A PROJECT REPORT

On

Image to pencil recognition using Machine learning and Open CV

Submitted in partial Fulfillment of Requirements for the degree of

Bachelor of Technology

In

Artificial Intelligence and Data Science

By

Kasak (02715611922)

Jyoti Rana (05415611922)

Under the supervision of

Dr. Archana Kumar (H.O.D. & Professor (AI & DS))

Mr. Ritesh Kumar (Assistant professor (AI & DS))

Department of Artificial Intelligence & Data Science

Dr. AKHILESH DAS GUPTA INSTITUTE OF PROFESSIONAL STUDIES

(A Unit of BBD Group)

Approved by AICTE and Affiliated with GGSIP University

FC-26, Shastri Park, New Delhi-110053

TABLE OF CONTENTS **Page**

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name: KASAK

Roll No.: 02715611922

Date:

Signature:

Name: JYOTI RANA

Roll No.: 05415611922

Date:

CERTIFICATE

This is to certify that the Project Report entitled **"Image to pencil recognition using Machine Learning Techniques and Open CV"** which is submitted by **Kasak & Jyoti Rana** in partial fulfillment of the requirement for the award of degree B. Tech. in the Department of Artificial Intelligence & Data Science of Dr. Akhilesh Das Gupta Institute of Professional Studies formerly known as ADGITM, New Delhi, is a record of the candidate's own work carried out by him under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Date: Supervisor

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Second Year. We owe a special debt of gratitude to his/her constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day. We also take the opportunity to acknowledge the contribution of for his/her full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:

Name: KASAK

Roll No.: 02715611922

Date:

Signature:

Name: JYOTI RANA

Roll No.: 05415611922

Date:

ABSTRACT

• The "Image to pencil recognition using Machine Learning Techniques and Open CV" project explores the intersection of image processing and artistic expression by developing an application that transforms digital images into realistic pencil sketches.

• This project aims to develop an algorithm or application that automatically transforms digital images into artistic sketches. It will bridge the gap between photography and drawing by analyzing the original image and generating a new image that captures its essence in a stylized, hand-drawn manner.

• The project involves preprocessing steps, feature extraction, and a carefully trained model to achieve a convincing and aesthetically pleasing pencil sketch effect.

• This system helps people understand how painters create pencil drawings, which is a fundamental art form for representing natural scenes.

• The primary objective is to make art accessible to a broad audience and democratize its use.

• It will bridge the gap between photography and drawing by analyzing the original image and generating a new image that captures its essence in a stylized, hand-drawn manner.

• This technology utilizes creativity to offer users a fun and artistic method to transform their photos into sketches. It converts digital images into realistic and aesthetically pleasing pencil sketches.

LIST OF KEYWORDS

- Image processing
- Pencil sketch
- Image-to-sketch conversion
- Grayscale conversion
- Edge enhancement
- Gaussian blurring
- Colour dodging
- Real-time
- Artistic exploration
- Image editing
- Content creation
- User-friendly application
- OpenCV
- Kivy framework
- Graphical user interface (GUI)
- Line Integral Convolution (LIC)
- Deep learning
- Artistic style transfer

LIST OF ABBREVIATIONS

- **AI**: Artificial Intelligence
- **AAI:** The Association for the Advancement of Artificial Intelligence
- **ECCV:** European Conference on Computer Vision
- **GUI:** Graphical User Interface
- **GMED:** Gradient Morphological Edge Detection
- **LIC:** Line Integral Convolution
- **OpenCV:** Open-Source Computer Vision Library
- **SBL:** Swing Bilateral LIC

CHAPTER -1

INTRODUCTION

1.1 Introduction of the problem: -

The ability to convert digital images into artistic representations like pencil sketch has garnered significant interest in recent years. These conversions hold applications in various domains, including artistic exploration, image editing, and content creation.

Traditional methods for creating pencil sketch involve manual drawing techniques that demand artistic skills and time investment. Conversely, digital image-to-sketch conversion techniques offer a streamlined approach, enabling users to generate artistic outputs from digital photographs with relative ease.

This research investigates the development of a user-friendly application that leverages established image processing techniques to achieve real-time image-to-pencil sketch conversion.

The application utilizes established image processing techniques, such as the OpenCV [10, 11] library in Python and the Kivy[12] framework, to create an interactive and user-friendly graphical user interface. The technique is faster, more efficient, and easier to use than traditional manual drawing techniques. Overall, the application provides a seamless user experience for generating artistic outputs from digital photographs

1.2 Some Related Definitons:

- **Machine learning[13]:**

Machine learning is a subset of artificial intelligence (AI) that focuses on enabling machines to learn from data and make predictions or decisions without being explicitly programmed to perform specific tasks.

- **Open CV:**

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It provides a comprehensive set of tools and functions for real-time image and video processing, including various algorithms for tasks such as object detection, image segmentation, facial recognition, and feature extraction.

- **Kivy:**

Kivy is an open-source Python framework for developing multi-touch applications. It enables developers to create cross-platform applications that run on desktop operating systems (such as Windows, macOS, and Linux) as well as mobile platforms (such as Android and iOS).

1.3 Problem statement:

Image to pencil recognition involves converting a digital image into a pencil sketch, mimicking the appearance of a hand-drawn sketch.

CHAPTER -2

LITERATURE SURVEY

- Saeko Takagi†, Noriyuki Matsuda, Masato Soga, Hirokazu Taki, Takashi Shima, Fujiichi Yoshimoto proposed an AI-powered system that has been developed to aid beginners in learning basic techniques of pencil drawing. The system assesses the user's sketches and provides guidance based on the motif data. It includes four subsystems, namely, motif feature extraction, sketch feature extraction, error identification, and advice generation and presentation. The advice is presented through a 3D model, which helps users better understand the errors in their sketches. The system aims to help beginners enhance their sketches by providing them with effective advice. [1]
- Jin Zhou and Baoxin Li's produced a new algorithm can turn personal photos into pencil sketch-like drawings. It involves gradient transformation and final smoothing to produce visually striking results. Their algorithm generates more aesthetically pleasing images than existing gradient estimation methods, even for complex images like human faces and intricate backgrounds. [2]
- Shuo Sun Tianjin, Dongwei Huang Tianjin proposed an enhanced method for generating automatic pencil drawings using Line Integral Convolution (LIC)[15]. The objective is to improve the quality of the images produced by refining the techniques used for image segmentation and texture direction detection. The proposed technique involves a more precise graph-based image segmentation algorithm and a region-based approach for creating white noise and texture directions. This results in pencil drawings that closely resemble real artistic styles. [3]
- Heekyung Yang, Yunmi Kwon, & Kyungha Min developed a swing bilateral LIC (SBL) method for creating pencil drawings in various styles , filter to control the direction and colour of pencil strokes, resulting in detailed drawings that resemble artistic techniques. The technique involves several steps, such as extracting linear features, generating a noise distribution, and determining noise values, to achieve the desired pencil drawing effects.[4].
- Jiang, Yifeng shows the system is implemented to generate pencil-drawing-style images from photographs. It includes two main stages. First, the input image

generates a stroke layer to represent the shapes on the image, imitating painters sketching the contours. Then it produces tonal textures, imitating the hatching process when painters depict brightness and shades with pencils. Finally, the two layers are combined to synthesize a non-photorealistic pencil drawing. [5]

- Yijun Li, Chen Fang, Aaron Hertzmann, Eli Shechtman, Ming-Hsuan Yang develops a two-branch model that learns separate filters for generating sketchy outlines and tonal shading from a collection of pencil drawings. The training data pairs are created by extracting clean outlines and tonal illustrations from original pencil drawings using image filtering techniques and the drawing styles are manually labelled. In addition, the model creates different pencil styles (e.g., line sketchiness and shading style) in a user-controllable manner. [6]
- HUAPING ZHOU, CHAO ZHOU, AND XIAOYAN WANG introduce a new algorithm called GMED that can be used to create pencil drawings from natural images. The algorithm uses gradient maps and morphological operations to extract lines, resulting in pencil drawings that are of higher quality compared to existing methods. The algorithm works by first extracting the gradient map of the input image, applying morphological operations to the gradient map, and then using texture filling and tone mapping to create a realistic pencil drawing effect. The researchers experimented with different line lengths and found that optimal line lengths could produce a hand-drawn effect without appearing messy. The main innovation of the GMED algorithm lies in its use of gradient maps and morphological operations to extract more natural and continuous lines in the final pencil drawings. [7]
- Zhengyan Tong, Xuanhong Chen, Bingbing Ni, Xiaohang Wang presents an automatic image-based approach for converting greyscale images to pencil sketches, in which strokes follow the image features. The algorithm first extracts a dense direction field automatically using Logical/Linear operators which embody the drawing mechanism. Next, a reconstruction approach based on a sampling and interpolation scheme is introduced to generate stroke paths from the direction field. Finally, pencil strokes are rendered along the specified paths with consideration of image tone and artificial illumination. As an important application, the technique is applied to render portraits from images with little user interaction. The

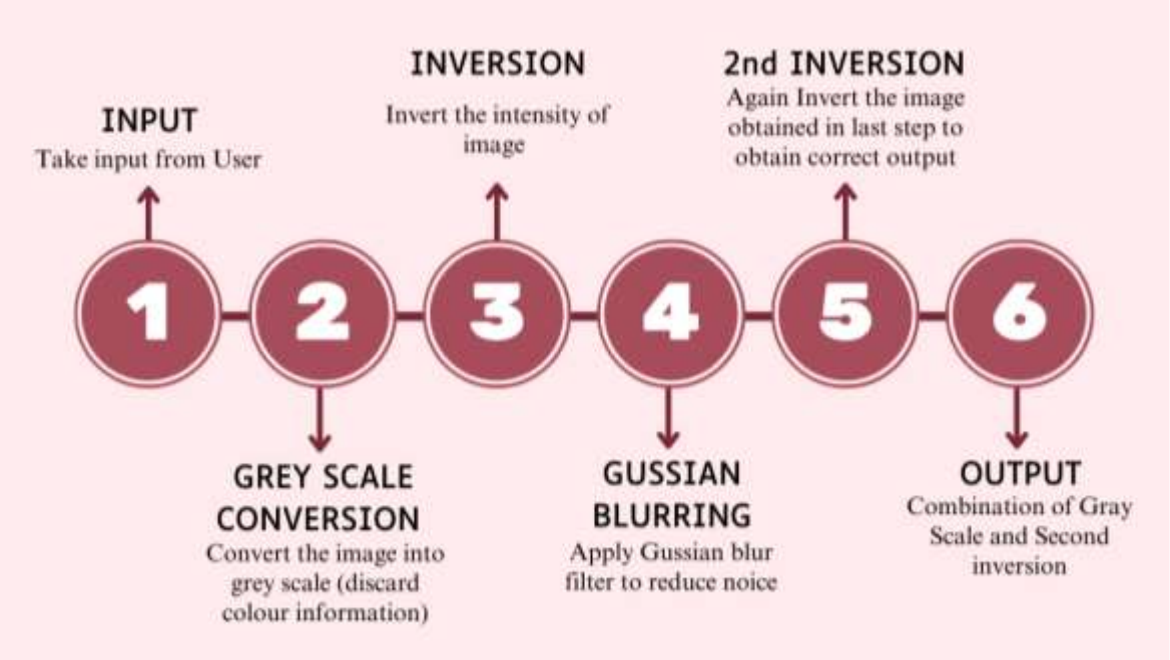
experimental results demonstrate that the approach can automatically achieve compelling pencil sketches from reference images.[8]

