

openCV

Tính năng:

- Xử lý và hiển thị Hình ảnh/ Video/ I/O (**core**, **imgproc**, **highgui**)
- Phát hiện các vật thể (**objdetect**, **features2d**, **nonfree**)
- Geometry-based monocular hoặc stereo computer vision (**calib3d**, **stitching**, **videostab**)
- Computational photography (**photo**, **video**, **superres**)
- Machine learning & clustering (**ml**, **flann**)
- CUDA acceleration (**gpu**)

Các module:

- **Core functionality (core)** – module nhỏ gọn để xác định cấu trúc dữ liệu cơ bản, bao gồm mảng đa chiều dày đặc và nhiều chức năng cơ bản được sử dụng bởi tất cả các module khác.
- **Image Processing (imgproc)** – module xử lý hình ảnh gồm cả lọc hình ảnh tuyến tính và phi tuyến (linear and non-linear image filtering), phép biến đổi hình học (chỉnh size, afin và warp phối cảnh, ánh xạ lại dựa trên bảng chung), chuyển đổi không gian màu, biểu đồ,...
- **Video Analysis (video)** – module phân tích video bao gồm các tính năng ước tính chuyển động, tách nền, và các thuật toán theo dõi vật thể.
- **Camera Calibration and 3D Reconstruction (calib3d)** – thuật toán hình học đa chiều cơ bản, hiệu chuẩn máy ảnh single và stereo (single and stereo camera calibration), dự đoán kiểu dáng của đối tượng (object pose estimation), thuật toán thư tín âm thanh nổi (stereo correspondence algorithms) và các yếu tố tái tạo 3D.
- **2D Features Framework (features2d)** – phát hiện các đặc tính nổi bật của bộ nhận diện, bộ truy xuất thông số, thông số đối chiếu.
- **Object Detection (objdetect)** – phát hiện các đối tượng và mô phỏng của các hàm được định nghĩa sẵn – predefined classes (vd: khuôn mặt, mắt, cổ, con người, xe hơi,...).
- **High-level GUI (highgui)** – giao diện dễ dùng để thực hiện việc giao tiếp UI đơn giản.
- **Video I/O (videoio)** – giao diện dễ dùng để thu và mã hóa video.
- **GPU** – Các thuật toán tăng tốc GPU từ các modul OpenCV khác.
- ... và một số module hỗ trợ khác, ví dụ như FLANN và Google test wrapper, Python binding, v.v.

Giáo trình học openCV: <https://aishack.in/category/opencv/>

<ul style="list-style-type: none">• K-Nearest Neighbors in OpenCV• Image Convolutions in OpenCV• K-Means clustering in OpenCV• OpenCV's C++ interface• Integral images in OpenCV• Mathematical Morphology in OpenCV• Using OpenCV on Windows• OpenCV vs VXL vs LTI: Performance Test• Transparent image overlays in OpenCV• Drawing Histograms in OpenCV• Accessing Histogram Data• Histograms with functions of OpenCV• Implementing SIFT in OpenCV• Subpixel corners in OpenCV• Efficiently accessing matrices• Memory layout of matrices of multi-dimensional objects	<ul style="list-style-type: none">• Hough circles in OpenCV• Hough transform in OpenCV• Cropping robotics arena boundaries: Implementation• Cropping robotics arena boundaries• 2D matrices with CvMat in OpenCV• Primitive structures in OpenCV• Installing and configuring OpenCV 2.0 on Windows• Capturing images with DirectX• HighGUI: Creating Interfaces• Capturing images• Filtering images• Hello World! with Images• Installing and Getting OpenCV running• Why OpenCV?• OpenCV Memory Management• Generating uniform noise
---	--

OpenCvSharp

Cv2

- Cv2 là static class, nên không cần khởi tạo đối tượng.
- Các hàm trong Cv2 thường có tên giống với hàm gốc trong OpenCV C++ (ImRead, CvtColor, GaussianBlur, Threshold, v.v.).
- Bạn cần using OpenCvSharp; ở đầu file để sử dụng Cv2.

-

Mat

Khởi tạo

- Khởi tạo

```
using OpenCvSharp;

Mat img = new Mat(); // Khởi tạo rỗng
Mat gray = new Mat(480, 640, MatType.CV_8UC1); // Ảnh xám 640x480
```

- img.Empty(): kiểm tra có ảnh hay không,

```
using OpenCvSharp;

Mat img = Cv2.ImRead("image.jpg");

if (img.Empty())
{
    Console.WriteLine("Không đọc được ảnh hoặc ảnh rỗng.");
}
else
{
    Cv2.ImShow("Ảnh", img);
    Cv2.WaitKey(0);
}
```

- Dùng sau khi đọc ảnh từ file, lấy frame từ webcam, Trước khi xử lý ảnh, không thao tác với Mat rỗng vì sẽ lỗi Runtime.

