

# **Cartoonify Image**

**A Project Report  
Submitted in the partial fulfilment for the award of**

**Master of Computer Application**

**Submitted to**



**DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY  
LUCKNOW (UP) INDIA**

**Submitted by**

**Nitish Rathi (2000290140080)**

**Under the Guidance of**

**Prof. Ankit Verma**



**DEPARTMENT OF Master of Computer Application  
KIET Group of Institution**

## **KIET Group of Institution**

### **CANDIDATE DECLARATION**

I hereby declare that the work, which is being presented in the project, towards the partial fulfilment of the requirement for the award of the degree of **Master of Computer Application**, at KIET Group of Institution, Ghaziabad (India) is an authentic record of my own work carried out during the period from January 2022 to March 2022, under the guidance of **Mr. SD Mishra, Department of Master of Computer Application, KIET Group of Institution, Ghaziabad.**

I have not submitted the matter embodied in this project for the award of any other degree or diploma.

**Nitish Rathi**                      **(2000290140080)**

Date:

Place: Ghaziabad

## **KIET Group of Institution**

### **CERTIFICATE**

This is to certify that the project submitted to **Dr. A.P.J.Abdul Kalam Technical University Lucknow (U.P.)** by Nitish Rathi is a partial fulfilment of the requirement for the award of degree of Master of Computer Application. The matter embodied is the actual work done by Nitish Rathi is a record of bona fide work done by him under my supervision.

**Guided By**

**Approved By**

**Forward by**

**(Mr.SD Mishra)**

**Asst. Professor**

**Academic**

(Master of Computer Application)

## **KIET Group of Institution**

### **CERTIFICATE OF APPROVAL**

This project work entitled submitted by Nitish Rathi is approved for the award of degree of **Master of Computer Application**.

**(Name of Internal Examiner)**

**(Name of External Examiner)**

**(Sign of Internal Examiner)**

**(Sign of External Examiner)**

**Date:**

**Date:**

## **KIET Group of Institution**

### **ACKNOWLEDGEMENT**

I express my deep sense of gratitude to **Mr. Ankit Verma**, Faculty of Master of Computer Application at KIET Group of Institution, Ghaziabad, Whose kindness valuable guidance and timely help encouraged me to complete this volume on a very crucial issue related to the work and who helped me in completing the project and he exchanged his interesting ideas, thoughts and made this project easy and accurate.

I express my thanks to **Dr. Ajay Kumar Shrivastav**, Head of Master of Computer Application Department for kindly intention.

I express my thanks to the authors whose works have been consulted by me during the project.

I would also thank my institution and my faculty members without whom this project would have been a distant reality.

I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go my own way without whom I would be unable to complete my project .

At last but not the least I want to thank my friends who appreciated me for my work and motivated me and finally to god who made all the things possible.

**Nitish Rathi**

**(2000290140080)**

## ABSTRACT

Since the client for animation pictures recovery framework focuses to get applicable pictures to inquiry picture from information base inside same item (for example a client has animation picture with object, for this situation the client will focus to get all significant picture with, along these lines A significant advance in animation picture recovery is characterizing the item inside animation picture, a proficient technique for objects extraction from animation pictures is presented; it depends on broad suppositions identified with shading and areas of items in animation pictures, the items are commonly gravitated toward the focal point of the picture, the foundation tones is the all the more much of the time gravitated toward the edges of animation picture, and the item colors is less touch for the edges. The cycles of shading quantization, seed filling and found the item apparition have been utilized. The aftereffects of led tests showed that the framework have promising effectiveness for extricating both single or multi object(s) lay in straightforward and complex foundations of animation pictures.

**Keywords:** Cartoon pictures, Cartoon Image Retrieval, k-Means applications.

# Table of Contents

Acknowledgements .....	v
1. Introduction about the project.....	8
1.1 Modules .....	8
1.2 Functional Requirement .....	8
1.3 Nonfunctional requirement .....	9
2. Introduction .....	10
2.1 Existing System .....	10
2.2 Drawback of existing system .....	10
2.3 Proposed system.....	10
2.4 Expected advantage of proposed system.....	11
2.4.1 Feasibility study .....	12
2.4.2 Technical Feasibility .....	12
2.4.3 Economic Feasibility.....	12
2.4.4 Behavioral Feasibility.....	13
3. Software and Hardware Specification .....	13
3.1 Hardware Specification .....	13
3.2 Software Specification.....	13
4. Software Development life cycle .....	14
5. Design.....	15
5.1 Logical Design.....	16
5.2 Physical Design .....	16
5.3 Module Design .....	17

5.4 Input Design.....	18
5.5 Output Design.....	19
5.6 Block Diagram.....	20
7. Feature of Operating System .....	26
8. Screenshot.....	29
9. Testing .....	34
10. Conclusion .....	37
10.1 Scope of Future Enhancement.....	38
11. Online Reference .....	39



## **1. Introduction about the project**

Social media is extensively used these days. And standing out in this online crowd has always been a to-do on every user's list on these social media platforms. Be it images, blog posts, artwork, tweets, memes, opinions and what not being used to seek attention of followers or friends to create influence or to connect with them on such social platforms. We aim to provide one such creative solution to their needs, which is applying cartoon like effects to their images. Users can later share these images on any social media platforms, messengers, keep it for themselves, share it with loved ones or do whatever they like with it. Nowadays almost everyone is registered in social networks. We keep online status updated every day, share photos and comments, follow our friends' news. To have a nice profile is a matter of prestige. You can use a photo of your own in a profile image, create an amusing avatar or turn your photo into a cartoon. With a pool of web applications available online, an image conversion to cartoon takes few clicks.