- Dong Wang, Guiqing Li, Chengying Gao, Shengwu Fu, and Yun Liang proposes a method for generating feature-preserving colour pencil drawings from photographs by enhancing lightness and reducing saturation The approach includes devising lightness enhancement and saturation reduction mappings to mimic the tonal style of colour pencil drawings, resulting in superior tone capture and feature preservation.[9]

CHAPTER -3

METHODOLOGY & TECHNOLOGY

The application leverages established image processing techniques to achieve the core functionality of converting digital images into visually appealing pencil sketches.


INPUT
Take input from User
INVERSION
Invert the intensity of image
2nd INVERSION
Again Invert the image obtained in last step to obtain correct output
1
2
3
4
5
6
GREY SCALE CONVERSION
Convert the image into grey scale (discard colour information)
GUSSIAN BLURRING
Apply Gussian blur filter to reduce noice
OUTPUT
Combination of Gray Scale and Second inversion

This section outlines the key steps involved in this process:

3.1 Preprocessing

- **Image Reading:** The process commences by reading the user-selected image using OpenCV's cv2.imread function. This function retrieves the image data from the specified file path and stores it as a NumPy [16] array for further processing within the application.
- **Grayscale Conversion[17]:** Since pencil sketches primarily focus on capturing shading and texture variations, the initial step involves converting the loaded colour image into grayscale. This step discards colour information and emphasizes intensity variations within the image. This lays the groundwork for generating a sketch that emphasizes edges and shading, mimicking the visual characteristics of a pencil drawing.

3.2 Edge Enhancement

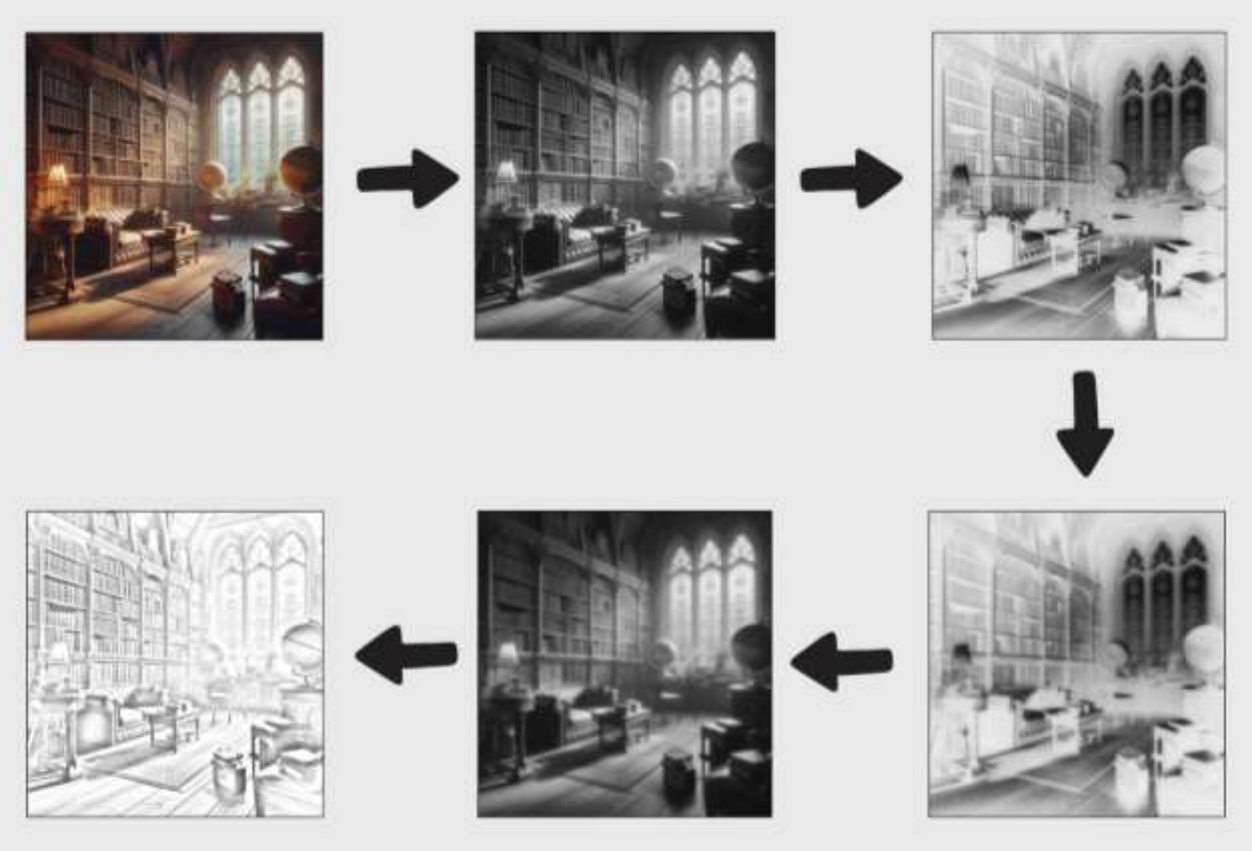
To achieve a prominent sketch-like appearance, the application employs techniques that enhance edges within the image. These edges are crucial for creating a visually appealing sketch as they define the overall structure and details of the converted output.

- **Image Inversion[18]:** The grayscale image undergoes an inversion process using a bitwise NOT operation (cv2.bitwise_not). This operation essentially flips the intensity values in the image, transforming bright areas into dark and vice versa. Inverting the grayscale image serves to enhance the prominence of edges within the grayscale data.
- **Gaussian Blurring[19]:** A Gaussian blur filter is applied to the inverted grayscale image using cv2.GaussianBlur. This step serves two primary purposes. Firstly, it acts as a noise reduction technique, smoothing out unwanted details and inconsistencies present in the image. Secondly, and more importantly for sketch generation.
- **Second Inversion:** Following the blur operation, an optional step involves inverting the image once again using bitwise NOT. This double inversion process can further amplify the edges within the image, creating a stronger foundation for the sketch generation stage.

3.3 Sketch Creation

The core concept behind generating the sketch lies in combining the original grayscale image with the processed, inverted, and blurred image. This is achieved by performing colour dodging using cv2.divide. In simpler terms, the dark edges in the blurred image act as a mask, darkening corresponding areas in the grayscale image. This results in a shading effect that creates a sense of depth and dimension within the sketch. Essentially, colour dodging leverages the inverted and blurred image to "dodge" away brightness from the original grayscale image, creating darker areas that resemble pencil strokes.

This multi-step process effectively transforms digital images into visually appealing pencil sketches. The interplay of grayscale conversion, edge enhancement, and colour dodging allows the application to capture the essence of the original image while infusing it with an artistic touch.



CHAPTER -4

RESULT ANALYSIS & DISCUSSION

4.1 Result:

The image-to-sketch conversion application successfully fulfils its objective of transforming digital photographs into visually striking pencil sketches. This section delves into the results achieved and explores the interplay of core image processing techniques that contribute to the artistic transformation.

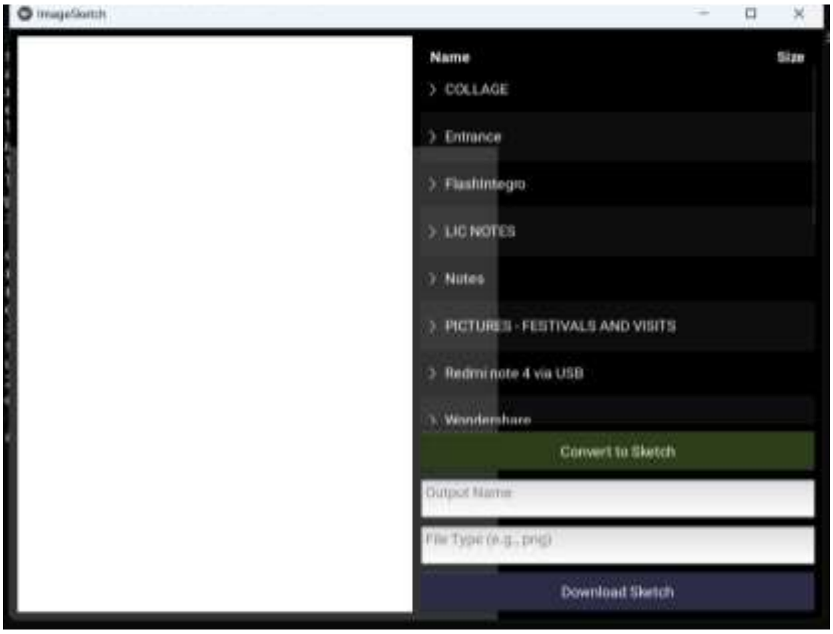
The application successfully bridges the gap between digital photographs and artistic expression, generating visually appealing pencil sketches. The core image processing techniques work in harmony to achieve this transformation. Grayscale conversion sets the stage by focusing on light and shadow variations, crucial for capturing the essence of a sketch.

Edge enhancement acts like an artistic brush, highlighting details and offering control through the blur level. A larger blur creates a more abstract and impressionistic feel, while a smaller blur retains finer details for a realistic sketch. Finally, colour dodging breathes life into the sketch by creating a shading effect, mimicking pencil strokes.

