**Image Style Conversion using OpenCV**

**A Mini Project Report submitted to MOHAN BABU UNIVERSITY**

**in Partial Fulfillment of the Requirements for the Award of the degree of**

#### BACHELOR OF TECHNOLOGY IN

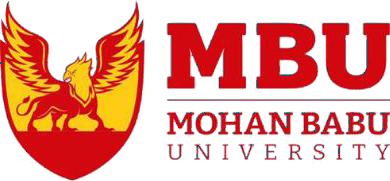
**COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)**

*Submitted by*

|  |  |
| --- | --- |
| **M CHANDANA REDDY** | **22102A030220** |
| **M ARSHIYA** | **22102A030221** |
| **M PAVAN BALA SUBASH REDDY** | **22102A030222** |
| **M RAMBANNI** | **22102A030223** |
| **N DURGA SREE** | **22102A030224** |

*Under the Guidance of*

**Mrs.Ch. Prathima** Assistant Professor Department of Data Science



**Department of Data Science SCHOOL OF COMPUTING**

MOHAN BABU UNIVERSITY

Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA 2024-25

MOHAN BABU UNIVERSITY

Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA 2024-25

**DEPARTMENT OF DATA SCIENCE**

**CERTIFICATE**

This is to certify that the mini project report entitled

### “Image Style Conversion using OpenCV”

is the Bonafide work done by

|  |  |
| --- | --- |
| **M CHANDANA REDDY** | **22102A030220** |
| **M ARSHIYA** | **22102A030221** |
| **M PAVAN BALA SUBASH REDDY** | **22102A030222** |
| **M RAMBANNI** | **22102A030223** |
| **N DURGA SREE** | **22102A030224** |

in the Department of **Data Science**, and submitted to Mohan Babu University, Tirupati in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering (Data Science) during the academic year 2024-2025. This work has been carried out under my supervision. The results of this mini project work have not been submitted to any university for the award of any degree or diploma.

***Guide: Head:***

Mrs. Ch. Prathima Dr. P.K. Gupta

Assistant Professor Professor & Head

Dept. of Data Science Dept. of Data Science

**INTERNAL EXAMINER EXTERNALEXAMINER**

# DEPARTMENT OF DATA SCIENCE

**Vision**

To become a Centre of Excellence in Data Science by imparting high quality education through teaching, training and research

# Mission

* To impart quality education in Computer Science and Engineering with specializations in Data Science by disseminating knowledge through contemporary curriculum, competent faculty and effective teaching-learning methodologies.
* Nurture research, innovation and entrepreneurial skills among students and faculty to contribute to the needs of industry and society.
* Inculcate professional attitude, ethical and social responsibilities for prospective and promising Engineering profession.
* Encourage students to engage in life-long learning by creating awareness of the contemporary developments in Computer Science and Engineering with specialization in Data Science.

### PROGRAM EDUCATIONAL OBJECTIVES

After few years of graduation, the graduates of B. Tech. CSE(DS) will:

|  |  |
| --- | --- |
| **PEO1**  **.** | Pursue higher studies in Computer Science, Data science or  Management. |
| **PEO2**  **.** | Become successful entrepreneurs or be employed by acquiring required  skill sets in the domains of Data Science and allied areas. |
| **PEO3**  **.** | Exhibit progression and effective adaptation to technological developments through life-long learning to address ever changing industrial requirements and follow ethical attitude in professional  practice. |

### PROGRAM SPECIFIC OUTCOMES

On successful completion of the Program, the graduates of B. Tech. CSE(DS) program will be able to:

|  |  |
| --- | --- |
| **PSO1.** | Apply appropriate data analytical techniques for building effective  decision-making systems. |
| **PSO2.** | Develop intelligent systems using novel Machine Learning and Artificial Intelligence techniques. |
| **PSO3.** | Design and develop efficient software systems using modern tools, techniques, and platforms to meet societal needs. |
| **PSO4.** | Apply suitable tools and techniques to build secure distributed systems. |

### PROGRAM OUTCOMES

On successful completion of the Program, the graduates of B.Tech. CSE (DS) Program will be able to:

|  |  |
| --- | --- |
| **PO1.** | **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the  solution of complex engineering problems. |
| **PO2.** | **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering  sciences. |
| **PO3.** | **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental  considerations. |
| **PO4.** | **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to  provide valid conclusions. |
| **PO5.** | **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the  limitations. |
| **PO6.** | **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering  practice. |
| **PO7.** | **Environment and sustainability**: Understand the impact of the  professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| **PO8.** | **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |

|  |  |
| --- | --- |
| **PO9.** | **Individual and team work**: Function effectively as an individual, and as  a member or leader in diverse teams, and in multidisciplinary settings. |
| **PO10**  **.** | **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear  instructions. |
| **PO11**  **.** | **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage  projects and in multidisciplinary environments. |
| **PO12**  **.** | **Life-long learning**: Recognize the need for, and have the preparation  and ability to engage in independent and life-long learning in the broadest context of technological change. |

We hereby declare that this project report titled **“IMAGE STYLE CONVERSION USING OPENCV”** is a genuine work carried out by us, in **B.Tech *(Computer Science and Engineering (Data Science))*** degree course of **Mohan Babu University, Tirupati** and has not been submitted to any other course or University for the award of any degree by us.

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the students 1.

2.

3.

4.

5.

6.

In today's digital world, the ability to modify and enhance images through artificial intelligence (AI) techniques has opened up creative and practical possibilities across various fields. This project, titled **"Image Style Conversion Using OpenCV"** focuses on developing an application that transforms ordinary images into artistic versions, such as cartoon styles, pencil sketches, and painting-like effects .Using OpenCV, an open-source computer vision library, this project applies advanced image processing techniques such as bilateral filtering, edge detection, color smoothing, and grayscale transformations. The user can select different artistic styles, and the system dynamically processes the uploaded image based on the selected style. Techniques like adaptive thresholding for sketches and bitwise masking for cartoon effects are implemented to achieve visually appealing results.The goal of the project is not only to explore the capabilities of AI and computer vision but also to provide an easy-to-use platform where users can enhance their photos in creative ways. This project highlights the fusion of AI-driven image manipulation with real-world applications, making artistic photo editing more accessible and intelligent.

Additionally, this project emphasizes the practical use of machine learning concepts such as feature extraction and transformation. By working on real-world images, the model improves the user’s understanding of how AI can interpret, modify, and recreate visual content. Future improvements could involve adding neural style transfer models to create even more sophisticated effects. Overall, this project is a step toward bridging creativity and technology through the power of AI and OpenCV.

**Keywords**: Artificial Intelligence, OpenCV, Image Processing, Style Conversion, Cartoonization,

Pencil Sketching, Feature Extraction, HD Visual Transformation.

