
    # video = str(vid.video)
    checkpoint_dir = 'filter/'+str(fil)
    video = './media/'+ video_name
    # checkpoint_dir = "checkpoint/generator_Hayao_weight"
    output = "output/"
    converted_video = output+video_name
    output_format='MP4V'
    test_real = tf.placeholder(tf.float32, [1, None, None, 3], name='test')
    with tf.variable_scope("generator", reuse=tf.AUTO_REUSE):
        test_generated = generator.G_net(test_real).fake
    saver = tf.train.Saver()
    # load video
vid = cv2.VideoCapture(video)
# vid_name = os.path.basename(video)
    total = int(vid.get(cv2.CAP_PROP_FRAME_COUNT))
fps = vid.get(cv2.CAP_PROP_FPS)
    width = int(vid.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(vid.get(cv2.CAP_PROP_FRAME_HEIGHT))
    codec = cv2.VideoWriter_fourcc(*output_format)
    tfconfig = tf.ConfigProto(allow_soft_placement=True)
    with tf.Session(config=tfconfig) as sess:
        ckpt = tf.train.get_checkpoint_state(checkpoint_dir)  # checkpoint file information
        if ckpt and ckpt.model_checkpoint_path:
            ckpt_name = os.path.basename(ckpt.model_checkpoint_path)  # first line
            saver.restore(sess, os.path.join(checkpoint_dir, ckpt_name))
            print(" [*] Success to read {}".format(os.path.join(checkpoint_dir, ckpt_name)))
        else:
            print(" [*] Failed to find a checkpoint")
            # return
        video_out = cv2.VideoWriter(os.path.join(output,video_name), codec, fps, (width,
height)
        pbar = tqdm(total=total, ncols=80)
        pbar.set_description(f"Making: {os.path.basename(video).rsplit('.', 1)[0] + video_name}")
```

```
while True:
        ret, frame = vid.read()
        if not ret:
            break
        frame = np.asarray(np.expand_dims(process_image(frame),0))
        fake_img = sess.run(test_generated, feed_dict={test_real: frame})
        fake_img = post_precess(fake_img, (width, height))
        video_out.write(cv2.cvtColor(fake_img, cv2.COLOR_BGR2RGB))
        pbar.update(1)
    pbar.close()
    vid.release()
video_out.release()
context = {'converted':converted_video}
return render(request,'user/user-video-result.html',context
```

TESTING

7.TESTING

INTRODUCTION:

Testing is the process of executing a program to find errors. To make our software perform well it should be error-free. If testing is done successfully it will remove all the errors from the software.

7.1 **TESTING METHODOLOGIES:**

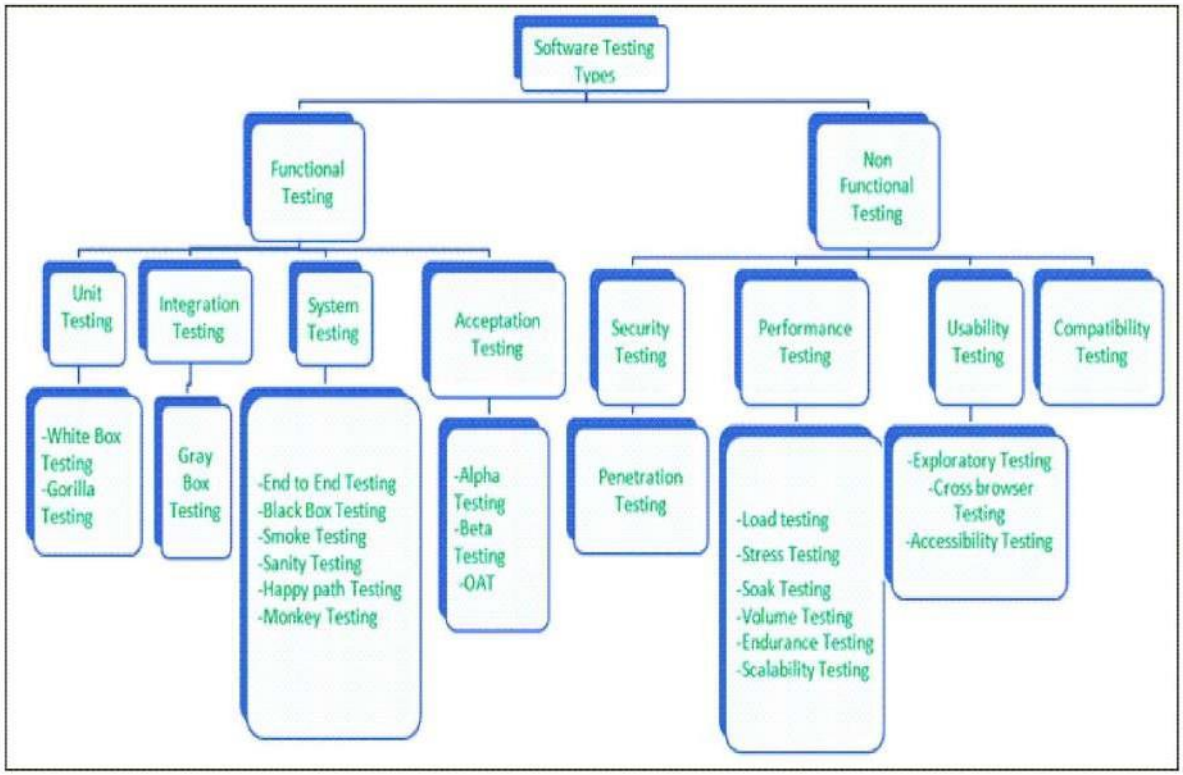
SYSTEM TESTING

Types of Software Testing: Different Testing Types with Details

We, as testers, are aware of the various types of Software Testing like Functional Testing, Non-Functional Testing, Automation Testing, Agile Testing, and their sub-types, etc.