### **1. Modules**

- Image
- Video

### **2. Need of Project:**

Creating a cartoon like effect is time and space consuming. Existing solutions to provide cartoon like effect to images are complex. Some solutions involve installing complex photo editing software like photoshop and other involve performing some task by user. Our research shows a website to carry out the task of Applying effects is more suitable, space efficient and takes minimum user efforts, for example toony photos is an existing website to perform such task but it is difficult to use as user has to mark down points & lines on the image to apply effects which is not user friendly also the options are limited. Hence there is a dire need for a website which is user friendly and performs the task of applying effects to images very well

## 2. Introduction:

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

## 1. Existing System:

Cartoon Effect- The majority of photo editing websites offer the so called Cartoon Effect. The main advantages of online photo to cartoon effect apps are simplicity and quickness. You'll have to upload a photo from your computer or from the web, find Cartoon Effect in the tool set or choose between styles or variants of this funny photo effect (like in case of [www.picturetopeople.org](http://www.picturetopeople.org), Kuso Cartoon ) and press the button Apply (or Go). The image processing varies from several seconds up to 1-2 minutes. However, as all quick online solutions these apps have drawbacks. A lot of photo online photo editing tools are rather humdrum because they are deprived of enhancement features. In these apps cartoonization is limited to 1-click operation. Besides, sometimes colors may become blurred and it leads to an unsatisfactory result. Such apps as [www.converttocartoon.com](http://www.converttocartoon.com), Photo.to, Any Making and others belong to this group. At the same time there are online photo editors with more advanced tools. They have a variety of adjustment options. For example, Be Funky helps you modify sketch brightness, contrast, smoothness and other details.

## 2. Drawbacks of Existing System

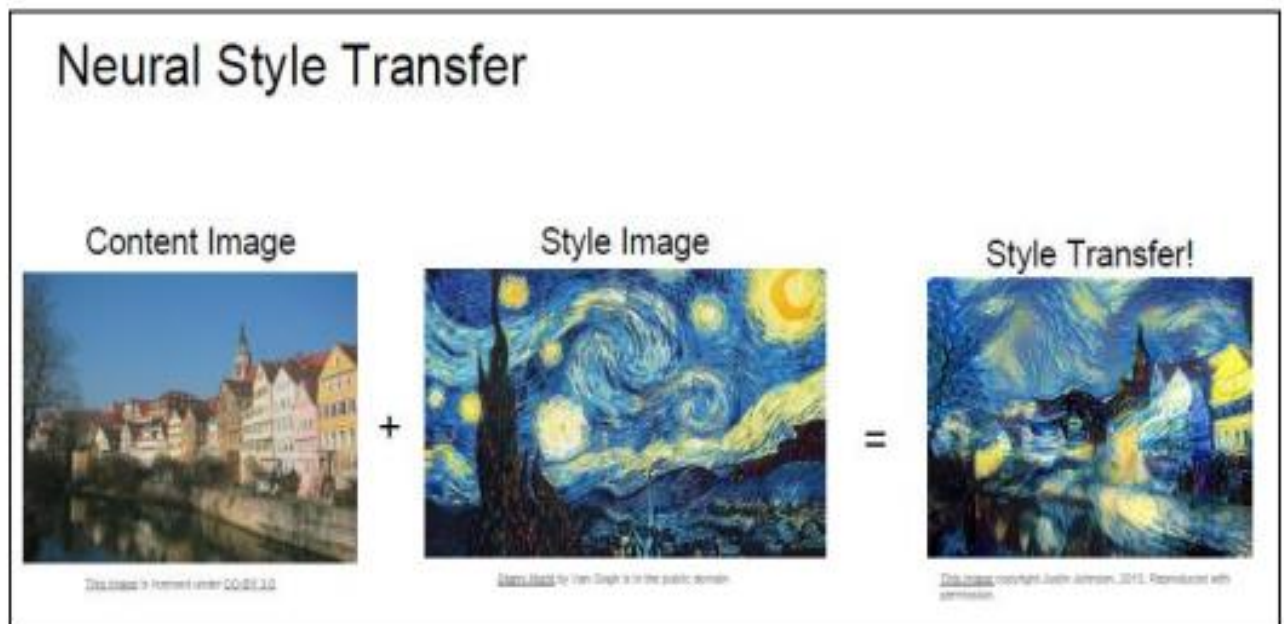
- Time consuming.
- Damage of machines due to lack of attention.

To avoid all these limitations and make the working more accurately the system needs to be computerized.

## 3. Proposed System:

We propose to use neural style transfer which is a machine learning algorithm, which involves two images, first is the input image from the user and second is the style image which is used to apply the style on the input image. Following are the examples of images generated using neural style transfer. We propose to create a website, which consists of image upload functionality using which the user can upload his image, the uploaded image is then processed by server using Neural style transfer algorithm and the resulting image is presented to the user on the website. Which then user can download & share. Neural fast style transfer is used by Apps such as <https://deepart.io>, Prisma, Artisto etc. We decided to choose this approach over traditional image filters (e.g. using image filters such as median & bilateral filters to posterize an Image) as Neural fast style transfer is quite new and challenging technique which uses machine learning & image processing to produce

various styled images based on variety of input & style images. The algorithm can be implemented in Python/JavaScript/Lua to perform neural style transfer. We will use Python to implement the backend and the front end of the website will be in HTML, CSS & JS. Basically, in Neural Style Transfer we have two images- style and content. We need to copy the style from the style image and apply it to the content image. By, style we basically mean, the patterns, the brushstrokes, etc. we will provide a set of style images which a user can use to apply different kinds of Cartoon like effects to his image.



#### **4. Expected Advantages of Proposed System:**

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

- Reduce the damages of the machines.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required.

