book

OpenCV

- introduction
- chapter 0
    - Introduction

      # Introduction

      Computer vision is the study of processing and analyzing images and videos for information.

      OpenCV handles tasks like filtering, object detection, face recognition, and deep learning. It works with TensorFlow, PyTorch, and Caffe. OpenCV is free and open source, compatible with Linux, Windows, and Mac OS X.

      OpenVINO optimizes deep learning models from various frameworks, deploys them on Intel hardware, and enhances performance and efficiency for computer vision applications on Intel devices.

      **

      # History of OpenCV

        - Beta 1999; (5 betas 2001-2005)
        - 1.0.0 version was released in 2006

    - 2.0.0 version was released in 2009 (C++)
    - 3.0.0 version was released in 2015 (fastest, stable)
    - 4.0.0 version was released in 2018 (deep learning)
    - 4.5.x version was released in 2021 (3D, RISC-V)
    - 5.0.0 version was released in 2023
- Install in Python

  ```
  pip install opencv-python
  ```

- Type in OpenCV

  # OpenCV type

    - CV_8U
        - 8-bit unsigned integers (0..255)
    - CV_8S
        - 8-bit signed integers (-128..127)
    - CV_16U
        - 16-bit unsigned integers (0..65535)
    - CV_16S
        - 16-bit signed integers (-32768..32767)
    - CV_32S
        - 32-bit signed integers (-2147483648..2147483647)
    - CV_32F

        - 32-bit floating-point numbers (-FLT_MAX..FLT_MAX, INF, NAN)
    - CV_64F
        - 64-bit floating-point numbers (-DBL_MAX..DBL_MAX, INF, NAN)

- chapter 1
    - Foundations

Basic

- OpenCV *imread* Function:
    - Channels are stored in BGR order by default.
    - Consider using the IMREAD_UNCHANGED flag to preserve the original image.
    - It is recommended to use a specific type, such as float32, to avoid potential issues.
    - Example usage: code 001,002,003,004, 005
- OpenCV *imshow* Function:
    - If the image type is not an integer, it may needs to be converted to an integer type or normalized before displaying.
    - To view the result, you must use the waitKey(x) function, where 'x' represents the delay time in milliseconds. Using waitKey(0) will

cause the window to wait until any key is pressed.

    - In larger projects, it is advisable to define a boolean variable to control the display of images throughout the program. This approach is useful for debugging purposes.
    - Example usage: code 001,002,003,004, 005
- OpenCV *imwrite* Function:
    - Example usage: code 006
    - Saving Images in Uncompressed Format: When working with images, it is recommended to save them in an uncompressed format whenever possible. This helps preserve the original quality and ensures that no additional compression artifacts are introduced. To save images in an uncompressed format using OpenCV, you can follow these tips:
        1. Choose an appropriate file format: Select a file format that supports uncompressed storage, such as TIFF (Tagged Image File Format) or BMP (Bitmap Image File). These formats store image data without any lossy compression.

2. Use the appropriate parameters: When saving the image, make sure to specify the appropriate parameters to ensure uncompressed storage. For example, in OpenCV, when using the imwrite() function, you can pass the appropriate parameters such as `cv2.IMWRITE_TIFF_COMPRESSION` or `cv2.IMWRITE_PNG_COMPRESSION` to specify the compression level. Set these parameters to their lowest values (e.g., 0 or None) to ensure uncompressed storage.
3. Preserve the original data: Before saving the image, ensure that the image data is in the desired format and has not undergone any compression or modification. If necessary, convert the image to the desired format or adjust its data type to ensure uncompressed storage.
   By following these guidelines, you can save images in an uncompressed format and maintain the original quality of

> > > the image data. This is especially important when working with critical applications that require high-fidelity images or when processing images for further analysis or research purposes.
>
> - OpenCV *resize* Function:
>   - Example usage: code 007
>   - The OpenCV `resize` function allows you to resize an image using different methods. You have the option to specify the desired size using `cv2.resize` or adjust the scaling factors `fx` and `fy`. There are three interpolation methods available: `INTER_CUBIC`, `INTER_LINEAR`, and `INTER_AREA`. It's worth noting that while `CUBIC` provides the best quality in most cases, it is slower compared to the other methods.

- chapter 1
- chapter 1
- chapter 1

Essential Python Tips And Tricks For advance computer vision Programmers
Essential Tips And Tricks For compiling code computer vision projects

- How to optimize your code for speed and efficiency
- How to debug your code and fix errors
- How to use OpenCV with other libraries and frameworks
- How to troubleshoot common problems
- How to extend the functionality of OpenCV with custom code