Nhập xuất

- Đọc ảnh vào Mat:

```
Mat img = Cv2.ImRead("image.jpg", ImreadModes.Color);
```

- Hiện thị ảnh:

```
Cv2.ImShow("Ảnh", img);
Cv2.WaitKey(0);
```

- Đọc ảnh từ webcam

```
using var cap = new VideoCapture(0);
Mat frame = new Mat();

while (true)
{
    cap.Read(frame);
    if (frame.Empty()) break;

    Cv2.ImShow("Webcam", frame);
    if (Cv2.WaitKey(1) == 27) break; // ESC để thoát
}
```

- Lưu ảnh ra đĩa: output là đường dẫn lưu (đuôi jpg,png,bmp,...)

```
Cv2.ImWrite("output.jpg", mat);
```

Xử lý ảnh cơ bản

- Resize:

```
Mat resized = new Mat();
Cv2.Resize(img, resized, new Size(300, 300));
```

- Cắt ROI:

```
Rect roi = new Rect(100, 100, 200, 200);
Mat cropped = new Mat(img, roi);
```

- Xoay ảnh:

```
Point2f center = new Point2f(src.Width / 2, src.Height / 2);
Mat rotationMatrix = Cv2.GetRotationMatrix2D(center, 45, 1); // Xoay 45 độ
Mat rotated = new Mat();
Cv2.WarpAffine(src, rotated, rotationMatrix, src.Size());
```

- Chuyển màu ảnh:

```
Mat gray = new Mat();
Cv2.CvtColor(img, gray, ColorConversionCodes.BGR2GRAY);

Mat hsv = new Mat();
Cv2.CvtColor(img, hsv, ColorConversionCodes.BGR2HSV);
```

- Chuyển ảnh Bitmap:

```
using OpenCvSharp.Extensions;
Bitmap bmp = BitmapConverter.ToBitmap(img);
```

- Truy cập pixel (gray):

```
byte pixel = gray.At<byte>(y, x);
gray.Set<byte>(y, x, 255); // Gán giá trị trắng
```

- Nhận diện cạnh:

```
Mat edges = new Mat();
Cv2.Canny(gray, edges, 50, 150);
```

- Làm mờ:
- Khử nhiễu:

```
Cv2.MedianBlur(src, blurred, 5); // Khử nhiễu muối tiêu
```

openCV in C#

Hàm	Mô tả	Hàm	Mô tả
Cv2.BitwiseAnd , Or , Not , Xor	Các phép toán pixel	Cv2.Line , Cv2.Circle , Cv2.Rectangle , Cv2.Ellipse	Vẽ các hình học cơ bản
Cv2.Add , Subtract , Multiply , Divide	Toán ma trận ảnh	Cv2.PutText	Viết chữ lên ảnh
Cv2.Compare , InRange	Lọc pixel theo điều kiện		
Cv2.CopyTo với mask	Sao chép một vùng có điều kiện		

Hàm	Mô tả	Hàm	Mô tả
Cv2.Threshold	Ngưỡng hóa ảnh (binarize ảnh)	Cv2.GaussianBlur	Làm mờ Gaussian
Cv2.AdaptiveThreshold	Ngưỡng hóa cục bộ	Cv2.MedianBlur	Làm mờ trung vị
Cv2.EqualizeHist	Cân bằng histogram (tăng tương phản ảnh xám)	Cv2.BilateralFilter	Giữ cạnh khi làm mờ
Cv2.Flip	Lật ảnh (ngang/dọc)	Cv2.Sobel , Laplacian	Phát hiện biên ảnh
Cv2.Rotate	Xoay nhanh 90/180/270 độ		
Cv2.Transpose	Chuyển vị ma trận (xoay chéo)		

Hàm	Mô tả
Cv2.FindContours	Tìm đường viền (contour) trong ảnh nhị phân
Cv2.DrawContours	Vẽ các contour
Cv2.HoughLinesP	Phát hiện đường thẳng
Cv2.HoughCircles	Phát hiện hình tròn
Cv2.CascadeClassifier	Phát hiện khuôn mặt bằng Haar cascade

Webcam live

```
using OpenCvSharp;

class Program
{
    static void Main()
    {
        using var capture = new VideoCapture(0); // webcam mặc định (0)
        if (!capture.IsOpened())
        {
            Console.WriteLine("Không mở được webcam.");
            return;
        }

        using var window = new Window("Webcam");
        using var frame = new Mat();

        while (true)
        {
            capture.Read(frame); // đọc một frame
            if (frame.Empty())
                break;

            window.ShowImage(frame);

            int key = Cv2.WaitKey(1);
            if (key == 27) // Nhấn ESC để thoát
                break;
        }
    }
}
```