##### **2.4.1 FEASIBILITY STUDY :**

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

## **2. TECHNICAL FEASIBILITY:**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

## **3. ECONOMIC FEASIBILITY**

The developing system must be justified by cost and benefit. Criteria to ensure that efforts concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

#### 4. BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

A. Hardware Requirements: → Windows 10, 64 bits PC or 64 Bit Mac OS X High Sierra computers → Any CPU (Intel i5/ i7/ Xeon recommended). → Nvidia GPUs (minimum 2GB Recommended) → At least 8 GB RAM, 10 GB HDD Free Space. B. Software Requirements: → Anaconda Framework for Python IDE & Packages (Recommended), Jupyter notebook, Tensorflow machine Learning library → Programming Language: Python

### 1. Software and Hardware Specifications

#### A. Hardware Requirements:

→ Windows 10, 64 bits PC or 64 Bit Mac OS X High Sierra computers

→ Any CPU (Intel i5/ i7/ Xeon recommended).

→ Nvidia GPUs (minimum 2GB Recommended)

→ At least 8 GB RAM, 10 GB HDD Free Space.

#### B. Software Requirements:

→ Anaconda Framework for Python IDE & Packages (Recommended), Jupyter notebook, Tensorflow machine Learning library

→ Programming Language: Python





#### 4.1 Software development life-cycle(SDLC) :

The systems development life cycle (SDLC), or software development process in systems information system and software engineering is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. In software engineering, the SDLC concept underpins many kinds methodologies form the framework for planning and controlling the creation of an information system: the software development process



A Systems Development Life Cycle (SDLC) adheres to important phases that are essential for developers, such as planning, analysing, design and implementation, and are explained in the section below. It includes evaluation of present system, information gathering, feasibility study and request approval. A number of system development life cycle (SDLC) models have

been created: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, and synchronize and stabilize. The oldest of these, and the best known, is the waterfall model : a sequence of stages in which the output of each stage becomes the input for the next. These stages can be characterized and divided up in different ways, including the following:

- **Systems analysis, requirements definition:** Defines project goals into defined functions and operation of the intended application. Analyzes end-user information needs.
- **Systems design:** Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation.
- **Development:** The real code is written here.
- **Integration and testing:** Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
- **Acceptance, installation, deployment:** The final stage of initial development, where the software is put into production and runs actual business.
- **Maintenance:** What happens during the rest of the software's life: changes, correction, additions, moves to a different computing platform and more. This is often the longest of the stages.

## 5. DESIGN:

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of

defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

## **1. LOGICAL DESIGN:**

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system – its data flows, file content, volumes, Frequencies etc.
- Prepares output specifications – that is, determines the format, content and Frequency of reports.
- Prepares input specifications – format, content and most of the input functions.
- Prepares edit, security and control specifications.
- Specifies the implementation plan.
- Prepares a logical design walk through of the information flow, output, input, Controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

## **2. PHYSICAL DESIGN:**

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- Design the physical system.
- Specify input and output media.
- Design the database and specify backup procedures.
- Design physical information flow through the system and a physical design
- Plan system implementation.
- Prepare a conversion schedule and target date.
- Determine training procedures, courses and timetable.
- Devise a test and implementation plan and specify any new hardware/software.
- Update benefits , costs , conversion date and system constraints

**Design/Specification activities:**

- Concept formulation.
- Problem understanding.
- High level requirements proposals.
- Feasibility study.
- Requirements engineering.
- Architectural design.

**4. INPUT DESIGN:**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.

- Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES:**

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be

free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

## **5. OUTPUT DESIGN :**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the Future.
- Signal important events, opportunities, problems, or warnings.

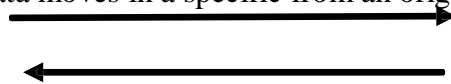
- Trigger an action.
- Confirm an action.

### **5.6 Data Flow Diagram:**

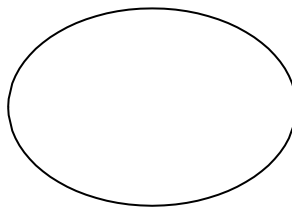
A **data flow diagram (DFD)** is a graphical representation of the "flow" of data through an [information system](#), modeling its *process* aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the [visualization](#) of [data processing](#) (structured design).

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).the basic notation used to create a DFD's are as follows:

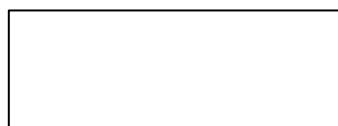
- 1. Dataflow:** data moves in a specific from an origin to a destination.



- 2. Process:** People ,procedures or device that use or produce data. The physical components not identified.