The interplay of these techniques delivers visually striking results. The application effectively transforms images, capturing their essence with a touch of artistic flair.

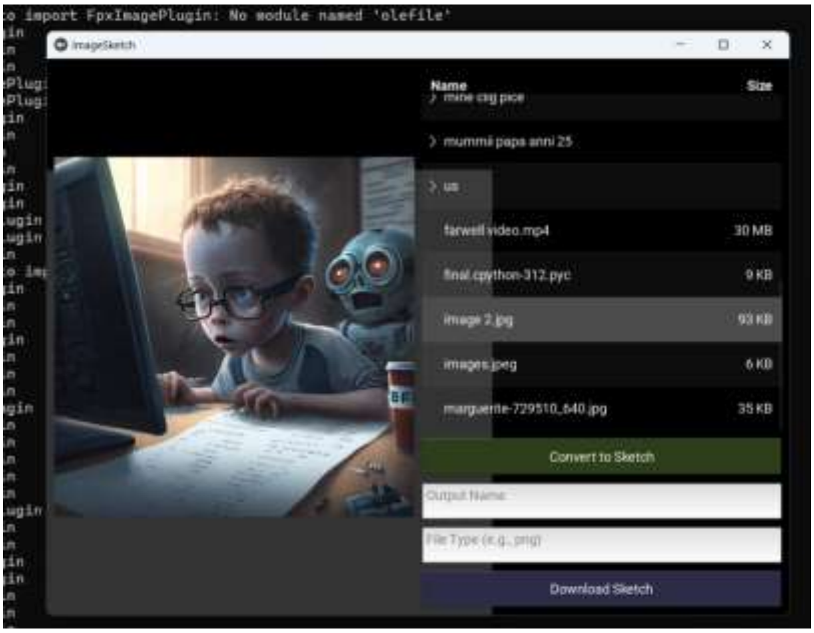
While the current focus is on pencil sketches, the future holds exciting possibilities for artistic exploration. Imagine the application evolving to create charcoal drawings or ink washes, empowering users to experiment with a diverse range of artistic styles.

Starting interface



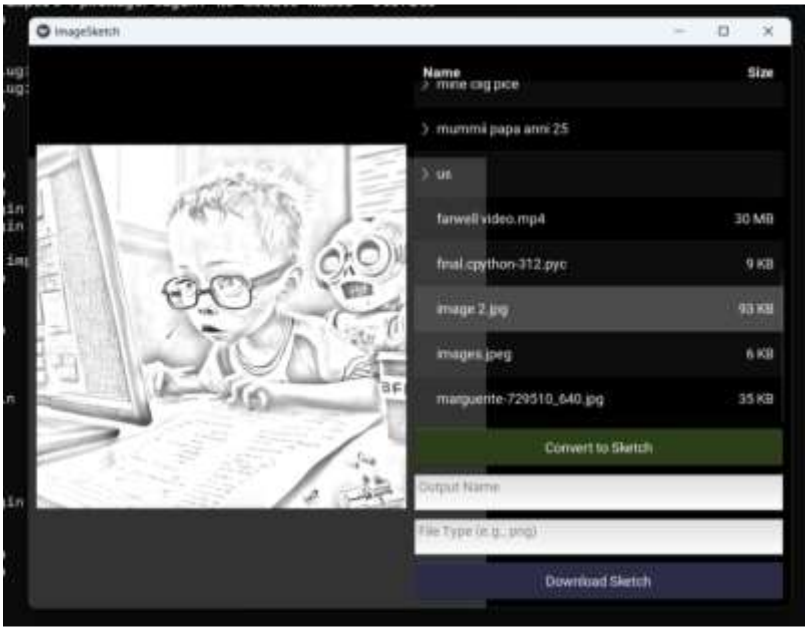
Name	Size
› COLLAGE	
› Entrance	
› FlashIntegro	
› LIC NOTES	
› Notes	
› PICTURES - FESTIVALS AND VISITS	
› Redmi note 4 via USB	
› Wondershare	

Image Selection


```
o import FpxImagePlugin: No module named 'olefile'
```

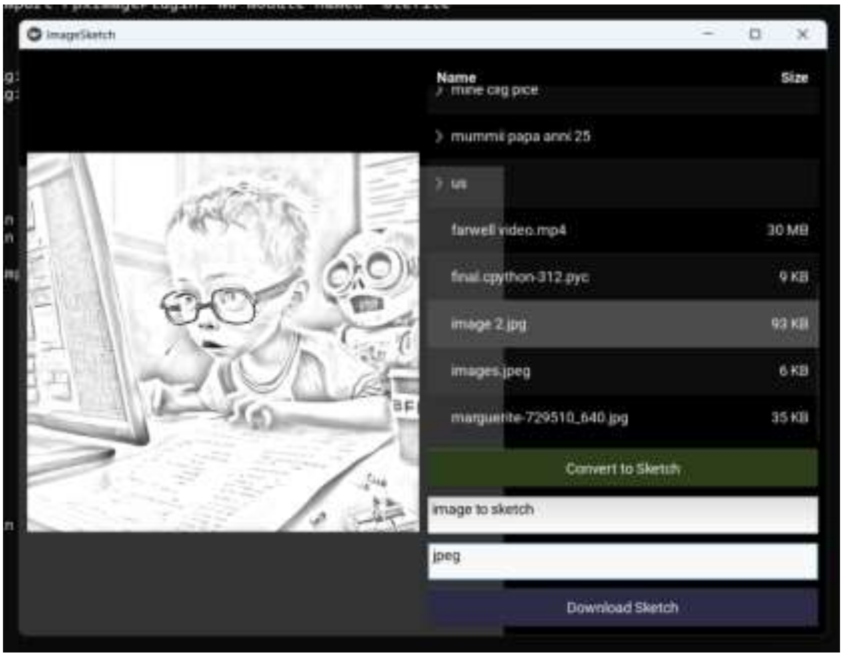
Name	Size
› mine cig pice	
› mummi papa anni 25	
› us	
farwell video.mp4	30 MB
final.cpython-312.pyc	9 KB
image 2.jpg	93 KB
images.jpeg	6 KB
marguerite-729510_640.jpg	35 KB

Image Conversion to Sketch



Name	Size
› mine cag pice	
› mummi papa anni 25	
› us	
farwell video.mp4	30 MB
final.cpython-312.pyc	9 KB
image 2.jpg	93 KB
images.jpeg	6 KB
marguerite-729510_640.jpg	35 KB

Saving file with User Specified Name & File Format



Name	Size
› mine cag pice	
› mummi papa anni 25	
› us	
farwell video.mp4	30 MB
final.cpython-312.pyc	9 KB
image 2.jpg	93 KB
images.jpeg	6 KB
marguerite-729510_640.jpg	35 KB

image to sketch

jpeg

4.2 Analysis and Discussion:

The implemented image processing techniques effectively collaborate to achieve the desired outcome of transforming digital images into pencil sketches. Grayscale conversion lays the groundwork by focusing on the core elements of a sketch – light and shadow variations.

Edge enhancement, through image inversion and Gaussian blurring, strengthens the prominence of key details and offers some artistic control over the level of detail and overall style through the blur kernel size adjustment. Finally, colour dodging leverages the processed image to create a shading effect, mimicking pencil strokes and infusing depth into the sketch.

This project demonstrates the feasibility of leveraging established image processing techniques to create user-friendly applications for artistic image manipulation. The ability to generate visually appealing pencil sketches from digital photographs opens doors to exciting avenues for both artistic exploration and practical applications:

- **Artistic Exploration:** The application empowers users with a basic tool for artistic expression. By experimenting with the blur kernel size, users can explore different artistic styles within the realm of pencil sketches. Additionally, future advancements could introduce functionalities for creating sketches that mimic various artistic styles, such as charcoal drawings or ink washes, further expanding the creative potential of the application.

- **Practical Applications:** The application's capabilities can be harnessed for various practical purposes. For instance, educators in art or visual communication courses could utilize it to create quick visual representations for teaching purposes. Content creators in fields like graphic design or web development might find the generated sketches valuable assets for their projects. Additionally, the application could be a fun and accessible tool for social media users to add an artistic touch to their content.

CHAPTER -5

CONCLUSION & FUTURE WORK

5.1 Conclusion:

The primary objective of the project was to utilize image processing techniques to transform digital photos into pencil sketches, with a focus on imitating the visual features of a pencil drawing. This involved converting colourful images into grayscale to achieve the desired effect.

The process of turning an image into a pencil sketch is a fascinating blend of modern technology and traditional artistry. This technique captures the intricate details and textures of the original image through a series of hand-drawn strokes that mimic the look of a pencil. The result is often nostalgic and artistic, evoking emotions and capturing the viewer's attention in a unique and captivating way. Whether used for personal enjoyment or professional projects, the conversion of images to pencil sketches is a beautiful way to bridge the gap between the digital and analog worlds.

This project involves converting colourful images into grayscale to achieve the desired effect. This demonstrates the effectiveness of established image processing techniques combined with the Kivy framework for creating an interactive and user-centric tool. The Kivy framework facilitates a seamless user experience, allowing users to interact with the application intuitively.

5.2 Future Work:

This project establishes a solid foundation for an image-to-sketch application. Here's a glimpse into some exciting possibilities for future development:

- **Fine-tuning the Look:**
  - Users can control shading intensity to personalize the sketch's appearance.
  - Experimenting with different blur levels and potentially colour palettes can further enhance artistic expression.
- **Artistic Exploration:**
  - Move beyond pencil sketches to explore techniques for charcoal or ink drawings.
  - This might involve incorporating additional image processing methods.

- **Advanced Techniques:**
  - Investigate ways to capture texture information, adding depth and detail to the sketches.
  - Deep learning approaches hold promise for artistic style transfer, allowing users to mimic renowned artists' styles.

These advancements can transform the application into a powerful tool for artistic exploration and personalized image manipulation.