## TABLE OF CONTENTS

|  |  |
| --- | --- |
| **Title** | **Page No.** |
| ABSTRACT | 8 |
| **CHAPTER 1: INTRODUCTION** | 10 |
| INTRODUCTION | 10 |
| STATEMENT OF THE PROBLEM | 11 |
| OBJECTIVES | 12 |
| LIMITATIONS | 12 |
| **CHAPTER 2: METHODOLOGY** | 13 |
| **CHAPTER 3: DESIGN** | 14 |
| REQUIREMENTS | 14 |
| SYSTEM DESIGN | 14-15 |
| TECHNOLOGIES USED | 16 |
| **CHAPTER 4: IMPLEMENTATION** | 17 |
| **CHAPTER: 5 PROPOSED SYSTEM ARCHITECTURE** | 18 |
| FLOW CHART | 19 |
| **CHAPTER 6: CODING AND RESULT** | 20 |
| CODE | 20 |
| RESULT | 21-24 |
| **CHAPTER 7: CONCLUSION AND FUTURE WORK** | 25 |
| CONCLUSION | 25 |
| FUTURE WORK | 25 |
| **CHAPTER 8: REFERENCES** | 26 |
| REFERENCES | 26 |

**CHAPTER- 1 INTRODUCTION**

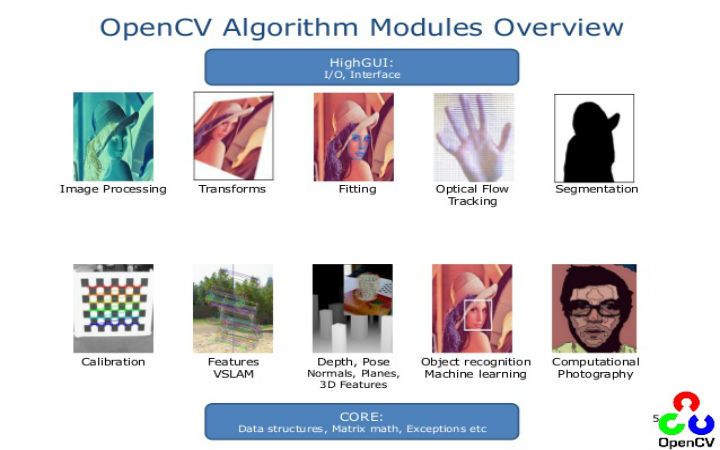
##### Introduction:

In the modern digital era, the demand for creative image editing and transformation has grown significantly across various industries such as entertainment, photography, social media, education, and design. Traditional image editing methods often require expert knowledge and manual effort, which can be time-consuming and expensive. With advancements in Artificial Intelligence (AI) and Computer Vision technologies, it is now possible to automate the process of modifying images with minimal human intervention.

This project, titled "Image Style Conversion using OpenCV," aims to develop an efficient system that can transform standard images into various artistic styles such as cartoon effects, pencil sketches, and high-definition (HD) enhancements. Leveraging **OpenCV**, a powerful open-source computer vision library, the system uses image processing techniques like bilateral filtering, adaptive thresholding, and edge detection to produce visually appealing artistic outputs.

The main goal of the project is to allow users to easily convert their photographs into different styles while maintaining clarity and preserving important details. By utilizing OpenCV, a library known for its ability to handle large datasets and efficiently execute image transformations, this system enables real-time or near real-time style transfer applications. The project demonstrates how AI-based automation can be effectively used to enhance creativity, reduce manual work, and make artistic tools accessible to everyone, even those without professional editing skills.

Through offering a range of style transformations, this project highlights the powerful combination of technology and art, made possible through the integration of computer vision, image processing techniques, and machine learning.



##### Problem Statement:

Traditional image editing tools require significant manual effort and expertise, making them time-consuming and inaccessible to many users. As demand for quick, creative image transformations grows across industries, there is a lack of efficient, automated solutions that allow users to easily convert images into artistic styles like cartoons, sketches, or HD enhancements. Current tools often fail to provide real-time processing or bulk editing capabilities, and the results often lack consistency. Moreover, users face challenges in preserving important image details while applying complex styles. This project, "Image Style Conversion using OpenCV," aims to solve these issues by leveraging image processing techniques such as bilateral filtering and edge detection to automate the style conversion process, making creative image editing more accessible, efficient, and user-friendly for all users. By providing an intuitive and quick solution, this system enables users to explore artistic transformations without needing professional editing skills.

##### Objectives:

* To develop an automated image style conversion system using OpenCV for transforming standard images into various artistic styles.
* To utilize image processing techniques such as bilateral filtering, edge detection, and adaptive thresholding for effective style transformation.
* To maintain image quality and preserve important details during the style conversion process.
* To offer users a range of artistic styles, including cartoon effects, pencil sketches, and HD enhancements.
* To ensure real-time or near real-time processing of images for quick style transformation.
* To make the system user-friendly and accessible to individuals with no professional image editing skills.
* To automate the image style conversion process, reducing the need for manual adjustments and fine-tuning.
* To provide flexibility in applying different styles to images while maintaining consistent output quality.

##### Limitations :

Complexity of Images: The system may not produce optimal results for detailed or intricate

images due to the limitations of current processing techniques.

Processing Time: Handling large or high-resolution images may lead to slower performance,

making real-time processing challenging.

Limited Style Options: The available artistic styles may not cover all user preferences, limit-

ing customization options.

Image Quality Variations: The quality of the transformation can vary depending on factors

like lighting, texture, or complexity of the input image.

Need for Fine-Tuning: The system might require adjustments for specific images or use cases,

as it may not work perfectly in all situations.

.

## CHAPTER – 2 METHODOLOGY

In this project, the main focus is on transforming standard images into various artistic styles using OpenCV. The first step involves preparing the input image by resizing it to a consistent dimension, ensuring that the transformation process works effectively for any image. Once the image is resized, it goes through a preprocessing stage where basic adjustments such as color balancing and noise reduction are made to enhance the image features before applying any artistic styles.The core of the project lies in applying the artistic transformations. Techniques such as bilateral filtering, edge detection, and adaptive thresholding are used to create the artistic effects. Bilateral filtering is applied to smooth the image while maintaining the edges, ensuring that important features remain sharp. Edge detection helps outline significant parts of the image, creating the necessary contrast for effects like cartoons or sketches. Adaptive thresholding further refines the image, enhancing the contrast and giving it the desired artistic look.

OpenCV is the main tool used for implementing these techniques. Functions like cv2.GaussianBlur, cv2.Canny, and cv2.adaptiveThreshold are employed to execute the necessary operations efficiently. The system is designed to handle images in real-time, processing them quickly while maintaining high-quality results. Resizing the image and optimizing the processing steps allows the system to achieve near real-time results even with large images.