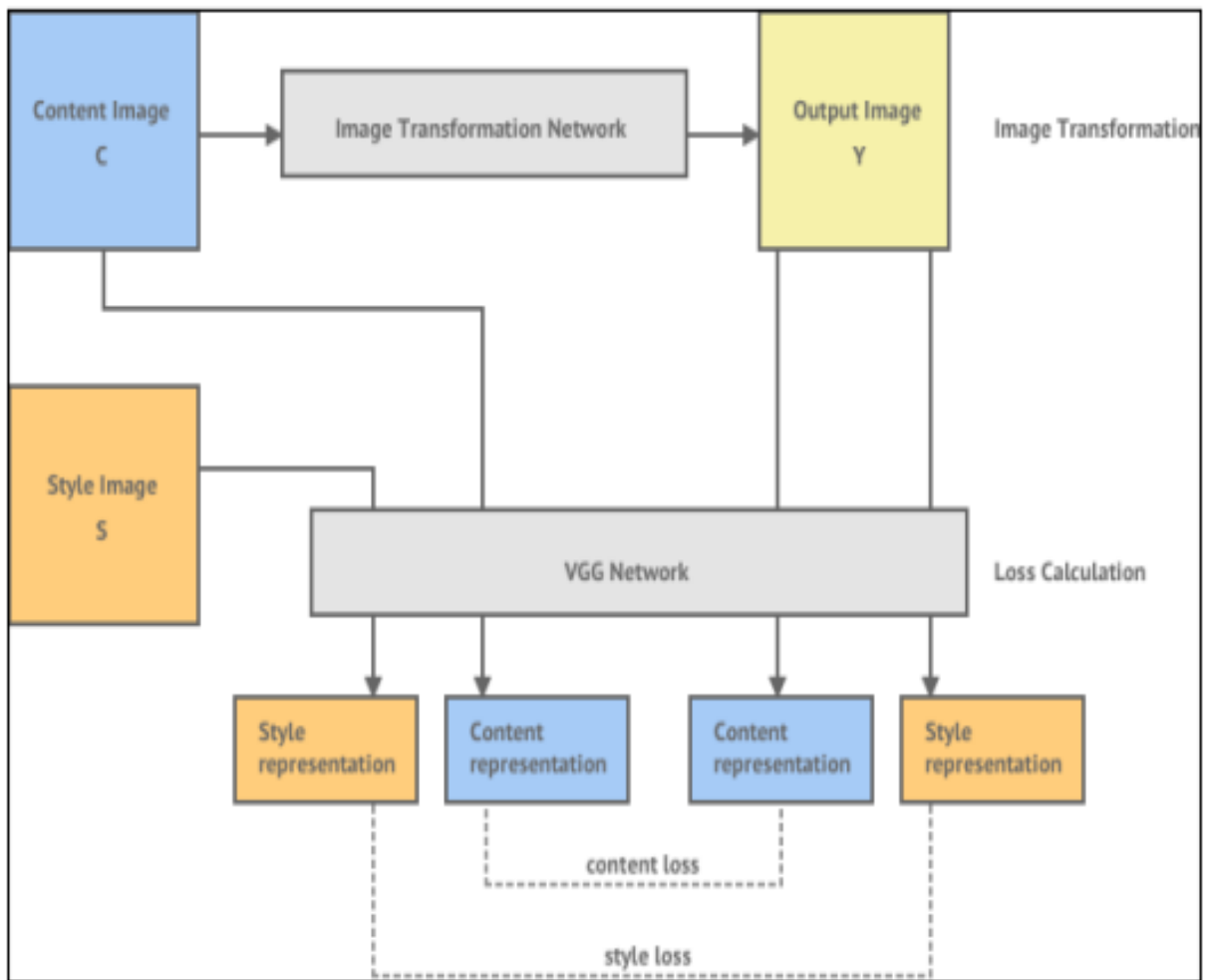


- 3. Source:** external source or destination of data, which may be people programs, organizations or other entities.

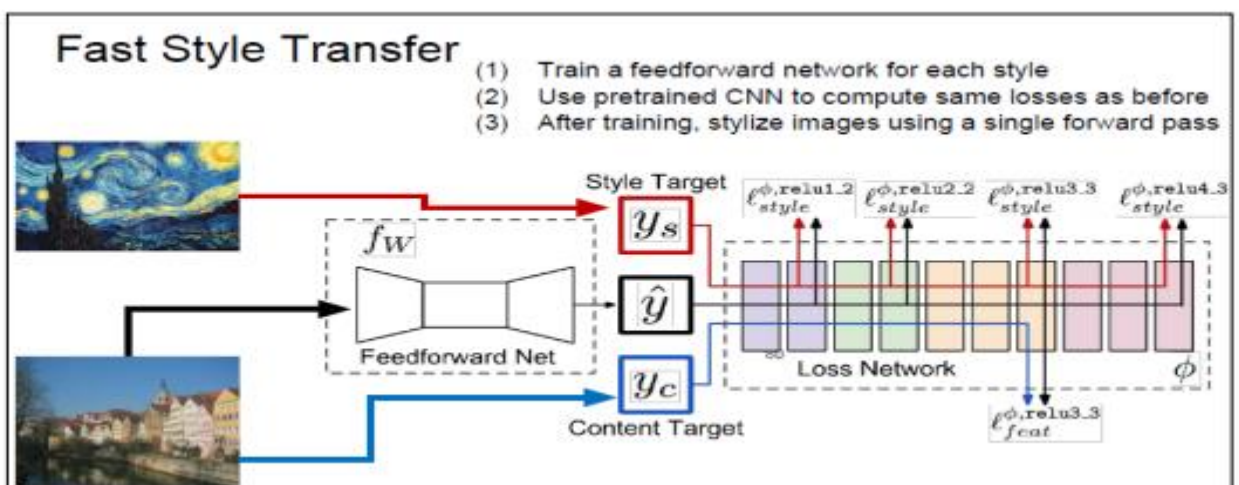


- 4 .Data source:** here data are store and referenced by a process in the system.