APPENDIX A:
RESEARCH PAPER

Image to pencil recognition using Machine Learning Techniques and Open CV

Dr. Archana Kumar *[1]
Professor, H.O.D
(AI & DS)

Kasak *[2]
arorakasak2005@gmail.com

Jyoti Rana *[2]
jyotiserana@gmail.com

Scholar B.Tech. (AI&DS) 2nd Year

Department of Artificial Intelligence and Data Science,
Dr. Akhilesh Das Gupta Institute of Professional Studies, New Delhi

Abstract: ***The "Image to Pencil Sketch" paper investigates the crossroads of image processing and artistic expression by creating an application that turns digital images into genuine pencil sketches. This paper aims to devise an algorithm or application that automatically converts digital images into artistic sketches. Examining the original image and generating a new image that captures its essence in a stylized, hand-drawn way, will bridge the gap between photography and drawing. The primary goal is to make art more accessible to a wider audience and democratize its use. This paperwork employs creativity to provide users with a fun and artistic way to transform their photos into sketches using machine learning and Open CV. It transforms digital images into realistic and aesthetically pleasing pencil sketches.***

Keywords- Image processing, Grayscale conversion, Edge enhancement, Gaussian blurring, Pencil sketch, Image-to-sketch conversion

Abbreviations -

OpenCV - Open-Source Computer Vision Library

AI – Artificial Intelligence

GUI - Graphical User Interface

GANs - Generative Adversarial Networks

VAEs - Variational Autoencoders

SBL - Swing Bilateral LIC

LIC - Line Integral Convolution

GMED - Gradient Maps and Morphological Operations

1. Introduction

The study examines the development of an application that can convert digital images into pencil sketches in real-time. These conversions have various applications, including artistic exploration, image editing, and content creation. Traditional techniques for creating pencil sketches demand artistic skills and considerable time, whereas digital image-to-sketch conversion techniques offer a streamlined approach. The application utilizes established image processing techniques, such as the OpenCV [1,2] library in Python and the Kivy framework [3], to create an interactive and user-friendly graphical user interface. The technique is faster, more efficient**,** and easier to use than traditional manual drawing techniques. Overall, the application provides a seamless user experience for generating artistic outputs from digital photographs.

1.1 Applications

A Python-based Image to Pencil Sketch application has diverse practical uses in various fields such as Graphic design Education Marketing and Advertising The main purpose of image to pencil sketch application is to express artistic vision. Converting images to pencil sketches is a great way for artists to express themselves and add their own creative flair to photographs. This process provides a chance to experiment with various techniques and textures, resulting in visually captivating artworks that are unique and appealing. It plays an important role in the campaigns and advertising materials It helps to evoke feelings of nostalgia, authenticity, and emotional resonance with audiences. They are often incorporated into advertisements, product illustrations, and branding materials to effectively convey brand messages and establish a connection with consumers.

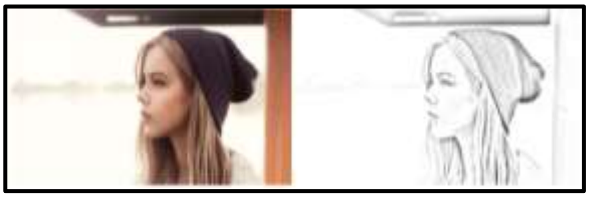


Fig.1 Image to pencil Sketch

1.2 Role of different fields

The technology and software development industry plays a vital role in the creation of the Image to Pencil Sketch by providing the necessary tools and frameworks. Ongoing research and development efforts in fields like computer vision, machine learning, and deep learning contribute to enhancing the app's algorithms. Collaboration with accessibility and inclusion organizations ensures that the app is designed to meet the specific needs of individuals with hearing impairments, incorporating their valuable guidance and support. Through these collective efforts, the technology, software development, accessibility, and education sectors collaborate to create an Image to Pencil Sketch app that promotes accessibility, inclusion, and effective conversion.

1.3 Recent Advancements in Image-to-Pencil Sketch Conversion

In recent years, Image to Pencil Sketch has witnessed notable advancements driven by. deep learning techniques such as GANs [4], VAEs [5]. Texture analysis and synthesis techniques, artistic style transfer, and user control are being developed. Researchers are also exploring real-time applications for more interactive user experiences. Artistic style transfer lets users mimic famous artists, while user-controlled parameters (detail, darkness, style) offer creative freedom. And help animators produce detailed sketches without spending countless hours drawing by hand. By creating high-quality pencil sketches from digital images, these advancements are paving the way for more sophisticated, user-friendly, and artistically expressive tools in the field of image-to-sketch conversion.

1.4 Challenges

Despite advancements, image-to-sketch conversion faces hurdles. Capturing fine details and achieving the desired balance between realism and artistic abstraction remain challenges. Real-time processing struggles with complex images and current methods cannot fully capture artistic intent. Greater user control over stroke style, element emphasis, and style mimicry is needed. Combining multiple images seamlessly and addressing ethical concerns regarding style mimicry are also areas for future research. These challenges hold the key to unlocking the full potential of image-to-sketch conversion**.**

2. Literature review

Several researches have contributed to advancements in “image-to-pencil sketch” recognition. In recent research [11], an AI-powered system was developed to help beginners learn basic pencil drawing techniques. It assesses the user's sketches and provides guidance based on the motif data through four subsystems, including motif feature extraction, sketch feature extraction, error identification, and advice generation and presentation. The advice is presented through a 3D model, which helps users understand the errors in their sketches. In a separate investigation [12], introduced a new algorithm that turns personal photos into pencil sketch-like drawings using gradient transformation and final smoothing for visually striking results. Further studies result in [13], in the generation of automatic pencil drawings using Line Integral Convolution (LIC) [8], the goal of the algorithm is to enhance image quality by improving image segmentation and texture direction detection techniques. It is achieved by a graph-based segmentation algorithm and a region-based approach for creating white noise and texture directions, resulting in pencil drawings that closely resemble real artistic styles. In another study [14], the SBL method is introduced which creates pencil drawings in various styles with a filter to control the direction and colour of pencil strokes. extracting linear features, generating a noise distribution, and determining noise values to achieve the desired pencil drawing effects. And one more study result [15] it involves two stages: generating a stroke layer to represent shapes and producing tonal textures to depict brightness and shades. These layers are combined to create a non-photorealistic pencil drawing. Another study [16] a two-branch model is developed to generate sketchy outlines and tonal shading from pencil drawings. Clean outlines and tonal illustrations are extracted from the original drawings, and the model creates different pencil styles in a user-controllable manner. The next study [17] the technique extracts a direction field, generates stroke paths, and renders pencil strokes with consideration of image tone and illumination. It can automatically produce pencil sketches from images with little user interaction. In one separate study [18] introduced a new algorithm called GMED which can create high-quality pencil drawings from natural images. It uses gradient maps and morphological operations to extract lines, and texture filling and tone mapping to create a realistic effect. One more study [19] proposes a method for generating feature-preserving colour pencil drawings from photographs by enhancing lightness and reducing saturation The approach includes devising lightness enhancement and saturation reduction mappings to mimic the tonal style of colour pencil drawings, resulting in superior tone capture and feature preservation.

3. Methodology

This research proposes a method for image-to-pencil sketch using OpenCV and machine learning techniques. The proposed method consists of the following steps:

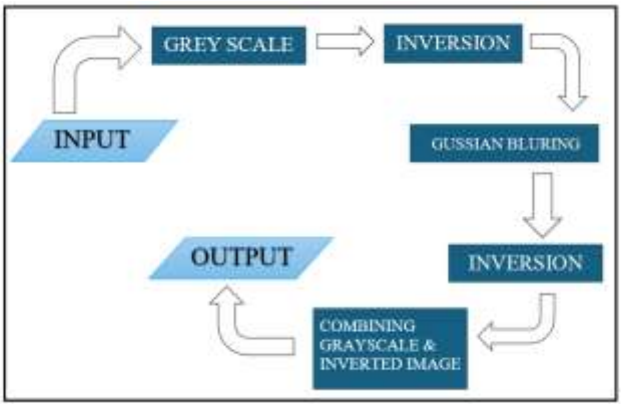


Fig.2 Flowchart of the steps followed.

3.1 Preprocessing

- **Image Reading:** The process starts by reading the user-selected image with OpenCV's cv2.imread function, which stores it as a NumPy [7] array for further processing.
- **Grayscale Conversion:** The loaded colour image is converted to grayscale to focus on shading and texture, discarding colour information. It helps in simplifying algorithms and as well eliminates the complexities related to computational requirements. [8]

3.2 Edge Enhancement

- **Image Inversion [9]:** The image is inverted using a bitwise NOT operation, which flips the intensity values.

- **Gaussian Blurring:** A Gaussian blur [10] filter is applied to the image for noise reduction and to preserve significant edges.

- **Second Inversion:** After blurring, the image is inverted twice with bitwise NOT, amplifying edges for sketch generation.

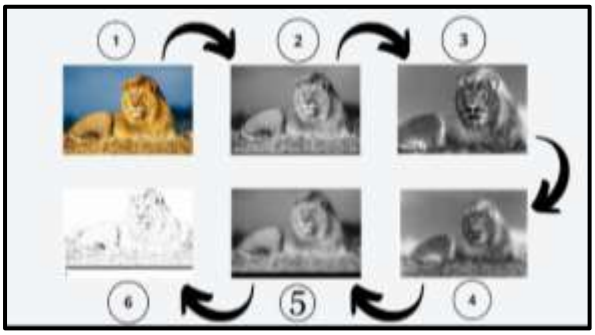


Fig.3 Input to Output Process

4. Conclusion

The primary objective of the paper was to utilize image processing techniques to transform digital photos into pencil sketches, with a focus on imitating the visual features of a pencil drawing. This involved converting colourful images into grayscale to achieve the desired effect. This demonstrates the effectiveness of established image processing techniques combined with the

Kivy framework for creating an interactive and user-centric tool. The Kivy framework facilitates a seamless user experience, allowing users to interact with the application intuitively.

5. Future scope

For future directions, the paper proposes the possibility of developing a mobile application that can explore more complex algorithms for detecting edges that will improve the details of the sketch. Furthermore, there are plans to expand research methods to include shading and texture variations, which will make the sketch appear more lifelike. Also, the team will focus on developing a user interface that permits the user to customize the sketch style, such as adjusting line thickness and contrast, which would be beneficial. These future steps will contribute to further advancements in Image to Pencil Sketch.