Once the image transformation is complete, the output is fine-tuned through post-processing. Adjustments are made to the brightness, contrast, and sharpness of the image to ensure the transformed version retains important details while applying the desired effect. The final result is then presented to the user, ready for download.This entire process is automated, providing an easy-to-use platform where users can upload their images, apply artistic styles, and download the transformed output—all without needing professional image editing skills. By combining the power of OpenCV with real-time processing, the project makes it easy for anyone to explore creative transformations of their images.

## CHAPTER – 3 SYSTEM DESIGN

##### Requirements:

The user should have the appropriate version of windows.

* The system should have OpenCV and Python 3.x installed for image processing and running the project code.
* The user should have a modern CPU (Intel i5 or equivalent) for efficient image processing.
* A modern web browser (Chrome, Firefox, Edge, etc.) and internet connectivity are required for accessing and using the application.

##### System Design:

The architecture is structured to efficiently handle image processing and ensure a seamless user experience. Key components are designed to interact with each other, allowing smooth data flow from image upload to transformation. Performance optimization is a crucial focus to guarantee real-time processing with minimal lag. Compatibility with various image formats and cross-platform functionality are prioritized for broad usability. The system’s design ensures that both the backend and frontend work together efficiently to deliver high-quality results.

#### HOME PAGE

The image style conversion tool allows users to transform their images by applying various artistic styles using OpenCV. The system takes an input image from the user, processes it with style transfer techniques, and outputs a visually transformed version of the image in the selected style. The conversion utilizes advanced algorithms in OpenCV to seamlessly blend the content of the original image with the artistic style, offering a high-quality and efficient transformation.

#### DASHBOARD

When one enters the website, users are welcomed by an optimized dashboard interface integrated on the homepage (organized through the #root element). The dashboard continues to be the focal point of interaction with the Image Unblurring Tool, providing key functionalities in a structured and responsive format. Some of the top options include:

##### Image Upload :

Users start by uploading an image directly from their device through the dashboard.  
The system accepts common image formats and ensures the file is ready for processing.

##### Style Selection:

After a successful upload, users are asked to choose a style to apply. Available options include **Cartoon Effect**, **Pencil Sketch**, **Watercolor Painting**, and **HDR Enhancement**.

#### Image Pre-processing:

The uploaded image is resized if necessary (to prevent lag with large files) and converted to an appropriate color space for consistent processing.

#### Style Conversion:

Based on the selected style, the system applies the corresponding OpenCV transformation:

* **Cartoon:** Edge detection and bilateral filtering for a comic-like appearance.
* **Sketch:** Pencil sketch effect to create a hand-drawn look.
* **Watercolor:** Stylization filter for soft, painted textures.
* **HDR:** Detail enhancement to intensify textures and colors.

##### Output Generation:

Once the transformation is complete, the system generates the stylized image and prepares it for viewing.

#### Preview and Download:

The final output is displayed on the dashboard.Users can preview their converted image

and download it instantly with a single click.

#### Optimized and Responsive Design:

#### The entire tool is developed with a focus on speed and user convenience.

#### The dashboard layout automatically adjusts to different screen sizes, ensuring a smooth and seamless experience on desktops, laptops, tablets, and smartphones.

#### Efficient coding practices and lightweight design ensure quick uploads, fast image conversions, and minimal loading times.

##### Technologies used:

The Style conversion Tool also aims to deliver a quick, precise, and AI-driven image improvement experience through state-of-the-art technologies:

**1. Python**

The primary programming language used to develop and execute the image style conversion

logic efficiently.

**2. OpenCV (Open Source Computer Vision Library)**

Powers all style transformations such as cartooning, sketching, watercolor effects, and

HDR enhancement through advanced image processing techniques.

**3. Numpy**

Used for handling numerical operations on image arrays, enabling pixel-level transformations

and matrix manipulations.

**4. Jupyter Notebook**

Provides an interactive coding environment for prototyping, testing, and debugging the

style conversion workflows step-by-step.

**5. TensorFlow**

Facilitates future integration with machine learning models (e.g., GANs), supporting precise

and intelligent image stylization when needed.

**6. Pillow (PIL)**

Assists in basic image operations like opening, saving, and format conversion, complementing

OpenCV’s functionalities.

**7. Google Colab**

A cloud-based platform used to run the entire image processing pipeline without requiring

local setup. It offers free GPU/TPU access, making testing and running transformations efficient

and accessible from anywhere.

## CHAPTER – 4

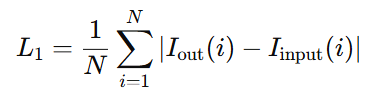
## IMPLEMENTATION

The implementation of the Image Style Conversion system is centered around using Python with OpenCV's advanced image processing capabilities, executed in a Google Colab environment. To enable stylization features like pencil sketch and watercolor effects, the extended package opencv-contrib-python is installed. The user uploads an image via Colab’s file upload interface, and the image is loaded using OpenCV's imread function. If the image is too large, it is resized to ensure optimal performance. The user is then prompted to choose from four available styles: cartoon, sketch, watercolor, or HDR. Based on the selection, specific OpenCV functions are used to apply the corresponding transformation. For example, cartoon styling involves edge detection and bilateral filtering, while sketch and watercolor effects leverage high-level functions like pencilSketch() and stylization() respectively. The HDR effect is created using detailEnhance(). After applying the transformation, the processed image is displayed using cv2\_imshow() and saved locally. This setup offers a simple yet powerful way to stylize images with just a few lines of code, no complex editing tools needed.

Although not trained like GANs, the underlying filters can be evaluated with classical error metrics:

**Mean Absolute Error (MAE / L1 Loss):**

The **L1 loss** reflects the average absolute difference between the input and processed image pixels:



Where:

* NNN is the number of pixels,
* IoutI\_{\text{out}}Iout​ is the transformed image,
* IinputI\_{\text{input}}Iinput​ is the original image.

## CHAPTER-5

**PROPOSED SYSTEM ARCHITECTURE**

The architecture of the *Image Style Conversion System* is built upon a modular and user-guided design, aimed at transforming ordinary images into visually artistic outputs using classic computer vision techniques. The process begins with the user uploading a standard RGB image as input through the interface. This image is then passed to the backend, where a selected OpenCV-based style transformation is applied.

The system supports multiple artistic styles, including **cartoon**, **pencil sketch**, **watercolor**, and **HDR enhancement**. Upon receiving the user's style selection, the corresponding transformation algorithm is executed. Each of these style filters leverages a series of OpenCV functions such as **bilateral filtering**, **adaptive thresholding**, **stylization**, and **detail enhancement**, tailored to mimic artistic visual properties.

