**AI MINI PROJECT**

**TOPIC: - Cartoonify an Image**

**TEAM MEMBERS: -**

**Hitik Sharma (RA1911003010863)**

**Ayushi Kumari (RA1911003010865)**

**Shreyansh Kumar Gupta (RA1911003010878)**

**Problem Statement: -**

To cartoonify an Image with OpenCV in Python

**IDE used**: VS Code

**Objective with Technical Depth: -**

The main aim of this project is to cartoonify the image provided by the user to the application. We have used various functionality of Python in OpenCV to implement the cartoonify effect.

Python functionalities such as Conversion of image to greyscale image for smooth conversion to cartoon effect.

We have imported the following modules:

* **OpenCV:**

OpenCV is a cross-platform library mainly aimed at real-time Computer Vision, written in C/C++, released in June 2000. It includes applications like video and image capturing and processing. It is majorly used in image transformation, object detection, face recognition, and many other amazing applications. It is extensively being used in Machine Learning, Artificial Intelligence, Internet of Thing, etc.

CV2 is imported to use OpenCV for image processing.

* **NumPy:**

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy is a Python package. It stands for 'Numerical Python'.

Images are stored and processed as numbers. These are taken as arrays. We use NumPy to deal with arrays.

* [**Tkinter**](https://tkdocs.com/)**:**

It is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python. The name Tkinter comes from Tk interface.

**AI/ML Algorithm used: -**

OpenCV ml module assumes that, each feature can be ordered

(i.e. its values are floating-point numbers that can be compared with each other and strictly ordered, i.e. sorted)

Or categorical (i.e. its value belongs to a fixed set of values that can be integers, strings etc.).

* **Normal Bayes Classifier:**

This simple classification model assumes that feature vectors from each class are normally distributed (though, not necessarily independently distributed). So, the whole data distribution function is assumed to be a Gaussian mixture, one component per class. Using the training data the algorithm estimates mean vectors and covariance matrices for every class, and then it uses them for prediction.

* **K-Nearest Neighbours:**

The algorithm caches all training samples and predicts the response for a new sample by analysing a certain number (K) of the nearest neighbours of the sample using voting, calculating weighted sum, and so on. The method is sometimes referred to as "learning by example" because for prediction it looks for the feature vector with a known response that is closest to the given vector.

* **Support Vector Machines:**

Originally, support vector machines (SVM) were a technique for building an optimal binary (2-class) classifier. Later the technique was extended to regression and clustering problems. SVM is a partial case of kernel-based methods. It maps feature vectors into a higher-dimensional space using a kernel function and builds an optimal linear discriminating function in this space or an optimal hyper- plane that fits into the training data. In case of SVM, the kernel is not defined explicitly. Instead, a distance between any 2 points in the hyper-space needs to be defined.

The solution is optimal, which means that the margin between the separating hyper-plane and the nearest feature vectors from both classes (in case of 2-class classifier) is maximal. The feature vectors that are the closest to the hyper-plane are called support vectors, which means that the position of other vectors does not affect the hyper-plane (the decision function).

**Functions Used:**

1. **askopenfilename():** While working with GUI one may need to open files and read data from it or may require to write data in that particular file. One can achieve this with the help of open () function (python built-in) but one may not be able to select any required file unless provides a path to that particular file in code.

With the help of GUI, you may not require to specify the path of any file but you can directly open a file and read its content.

1. **imread(photo):** cv2.imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.
2. **cvtColor(img, cv2.COLOR\_BGR2GRAY):** cv2.cvtColor() method is used to convert an image from one color space to another. There are more than 150 color-space conversion methods available in OpenCV.
3. **medianBlur(grey, 5):** The Median blur operation is similar to the other averaging methods. Here, the central element of the image is replaced by the median of all the pixels in the kernel area. This operation processes the edges while removing the noise.
4. **adaptiveThreshold(grey, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C, cv2.THRESH\_BINARY, 9, 9):** Adaptive thresholding is the method where the threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions.

In OpenCV, you can perform Adaptive threshold operation on an image using the method adaptiveThreshold() of the Imgproc class.

1. **bilateralFilter(img, 9, 250, 250):** Bilateral filter is used for smoothing the image.
2. **bitwise\_and(color, color, mask = edges):** bitwise and is used for applying the edges over the smoothen image.
3. **imshow("Image", img):** cv2.imshow() method is used to display an image in a window. The window automatically fits to the image size.

1. **imwrite("cartoon.jpg", cartoon):** cv2.imwrite() method is used to save an image to any storage device. This will save the image according to the specified format in current working directory.
2. **waitKey(0):**  waitkey() function of Python OpenCV allows users to display a window for given milliseconds or until any key is pressed. It takes time in milliseconds as a parameter and waits for the given time to destroy the window, if 0 is passed in the argument it waits till any key is pressed.
3. **destroyAllWindows():** Python Opencv destroyAllWindow() function allows users to destroy all windows at any time. It doesn’t take any parameters and doesn’t return anything. It is similar to destroyWIndow(). destroyWindow() only destroys a specific window but in the case of destroyAllWindow() it destroys all windows.

