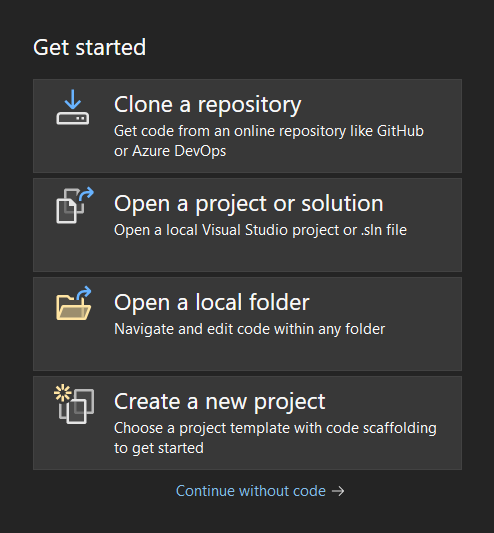
**Trích xuất ảnh từ video bằng Winform (C#)**

# 1. Tạo mới dự án Winform

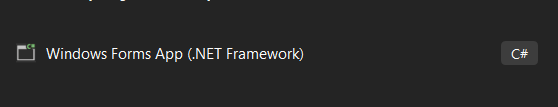
Mở lên ứng dụng Visual Studio 2022



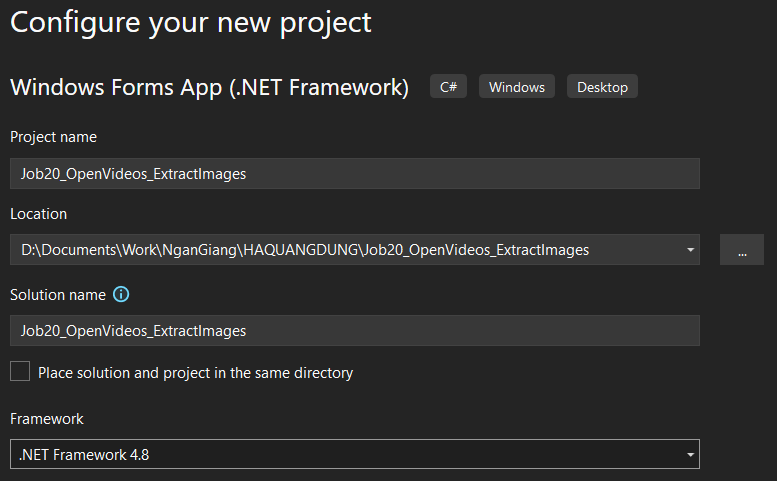
Chọn **Create a new project** để tạo mới một dự án



Chọn **Windows Forms App (.NET Framework)**



Đặt tên cho dự án



# 2. Cài đặt các thư viện cần thiết

## 2.1. Giới thiệu về thư viện EmguCV

**EmguCV** là một thư viện mã nguồn mở được phát triển dựa trên thư viện OpenCV, nhằm hỗ trợ việc tích hợp OpenCV vào các ứng dụng .NET. EmguCV cung cấp một giao diện lập trình ứng dụng (API) cho C# và các ngôn ngữ .NET khác, giúp các nhà phát triển dễ dàng sử dụng các chức năng xử lý hình ảnh và thị giác máy tính của OpenCV trong môi trường .NET.

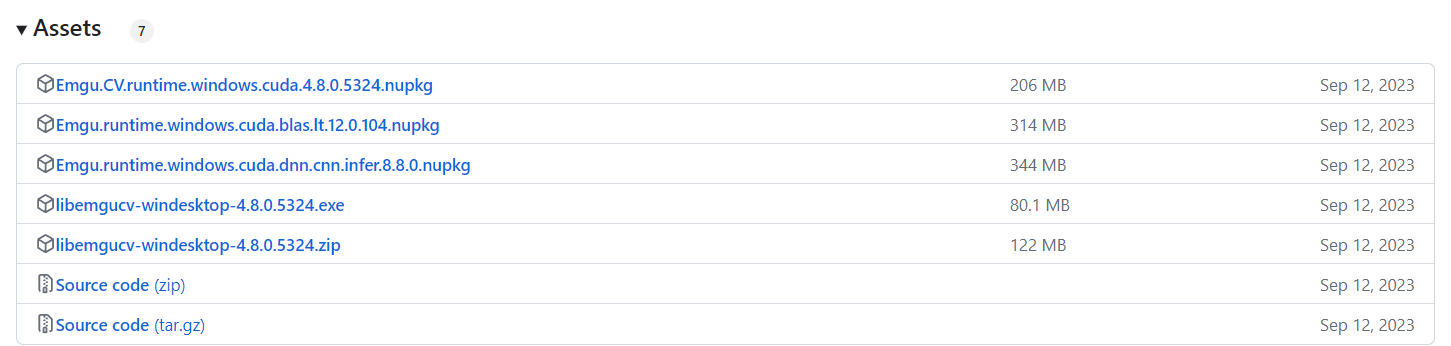
Dưới đây là một số điểm quan trọng về EmguCV:

* **Tích hợp với .NET**: EmguCV cung cấp các liên kết và bao gồm tất cả các tính năng của OpenCV trong môi trường .NET, giúp người lập trình sử dụng OpenCV mà không cần chuyển đổi giữa các ngôn ngữ lập trình.
* **Dễ sử dụng**: EmguCV được thiết kế để đơn giản hóa việc sử dụng OpenCV trong ứng dụng .NET. API của nó giữ nguyên cấu trúc và tính linh hoạt của OpenCV mà không làm cho việc tích hợp trở nên phức tạp.
* **Hỗ trợ nền tảng đa dạng**: EmguCV hỗ trợ nền tảng .NET, bao gồm WinForms, WPF, ASP.NET, Xamarin, và nhiều loại dự án .NET khác. Điều này giúp bạn tích hợp các tính năng xử lý hình ảnh vào các loại ứng dụng khác nhau.
* **Cập nhật theo phiên bản OpenCV**: EmguCV thường được cập nhật để phản ánh các cập nhật mới nhất từ OpenCV, giữ cho người dùng có thể tận dụng các tính năng mới và sửa lỗi.