Each type of testing has its own features, advantages, and disadvantages as well. However, in this tutorial, we have covered mostly each and every type of software testing which we usually use in our day-to-day testing life.

Different types of software Testings



Functional Testing

There are four main types of functional testing.

#1) Unit Testing

Unit testing is a type of software testing which is done on an individual unit or component to test its corrections. Typically, Unit testing is done by the developer at the application development phase. Each unit in unit testing can be viewed as a method, function, procedure, or object. Developers often use test automation tools such as NUnit, Xunit, JUnit for the test execution.

Unit testing is important because we can find more defects at the unit test level.

For example, there is a simple calculator application. The developer can write the unit test to check if the user can enter two numbers and get the correct sum for addition functionality.

a) White Box Testing

White box testing is a test technique in which the internal structure or code of an application is visible and accessible to the tester. In this technique, it is easy to

find loopholes in the design of an application or fault in business logic. Statement coverage and decision coverage/branch coverage are examples of white box test techniques.

b) Gorilla Testing

Gorilla testing is a test technique in which the tester and/or developer test the module of the application thoroughly in all aspects. Gorilla testing is done to check how robust your application is.

For example, the tester is testing the pet insurance company's website, which provides the service of buying an insurance policy, tag for the pet, Lifetime membership. The tester can focus on any one module, let's say, the insurance policy module, and test it thoroughly with positive and negative test scenarios.

#2) Integration Testing

Integration testing is a type of software testing where two or more modules of an application are logically grouped together and tested as a whole. The focus of this type of testing is to find the defect on interface, communication, and data flow among modules. Top-down or Bottom-up approach is used while integrating modules into the whole system.

This type of testing is done on integrating modules of a system or between systems. **For example,** a user is buying a flight ticket from any airline website. Users can see flight details and payment information while buying a ticket, but flight details and payment processing are two different systems. Integration testing should be done while integrating of airline website and payment processing system.

a) Gray box testing

As the name suggests, gray box testing is a combination of white-box testing and black-box testing. Testers have partial knowledge of the internal structure or code of an application.

#3) System Testing

System testing is types of testing where tester evaluates the whole system against the specified requirements.

a) End to End Testing

It involves testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

For example, a tester is testing a pet insurance website. End to End testing involves testing of buying an insurance policy, LPM, tag, adding another pet, updating credit card information on users' accounts, updating user address information, receiving order confirmation emails and policy documents.

b) Black Box Testing

Blackbox testing is a software testing technique in which testing is performed without knowing the internal structure, design, or code of a system under test. Testers should focus only on the input and output of test objects.

Detailed information about the advantages, disadvantages, and types of Black Box testing can be found here.

c) Smoke Testing

Smoke testing is performed to verify that basic and critical functionality of the system under test is working fine at a very high level.

Whenever a new build is provided by the development team, then the Software Testing team validates the build and ensures that no major issue exists.

testing team will ensure that the build is stable, and a detailed level of testing will be carried out further.

For example, tester is testing pet insurance website. Buying an insurance policy, adding another pet, providing quotes are all basic and critical functionality of the application. Smoke testing for this website verifies that all these functionalities are working fine before doing any in-depth testing.

d) Sanity Testing

Sanity testing is performed on a system to verify that newly added functionality or bug fixes are working fine. Sanity testing is done on stable build. It is a subset of the regression test.

For example, a tester is testing a pet insurance website. There is a change in the discount for buying a policy for second pet. Then sanity testing is only performed on buying insurance policy module.

e) Happy path Testing

The objective of Happy Path Testing is to test an application successfully on a positive flow. It does not look for negative or error conditions. The focus is only on valid and positive inputs through which the application generates the expected output.