6. References

[1] About - OpenCV (https://opencv.org/about/)

[2] What is OpenCV? The Complete Guide (2024) - viso.ai (https://viso.ai/computer-vision/opencv/)

[3] Introduction to Kivy ; A Cross-platform Python Framework - GeeksforGeeks (https://www.geeksforgeeks.org/introduction-to-kivy/)

[4]Introduction | Machine Learning | Google for Developers (https://developers.google.com/machine-learning/gan)

[5] Understanding Variational Autoencoders (VAEs) | by Joseph Rocca | Towards Data Science (https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73?gi=295d841ff1fb)

[6] LIC (Line Integral Convolution) / LIC Source Code (zhanpingliu.org) (http://www.zhanpingliu.org/research/flowvis/lic/lic.htm)

[7] numpy · PyPI (https://pypi.org/project/numpy/)

[8] Why to use Grayscale Conversion during Image Processing? (isahit.com) (https://www.isahit.com/blog/why-to-use-grayscale-conversion-during-image-processing)

[9] Image Processing Algorithms Part 7: Colour Inversion And Solarisation | Dreamland Fantasy Studios (dfstudios.co.uk) (https://www.dfstudios.co.uk/articles/programming/image-programming-

algorithms/image-processing-algorithms-part-7-colour-inversion-solarisation/)

[10] Image Processing with Python: Blurring Images (datacarpentry.org) (https://datacarpentry.org/image-processing/06-blurring.html)

[11] Saeko Takagi†, Noriyuki Matsuda, Masato Soga, Hirokazu Taki, Takashi Shima, Fujiichi Yoshimoto "An Educational Tool for Basic Techniques in Beginner's Pencil Drawing" Proceedings of Computer Graphics International Conference, CGI · August 2003.

[12]Jin Zhou, Baoxin Li "AUTOMATIC GENERATION OF PENCIL-SKETCH LIKE DRAWINGS" 0-7803-9332-5/05/$20.00 ©2005 IEEE.

[13] Shuo Sun Tianjin, Dongwei Huang Tianjin "Efficient Region-Based Pencil Drawing" WSCG '2007: Full s Proceedings University of West Bohemia Plzen Czech Republic, January 29 – February 1, 2007, p. 279-286.

[14] Heekyung Yang, Yunmi Kwon, & Kyungha Min "A Stylized Approach for Pencil Drawing from Photographs" Eurographics Symposium on Rendering 2012 Fredo Durand and Diego Gutierrez.

[15] Jiang, Yifeng "Combining Sketch and Tone for Pencil Drawing Rendering" Rensselaer Polytechnic Institute Department of Electrical, Computer, and Systems Engineering ECSE 4540: Introduction to Image Processing, Spring 2015.

[16]Yijun Li, Chen Fang, Aaron Hertzmann, Eli Shechtman, Ming-Hsuan Yang "Im2Pencil: Controllable Pencil Illustration from Photographs". IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1525-1534.

[17] Zhengyan Tong, Xuanhong Chen, Bingbing Ni, Xiaohang Wang "Sketch Generation with Drawing Process Guided by Vector Flow and Grayscale" The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21).

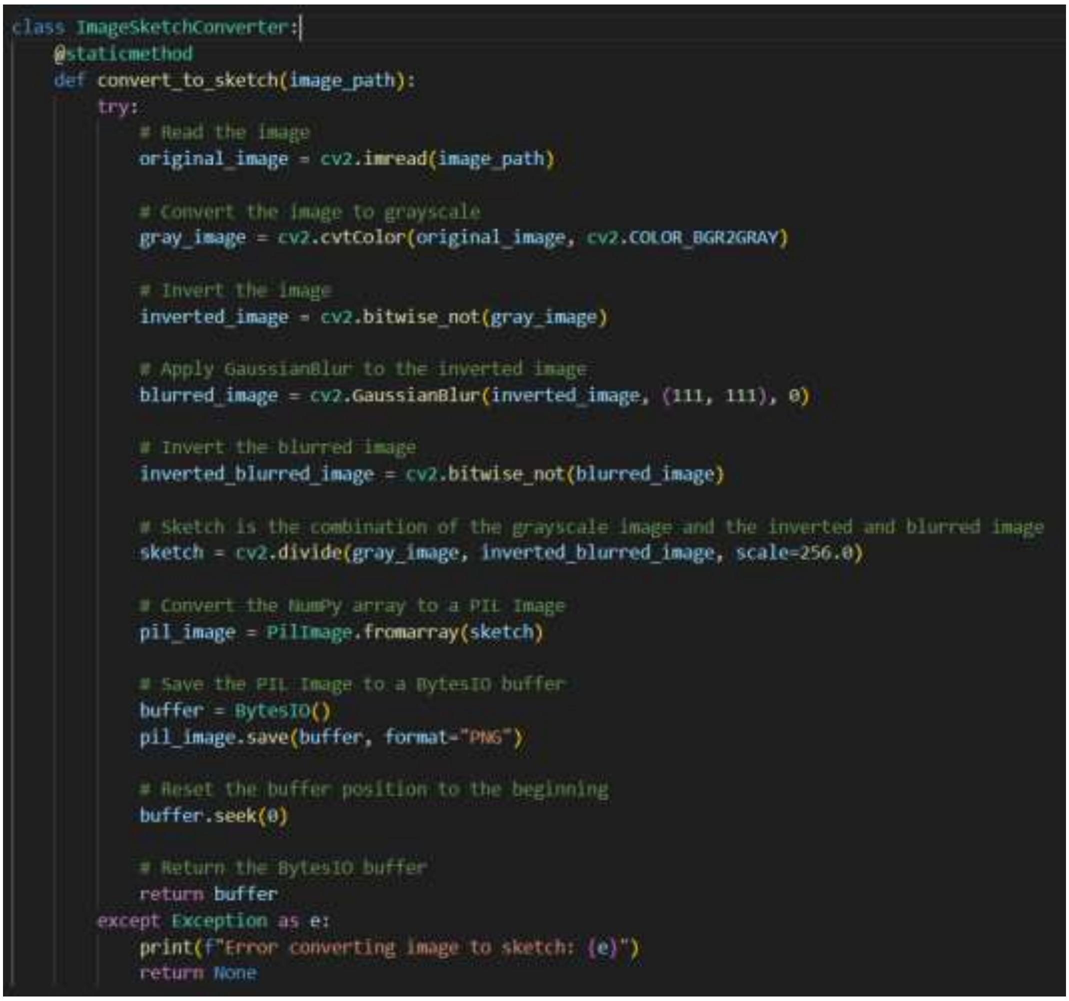
[18] HUAPING ZHOU, CHAO ZHOU, AND XIAOYAN WANG "Pencil Drawing Generation Algorithm Based on GMED" Digital Object Identifier 10.1109/ACCESS.2021.3065428.

[19] Dong Wang, Guiqing Li, Chengying Gao, Shengwu Fu, and Yun Liang, "Feature-preserving colour pencil drawings from photographs" Computational Visual Media, https://doi.org/10.1007/s41095-022-0320-6 Vol. 9, No. 4, December 2023, 807–825

APPENDIX B:
SURVEY DATA OR TYPICAL PART OF SOURCE CODE

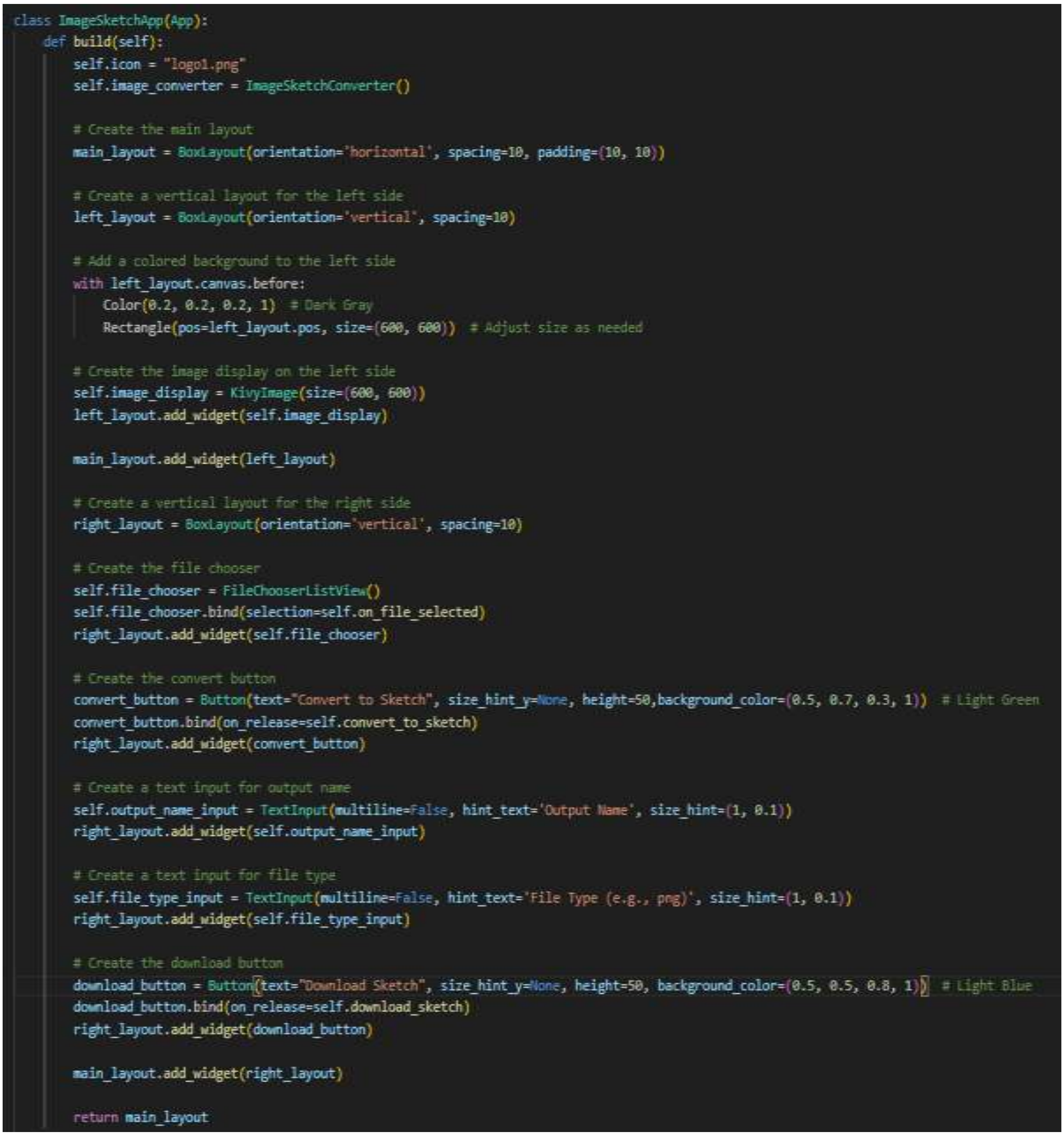
Typical Part of Source Code:

1. **Imports:**
   - **os:** for interacting with the operating system (e.g., file paths).
   - **cv2:** OpenCV library for image processing tasks (e.g., reading, converting, blurring).
   - **numpy:** NumPy library for numerical computations (often used with OpenCV).
   - **PIL (Pillow Fork):** for handling image formats (e.g., creating PIL Image from NumPy array).
   - **io:** for working with input/output streams (e.g., BytesIO buffer for image data).
   - **kivy:** Kivy framework for building the user interface (e.g., buttons, image display).
2. **Classes:**
   - **ImageSketchConverter:** This class encapsulates the logic for converting an image to a sketch. It has a static method convert_to_sketch that takes an image path, performs the conversion steps and returns a BytesIO buffer containing the sketch image data.



