**Mini Project Report on**



**CARTOONIFY IMAGE USING MACHINE LEARNING**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name**  **University Roll No.**

**Sohard Behl 2019151**

***Under the Mentorship of***

**Dr. Vikas Tripathi**

**Associate Dean (R&D)**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January 2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Cartoonify Image using Machine Learning”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Vikas Tripathi, Associate Dean (R&D)**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name University Roll no **signature**

Sohard Behl 2019151

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1** |
| Chapter 2 | Literature Survey | **4** |
| Chapter 3 | Methodology | **6** |
| Chapter 4 | Result and Discussion | **9** |
| Chapter 5 | Conclusion and Future Work | **11** |
|  | References | **12** |

**Chapter 1**

**Introduction**

This mini project is based on an application of Machine Learning i.e., Cartoonifying image. It makes use of different machine learning tools to perform the processes that are further discussed in the report.

* 1. **Introduction**

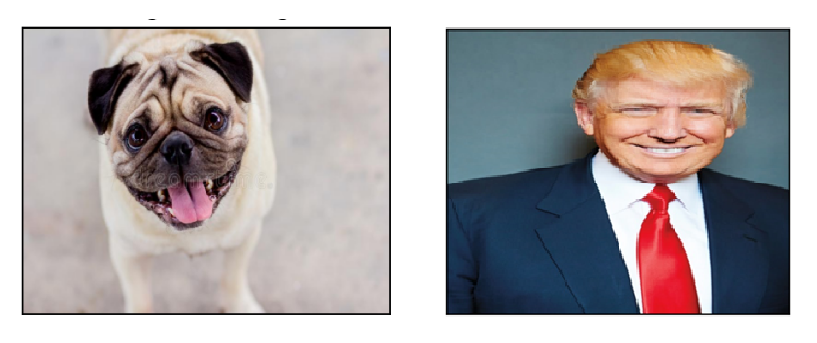
In this project of Cartoonify image using ML, a very interesting application of Machine Learning comes into play. This project makes use of Python as the programming language to develop this application. Furthermore, we use OpenCV (Open Computer Vision) which is a machine learning tool to execute different pre-defined processes on images and videos. This is the most essential library of this application. We also learn to use different libraries like matplot, numpy, easygui, tkinter in the midst of building this project. All of these libraries and the tools that they provide are used to convert our input image into its cartoon output. We will now discuss the requirements of this project in detail.

* 1. **Requirements**

The requirements of the application are discussed below. All of these include the libraries that need to be installed for creating this project.

1. Python
2. OpenCV
3. Matplotlib
4. Tkinter
5. Easygui
6. Numpy
7. OS
   * 1. **Python:** Python is used as the programming language to build this application. We import all the required libraries using python and all development is done on python interpreter. Python language is very much preferred among other programming languages in applications that involve machine learning and artificial intelligence and . It is widely used as it is portable, open source, provides features like ease-to-code, ease-to-read, etc.
     2. **OpenCV:** OpenCV stands for Open Computer Vision. It is used mainly used in the areas of machine learning and artificial intelligence. It implements a discipline known as computer vision that involves understanding 3d with help of 2d images that are provided to it. There are various features of OpenCV library like reading, writing images, videos, feature detection, motion estimation, processing on images and videos, etc. It is made for different platforms like linux, windows, macOS and prominently used via python, java, C++.
     3. **Matplotlib:** Matplotlib is used in executing different tasks. It is used to visualize data in form of arrays, matrices, etc. It can plot graphs, piecharts, histograms, line graphs and many more. It is built on numpy array and can generate complex grids and subplots in a semantic way.
     4. **Tkinter:** Tkinter is a python library used for GUI (Graphical User Interface). It is very commonly used among different GUI methods and it is the only framework completely built in python. It is represented as “tk” in programs. It facilitates a swift and easy way to create GUI applications.
     5. **Easygui:** Used for GUI programming and can be invoked by simple function calls. It is not event-driven like other GUI libraries and generators.
     6. **Numpy:** It is alibrary used prominently in various applications of python. Numpy helps in working over data in the form of matrices, arrays, linear algebra and many more. In our project, it is used in converting images into array format to make the processing over the images easy. It is used in-sync with matplotlib in our application.
     7. **OS:** Used for handling paths like reading images and saving images in our project.
   1. **Sample Input and Output images**

The following is a representation of an image that is inputted in our application and how the application processes the image and provides us the final output image after all the inbuilt definitions that are made in the project. An important point to be noted is that the application of our project depends majorly on the quality of image that is fed to it and also its resolution and aspect ratio. It converts image into its cartoonified version and utilizes machine learning to produce an output that resembles the initial image but changes different features of the image. Here we have an input image and then the output image that is derived from the application.