## 5.7 BLOCK DIAGRAM



## 5.8 PROCESS DIAGRAM -

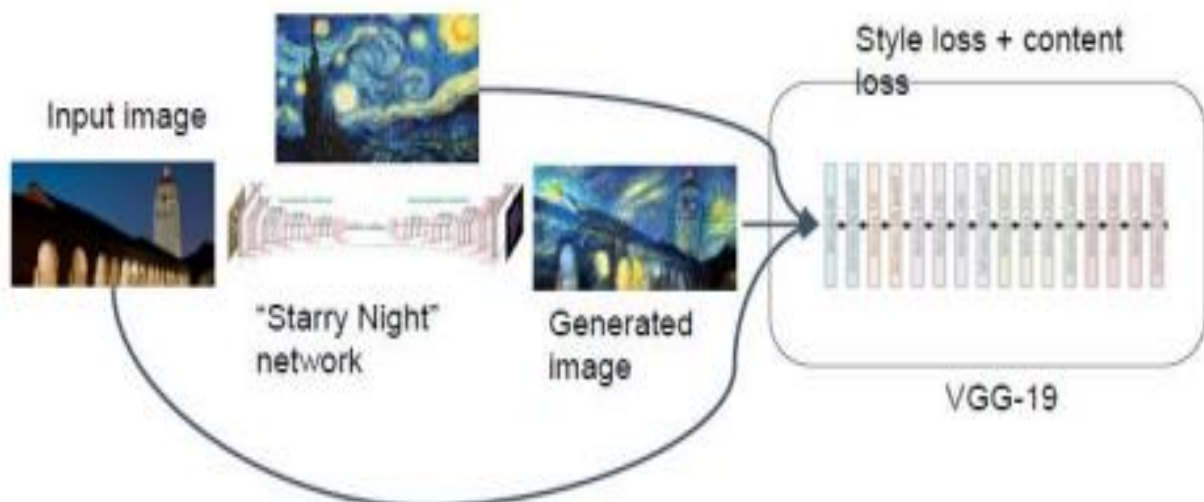




## 6. ALGORITHM –

Our implementation uses TensorFlow to train a fast style transfer network. We use roughly the same transformation network as described in Justin Johnson et. al, except that batch normalization is replaced with Ulyanov's instance normalization. We use a loss function close to the one described in Gatys, using VGG19 instead of VGG16 and typically using "shallower" layers than in Johnson's implementation (e.g. we use relu1\_1 rather than relu1\_2). Empirically, this results in larger scale style features in transformations.

- Overview of per-style-per-model "Fast" algorithm:



## 8. CODE SNIPPET –

### App.py

```
import os
import io
import uuid
import sys
import yaml
import traceback

with open('./config.yaml', 'r') as fd:
    opts = yaml.safe_load(fd)

sys.path.insert(0, './white_box_cartoonizer/')

import cv2
from flask import Flask, render_template, make_response, flash
import flask
from PIL import Image
import numpy as np
import skvideo.io
if opts['colab-mode']:
    from flask_ngrok import run_with_ngrok #to run the application on colab using
    ngrok

from cartoonize import WB_Cartoonize

if not opts['run_local']:
    if 'GOOGLE_APPLICATION_CREDENTIALS' in os.environ:
        from gcloud utils import upload_blob, generate_signed_url, delete_blob,
        download_video
    else:
        raise Exception("GOOGLE_APPLICATION_CREDENTIALS not set in environment
        variables")
    from video_api import api_request
    # Algorithmia (GPU inference)
    import Algorithmia

app = Flask(__name__)
if opts['colab-mode']:
    run_with_ngrok(app) #starts ngrok when the app is run

app.config['UPLOAD_FOLDER_VIDEOS'] = 'static/uploaded_videos'
app.config['CARTOONIZED_FOLDER'] = 'static/cartoonized_images'
```

```

app.config['OPTS'] = opts

## Init Cartoonizer and load its weights
wb_cartoonizer =
WB_Cartoonize(os.path.abspath("white_box_cartoonizer/saved_models/"), opts['gpu'])

def convert_bytes_to_image(img_bytes):
    """Convert bytes to numpy array

    Args:
        img_bytes (bytes): Image bytes read from flask.

    Returns:
        [numpy array]: Image numpy array
    """

    pil_image = Image.open(io.BytesIO(img_bytes))
    if pil_image.mode=="RGBA":
        image = Image.new("RGB", pil_image.size, (255,255,255))
        image.paste(pil_image, mask=pil_image.split()[3])
    else:
        image = pil_image.convert('RGB')

    image = np.array(image)

    return image

@app.route('/')
@app.route('/cartoonize', methods=["POST", "GET"])
def cartoonize():
    opts = app.config['OPTS']
    if flask.request.method == 'POST':
        try:
            if flask.request.files.get('image'):
                img = flask.request.files["image"].read()

                ## Read Image and convert to PIL (RGB) if RGBA convert
                appropriately
                image = convert_bytes_to_image(img)

                img_name = str(uuid.uuid4())

                cartoon_image = wb_cartoonizer.infer(image)

                cartoonized_img_name =
os.path.join(app.config['CARTOONIZED_FOLDER'], img_name + ".jpg")
                cv2.imwrite(cartoonized_img_name, cv2.cvtColor(cartoon_image,
cv2.COLOR_RGB2BGR))

                if not opts["run_local"]:
                    # Upload to bucket

```

```

        output_uri = upload_blob("cartoonized_images",
cartoonized_img_name, img_name + ".jpg", content_type='image/jpg')

        # Delete locally stored cartoonized image
        os.system("rm " + cartoonized_img_name)
        cartoonized_img_name = generate_signed_url(output_uri)

    return render_template("index_cartoonized.html",
cartoonized_image=cartoonized_img_name)

if flask.request.files.get('video'):

    filename = str(uuid.uuid4()) + ".mp4"
    video = flask.request.files["video"]
    original_video_path =
os.path.join(app.config['UPLOAD_FOLDER_VIDEOS'], filename)
    video.save(original_video_path)

    modified_video_path =
os.path.join(app.config['UPLOAD_FOLDER_VIDEOS'], filename.split(".")[0] +
"_modified.mp4")

    ## Fetch Metadata and set frame rate
    file_metadata = skvideo.io.ffprobe(original_video_path)
    original_frame_rate = None
    if 'video' in file_metadata:
        if '@r_frame_rate' in file_metadata['video']:
            original_frame_rate =
file_metadata['video']['@r_frame_rate']

    if opts['original_frame_rate']:
        output_frame_rate = original_frame_rate
    else:
        output_frame_rate = opts['output_frame_rate']

    output_frame_rate_number = int(output_frame_rate.split('/')[0])

    #change the size if you want higher resolution :
    #####
    # Recommnded width_resize #
    #####
    #width_resize = 1920 for 1080p: 1920x1080.
    #width_resize = 1280 for 720p: 1280x720.
    #width_resize = 854 for 480p: 854x480.
    #width_resize = 640 for 360p: 640x360.
    #width_resize = 426 for 240p: 426x240.
    width_resize=opts['resize-dim']

    # Slice, Resize and Convert Video as per settings
    if opts['trim-video']:

```

```

#change the variable value to change the time_limit of video
(In Seconds)

time_limit = opts['trim-video-length']
if opts['original_resolution']:
    os.system("ffmpeg -hide_banner -loglevel warning -ss 0 -i
'{}' -t {} -filter:v scale=-1:-2 -r {} -c:a copy
'{}'".format(os.path.abspath(original_video_path), time_limit,
output_frame_rate_number, os.path.abspath(modified_video_path)))
else:
    os.system("ffmpeg -hide_banner -loglevel warning -ss 0 -i
'{}' -t {} -filter:v scale={}:-2 -r {} -c:a copy
'{}'".format(os.path.abspath(original_video_path), time_limit, width_resize,
output_frame_rate_number, os.path.abspath(modified_video_path)))
else:
    if opts['original_resolution']:
        os.system("ffmpeg -hide_banner -loglevel warning -ss 0 -i
'{}' -filter:v scale=-1:-2 -r {} -c:a copy
'{}'".format(os.path.abspath(original_video_path), output_frame_rate_number,
os.path.abspath(modified_video_path)))
    else:
        os.system("ffmpeg -hide_banner -loglevel warning -ss 0 -i
'{}' -filter:v scale={}:-2 -r {} -c:a copy
'{}'".format(os.path.abspath(original_video_path), width_resize,
output_frame_rate_number, os.path.abspath(modified_video_path)))

audio_file_path = os.path.join(app.config['UPLOAD_FOLDER_VIDEOS'],
filename.split(".")[0] + "_audio_modified.mp4")
os.system("ffmpeg -hide_banner -loglevel warning -i '{}' -map 0:1 -
vn -acodec copy -strict -2 '{}'".format(os.path.abspath(modified_video_path),
os.path.abspath(audio_file_path)))

if opts["run_local"]:
    cartoon_video_path =
wb_cartoonizer.process_video(modified_video_path, output_frame_rate)
else:
    data_uri = upload_blob("processed_videos_cartoonize",
modified_video_path, filename, content_type='video/mp4',
algo_unique_key='cartoonizeinput')
    response = api_request(data_uri)
    # Delete the processed video from Cloud storage
    delete_blob("processed_videos_cartoonize", filename)
    cartoon_video_path = download_video('cartoonized_videos',
os.path.basename(response['output_uri']),
os.path.join(app.config['UPLOAD_FOLDER_VIDEOS'], filename.split(".")[0] +
"_cartoon.mp4"))

## Add audio to the cartoonized video
final_cartoon_video_path =
os.path.join(app.config['UPLOAD_FOLDER_VIDEOS'], filename.split(".")[0] +
"_cartoon_audio.mp4")

```

```

        os.system("ffmpeg -hide_banner -loglevel warning -i '{}' -i '{}' -
codec copy -shortest '{}'.format(os.path.abspath(cartoon_video_path),
os.path.abspath(audio_file_path), os.path.abspath(final_cartoon_video_path)))

        # Delete the videos from local disk
        os.system("rm {} {} {}".format(original_video_path,
modified_video_path, audio_file_path, cartoon_video_path))

        return render_template("index_cartoonized.html",
cartoonized_video=final_cartoon_video_path)

    except Exception:
        print(traceback.print_exc())
        flash("Our server hiccuped :/ Please upload another file! :)")
        return render_template("index_cartoonized.html")
    else:
        return render_template("index_cartoonized.html")

if __name__ == "__main__":
    # Commemnt the below line to run the Appication on Google Colab using ngrok
    if opts['colab-mode']:
        app.run()
    else:
        app.run(debug=False, host='127.0.0.1', port=int(os.environ.get('PORT',
8080)))

```

## CARTOONIFY.PY

```

import os
import uuid
import time
import subprocess
import sys

import cv2
import numpy as np
import skvideo.io
try:
    import tensorflow.compat.v1 as tf
except ImportError:
    import tensorflow as tf

import network
import guided_filter

class WB Cartoonize:
    def __init__(self, weights_dir, gpu):
        if not os.path.exists(weights_dir):

```

```

        raise FileNotFoundError("Weights Directory not found, check path")
    self.load_model(weights_dir, gpu)
    print("Weights successfully loaded")

def resize_crop(self, image):
    h, w, c = np.shape(image)
    if min(h, w) > 720:
        if h > w:
            h, w = int(720*h/w), 720
        else:
            h, w = 720, int(720*w/h)
    image = cv2.resize(image, (w, h),
                       interpolation=cv2.INTER_AREA)
    h, w = (h//8)*8, (w//8)*8
    image = image[:h, :w, :]
    return image

def load_model(self, weights_dir, gpu):
    try:
        tf.disable_eager_execution()
    except:
        None

    tf.reset_default_graph()

    self.input_photo = tf.placeholder(tf.float32, [1, None, None, 3],
name='input_image')
    network_out = network.unet_generator(self.input_photo)
    self.final_out = guided_filter.guided_filter(self.input_photo, network_out,
r=1, eps=5e-3)

    all_vars = tf.trainable_variables()
    gene_vars = [var for var in all_vars if 'generator' in var.name]
    saver = tf.train.Saver(var_list=gene_vars)

    if gpu:
        gpu_options = tf.GPUOptions(allow_growth=True)
        device_count = {'GPU':1}
    else:
        gpu_options = None
        device_count = {'GPU':0}

    config = tf.ConfigProto(gpu_options=gpu_options, device_count=device_count)

    self.sess = tf.Session(config=config)

    self.sess.run(tf.global_variables_initializer())
    saver.restore(self.sess, tf.train.latest_checkpoint(weights_dir))

def infer(self, image):
    image = self.resize_crop(image)