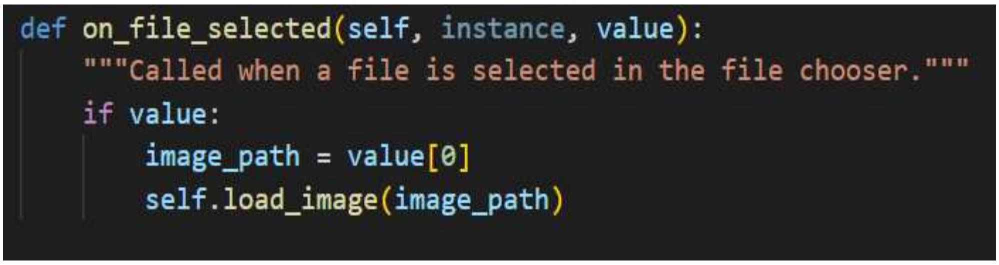
```
class ImageSketchConverter:
    @staticmethod
    def convert_to_sketch(image_path):
        try:
            # Read the image
            original_image = cv2.imread(image_path)

            # Convert the image to grayscale
            gray_image = cv2.cvtColor(original_image, cv2.COLOR_BGR2GRAY)

            # Invert the image
            inverted_image = cv2.bitwise_not(gray_image)

            # Apply GaussianBlur to the inverted image
            blurred_image = cv2.GaussianBlur(inverted_image, (111, 111), 0)

            # Invert the blurred image
            inverted_blurred_image = cv2.bitwise_not(blurred_image)

            # Sketch is the combination of the grayscale image and the inverted and blurred image
            sketch = cv2.divide(gray_image, inverted_blurred_image, scale=256.0)

            # Convert the NumPy array to a PIL Image
            pil_image = PilImage.fromarray(sketch)

            # Save the PIL Image to a BytesIO buffer
            buffer = BytesIO()
            pil_image.save(buffer, format="PNG")

            # Reset the buffer position to the beginning
            buffer.seek(0)

            # Return the BytesIO buffer
            return buffer
        except Exception as e:
            print(f"Error converting image to sketch: {e}")
            return None
```

- ImageSketchApp: This class inherits from kivy.app.App and is responsible for building the application's user interface and handling user interactions. It defines the following methods:
  - build: Creates the main layout with two sub-layouts (left for image display, right for controls).



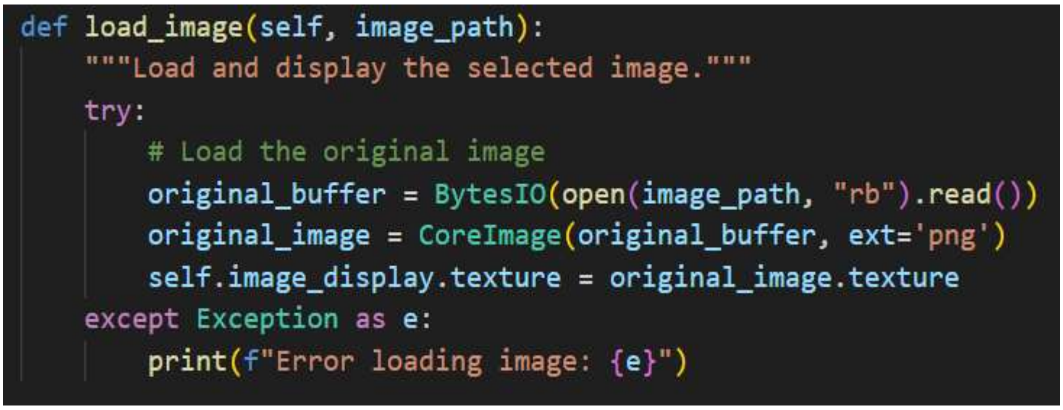
```
class ImageSketchApp(App):
    def build(self):
        self.icon = "logo1.png"
        self.image_converter = ImageSketchConverter()

        # Create the main layout
        main_layout = BoxLayout(orientation='horizontal', spacing=10, padding=(10, 10))

        # Create a vertical layout for the left side
        left_layout = BoxLayout(orientation='vertical', spacing=10)

        # Add a colored background to the left side
        with left_layout.canvas.before:
            Color(0.2, 0.2, 0.2, 1)  # Dark Gray
            Rectangle(pos=left_layout.pos, size=(600, 600))  # Adjust size as needed

        # Create the image display on the left side
        self.image_display = KivyImage(size=(600, 600))
        left_layout.add_widget(self.image_display)

        main_layout.add_widget(left_layout)

        # Create a vertical layout for the right side
        right_layout = BoxLayout(orientation='vertical', spacing=10)

        # Create the file chooser
        self.file_chooser = FileChooserListView()
        self.file_chooser.bind(selection=self.on_file_selected)
        right_layout.add_widget(self.file_chooser)

        # Create the convert button
        convert_button = Button(text="Convert to Sketch", size_hint_y=None, height=50,background_color=(0.5, 0.7, 0.3, 1))  # Light Green
        convert_button.bind(on_release=self.convert_to_sketch)
        right_layout.add_widget(convert_button)

        # Create a text input for output name
        self.output_name_input = TextInput(multiline=False, hint_text='Output Name', size_hint=(1, 0.1))
        right_layout.add_widget(self.output_name_input)

        # Create a text input for file type
        self.file_type_input = TextInput(multiline=False, hint_text='File Type (e.g., png)', size_hint=(1, 0.1))
        right_layout.add_widget(self.file_type_input)

        # Create the download button
        download_button = Button(text="Download Sketch", size_hint_y=None, height=50, background_color=(0.5, 0.5, 0.8, 1))  # Light Blue
        download_button.bind(on_release=self.download_sketch)
        right_layout.add_widget(download_button)

        main_layout.add_widget(right_layout)

        return main_layout
```

- on_file_selected: Called when a file is chosen in the file chooser, updates the image display.



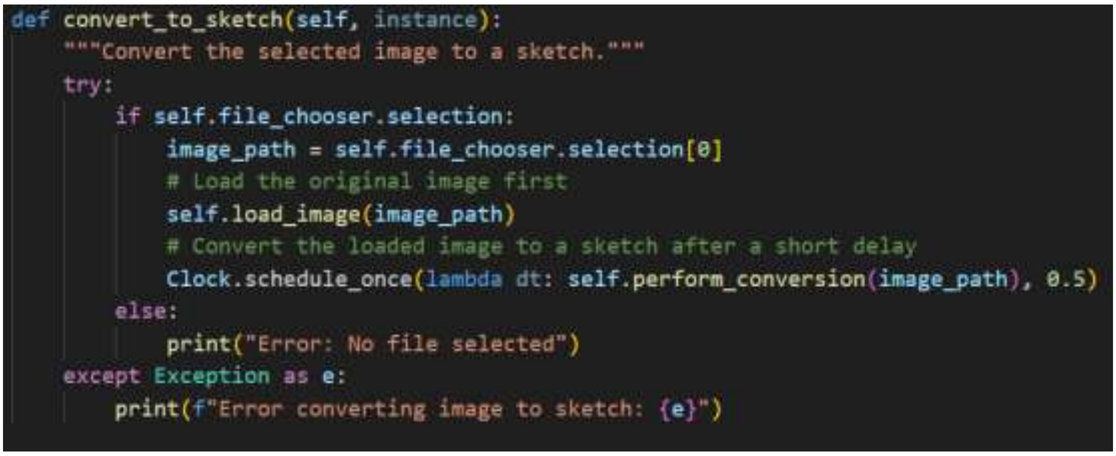
```python
def on_file_selected(self, instance, value):
    """Called when a file is selected in the file chooser."""
    if value:
        image_path = value[0]
        self.load_image(image_path)
```

- load_image: Loads the selected image and displays it.



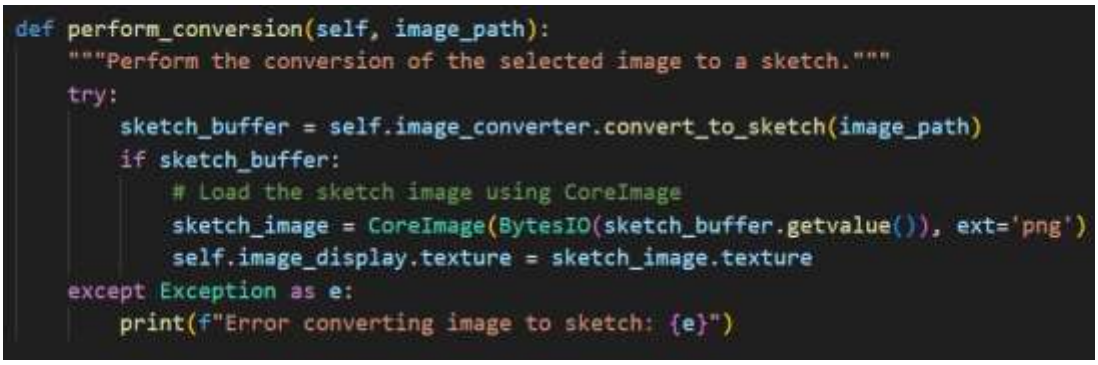
```python
def load_image(self, image_path):
    """Load and display the selected image."""
    try:
        # Load the original image
        original_buffer = BytesIO(open(image_path, "rb").read())
        original_image = CoreImage(original_buffer, ext='png')
        self.image_display.texture = original_image.texture
    except Exception as e:
        print(f"Error loading image: {e}")
```

- convert_to_sketch: Initiates the image conversion process with a delay using Clock.schedule_once.