**Figure 1.1 Original image**



**Figure 1.1 Final image**

**Chapter 2**

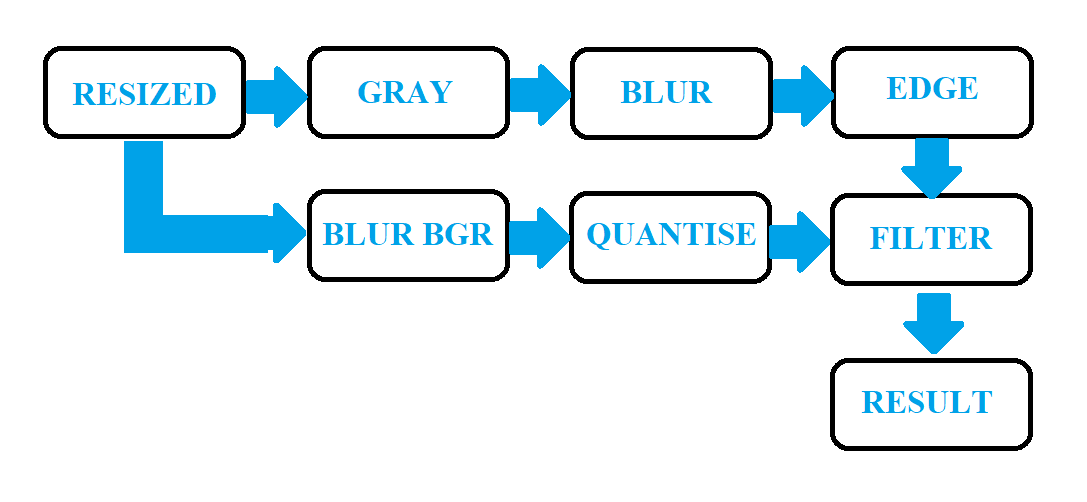
**Literature Survey**

* It is very important to analyze data in modern world where everything revolves around data. Large amount of data is available everywhere and there is a need of developing algorithms based on the data analysis. All of this is achieved by machine learning and data mining. Machine learning is used to design data algorithms that rely on relationships between data and data trends. Machine learning is implemented and used in various fields like image processing, deconvulation, intrusion detection, retrieving information, gaming, malware detection, marketing, bioinformatics and many more. [1]
* In recent years, computer applications have adapted to machine learning from single data processing. All of this is due to availability of massive amount of data. There are 3 fundamental categories of machine learning namely, supervised, unsupervised and reinforcement. Computer vision, neural networks, deep learning. [2]
* OpenCV can be used to perform different applications like reading and writing images and videos, performing feature detection, capturing and saving videos, different processing over images like filtering, masking, transformations, scaling, changing color spaces. Detection of objects both living and non-living is included in OpenCV. Analyze video like estimation of motion, subtracting background, tracking objects. Live video applications. [3]
* Image segmentation is one of the complicated works than other vision techniques in computer vision. Deep learning has impacted this field and this deep learning that is associated with GAN (Generated Adverserial Networks) provides us with remarkable outcomes on image segmentation. Image processing, texture transfer, face detection, 3d object generation, traffic controlling are major applications of GAN. It is developed using deep learning and can be considered as a secondary field of ML algorithms. Deep Neural Networks (DNNs) are used in conditional GANs. [4]
* There are 3 white-box representations from images that are proposed by the authors using GAN for cartoonization of images. This white box technique requires to upload an image and then perform various GAN techniques that have been developed to impart cartoon effects on the inputted image. These representations are namely, surface representation, structure representation and texture representation. [5]
* Tailored tools are being employed in developing methods for debugging and refining models in Machine Learning. All of this is very much essential in development for creating black box models like deep neural networks and random forests. It is beneficial for partitions that are subjected to data visualization and engineering. A developer can find partitions and mis predicted instances using these techniques. [6]

**Chapter 3**

**Methodology**

There is a series of methods that are implemented in this project regarding cartoonifying image. The methods are developed and tested on a lot of samples specifically for this application in order to ensure its swift functioning. The methodology is explained briefly in form of flowchart below:



By the above flowchart, we can further explore all the methods that are involved in the development of this application.

**3.1 Importing libraries**

We will first import all the libraries that are required in building the project. The libraries that are required have been listed in chapter 1 above.

**3.2 Defining upload function**

Upload function will be defined that will store the path of the image using easygui library. The stored path of the image will be passed as an argument to Cartoonify function. Upload function will work simultaneously with filedialog tool of tkinter library.

**3.3 Defining Cartoonify function**

The Cartoonify function is divided into a number of smaller sections or methods that have been already declared in the flowchart.

**3.3.1 Reading:** We read the image from the image path provided. The method is executed by imread function. The read image is stored in img.

**3.3.2 Resizing:** The image that is traced back from the image path provided is by the upload function is resized to dimension of 480x360. The resized image provides a better chance for performing processes over the image. The img is now changed to resize.