Nhận diện mắt

Nhận diện khuôn mặt

- File haarcascade_frontalface_default.xml
- NuGet: OpenCvSharp4, OpenCvSharp4.runtime.win

```
using OpenCvSharp;

class FaceDetection
{
    static void Main()
    {
        // Load mô hình Haar cascade
        var faceCascade = new CascadeClassifier("haarcascade_frontalface_default.xml");

        using var capture = new VideoCapture(0);
        using var window = new Window("Face Detection");
        using var frame = new Mat();
        using var gray = new Mat();

        while (true)
        {
            capture.Read(frame);
            if (frame.Empty()) break;

            // Chuyển sang ảnh xám
            Cv2.CvtColor(frame, gray, ColorConversionCodes.BGR2GRAY);

            // Phát hiện khuôn mặt
            Rect[] faces = faceCascade.DetectMultiScale(gray, 1.1, 4, HaarDetectionType.ScaleImage);

            // Vẽ khung quanh khuôn mặt
            foreach (var face in faces)
            {
                Cv2.Rectangle(frame, face, Scalar.Red, 2);
            }

            window.ShowImage(frame);
            if (Cv2.WaitKey(1) == 27) break; // ESC để thoát
        }
    }
}
```

Nhận diện nụ cười

openCV in C#

```
using OpenCvSharp;

class EyeDetection
{
    static void Main()
    {
        var faceCascade = new CascadeClassifier("haarcascade_frontalface_default.xml");
        var eyeCascade = new CascadeClassifier("haarcascade_eye.xml");

        using var capture = new VideoCapture(0);
        using var window = new Window("Nhận diện mắt");
        using var frame = new Mat();
        using var gray = new Mat();

        while (true)
        {
            capture.Read(frame);
            if (frame.Empty()) break;

            Cv2.CvtColor(frame, gray, ColorConversionCodes.BGR2GRAY);

            // Phát hiện khuôn mặt
            var faces = faceCascade.DetectMultiScale(gray, 1.1, 4);

            foreach (var face in faces)
            {
                // Vẽ khung khuôn mặt
                Cv2.Rectangle(frame, face, Scalar.Red, 2);

                // Cắt vùng mặt để tìm mắt
                var faceROI = new Mat(gray, face);
                var eyes = eyeCascade.DetectMultiScale(faceROI, 1.1, 4);

                foreach (var eye in eyes)
                {
                    var eyeRect = new Rect(
                        face.X + eye.X,
                        face.Y + eye.Y,
                        eye.Width,
                        eye.Height
                    );
                    Cv2.Rectangle(frame, eyeRect, Scalar.Blue, 2);
                }
            }

            window.ShowImage(frame);
            if (Cv2.WaitKey(1) == 27) break; // ESC để thoát
        }
    }
}
```

```
using OpenCvSharp;

class SmileDetection
{
    static void Main()
    {
        var faceCascade = new CascadeClassifier("haarcascade_frontalface_default.xml");
        var smileCascade = new CascadeClassifier("haarcascade_smile.xml");

        using var capture = new VideoCapture(0);
        using var window = new Window("Nhận diện nụ cười");
        using var frame = new Mat();
        using var gray = new Mat();

        while (true)
        {
            capture.Read(frame);
            if (frame.Empty()) break;

            Cv2.CvtColor(frame, gray, ColorConversionCodes.BGR2GRAY);

            var faces = faceCascade.DetectMultiScale(gray, 1.1, 4);

            foreach (var face in faces)
            {
                Cv2.Rectangle(frame, face, Scalar.Blue, 2);

                var faceROI = new Mat(gray, face);
                var smiles = smileCascade.DetectMultiScale(faceROI, 1.7, 20); // Tham số quan trọng!

                foreach (var smile in smiles)
                {
                    var smileRect = new Rect(
                        face.X + smile.X,
                        face.Y + smile.Y,
                        smile.Width,
                        smile.Height
                    );
                    Cv2.Rectangle(frame, smileRect, Scalar.Green, 2);
                    Cv2.PutText(frame, "Smile 😊", new Point(face.X, face.Y - 10),
                        HersheyFonts.HersheySimplex, 0.8, Scalar.Yellow, 2);
                    break; // chỉ cần phát hiện 1 nụ cười là đủ
                }
            }

            window.ShowImage(frame);
            if (Cv2.WaitKey(1) == 27) break; // ESC để thoát
        }
    }
}
```

Trình biên soạn IDE	Ngôn ngữ LT	Trình biên dịch	Trình quản lý gói	Cấu hình tạo file build
VS code	C, C++, C#	MGW	Extension	Cmake
Pycharm	Python		pip	
	Reac Native		npm	
	.NET		NuGet	