## **Core Module in OpenCV**

The core module is like the foundation of OpenCV. Almost everything in OpenCV depends on it. It contains the most basic stuff like data structures, array operations, math functions, and helper tools.

### **1. Basic Data Structures**

* Mat (Matrix) – This is the main way OpenCV stores images. An image is basically a matrix (2D or 3D array) of pixel values.  
   In Python, we usually work with NumPy arrays instead of directly using Mat.
* Scalar – Stores a 4-element value (often used for colors, like (B, G, R, Alpha)).
* Point – Stores a point with x and y coordinates (for marking positions in an image).
* Size – Holds width and height of an image.
* Rect – Defines a rectangular area (used for cropping or marking regions).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### **2. Bitwise Operations**

These work directly on the binary representation of pixels.

* AND: Keeps only the overlapping (white) parts of two images/masks.
* OR: Combines white areas from both images.
* XOR: Keeps non-overlapping white parts.
* NOT: Inverts the color of the image (white becomes black, black becomes white).

Example:

masked = cv2.bitwise\_and(img, img, mask=mask)

#### **3. Copying and Cloning Images**

* Copying just duplicates the array so we can edit one without affecting the original.
* Cloning can be deep copy (completely independent) or shallow copy (linked to the same data).

Example:

copy\_img = img.copy()

#### **4. Splitting and Merging Color Channels**

* Color images have 3 channels (Blue, Green, Red in OpenCV’s BGR order).

Example:  
b, g, r = cv2.split(img)

Or merge them back:  
Example:  
merged = cv2.merge((b, g, r))

This is useful when we want to work only on one color channel (Red or blue or green)

#### **5. Reshaping Data**

* Reshaping changes the array’s shape without altering the data.
* Example: turning a 2D grayscale image into a 1D array for machine learning.

#### **6. Masking Parts of an Image**

* Masking is a technique to keep functioning the process only at specific parts in the image.
* White (255) → process here.
* Black (0) → leave it alone.
* Example: Blur only a person’s face in a photo while keeping the rest sharp.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7. Utility Functions**

Some helpful tools in core:

* cv2.minMaxLoc() – Finds minimum and maximum pixel values in an image.
* cv2.mean() – Gets average color of an image.
* cv2.flip() – Flips an image with mirror image.
* cv2.addWeighted() – Blends two images together.
* cv2.inRange() – Checks if pixel values are within a certain range (useful for color detection).