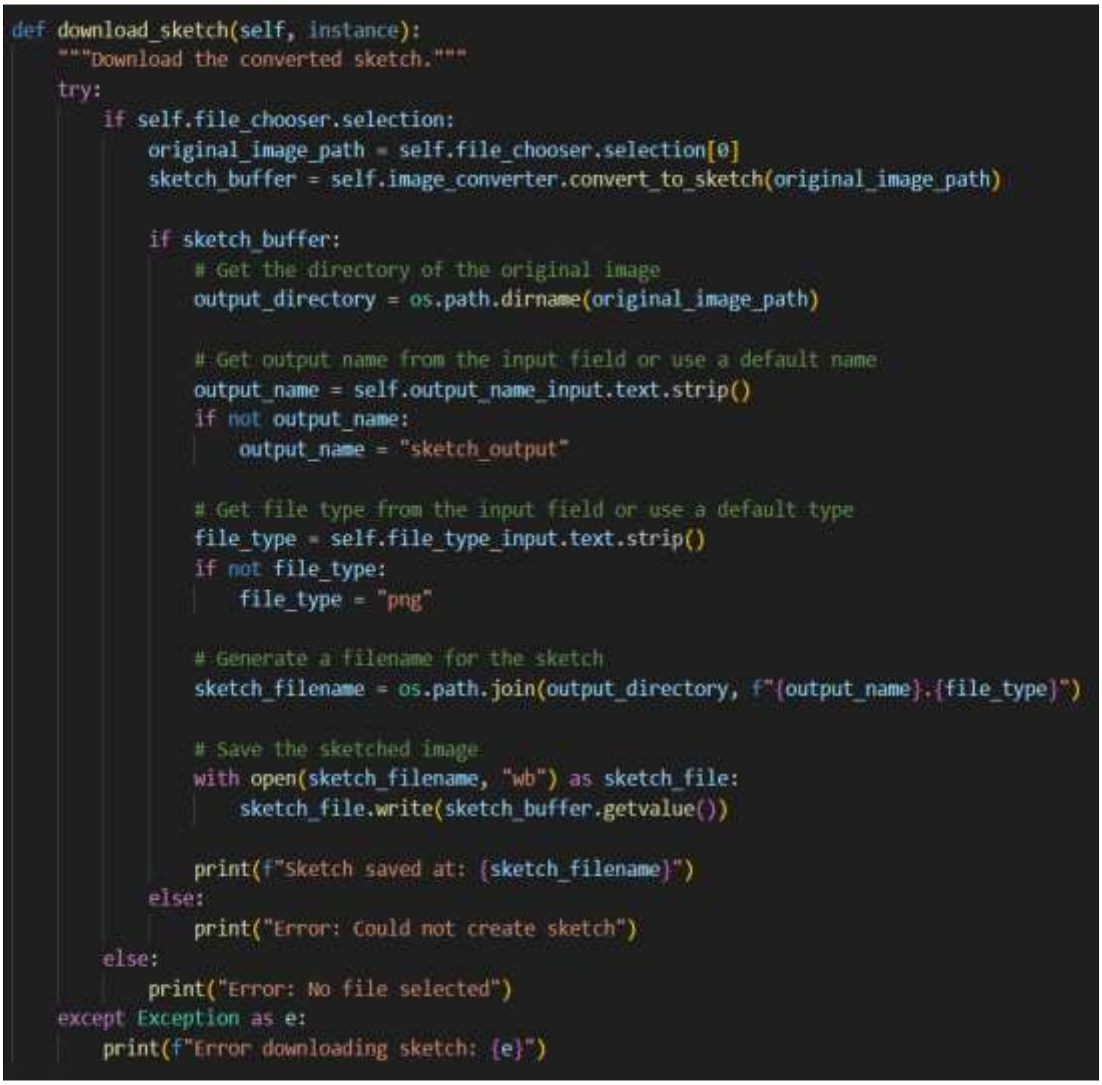
```python
def convert_to_sketch(self, instance):
    """Convert the selected image to a sketch."""
    try:
        if self.file_chooser.selection:
            image_path = self.file_chooser.selection[0]
            # Load the original image first
            self.load_image(image_path)
            # Convert the loaded image to a sketch after a short delay
            Clock.schedule_once(lambda dt: self.perform_conversion(image_path), 0.5)
        else:
            print("Error: No file selected")
    except Exception as e:
        print(f"Error converting image to sketch: {e}")
```

- perform_conversion: Performs the actual conversion using image_converter and updates the image display with the sketch.



```python
def perform_conversion(self, image_path):
    """Perform the conversion of the selected image to a sketch."""
    try:
        sketch_buffer = self.image_converter.convert_to_sketch(image_path)
        if sketch_buffer:
            # Load the sketch image using CoreImage
            sketch_image = CoreImage(BytesIO(sketch_buffer.getvalue()), ext='png')
            self.image_display.texture = sketch_image.texture
    except Exception as e:
        print(f"Error converting image to sketch: {e}")
```

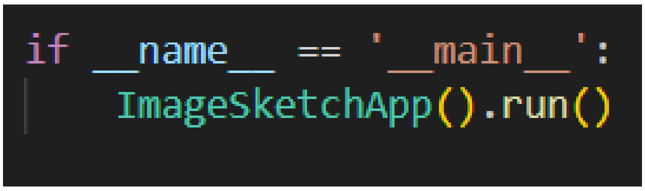
- download_sketch: Downloads the converted sketch based on user-provided output name and file type or defaults.



```python
def download_sketch(self, instance):
    """Download the converted sketch."""
    try:
        if self.file_chooser.selection:
            original_image_path = self.file_chooser.selection[0]
            sketch_buffer = self.image_converter.convert_to_sketch(original_image_path)

            if sketch_buffer:
                # Get the directory of the original image
                output_directory = os.path.dirname(original_image_path)

                # Get output name from the input field or use a default name
                output_name = self.output_name_input.text.strip()
                if not output_name:
                    output_name = "sketch_output"

                # Get file type from the input field or use a default type
                file_type = self.file_type_input.text.strip()
                if not file_type:
                    file_type = "png"

                # Generate a filename for the sketch
                sketch_filename = os.path.join(output_directory, f"{output_name}.{file_type}")

                # Save the sketched image
                with open(sketch_filename, "wb") as sketch_file:
                    sketch_file.write(sketch_buffer.getvalue())

                print(f"Sketch saved at: {sketch_filename}")
            else:
                print("Error: Could not create sketch")
        else:
            print("Error: No file selected")
    except Exception as e:
        print(f"Error downloading sketch: {e}")
```

3. **Main Execution:**

- The if __name__ == '__main__': block ensures the code within this block only executes when the script is run directly (not imported as a module).
- ImageSketchApp().run(): Creates an instance of ImageSketchApp and starts the Kivy application.



```python
if __name__ == '__main__':
    ImageSketchApp().run()
```

REFERENCES

[1] Saeko Takagi†, Noriyuki Matsuda, Masato Soga, Hirokazu Taki, Takashi Shima, Fujiichi Yoshimoto *"An Educational Tool for Basic Techniques in Beginner's Pencil Drawing"* Proceedings of Computer Graphics International Conference, CGI · August 2003.

[2] Jin Zhou, Baoxin Li *"AUTOMATIC GENERATION OF PENCIL-SKETCH LIKE DRAWINGS"* 0-7803-9332-5/05/$20.00 ©2005 IEEE.

[3] Shuo Sun Tianjin, Dongwei Huang Tianjin *"Efficient Region-Based Pencil Drawing"* WSCG '2007: Full s Proceedings University of West Bohemia Plzen Czech Republic, January 29 – February 1, 2007, p. 279-286.

[4] Heekyung Yang, Yunmi Kwon, & Kyungha Min *"A Stylized Approach for Pencil Drawing from Photographs"* Eurographics Symposium on Rendering 2012 Fredo Durand and Diego Gutierrez.

[5] Jiang, Yifeng *"Combining Sketch and Tone for Pencil Drawing Rendering"* Rensselaer Polytechnic Institute Department of Electrical, Computer, and Systems Engineering ECSE 4540: Introduction to Image Processing, Spring 2015.

[6] Yijun Li, Chen Fang, Aaron Hertzmann, Eli Shechtman, Ming-Hsuan Yang *"Im2Pencil: Controllable Pencil Illustration from Photographs".* IEEE/CVF Conference 7 on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1525-1534.

[7] HUAPING ZHOU, CHAO ZHOU, AND XIAOYAN WANG *"Pencil Drawing Generation Algorithm Based on GMED"* Digital Object Identifier 10.1109/ACCESS.2021.3065428.

[8] Zhengyan Tong, Xuanhong Chen, Bingbing Ni, Xiaohang Wang *"Sketch Generation with Drawing Process Guided by Vector Flow and Grayscale"* The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21).

[9] Dong Wang, Guiqing Li, Chengying Gao, Shengwu Fu, and Yun Liang, *"Feature-preserving colour pencil drawings from photographs"* Computational Visual Media, https://doi.org/10.1007/s41095-022-0320-6 Vol. 9, No. 4, December 2023, 807–825.