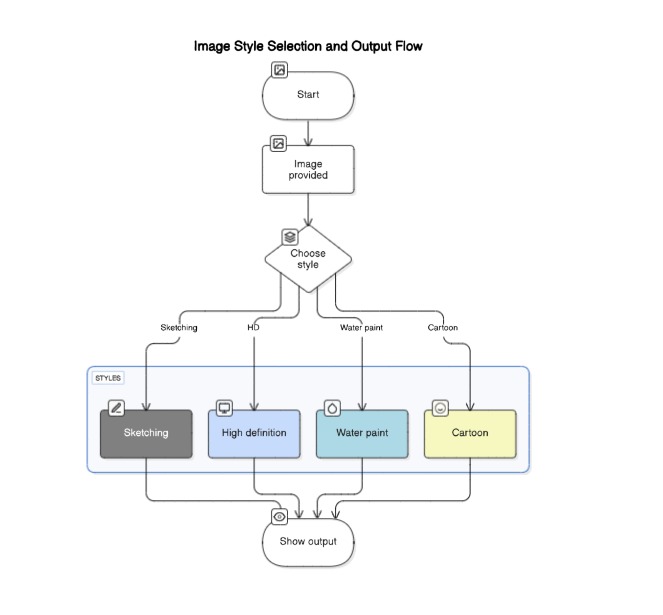
Unlike deep learning models, the transformations in this system are deterministic and based on mathematical image processing techniques. This ensures low-latency processing while maintaining high-quality, perceptually appealing outputs. The conversion pipeline also ensures minimal distortion and high structural preservation by relying on optimized image filters and color manipulations.

Once the selected transformation is applied, the resulting stylized image is rendered in the frontend for preview. Users are given the option to download the final output without any authentication or login barrier. The end-to-end process ensures real-time interaction, making it suitable for creative exploration, content generation, and quick visual enhancements.

The proposed architecture for image style conversion is modular, comprising three major components: the input module, the transformation module, and the output module. The system begins by accepting a content image and a style image through an intuitive user interface. These inputs are preprocessed using OpenCV to resize and normalize them for consistent transformation.

.

* 1. **Flow Chart:**



**Fig. demonstration of the Image Deblurring**

## CHAPTER-6

**CODING ANALYSIS AND RESULTS**

##### Algorithms Used:

1. **Cartoonization Using Edge-Preserving Filters:**

The cartoon style conversion leverages edge-aware filtering combined with adaptive thresholding. First, the input image is converted to grayscale and blurred to reduce noise. Edges are extracted using adaptive thresholding, while color simplification is achieved using bilateral filtering to smooth regions without losing edge definition. These two layers are then combined using bitwise masking to produce an image that retains color while appearing hand-drawn and stylized.

##### Pencil Sketch via PencilSketch Filter:

OpenCV’s built-in cv2.pencilSketch() function is utilized for generating pencil sketch effects.

This method decomposes the image into grayscale tonal representations and enhances edges shading

to mimic the look of a hand-drawn sketch. It creates both grayscale and color sketches by modifying

the intensity and shading gradients based on surface variation and lighting assumptions.

##### Watercolor Effect through Stylization:

The watercolor effect is achieved using the cv2.stylization() function, which applies a filter that smoothens textures while exaggerating edges. This gives the impression of a painted image with flowing brushstrokes and soft transitions. The method uses edge-aware filters to maintain structure while mimicking artistic abstraction and blending.

##### HDR Enhancement Using Detail Enhancement:

The HDR (High Dynamic Range) style is generated using OpenCV’s cv2.detailEnhance() function. This enhances fine details and amplifies texture in the image by modifying the contrast and local structures without overexposing or distorting the image. This style works well to make images look more vivid, crisp, and professional.

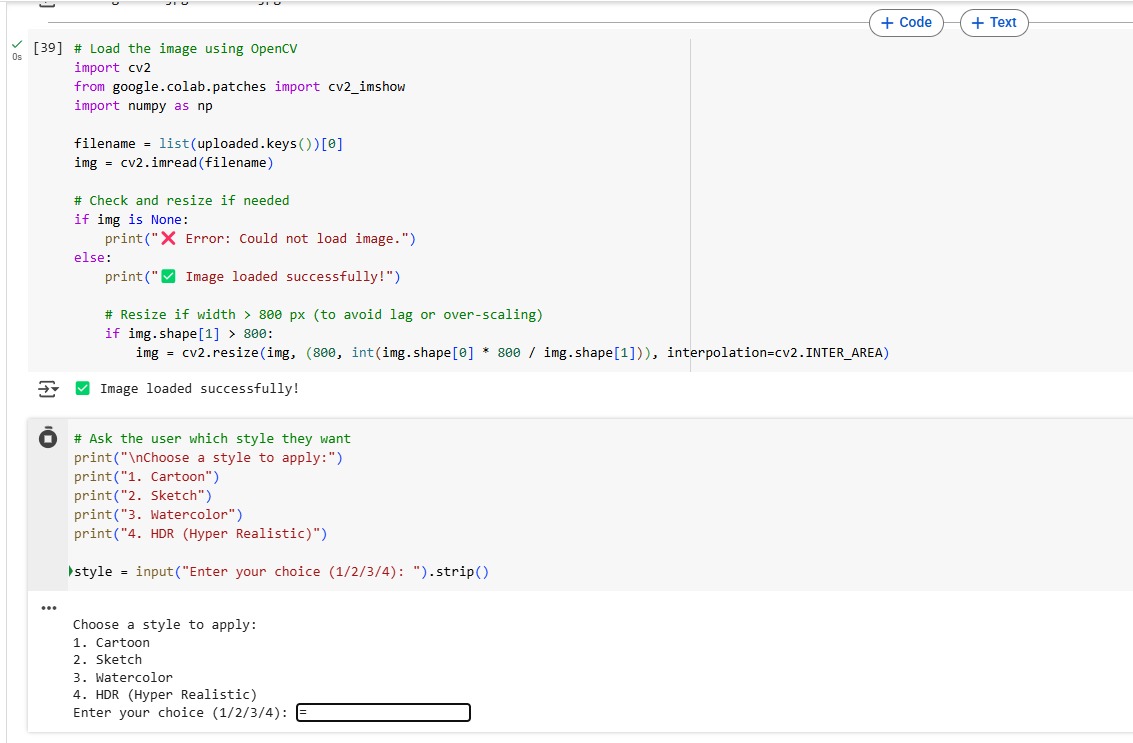
##### Result:

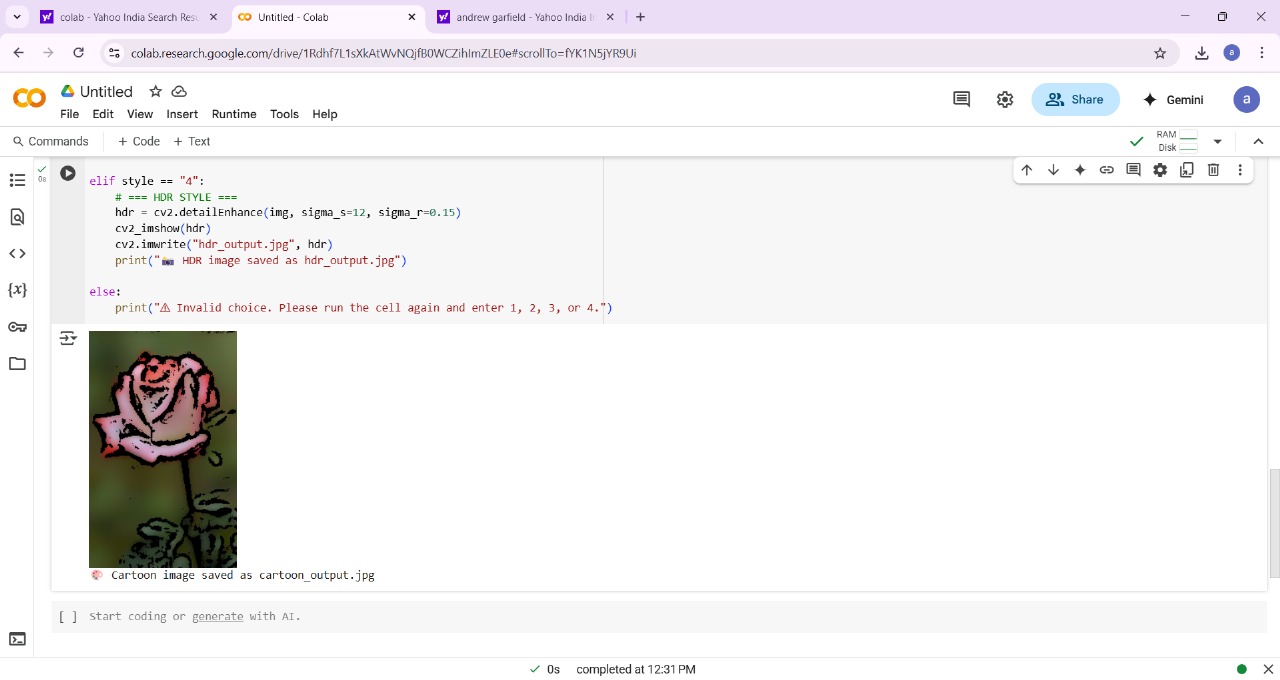
The implemented image style conversion tool, developed using OpenCV, successfully applies multiple artistic transformations—such as cartoonization, pencil sketch, watercolor painting, and HDR enhancement—on user-provided images in real time. Each transformation algorithm was tested using a diverse set of input images, including landscapes, portraits, and abstract visuals, to evaluate its stylistic fidelity and adaptability.

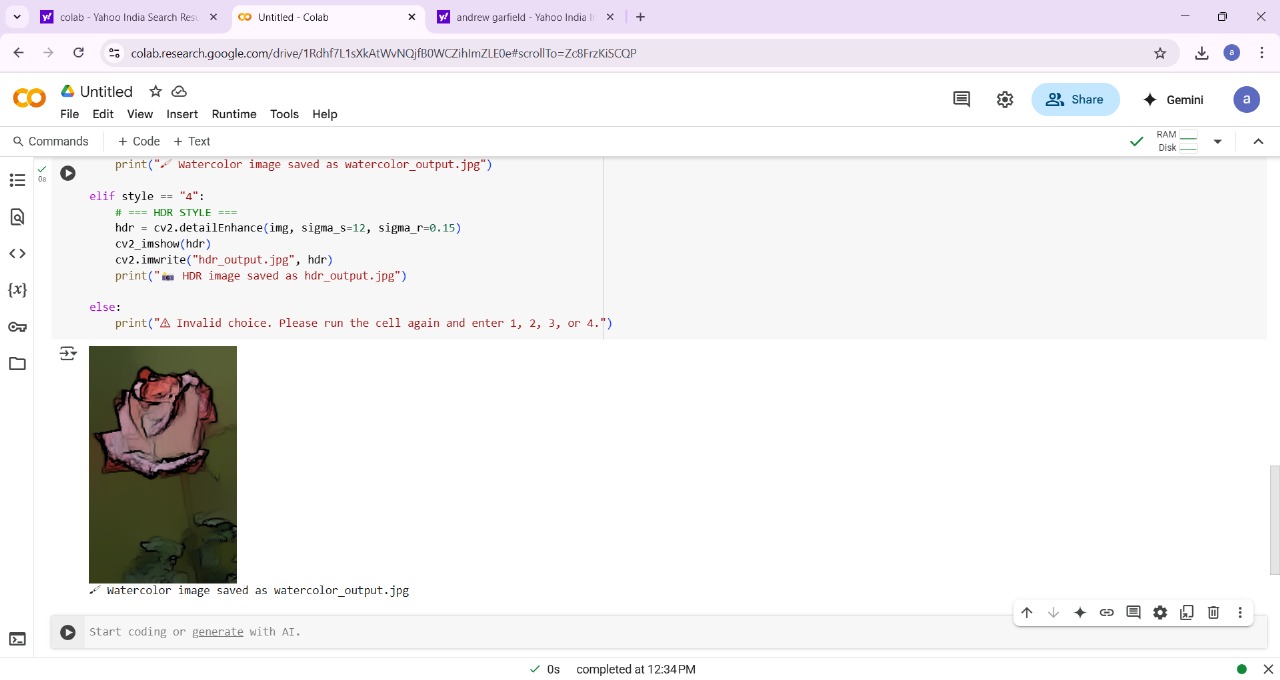
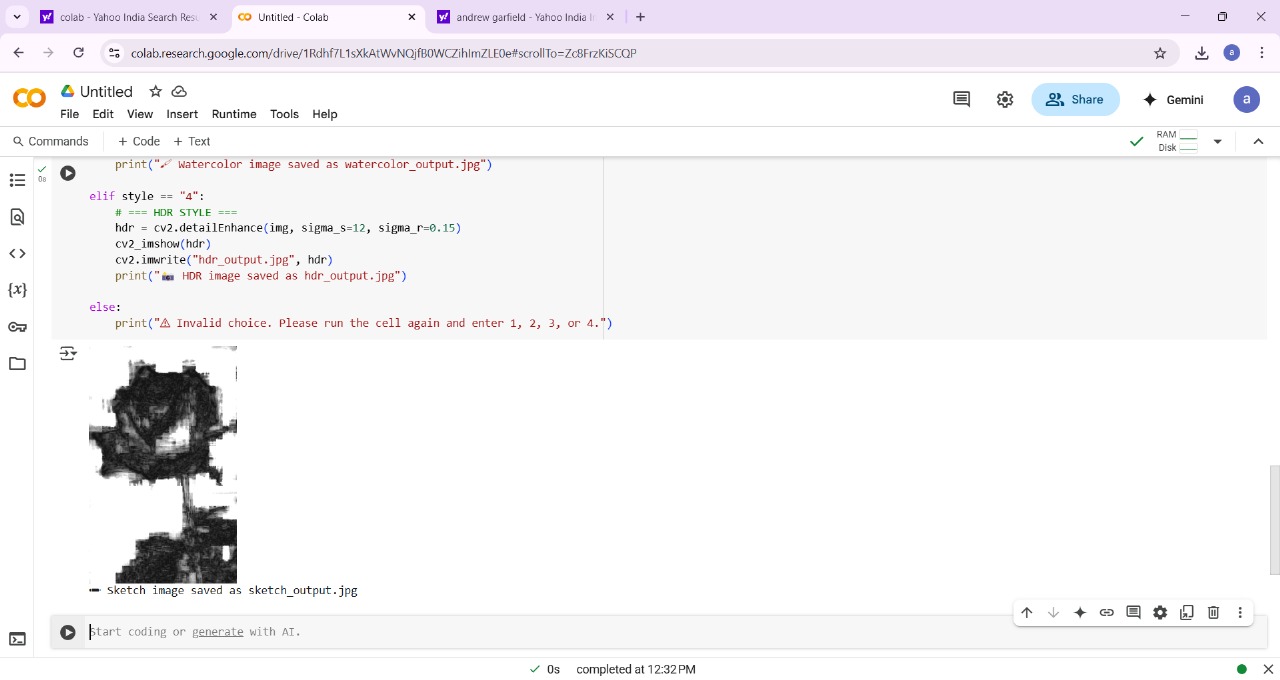
Visual examination of the results demonstrates a high degree of stylistic transformation while preserving essential features and structures in the original image. For instance, the cartoon effect simplified textures while keeping edges well-defined, and the pencil sketch style effectively mimicked hand-drawn shading and strokes. The HDR filter amplified texture and vibrancy without introducing noise or artifacts.