The resized image is of BGR (Blue Green Red) color space and now we will work on 2 different directions to obtain the final output.

**3.3.3 Grayscaling:** The resized image is now converted into grayscale image which is achieved by cvtColor function of OpenCV. It requires the source image and the argument that will specify which color space is needed to be achieved.

**3.3.3 Blurring:** The grayscale image is now blurred using GaussianBlur method. There are other blurring techniques that can be used based upon our requirements depending upon the extent of blur but for now blurring can be achieved by Gaussian blur. The arguments required are the source image and the kernel size of the blurring and the standard deviation.

**3.3.3 Detecting edges:** There are various methods that can be used for edge detection like Laplacian, Sobel but we will implement adaptive thresholding method in our application to achieve the desired output. This output will be used in the further processing of the image.

**3.3.4 Colored blurring:** Wewill make use of the original colored resize image and implement medianblur technique that requires the source image and kernel size as arguments.

**3.3.5 Quantisation:** Quantisation in Computer Vision generally refers to inculcating different color spaces and color channels and organizing the same palettes together as a cluster. We have defined a separate function to perform the quantisation for us. For this we convert our image into an array of numbers using numpy library. We then perform our operations like iterating the array and providing accuracy to its operation. The array is then again converted into image and returned. Higher the value of iteration, more colors will be visible in the image.

**3.3.5 Filtering:** Thequantized image is now filtered using bilateralfilter method. We pass the source image and the required kernel sizes as arguments. Higher the kernel size, more filteration will be done. Filter refers to smoothening in layman terms.

**3.3.5 Masking:** The filtered image is now masked over by the edged image from our previous method and the output obtained is our final cartoonified image.

**3.4 Plotting the images**

The images obtained are now plotted using matplotlib which is also a machine learning tool as discussed in the above-mentioned chapters. Since, matplotlib uses RGB format so we first convert all of our images to RGB format from BGR format. We then use subplot function to plot all images into a more symmetrical format.

**3.5 Pop-up menu and Dialog box**

Now we make use of another machine learning tool that is tkinter library which facilitates the designing of the dialog box that pops up for selecting the image. It works in sync with the upload function.

**3.6 Important Terminologies**

**3.6.1 Kernel size:** It is defined as a specific portion of an image and the blurring techniques are applied across the kernel sizes. Kernel include pixels and the effect takes place on the center pixel based upon the type of blurring like gaussian, median, average.