```

```

batch_image = image.astype(np.float32)/127.5 - 1
batch_image = np.expand_dims(batch_image, axis=0)

## Session Run
output = self.sess.run(self.final_out, feed_dict={self.input_photo:
batch_image})

## Post Process
output = (np.squeeze(output)+1)*127.5
output = np.clip(output, 0, 255).astype(np.uint8)

return output

def process_video(self, fname, frame_rate):
    ## Capture video using opencv
    cap = cv2.VideoCapture(fname)

    target_size = (int(cap.get(3)),int(cap.get(4)))
    output_fname = os.path.abspath('{}-{}-
{}.mp4'.format(fname.replace(os.path.basename(fname),
''),str(uuid.uuid4())[:7],os.path.basename(fname).split('.')[0]))

    out = skvideo.io.FFmpegWriter(output_fname, inputdict={'-r':frame_rate},
outputdict={'-r':frame_rate})

    while True:
        ret, frame = cap.read()

        if ret:

            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

            frame = self.infer(frame)

            frame = cv2.resize(frame, target_size)

            out.writeFrame(frame)

        else:
            break
    cap.release()
    out.close()

    final_name = '{}final_{}'.format(fname.replace(os.path.basename(fname),
''), os.path.basename(output_fname))

    p = subprocess.Popen(['ffmpeg', '-i', '{}'.format(output_fname), "-pix_fmt",
"yuv420p", final_name])
    p.communicate()
    p.wait()

    os.system("rm "+output_fname)

```



```
        return final_name

if __name__ == '__main__':
    gpu = len(sys.argv) < 2 or sys.argv[1] != '--cpu'
    wbc = WB_Cartoonize(os.path.abspath('white_box_cartoonizer/saved_models'), gpu)
    img = cv2.imread('white_box_cartoonizer/test.jpg')
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    cartoon_image = wbc.infer(img)
    import matplotlib.pyplot as plt
    plt.imshow(cartoon_image)
    plt.show()
```





## **7. FEATURES OF X :**

This project work is done on the windows 10 professional, which is the operating system. An operating system is a set of software tools designed to make it easy for people or programmers to make optimum use of the computer. People who use computers have different levels of needs and interest. These peoples can be separated can be two groups, users and programmers. The user wants a convenient set of commands to manage files of data or programs, copy and run application package while a programmer used as a set of tools that can be held together and debug programs.

No matter where you are working, your computer will be easier to use and manage, because Microsoft Windows 10 Professional is more compatible and more powerful than any workstation you've used before. The main features of Windows 10 Professional operating system are

- Easier to use.
- Easier to manage
- More compatible
- More powerful

### **EASIER TO USE**

With Windows 10 Professional, you have faster access to information, and you are able to accomplish tasks more quickly and easily.

Windows 10 Professional makes it easier to:

- Work with files.
- Find information.
- Personalize your computing environment.
- Work on the web.
- Work remotely

### **EASIER TO MANAGE**

You and your network administrators can work more efficiently now, because many of the most common computer-management tasks are automated are streamlined with Windows 10 Professional.

With Windows 10, your workstation will be easier to:

- Set up.
- Administrator
- Support.

### **MORE COMPATIBLE:**

Windows 10 Professional offers increased compatibility with different types of network and with a wide array of legacy hardware and software.

Windows 10 also provides:

- Improved driver support.
- Increased support for new-generation hardware multimedia technologies.

### **MORE POWERFUL:**

For all your computing needs, Windows 10 Professional provides:

- Industrial-strength reliability
- The highest level of security
- Powerful performance

Windows 10 also contains the following features:

### **PORTABILITY:**

- Windows file protection protects core system files from being overwritten by application installs.
- Driver certification provides safeguards to assure you that device drivers have not been tampered with and reduces your risk of installing non-certified drivers.
- Full 32 bit operating system minimizes the chance of application failures and unplanned reboots.