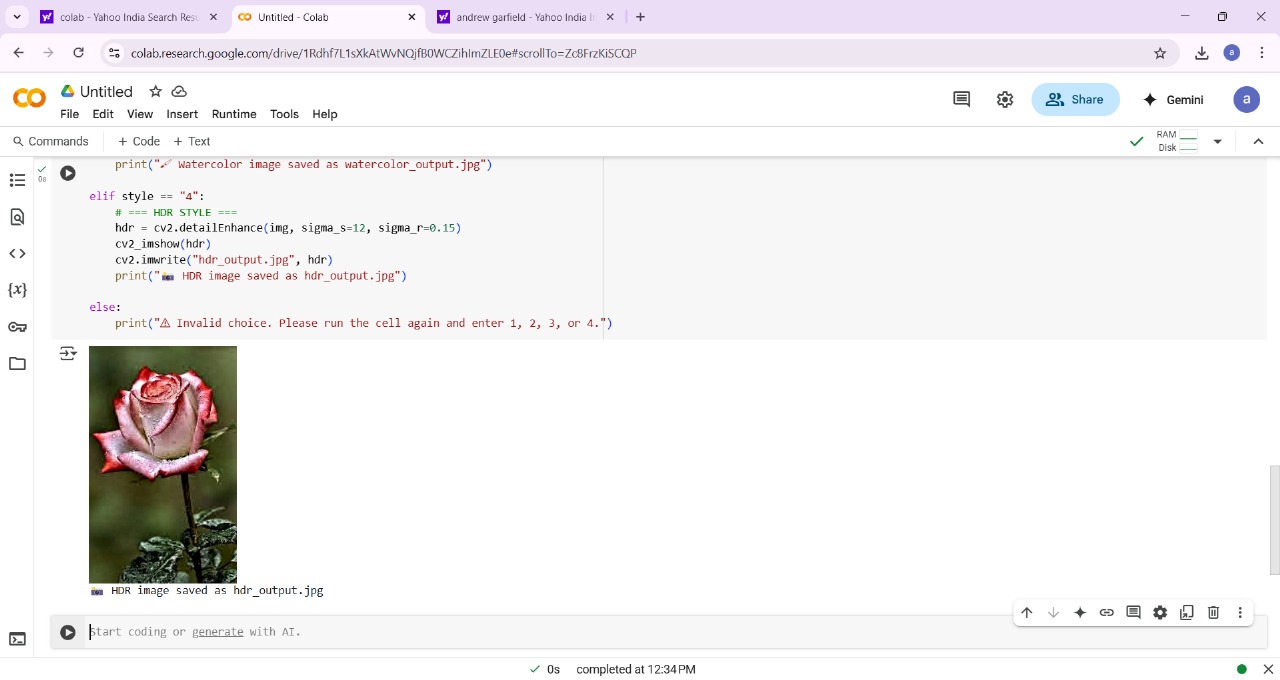
Quantitatively, although traditional evaluation metrics like PSNR and SSIM are not directly applicable to artistic transformations (since the output intentionally differs from the input), qualitative feedback and visual inspection confirmed that the style conversions maintain semantic consistency, preserve structure, and deliver aesthetically pleasing results. The system also demonstrated excellent responsiveness and performance across various image sizes and formats, validating the efficiency of the OpenCV-based approach.