f) Monkey Testing

Monkey Testing is carried out by a tester, assuming that if the monkey uses the application, then how random input and values will be entered by the Monkey without any knowledge or understanding of the application.

The objective of Monkey Testing is to check if an application or system gets crashed by providing random input values/data. Monkey Testing is performed

randomly, no test cases are scripted, and it is not necessary to be aware of the full functionality of the system.

#4) Acceptance Testing

Acceptance testing is a type of testing where client/business/customer test the software with real time business scenarios.

The client accepts the software only when all the features and functionalities work as expected. This is the last phase of testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

a) Alpha Testing

Alpha testing is a type of acceptance testing performed by the team in an organization to find as many defects as possible before releasing software to customers.

For example, the pet insurance website is under UAT. UAT team will run real-time scenarios like buying an insurance policy, buying annual membership, changing the address, ownership transfer of the pet in a same way the user uses the real website. The team can use test credit card information to process payment-related scenarios.

b) Beta Testing

Beta Testing is a type of software testing which is carried out by the clients/customers. It is performed in the **Real Environment** before releasing the product to the market for the actual end-users.

Beta Testing is carried out to ensure that there are no major failures in the software or product, and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software.

Usually, this testing is typically done by the end-users. This is the final testing done before releasing the application for commercial purposes. Usually, the Beta

version of the software or product released is limited to a certain number of users in a specific area.

So, the end-user uses the software and shares the feedback with the company. The company then takes necessary action before releasing the software worldwide.

c) Operational acceptance testing (OAT)

Operational acceptance testing of the system is performed by operations or system administration staff in the production environment. The purpose of operational acceptance testing is to make sure that the system administrators can keep the system working properly for the users in a real-time environment.

The focus of the OAT is on the following points:

- Testing of backup and restore.
- Installing, uninstalling, upgrading software.
- The recovery process in case of natural disaster.
- User management.
- Maintenance of the software.

Non-Functional Testing

There are four main types of functional testing.

#1) Security Testing

It is a type of testing performed by a special team. Any hacking method can penetrate the system.

Security Testing is done to check how the software, application, or website is secure from internal and/or external threats. This testing includes how much software is secure from malicious programs, viruses and how secure & strong the authorization and authentication processes are.

It also checks how software behaves for any hacker's attack & malicious programs and how software is maintained for data security after such a hacker attack.

a) Penetration Testing

Penetration Testing or Pen testing is the type of security testing performed as an authorized cyberattack on the system to find out the weak points of the system in terms of security.

Pen testing is performed by outside contractors, generally known as ethical hackers. That is why it is also known as ethical hacking. Contractors perform different operations like SQL injection, URL manipulation, Privilege Elevation, session expiry, and provide reports to the organization.

Notes: Do not perform the Pen testing on your laptop/computer. Always take written permission to do pen tests.

#2) Performance Testing

Performance testing is testing of an application's stability and response time by applying load.

The word stability means the ability of the application to withstand in the presence of load. Response time is how quickly an application is available to users. Performance testing is done with the help of tools. Loader.IO, JMeter, LoadRunner, etc. are good tools available in the market.

a) Load testing

Load testing is testing of an application's stability and response time by applying load, which is equal to or less than the designed number of users for an application.

For example, your application handles 100 users at a time with a response time of 3 seconds, then load testing can be done by applying a load of the maximum of 100 or less than 100 users. The goal is to verify that the application is responding within 3 seconds for all the users.

b) Stress Testing

Stress testing is testing an application's stability and response time by applying load, which is more than the designed number of users for an application.

For example, your application handles 1000 users at a time with a response time of 4 seconds, then stress testing can be done by applying a load of more than 1000 users. Test the application with 1100,1200,1300 users and notice the response time. The goal is to verify the stability of an application under stress.

c) Scalability Testing

Scalability testing is testing an application's stability and response time by applying load, which is more than the designed number of users for an application.

For example, your application handles 1000 users at a time with a response time of 2 seconds, then scalability testing can be done by applying a load of more than 1000 users and gradually increasing the number of users to find out where exactly my application is crashing.

Let's say my application is giving response time as follows:

- 1000 users -2 sec
- 1400 users -2 sec
- 4000 users -3 sec
- 5000 users - 45 sec
- 5150 users- crash – This is the point that needs to identify in scalability testing

d) Volume testing (flood testing)

Volume testing is testing an application's stability and response time by transferring a large volume of data to the database. Basically, it tests the capacity of the database to handle the data.

e) Endurance Testing (Soak Testing)

Endurance testing is testing an application's stability and response time by applying load continuously for a longer period to verify that the application is working fine.

For example, car companies soak testing to verify that users can drive cars continuously for hours without any problem.