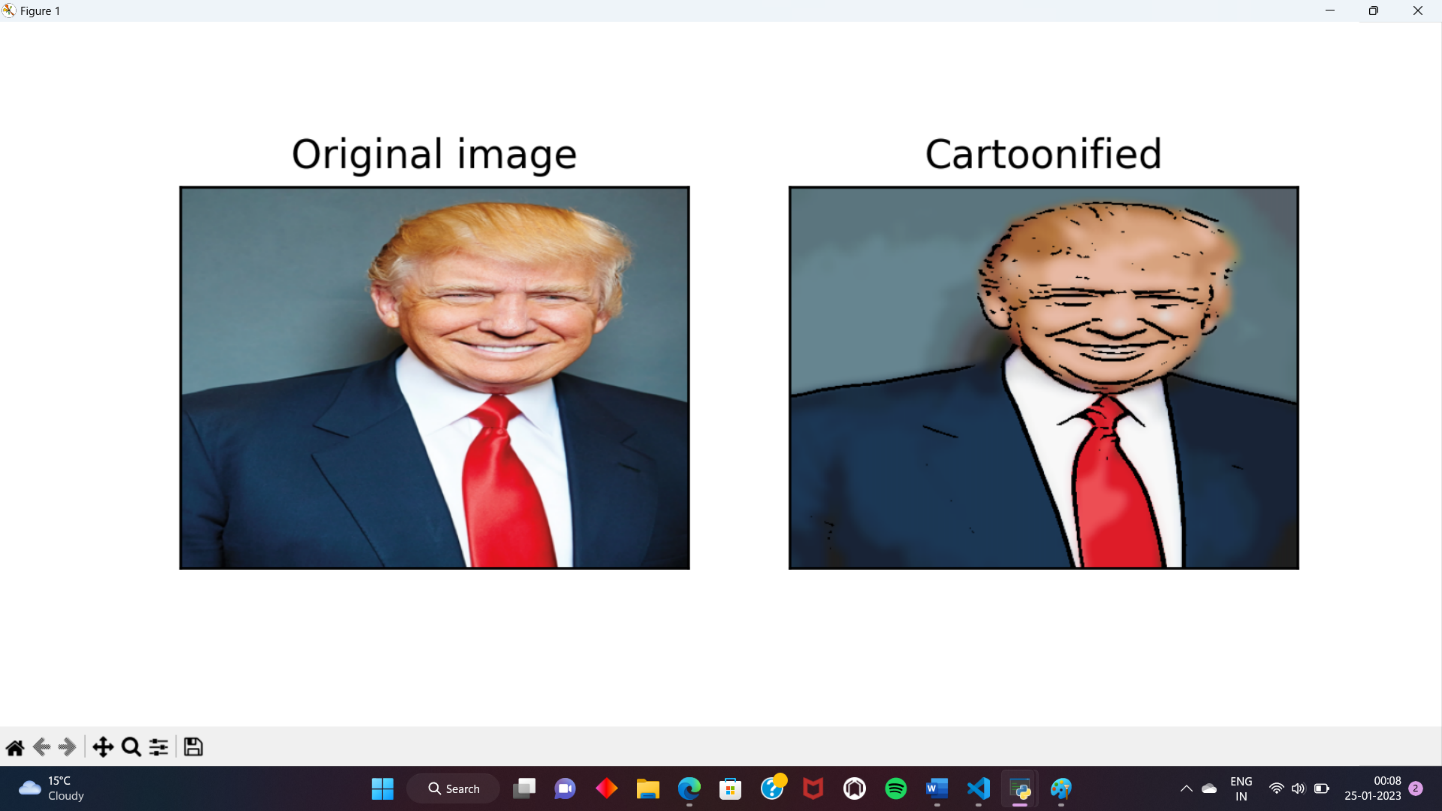
**Chapter 4**

**Result and Discussion**

The result and discussion based on the inferences is done in the chapter below.

**4.1 Input image and Cartoonified image**

This output window is the first subplot done by matplot library. It shows the inputted image and its cartoonified output.



**4.2 Intermediate steps**

All the methods that are used in the application to provide the final output that constitute of the intermediate steps are plotted below.

****

**4.3 Inference and Discussion**

The intermediate steps clearly show the methods that were discussed above in methodology. All those methods come together to provide a cartoonified version of the initially inputted image, the masked image being the final output.

One important point to be kept in mind while uploading or selecting an image for undergoing cartoonification in the application developed under this project is that it needs to be of a significantly good resolution and of a proper aspect ratio.

****

The final images received from the 2 steps are masked over each other via bitwise and operation to get the final image.

Moreover, we learnt about different blurring techniques like average blurring, median blurring, gaussian blurring and bilateral filtering. There are various thresholding and edge detection methods also provided by the OpenCV as well.

We also learnt about kernels, eroding, contours, histogram computations and other essential terminologies involved in Computer Vision.

**Chapter 5**

**Conclusion and Future Work**

**5.1 Conclusion**

After all the Machine Learning Techniques and tools that are used in this project, we are able to conclude that this project of cartoonifying an image using Machine Learning is working well and provides favourable output. We made use of Machine Learning tools like OpenCV, Matplotlib, Tkinter to execute our methodologies and techniques in achieving our goal of this project. We also learnt about different terminologies involved in the techniques discussed and also went through the research work of many authors regarding the same field of interest.

**5.2 Future Work**

For future work on this project, I would like to make it much more interactive with the user and provide other options as per the user’s requirements. It can also be developed into being adaptive to live video feed and work on it in Realtime as well. Along with that Generated Adverserial Networks (GANs) can also be inculcated within this project.

**References**

[1] S. Angra and S. Ahuja, "Machine learning and its applications: A review," 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, Andhra Pradesh, India, 2017, pp. 57-60, doi: 10.1109/ICBDACI.2017.8070809.

[2] S. V. Mahadevkar et al., "A Review on Machine Learning Styles in Computer Vision—Techniques and Future Directions," in IEEE Access, vol. 10, pp. 107293-107329, 2022, doi: 10.1109/ACCESS.2022.3209825.

[3] <https://www.tutorialspoint.com/opencv/opencv_overview.htm>

[4] Alankrita Aggarwal, Mamta Mittal, Gopi Battineni, Generative adversarial network: An overview of theory and applications, International Journal of Information Management Data Insights, Volume 1, Issue 1, 2021, 100004, ISSN 2667-0968,

https://doi.org/10.1016/j.jjimei.2020.100004.

(<https://www.sciencedirect.com/science/article/pii/S2667096820300045>)

[5] X. Wang and J. Yu, "Learning to Cartoonize Using White-Box Cartoon Representations," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 8087-8096, doi: 10.1109/CVPR42600.2020.00811.

[6] P. Rasouli and I. C. Yu, "Explainable Debugger for Black-box Machine Learning Models," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-10, doi: 10.1109/IJCNN52387.2021.9533944.

[7]<https://www.udemy.com/share/101WfW3@_ET_QxehgxOf7iTQfQuFsmPRR7KLXS28lFBR9oLJgOLXc-X-pp_oOkq8CiHNUWNXVQ==/>