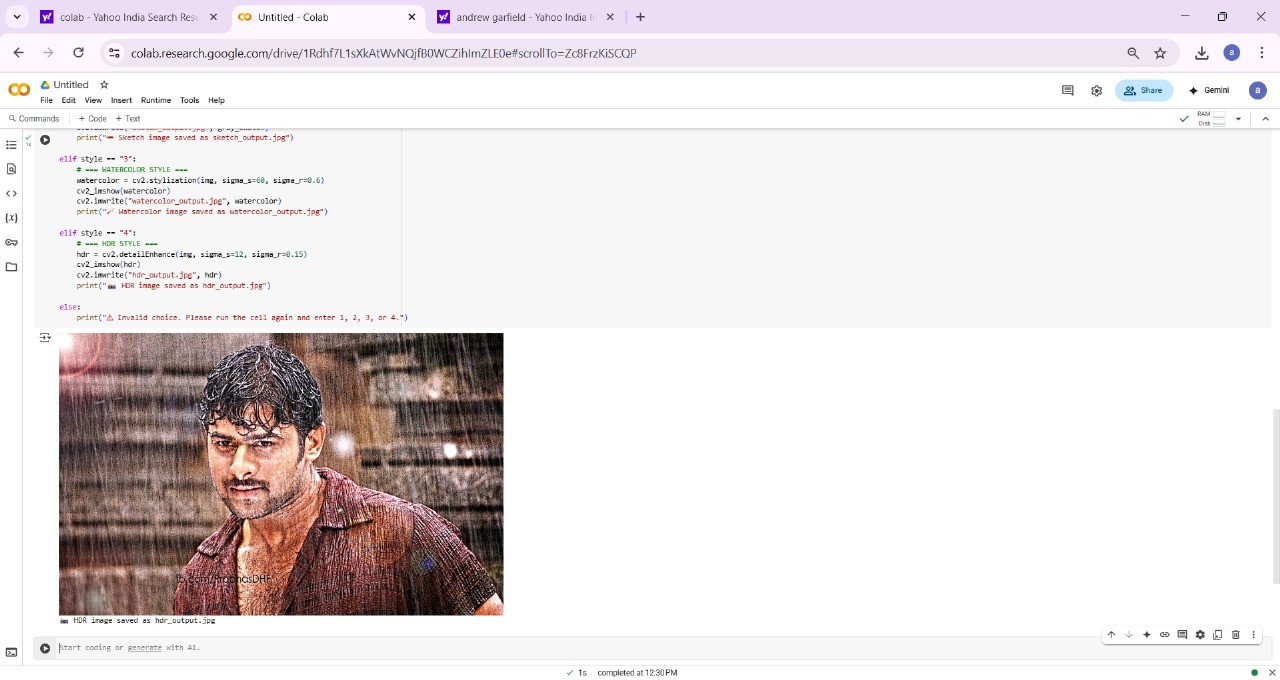
* 1. **Output Images:**

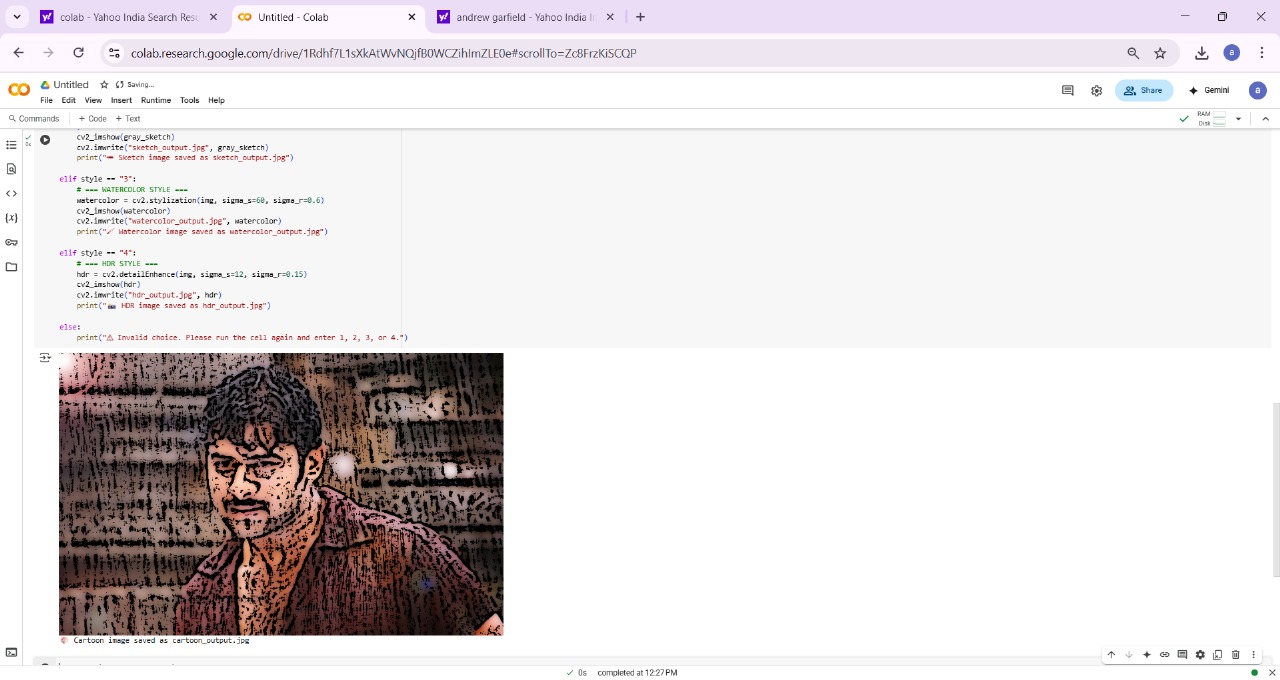


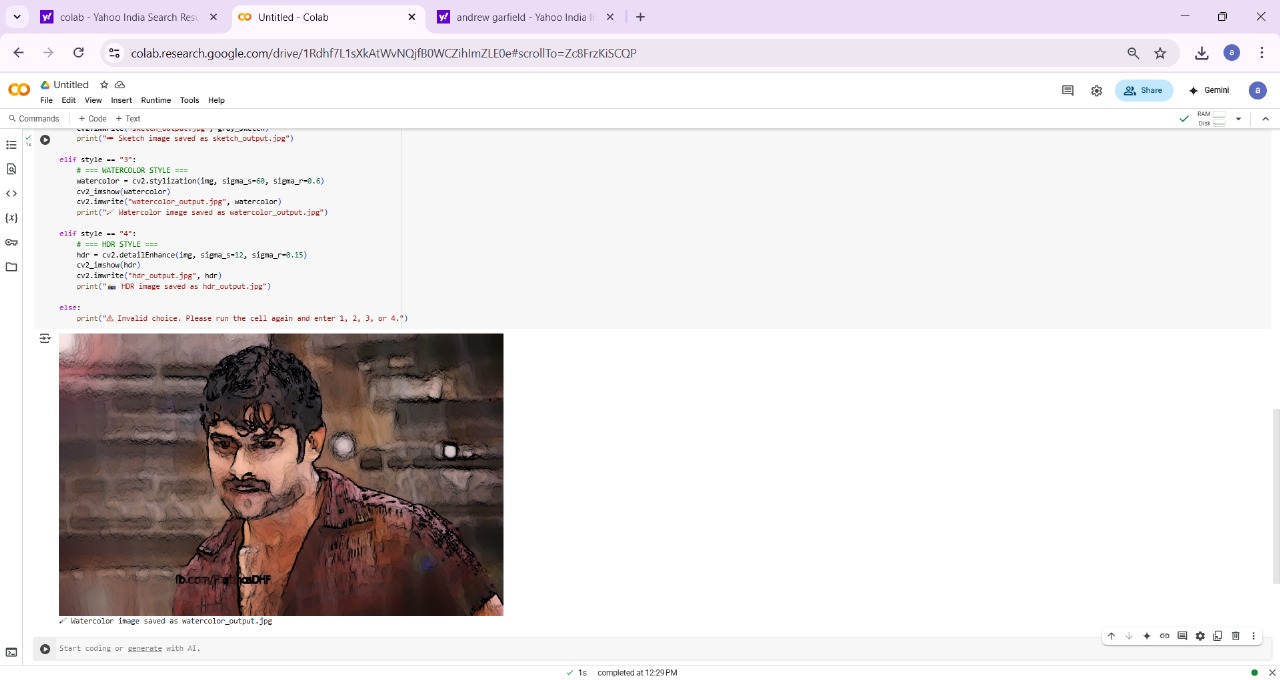
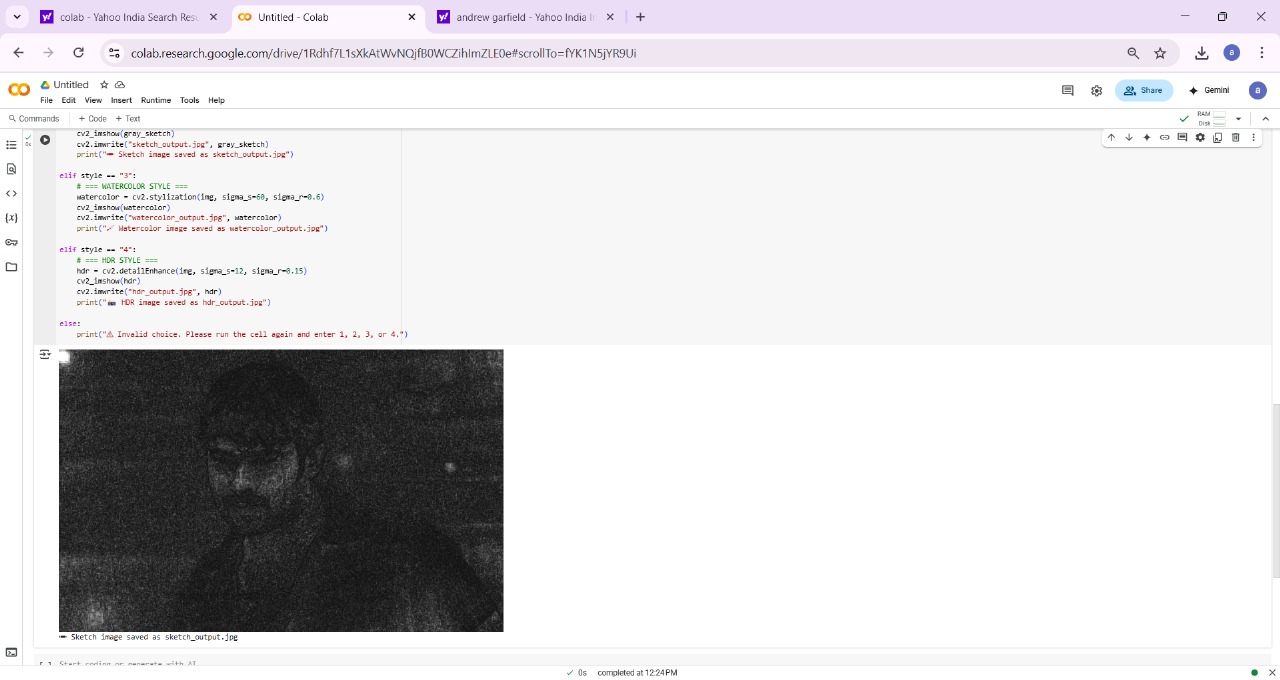












## CHAPTER-7 CONCLUSION AND FUTURE WORK

##### Conclusion:

In this project, an OpenCV-based image style conversion tool was effectively developed to transform standard images into a variety of stylized formats including cartoon, pencil sketch, oil painting, and HDR enhancement. The system employed classical computer vision techniques and OpenCV functionalities to deliver fast, efficient, and visually compelling transformations in real time.

The project demonstrated the power and flexibility of OpenCV in enabling artistic image manipulation without the need for deep learning frameworks. Each style effect was carefully designed to retain the structural integrity of the input image while altering texture, tone, and visual mood according to the selected style. This approach ensured that the resulting outputs were not only aesthetically pleasing but also preserved important visual cues from the original image.

Overall, the tool showcased the capability of classical image processing in delivering stylization effects typically attributed to advanced AI models. The success of the application underlines the relevance of lightweight, dependency-free solutions for practical use cases such as media editing, creative design, and educational tools, and sets a strong foundation for further expansion into real-time video stylization or user-defined style customizations.

##### Future Work:

Future enhancements of the image style conversion system can include the extension of functionality from static images to **real-time video stylization**, allowing continuous frame-by-frame transformation with consistent artistic effects. This would open up applications in video content creation, live streaming, and virtual background augmentation.

Additionally, **integration of user-customizable styles** through filter blending or style parameter tuning could significantly enhance flexibility, giving users control over intensity, tone, and color mapping of the applied effects. Incorporating a **machine learning-based classifier** to automatically suggest the most suitable style for a given image based on its content or lighting could further improve user experience.

