

A series of thin, black, overlapping geometric lines forming various polygons and shapes, primarily located in the upper left and center of the page.

PENCIL SKETCH EFFECT

TRANSFORMING DIGITAL IMAGES INTO
ARTISTIC SKETCHES

DONE BY:

RISHITHA T – E0322026

PROJECT OVERVIEW

What is Pencil Sketch Effect?

A computer vision application that converts regular photographs into pencil sketch-style artwork using image processing techniques.

Key Features:

- Grayscale pencil sketch conversion
- Colored pencil sketch effect
- Real-time webcam sketch filter
- Save and export functionality



TECHNOLOGIES USED

Programming Language:

- Python 3.x

Libraries:

- **OpenCV (cv2)** - Image processing and computer vision operations
- **NumPy** - Numerical computations and array manipulation

Hardware Requirements:

- Webcam (optional, for real-time feature)
- Standard computer with Python support



CORE CONCEPTS

Image Processing Techniques:

1. **Grayscale Conversion** - Reducing color complexity
2. **Image Inversion** - Creating negative images
3. **Gaussian Blur** - Smoothing and softening edges
4. **Image Division** - Dodge blend technique
5. **Color Space Manipulation** - BGR to Gray conversions

ALGORITHM – GRAYSCALE SKETCH

Step-by-Step Process:

1. **Load Image** - Read input image using OpenCV
2. **Convert to Grayscale** - Remove color information
3. **Invert Grayscale** - Create negative image
4. **Apply Gaussian Blur** - Blur the inverted image (21x21 kernel)
5. **Invert Blurred Image** - Create second inversion
6. **Divide Images** - Gray image \div Inverted blurred image
7. **Scale Result** - Multiply by 256 for proper intensity
8. **Result:** Realistic pencil sketch appearance

ALGORITHM – COLORED SKETCH

Using OpenCV's Built-in Function:

```
cv2.pencilSketch(image, sigma_s, sigma_r, shade_factor)
```

Parameters:

1. **sigma_s (60)** - Spatial window size, controls smoothness
2. **sigma_r (0.07)** - Range window size, controls color preservation
3. **shade_factor (0.05)** - Controls shading intensity

Output: Both grayscale and colored sketch versions

FEATURES – IMAGE CONVERSION

Static Image Processing:

- ✓ Load images from any path
- ✓ Support for multiple formats (JPG, PNG, BMP, TIFF)
- ✓ Side-by-side comparison view
- ✓ Optional save functionality
- ✓ Automatic file extension handling
- ✓ High-quality output preservation

Use Cases:

- Portrait sketching
- Artistic photo transformations
- Digital art creation

FEATURES – REALTIME WEBCAM

Live Sketch Effect:

- ✓ Real-time video processing
- ✓ Side-by-side original and sketch view
- ✓ Interactive screenshot capture (Press 'S')
- ✓ Smooth frame processing
- ✓ Easy exit option (Press 'Q')

Applications:

- Live demonstrations
- Video conferencing filters
- Interactive art installations

USER INTERFACE

Interactive Menu System:

Three main options:

- **Image to Pencil Sketch** - Convert static images
- **Image to Colored Sketch** - Create colored artwork
- **Webcam Pencil Sketch** - Real-time processing

User-Friendly Features:

Clear menu navigation

Input validation

Path handling (with quote removal)

Save prompts

Status messages

CODE ARCHITECTURE

Modular Design:

Function 1: `pencil_sketch()`

- Basic grayscale sketch conversion
- Manual image processing pipeline
- Single output image

Function 2: `pencil_sketch_colored()`

- Advanced colored sketch
- Uses OpenCV's `pencilSketch` function
- Dual output (gray and color)

Function 3: `webcam_pencil_sketch()`

- Real-time video processing
- Frame-by-frame conversion
- Interactive controls

TECHNICAL IMPLEMENTATION

Image Division Technique:

The core formula creates the sketch effect:

$\text{Sketch} = (\text{Gray Image} / \text{Inverted Blurred Image}) \times 256$

Why it works:

- Highlights edges and details
- Simulates pencil strokes
- Creates light/shadow contrast
- Mimics hand-drawn appearance

Gaussian Blur Purpose:

- Smooths harsh transitions
- Reduces noise
- Creates softer sketch lines

INPUT/OUTPUT EXAMPLES

Input Requirements:

- Any standard image format
- Clear, well-lit photographs work best
- Portrait or landscape orientation
- Minimum recommended: 640x480 resolution

Output Quality:

- Same resolution as input
- Grayscale or colored options
- JPEG format (default)
- Maintains aspect ratio



ADVANTAGES

Benefits of the Application:

- ✓ **Easy to Use** - Simple menu-driven interface
- ✓ **Fast Processing** - Efficient OpenCV algorithms
- ✓ **Versatile** - Multiple sketch styles available
- ✓ **Real-time Capable** - Live webcam processing
- ✓ **Free & Open Source** - Uses open-source libraries
- ✓ **Customizable** - Adjustable parameters
- ✓ **No Internet Required** - Works offline

LIMITATIONS & FUTURE ENHANCEMENTS

Potential Improvements:

- Add GUI using Tkinter or PyQt
- Batch processing for multiple images
- Adjustable parameters via sliders
- Additional artistic filters (oil painting, watercolor)
- Video file processing support
- Mobile app development
- AI-enhanced sketch generation

Current Limitations:

- Limited to predefined parameters
- No GUI (command-line only)
- Single image processing at a time
- Fixed blur kernel size

APPLICATIONS

Real-World Use Cases:

- **Digital Art** - Create artistic renditions of photos
- **Photography** - Add creative effects to portraits
- **Education** - Teach image processing concepts
- **Content Creation** - Social media filters
- **Profile Pictures** - Unique artistic avatars
- **Printing** - Prepare images for print sketches
- **Professional** - Quick sketch mockups for design

INSTALLATION & SETUP

Prerequisites:

`pip install opencv-python`

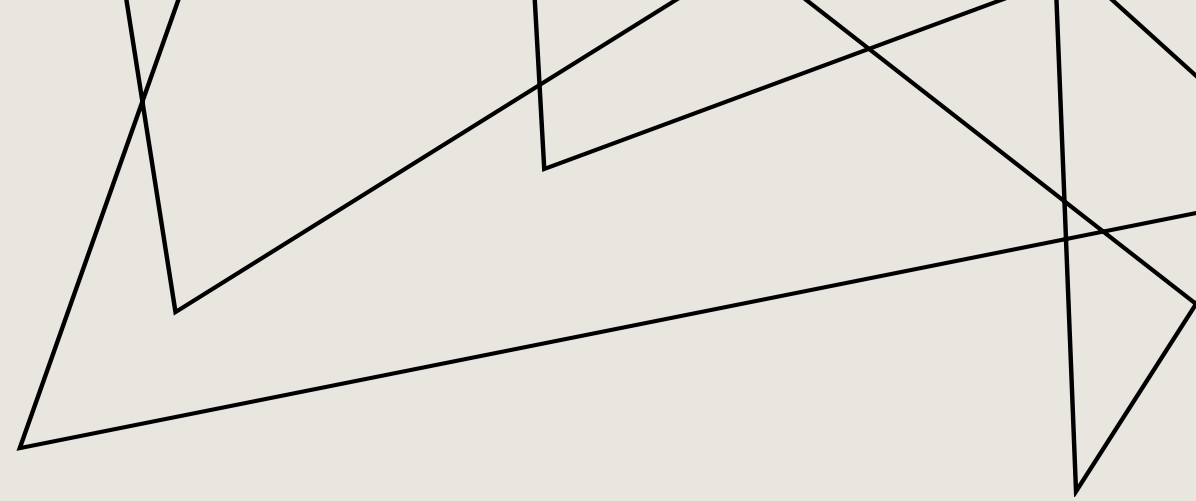
`pip install numpy`

Running the Program:

- Install required libraries
- Save the Python script
- Run: `python pencil_sketch.py`
- Follow on-screen menu prompts

System Requirements:

- Python 3.6 or higher
- 4GB RAM (minimum)
- Webcam (for real-time feature)