#3) Usability Testing

For example, there is a mobile app for stock trading, and a tester is performing usability testing. Testers can check the scenario like if the mobile app is easy to operate with one hand or not, scroll bar should be vertical, background colour of the app should be black and price of and stock is displayed in red or green colour.

The main idea of usability testing of this kind of app is that as soon as the user opens the app, the user should get a glance at the market.

a) Exploratory testing

Exploratory Testing is informal testing performed by the testing team. The objective of this testing is to explore the application and look for defects that exist

in the application. Testers use the knowledge of the business domain to test the application. Test charters are used to guide the exploratory testing.

b) Cross browser testing

Cross browser testing is testing an application on different browsers, operating systems, mobile devices to see look and feel and performance.

Why do we need cross-browser testing? The answer is different users use different operating systems, different browsers, and different mobile devices. The goal of the company is to get a good user experience regardless of those devices.

Browser stack provides all the versions of all the browsers and all mobile devices to test the application. For learning purposes, it is good to take the free trial given by browser stack for a few days.

c) Accessibility Testing

The aim of Accessibility Testing is to determine whether the software or application is accessible for disabled people or not.

Here, disability means deafness, colour blindness, mentally disabled, blind, old age, and other disabled groups. Various checks are performed, such as font size for visually disabled, colour and contrast for colour blindness, etc.

#4) Compatibility testing

This is a testing type in which it validates how software behaves and runs in a different environment, web servers, hardware, and network environment.

7.2 TEST CASES:

S. NO	Test Case ID	Test Case name	Test case Description	Steps to be Executed	Expected Result	Actual Result	Test Case Status Pass/Fail
1	User Login	Validate the login of User	To verify User login with user credentials	Enter user credentials and click on login	Login Successful	Login Successful	pass
2	Upload the image	Image uploaded	Take image from anywhere	Click on upload button	Image uploaded successfully	Image uploaded successfully	pass
3	Image conversion	To Convert Image as cartoon	Take uploaded image	Click on convert button	Display image as cartoon	Successfully display the cartoon	pass
4	Upload the video	Video uploaded	Take the video	Click on upload button	Video uploaded successfully	Video uploaded successfully	pass
5	Video conversion	To convert video as Gif	Take the uploaded video	Clock on convert button	Successfully video converted as Gif or sticker	Successfully video converted as Gif or sticker	pass