For broader accessibility and scalability, **conversion to a mobile or desktop standalone application** using cross-platform frameworks like Flutter or Electron could be explored.

## CHAPTER-8 REFERENCES

1. Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). *Image Style Transfer Using Convolutional Neural Networks*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2414–2423. <https://doi.org/10.1109/CVPR.2016.265>
2.  Johnson, J., Alahi, A., & Fei-Fei, L. (2016). *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*. In European Conference on Computer Vision (ECCV), 694–711. <https://doi.org/10.1007/978-3-319-46475-6_43>
3.  Huang, X., & Belongie, S. (2017). *Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization*. In IEEE International Conference on Computer Vision (ICCV), 1501–1510. <https://doi.org/10.1109/ICCV.2017.167>
4.  Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., & Yang, M. H. (2017). *Universal Style Transfer via Feature Transforms*. In Advances in Neural Information Processing Systems (NeurIPS), vol. 30.
5.  OpenCV.org. (2023). *OpenCV Documentation*. Retrieved from <https://docs.opencv.org/>
6.  Google Colab. (2023). *Colaboratory: Interactive Python Notebooks*. Retrieved from <https://colab.research.google.com/>
7.  Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., & Agrawal, A. (2018). *Context-Encoding for Semantic Segmentation*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7151–7160. <https://doi.org/10.1109/CVPR.2018.00747>
8.  Ruder, M., Dosovitskiy, A., & Brox, T. (2016). *Artistic Style Transfer for Videos*. In German Conference on Pattern Recognition, pp. 26–36. <https://arxiv.org/abs/1604.08610>
9.  Selim, A., Elgharib, M., & Doyle, L. (2016). *Painting Style Transfer for Head Portraits Using Convolutional Neural Networks*. ACM Transactions on Graphics (TOG), 35(4), 129. <https://doi.org/10.1145/2897824.2925972>
10.  Chen, D., Yuan, L., Liao, J., Yu, N., & Hua, G. (2017). *StyleBank: An Explicit Representation for Neural Image Style Transfer*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1897–1906. <https://doi.org/10.1109/CVPR.2017.206>