**Image MOBILITY**

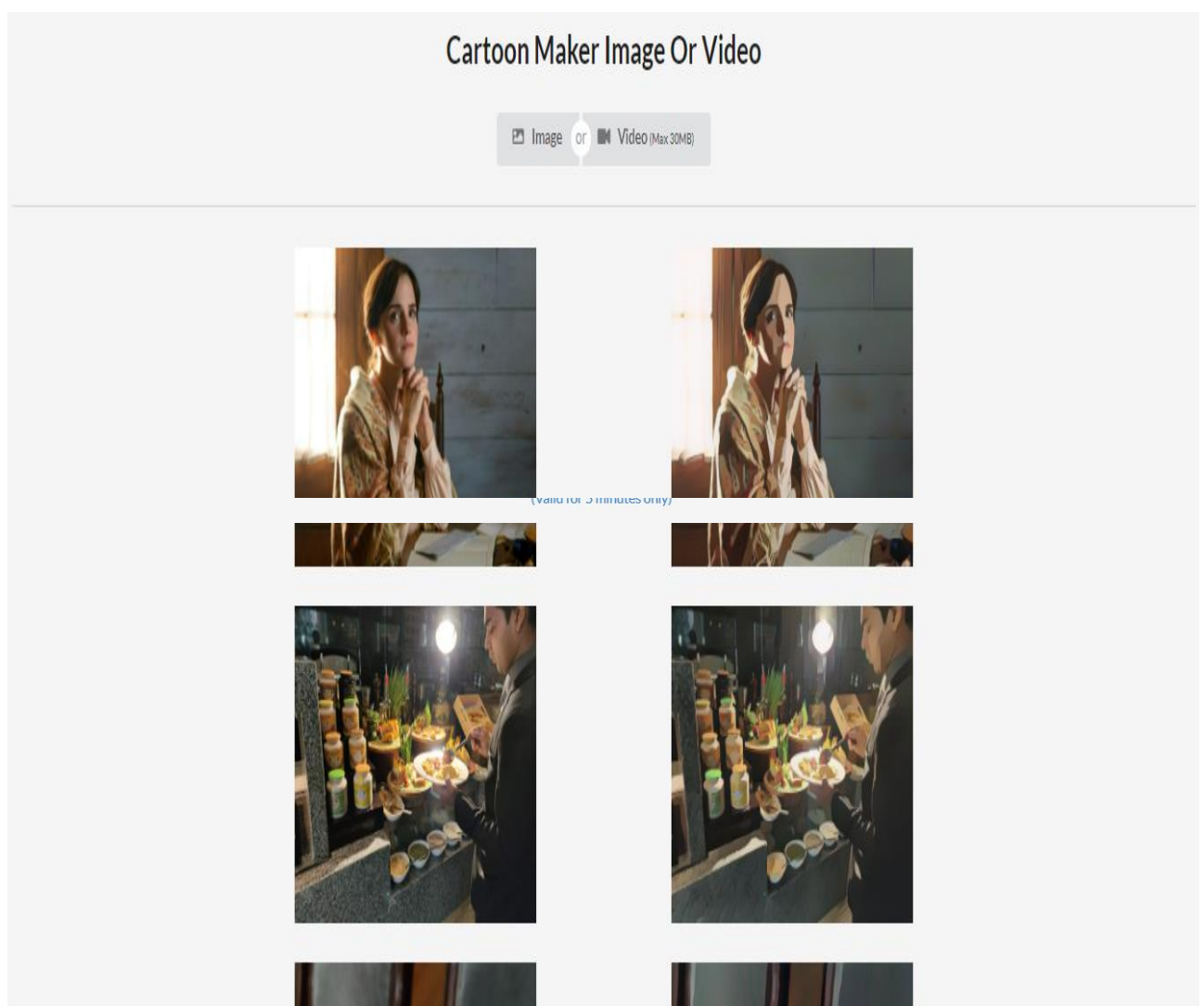
- Hibernate turns off your computer and monitors after a predetermined time while retaining your desktop on disk.
- Offline viewing makes entire WebPages with graphics available for viewing offline
- Synchronization manager allows you to compare and update your offline files and folders with those on the network.
- Smart battery gives you a more accurate view of your battery's life enabling you to reduce power to specify functions to extend your battery power.
- Hot docking tells you dock or undock your notebook computer without changing hardware configuration or rebooting.
- Universal Serial Bus (USB) lets you connect and disconnect a wide array of peripherals such as joysticks, scanners and camcorders without configuring or rebooting your computer.

**MAINTAINABILITY**

- System preparation tool (sys prep) helps administrators clone computer configuration systems and applications.
- Set up manager provides a graphical wizard that guides administrators in designing installation scripts.
- Multilingual support allows users to easily create, read and edit documentation in hundreds of languages.
- Windows 10 server offers 25% faster performance
- Windows 10 can support to 4GB of Ram and two symmetric multiprocessors.
- Encrypting file system (EFS) encrypts each file with a randomly generated key.
- IP Security (IP Sec) support protected data transmitted across a network.
- Kerberos support provides industry standard high-strength authentication with a fast, single login to windows 10 enterprise resources.

**Image INTERNET CAPABILITY**

- Internet Information Services (IIS) 5.0 includes web and FTP server support, as well as support for Front-page transactions, Active Server Pages (ASP) and database connections.
- Windows 10 has strong development platform support for dynamic HTML behaviors and XML.
- Intelliforms alleviates the tedious of filling out forms on the web by automatically entering your name, address or other information that you have securely stored on your computer.
- Automated proxy automatically locates a proxy server configures Internet Explorer 5.0 to connect to the internet through the server.

**8. Screen Shots**

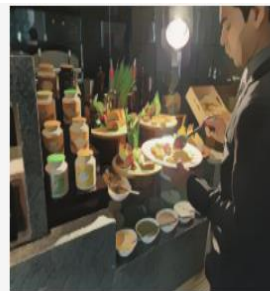
# Cartoon Maker Image Or Video

 Image or  Video (Max 30MB)



 Download

(Valid for 5 minutes only)















## 9. Testing :

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified?. Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Validation : Are we doing the right job?

Verification : Are we doing the job right?

Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are

- Testing is a process of executing a program with the intend of findingan error.
- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncovered errors in the software also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## **9.1 TEST PLAN**

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with

letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

### **9.1.1 UNIT TESTING**

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.



## 10. Conclusion :

Thus we have shown that how image can be converted to cartoon. We also stated the examples on how image is converted to cartoon. Hardware and software requirements of image to cartoon conversion are also shown in this paper. The systematic working of image to cartoon conversion and respective algorithm and formulae is shown with neat diagram in this paper. It is capable of producing exactly the effect specified above, and a wide range of input images will yield satisfactory results. However, the algorithm is not perfect, and it does not respond predictably across all inputs. Given the variety of input images, it would be unrealistic to expect a one-size-fits-all approach to produce consistent results. In the future, an adaptive algorithm would probably be better suited to providing a consistent effect. Such an algorithm would recognize image types (portrait, indoor scene, outdoor scene for instance) and then change the edge detection parameters to better suit that particular image. Also we have stated challenges and problems one can face while cartoonifying the captured image. In this paper we have also discussed need and scope of cartoonifying the content image.

## **2. Scope for Future Enhancement:**

User will be provided with a set of pretrained style images to choose from. Based on the chosen style and the content image provided by the user, the Resulting image with cartoon like effect is generated by the program. The implementation is based on of the combination of Gatys' A Neural Algorithm of Artistic Style, Johnson's Perceptual Losses for Real-Time Style Transfer and SuperResolution, and Ulyanov's Instance Normalization

### **ONLINE REFERENCE:**

- [www.wikipedia.com](http://www.wikipedia.com)
- [www.stackoverflow.com](http://www.stackoverflow.com)
- [www.tensorflow\[documentation.com](http://www.tensorflow[documentation.com)
- [www.geeksforgeeks.com](http://www.geeksforgeeks.com)
- [www.opencvdocumentation.com](http://www.opencvdocumentation.com)