6	Video conversion	To convert video as Gif	Take the uploaded 5min video	Clock on convert button	It does't Converted as Gif	It only converts low length videos	fail

7	My Profile	For changing the user details	For updating the user profile	Click on my profile and enter new details	Profile Updated Successfully.	Profile Updated Successfully.	pass
8	Logout	User Logout	Logout from the user profile	Logout from the user account.	Logout Successfully.	Logout Successfully.	pass
9	View Users	View Users	To display the registered users	Click on View Users	Displaying the Registered Users	Displaying the Registered Users	pass
10	logout	logout	Logout from the page	Click on logout	Logout successful	Logout successful	pass

Test Result: All the test cases mentioned above passed successfully. But one thing is limited capacity for video coversion.No defects encountered.

SCREENSHOTS

8.OUTPUT SCREENS

Figure 8.1: Command Execution :

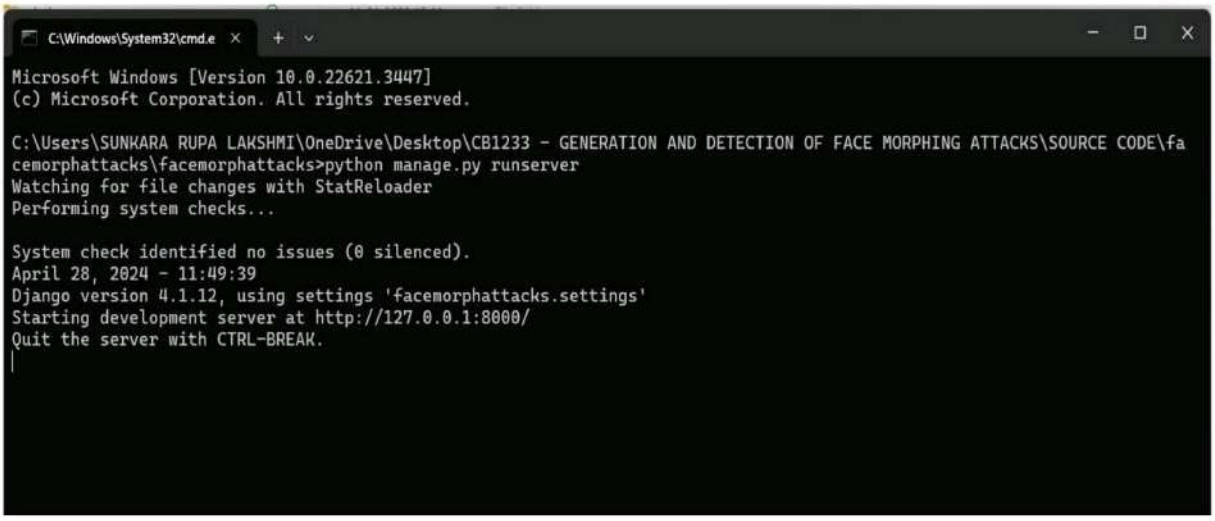


Figure 8.2: Home Page

Figure 8.3: User Registration Details

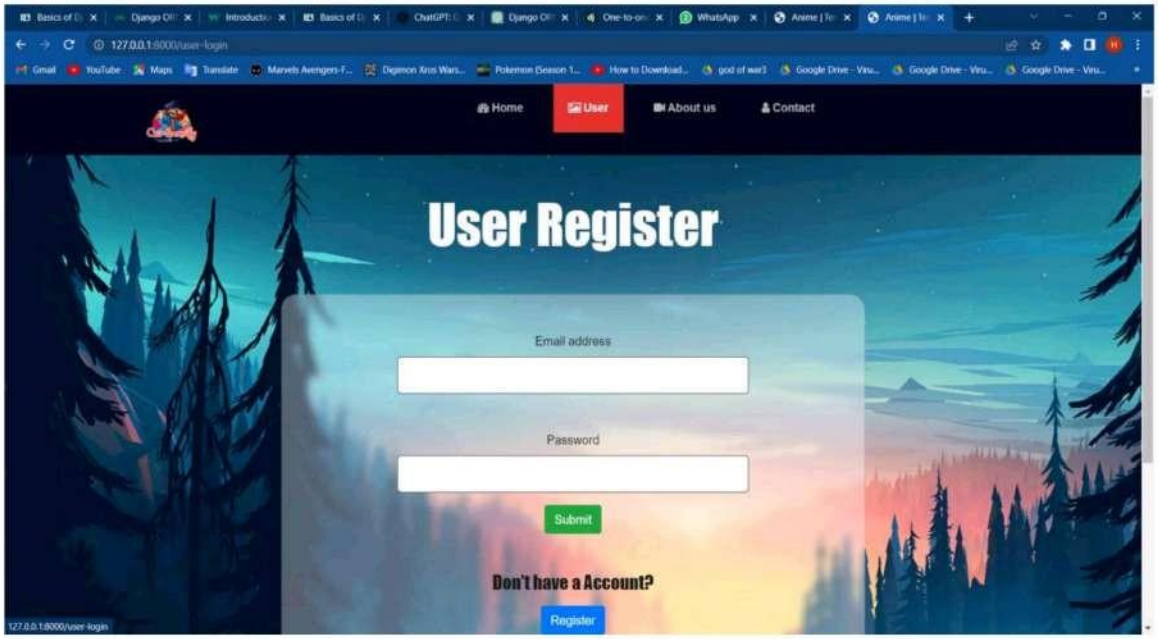


Figure 8.4: My profile page

Figure 8.5: User Login

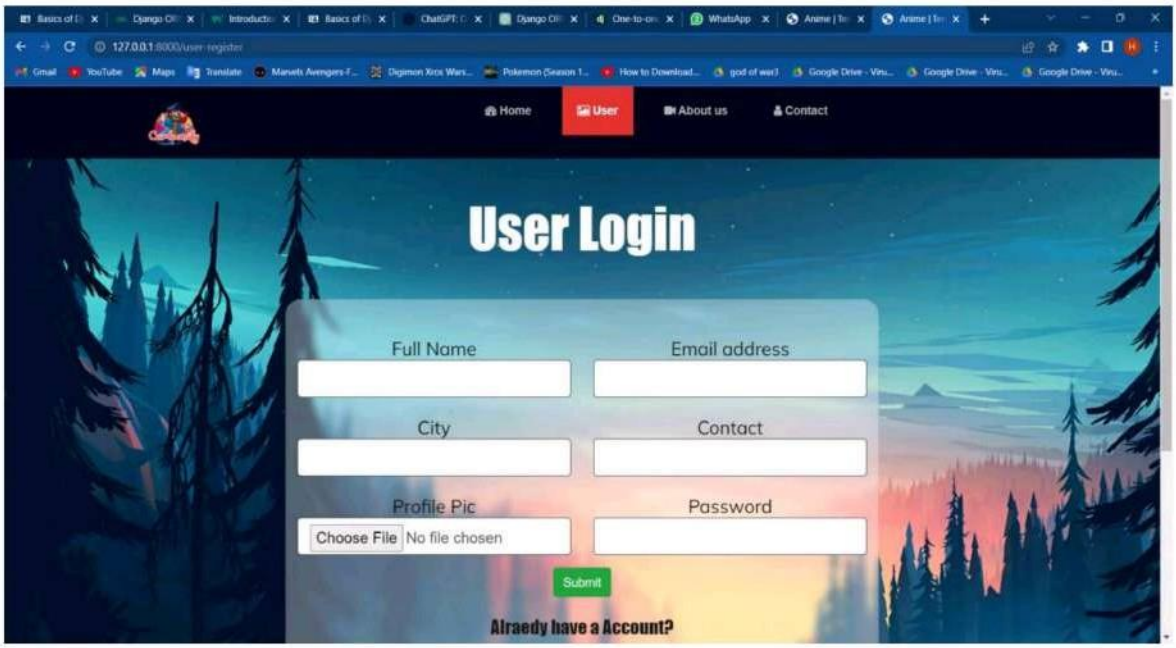


Figure 8.6: About page

Figure 8.7: User page

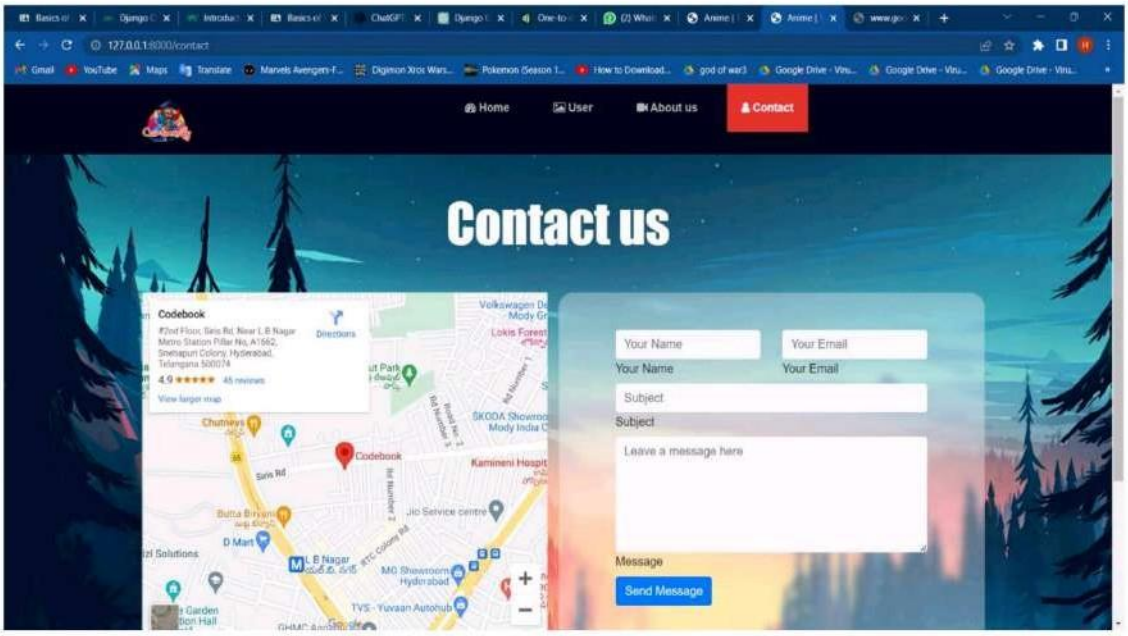


Figure 8.8: Dashboard

Figure 8.9: applying for cartoon filters

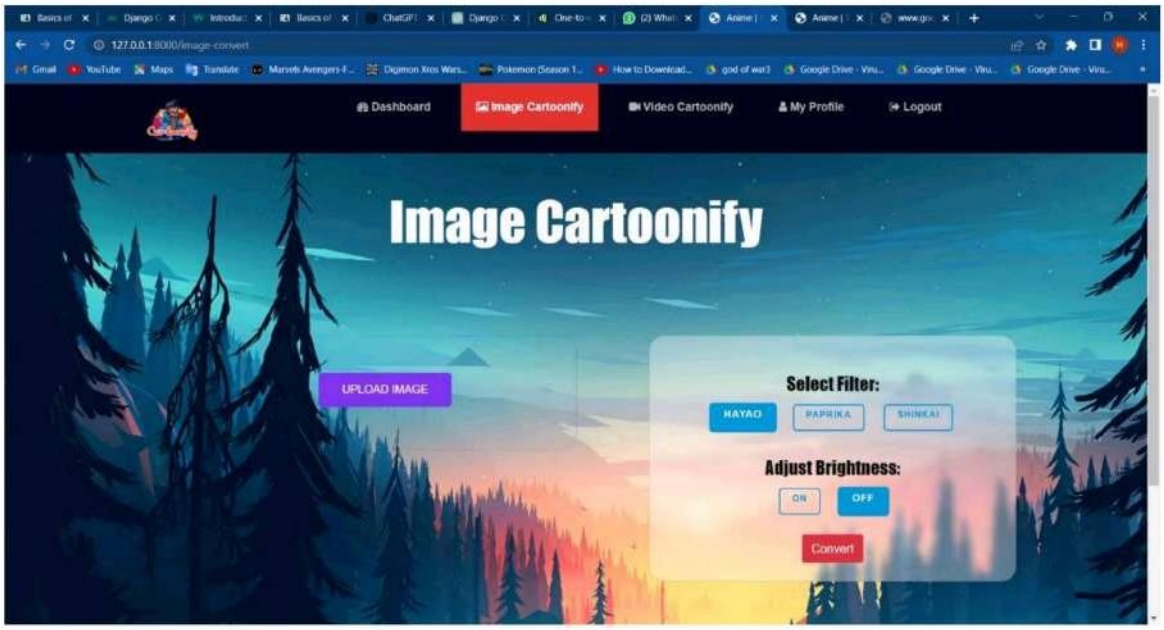


Figure 8.10: Cartoon Result

Figure 8.11: Video Cartoonify

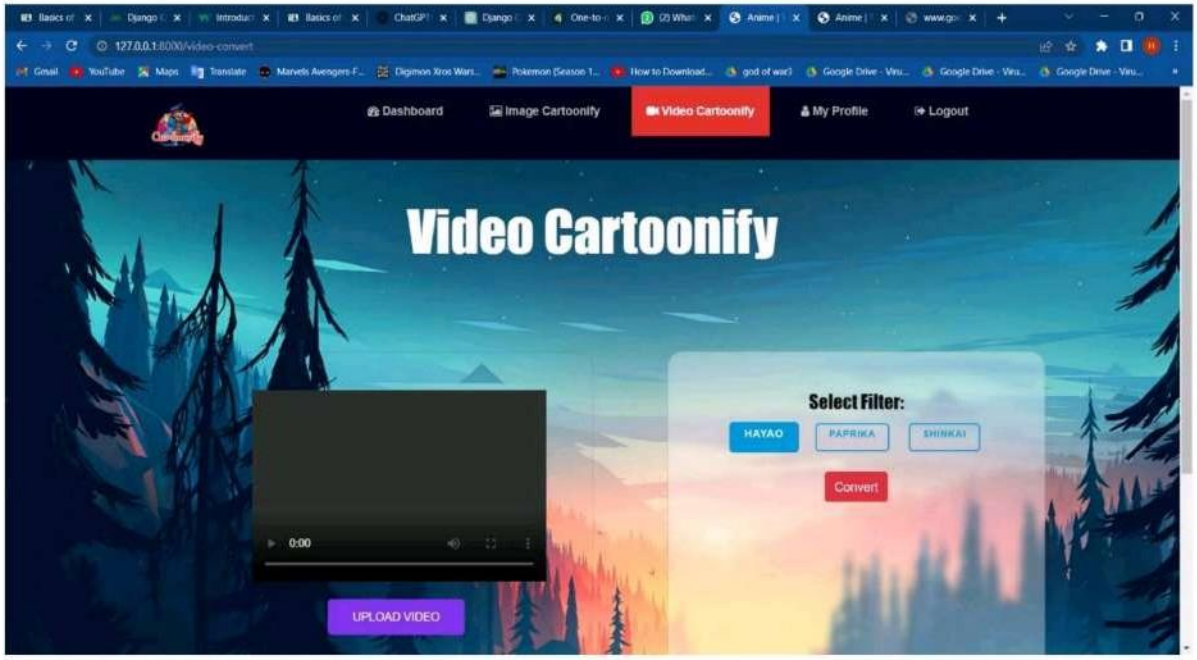
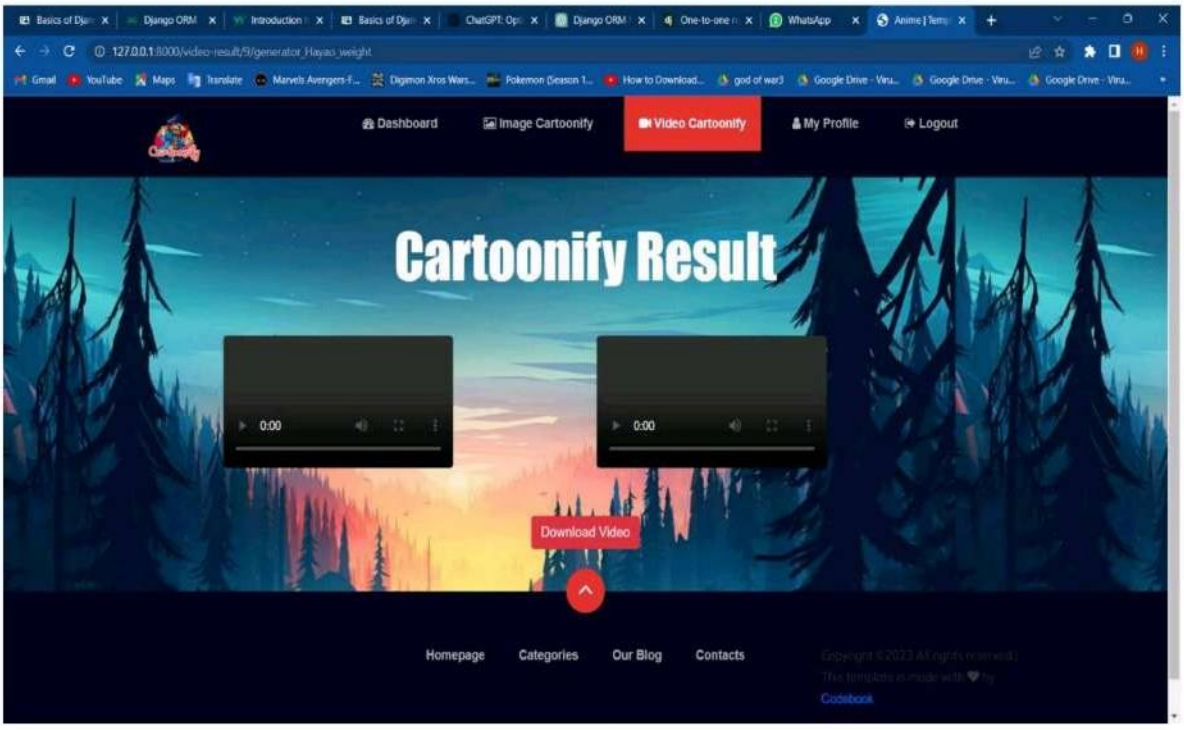


Figure 8.12: Cartoon result