DEMO SCREENSHOTS

Original image vs. Pencil sketch comparison



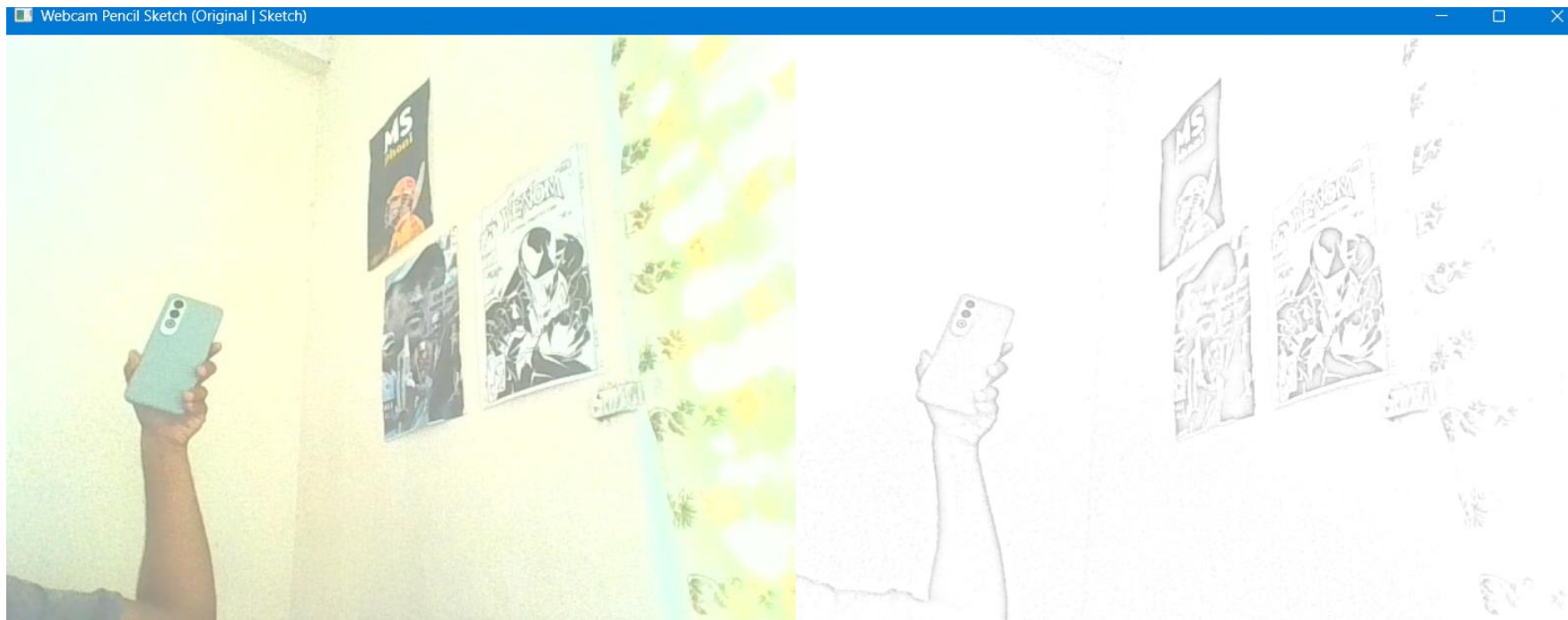
DEMO SCREENSHOTS

Colored sketch output example:



DEMO SCREENSHOTS

Webcam real-time processing screenshot





CONCLUSION

Project Summary:

Successfully implemented a pencil sketch effect application using computer vision techniques. The project demonstrates practical applications of image processing algorithms and provides multiple interaction modes for users.

Key Achievements:

- ✓ Three distinct processing modes
- ✓ Real-time video capabilities
- ✓ User-friendly interface
- ✓ Efficient algorithm implementation
- ✓ Flexible output options



REFERENCES

Technologies & Resources:

- OpenCV Documentation: docs.opencv.org
- NumPy Library: numpy.org
- Python Official Site: python.org
- Image Processing Tutorials
- Computer Vision Research Papers

Learning Resources:

- OpenCV Python Tutorials
- Digital Image Processing Fundamentals
- Computer Vision Applications

A series of white, thin, overlapping geometric lines on a black background, forming various polygons and intersecting points, primarily located on the left side of the slide.

THANK YOU