[10] About - OpenCV (https://opencv.org/about/)

[11] What is OpenCV? The Complete Guide (2024) - viso.ai (https://viso.ai/computervision/opencv/)

[12] Introduction to Kivy ; A Cross-platform Python Framework - GeeksforGeeks (https://www.geeksforgeeks.org/introduction -to-kivy/)

[13]Introduction | Machine Learning | Google for Developers Fig.3 Input to Output Process 6 (https://developers.google.com/machinelearning/gan)

[14] Understanding Variational Autoencoders (VAEs) | by Joseph Rocca | Towards Data Science (https://towardsdatascience.com/understandi ng-variational-autoencoders-vaesf70510919f73?gi=295d841ff1fb)

[15] LIC (Line Integral Convolution) / LIC Source Code (zhanpingliu.org) (http://www.zhanpingliu.org/research/flowvi s/lic/lic.htm)

[16] numpy · PyPI (https://pypi.org/project/numpy/)

[17] Why to use Grayscale Conversion during Image Processing? (isahit.com) (https://www.isahit.com/blog/why-to-usegrayscale-conversion-during-imageprocessing)

[18] Image Processing Algorithms Part 7: Colour Inversion And Solarisation | Dreamland Fantasy Studios (dfstudios.co.uk) (https://www.dfstudios.co.uk/articles/progra mming/image-programmingalgorithms/image-processing-algorithmspart-7-colour-inversion-solarisation/)

[19] Image Processing with Python: Blurring Images (datacarpentry.org) (https://datacarpentry.org/imageprocessing/06-blurring.html)

DR. AKHILESH DAS GUPTA INSTITUTE OF PROFESSIONAL STUDIES

(FORMERLY ADGITM)

Approved by AICTE & BCI and Affiliated to GGSIPU

FC-26, Shastri Park, New Delhi-110053

National Conference on Bridging Innovations for a Sustainable Future in Electrical, Mechanical, Electronics and Communication Engineering – 2024

NCBISF-24

CERTIFICATE

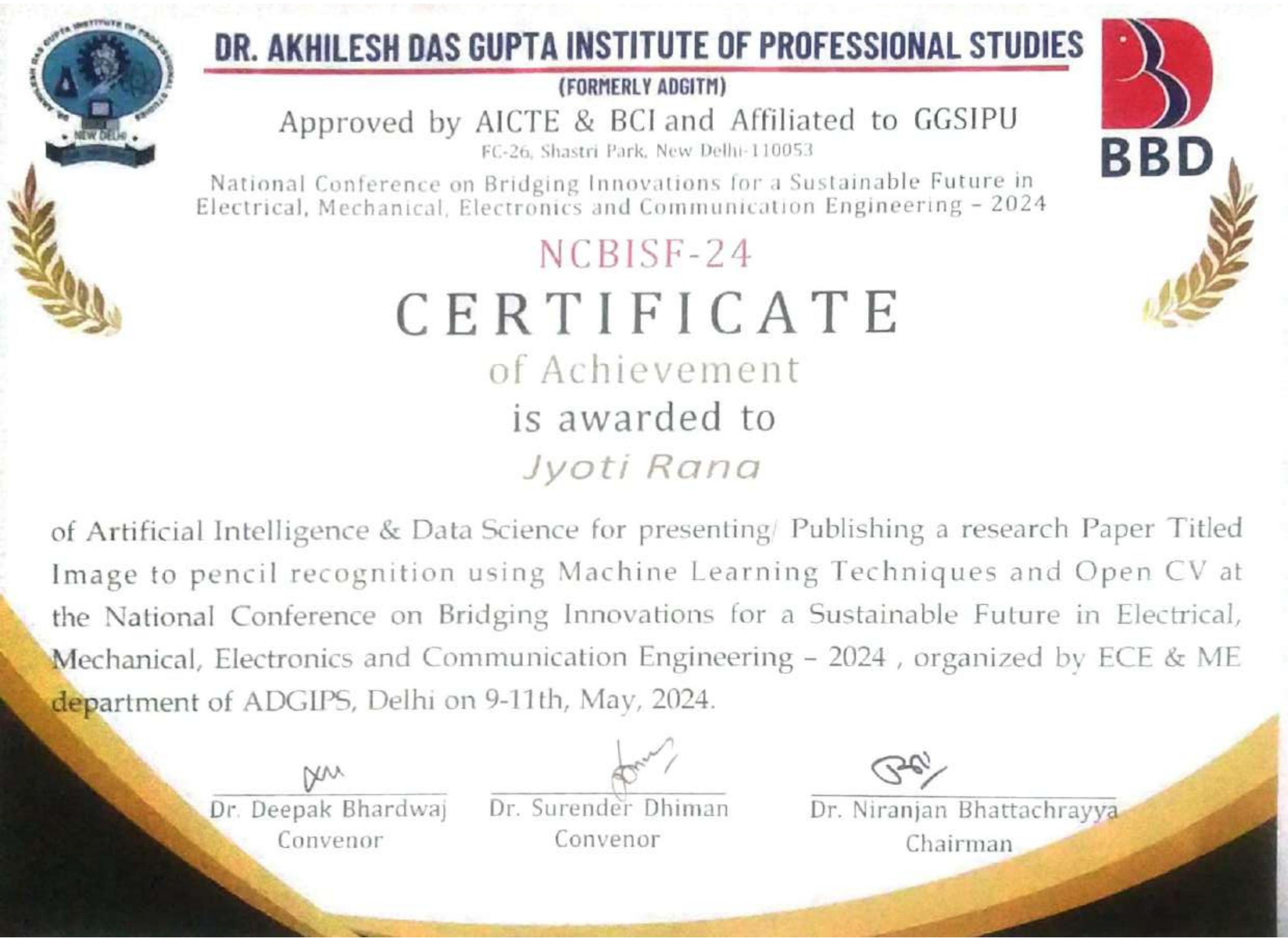
of Achievement

is awarded to

Kasak

of Artificial Intelligence & Data Science for presenting/ Publishing a research Paper Titled Image to pencil recognition using Machine Learning Techniques and Open CV at the National Conference on Bridging Innovations for a Sustainable Future in Electrical, Mechanical, Electronics and Communication Engineering – 2024 , organized by ECE & ME department of ADGIPS, Delhi on 9-11th, May, 2024.

Dr. Deepak Bhardwaj Convenor	Dr. Surender Dhiman Convenor	Dr. Niranjan Bhattachrayya Chairman



DR. AKHILESH DAS GUPTA INSTITUTE OF PROFESSIONAL STUDIES

(FORMERLY ADGITM)

Approved by AICTE & BCI and Affiliated to GGSIPU

FC-26, Shastri Park, New Delhi-110053

National Conference on Bridging Innovations for a Sustainable Future in
Electrical, Mechanical, Electronics and Communication Engineering - 2024

NCBISF-24

CERTIFICATE

of Achievement

is awarded to

Jyoti Rana

of Artificial Intelligence & Data Science for presenting/ Publishing a research Paper Titled Image to pencil recognition using Machine Learning Techniques and Open CV at the National Conference on Bridging Innovations for a Sustainable Future in Electrical, Mechanical, Electronics and Communication Engineering - 2024 , organized by ECE & ME department of ADGIPS, Delhi on 9-11th, May, 2024.

Dr. Deepak Bhardwaj Convenor	Dr. Surender Dhiman Convenor	Dr. Niranjan Bhattachrayya Chairman

Image to Pencil Sketch

KASAK JYOTI RANA

arorakasak2005@gmail.com jyotiserana2005@gmail.com

Dr. Akhilesh Das Gupta Institute of Professional Studies, GGSIPU

CASE ANALYSIS

The Image to Pencil Sketch Conversion Bridges the Gap Between Photography and Drawing: It allows you to capture the essence of a photograph in a stylized, hand-drawn manner. with hearing impairments. Image to Pencil Sketch is a fundamental task in the Computer Vision, Image Processing, Machine Learning, and Artificial Intelligence (AI) field. In this case analysis, we will examine a project focused on utilizing deep learning techniques for image-to-pencil conversion. The project aims to develop a system that automatically converts digital images into pencil sketch representations using image processing techniques. The primary goal is to make art more accessible to a wider audience and democratize its use.

The project needs a way to load digital images in a common format (e.g., JPEG, PNG). The project utilizes techniques for converting the image to grayscale, detecting edges, applying blur for shading, and potentially adjusting contrast. The project can save the generated sketch in a suitable image format.

Sketches are commonly encountered in various applications, such as Fashion Design, Forensic Sketching, Technical Communication, Product Design, Development, Landscape Architecture & Urban Planning. Developing an accurate image-to-sketch conversion system addresses a real-world problem and has practical applications across different domains. Image-to-pencil conversion has practical implications for various developments in fields for example Architects can convert digital renderings or photos of existing structures into sketches. This can generate historical illustrations, create preliminary design proposals, or communicate design ideas to clients who might find 3D models difficult to interpret. Fashion designers can convert photographs of models or clothing into pencil sketches. This can be used for creating initial design concepts, exploring garment details, and generating mood boards for presentations. These practical implications demonstrate the potential impact of Image to Pencil sketch conversion in various sectors, improving efficiency, accuracy, and automation in tasks.

An image-to-pencil sketch project offers a practical application of image processing techniques. By successfully converting digital images to artistic sketches, the project can contribute to creative workflows and enhance photo editing capabilities. Further research can explore advanced algorithms for style control, real-time processing, and integration with design software.

Image to Pencil Sketch

JYOTI RANA KASAK

jyotiserana2005@gmail.com arorakasak2005@gmail.com

Dr. Akhilesh Das Gupta Institute of Professional Studies, GGSIPU

IMPACT ANALYSIS

The project “Image to Pencil Sketch Conversion” brings several benefits to mankind and society and has significant implications and potential impact in various fields. Some major impact zones are:

Visual Communication: Pencil sketches often convey emotions, moods, and messages distinctively compared to digital images. They can be used in visual storytelling, graphic design, and other forms of communication to evoke specific responses from the audience.

Comic Book & Manga Creation: Artists can use image-to-sketch conversion to create base layers for comic panels or manga pages. This can save time on initial layouts and allow for quicker exploration of compositions and character poses based on reference photos.

Rapid Prototyping & Design Iteration: Product designers can quickly convert digital mock-ups or photos of physical objects into sketches. This allows for faster iteration of design ideas without manual sketching, facilitating exploration and communication with clients.

Architectural Visualization: Architects can convert digital renderings or photos of existing structures into sketches. This can be used for generating historical illustrations, creating preliminary design proposals, or communicating design ideas to clients who might find 3D models difficult to interpret.

Fashion Trend Analysis & Forecasting: Fashion analysts can use image-to-sketch tools to convert photos of street style or runway investigates simplified sketches.

In conclusion, image-to-pencil sketch conversion technology offers a valuable tool for various creative and practical applications. It empowers users to transform digital images into artistic sketches, fostering creative expression in photo editing, fashion design, and artistic endeavors. Additionally, the technology holds promise for streamlining workflows in product design, architecture, education, and content creation.