**Execution and GitHub upload: -**

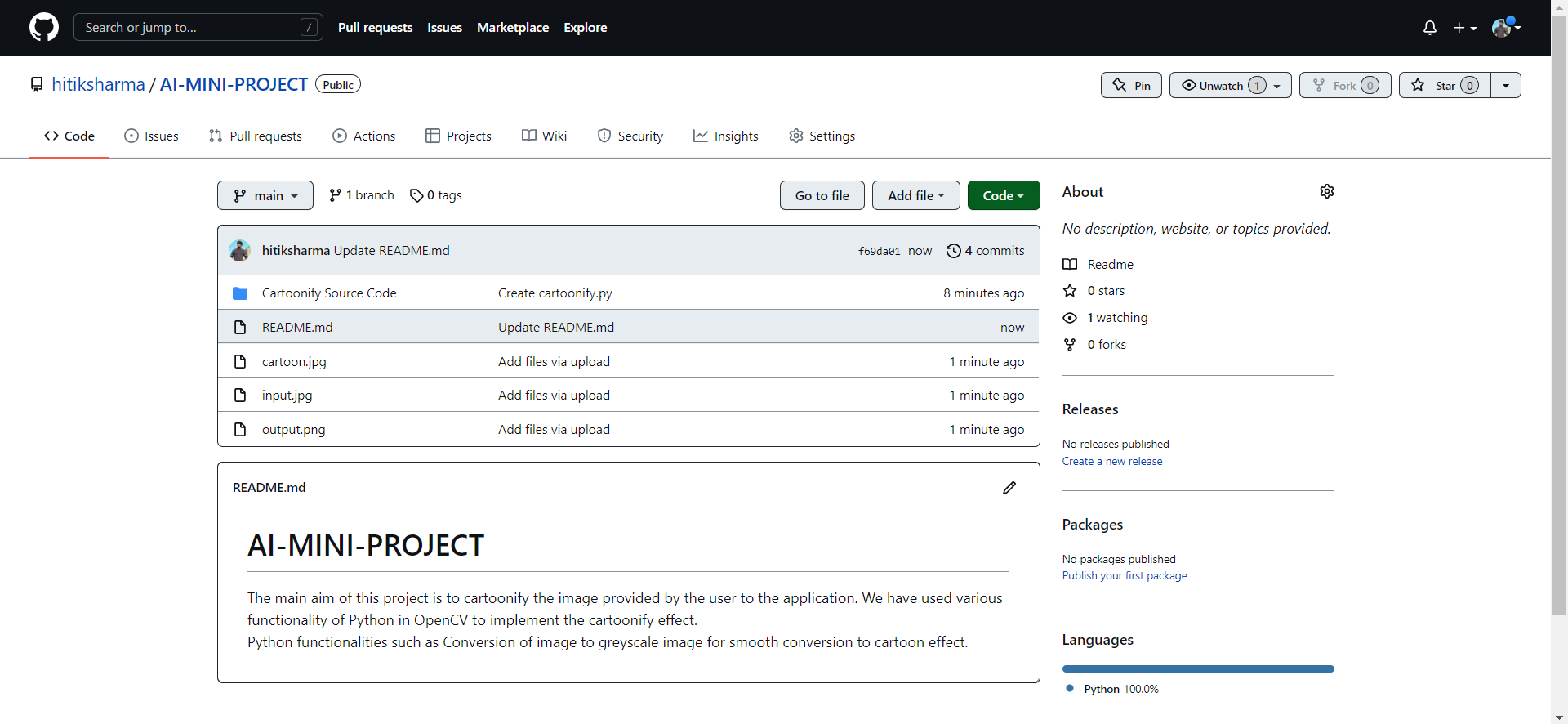
To convert an image to a cartoon, multiple transformations are done: -

* Convert the image to a Grayscale image.
* The Grayscale image is smoothened.
* Extract the edges in the image.
* Form a colour image and mask it with edges.

This creates a beautiful cartoon image with edges and lightened colour of the original image.

**Github Link:**

**https://github.com/hitiksharma/AI-MINI-PROJECT**





**Source Code: -**

import cv2

import numpy as np

from tkinter.filedialog import \*

photo = askopenfilename()

img = cv2.imread(photo)

grey = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

grey = cv2.medianBlur(grey, 5)

edges = cv2.adaptiveThreshold(grey, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C, cv2.THRESH\_BINARY, 9, 9)

#cartoonize

color = cv2.bilateralFilter(img, 9, 250, 250)

cartoon = cv2.bitwise\_and(color, color, mask = edges)

cv2.imshow("Image", img)

cv2.imshow("Cartoon", cartoon)

#save

cv2.imwrite("cartoon.jpg", cartoon)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Purpose of the Problem Statement (societal benefit): -**

* **Social Media Outburst:** People can cartoonify there image and upload it on various social media Handles (such as Instagram, Facebook) and this makes them socially active and they feel independent and open to the society.
* **Personality Development:** Using this people can be more open to the world and this can help them build their Professional skills that is people get more interactive to other people around and develop their personality.
* Nowadays, In the invitation cards and poster cards etc... often has cartoonify images in them as it gives modern look.
* Cartoonify an image using adobe photoshop is a very tedious task which our project solves within few lines of code using various Artificial Intelligence algorithms.
* Our project here provides a cartoon reference which makes the creation of digital art very easy.
* Our project can be used to create Caricature art easily which is nowadays used by Bollywood magazines and news outlets.