CONCLUSION AND FUTURE RECOMMENDATIONS

9.1 CONCLUSION

The present methods for converting real-world images or videos into cartoon versions which were proposed by other systems compromise the quality of the image. In the case of video, the audio present in the video file is lost in the resultant cartoon version of the video. We purpose a system that helps in cartoonization of images and videos with the help of Generative Adversarial Models (GANs). To implement this, real-world image files are denoised and then passed through the GAN model which generates the desired cartoonized image. The video is cartoonized by dividing the video into multiple image frames and simultaneously extracting audio from the image file. Each image frame is denoised and passed through the GAN to generate frames of cartoonized image. Those cartoonized image frames are aggregated and converted into video which is aggregated with the audio file to obtain the cartoonized videos. After the implementation of the project, the desired cartoon version of images and videos was achieved. Also the loss of audio has been solved using this approach. Hence, the GAN based Cartoonization model helps in saving time to convert the real-time image/videos into their cartoon version with less noise and better quality.

REFERENCES

REFERENENCES

[1] Y. Chen, Y.-K. Lai, Y.-J. Liu, "Transforming photos to comics using convolutional neural networks", International Conference on Image Processing, 2017.

[2] Yang Chen, Yu-Kun Lai, Yong-Jin Liu, "CartoonGAN: Generative Adversarial Networks for Photo Cartoonization",2020.

[3] I..Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672– 2680.

[4] Akanksha Apte, Ashwathy Unnikrishnan, NavjeevanBomble, Prof. SachinGavhane," Transformation of Realistic Images and Videos into Cartoon Images ",2020

[5] Bee Lim, Sanghyun Son, Heewon Kim Seungjun, Nah Kyoung Mu Lee,"Enhanced Deep Residual Networks for Single Image Super-Resolution", arXiv:1707.02921v1 [cs.CV], 10 Jul 2017

[6] Chao Dong, Chen Change Loy, Xiaoou Tang, "Accelerating the Super-Resolution Convolutional Neural Network", arXiv:1608.00367v1 [cs.CV], 1 Aug 2016

[7] https://opencv pythontutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_intro/py_intro.html

[8] https://docs.opencv.org/3.4/d1/d79/groupphotodenoise.html#ga03aa4189fc3e31dafd638d90de335617

[9] https://www.geeksforgeeks.org/generative-adversarial-network-gan/

[10] https://developers.google.com/machine-learning/gan/gan_structure

BIBILOGRAPHY

- W3 School – (https://www.w3schools.com/python/)
- Geek for Geeks – (https://www.geeksforgeeks.org/python-programming-language/learn-python-tutorial/)
- Python Official Documentation – (https://docs.python.org/3/tutorial/)
- Tutorials Point – (https://www.tutorialspoint.com/python/index.htm)
- Real Python – (https://realpython.com/)
- Django for Beginners – (https://djangoforbeginners.com/introduction/)
- Guru99 – (https://www.guru99.com/django-tutorial.html)