## 2.2. Hướng dẫn cài đặt thư viện EmguCV

### 2.2.1. Cài đặt từ trang chủ git Emgu

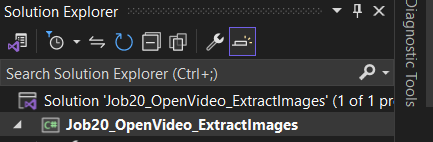
Link download: <https://github.com/emgucv/emgucv/releases/tag/4.8.0>

  
Bạn hãy chọn bản exe và cài đặt như ứng dụng bình thường

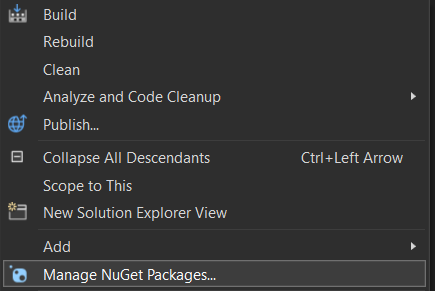
### 2.2.2. Cài trong Visual Studio

**Bước 1**: Mở dự án của bạn trong Visual Studio. (Bỏ qua nếu bạn đang trong dự án của mình)

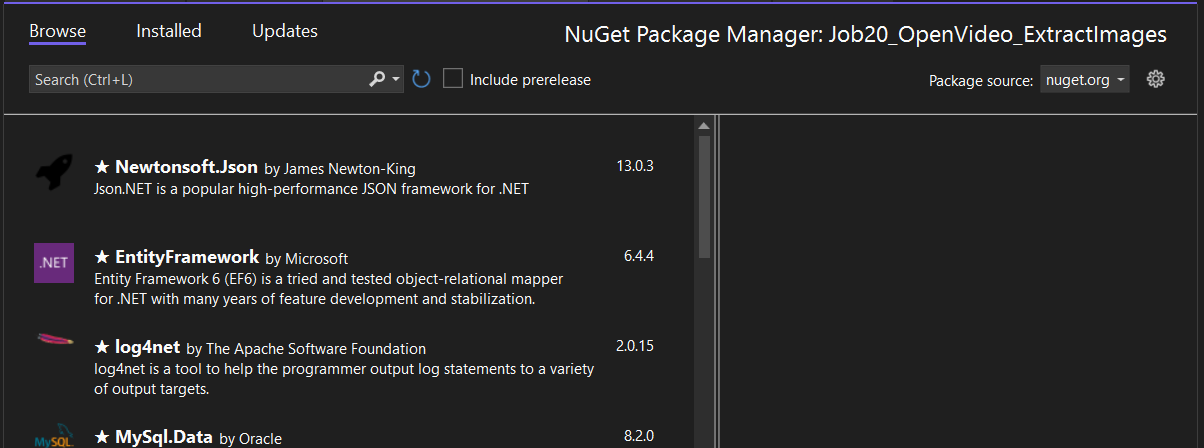
**Bước 2**: Click chuột phải vào tên dự án trong khung "Solution Explorer"



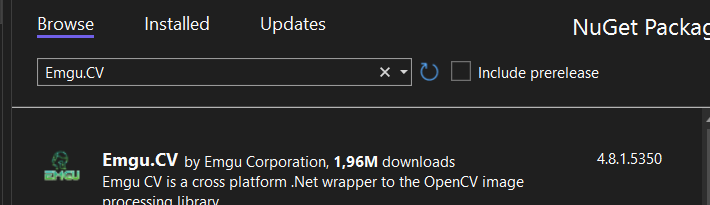
và chọn "Manage NuGet Packages" từ menu ngữ cảnh.



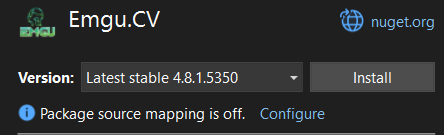
**Bước 3:** Trong cửa sổ "NuGet Package Manager", chọn tab "Browse".



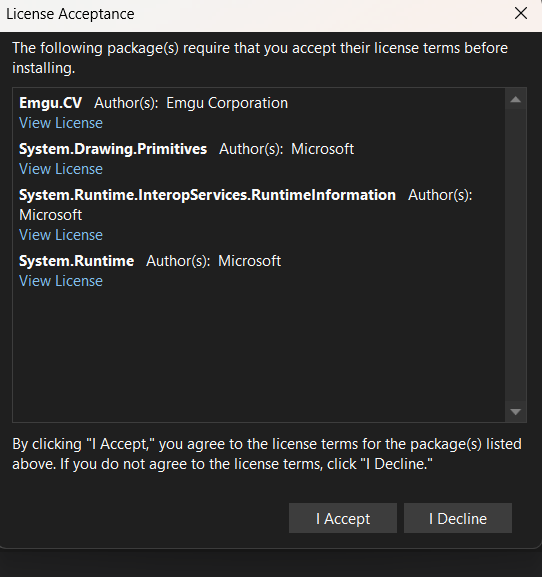
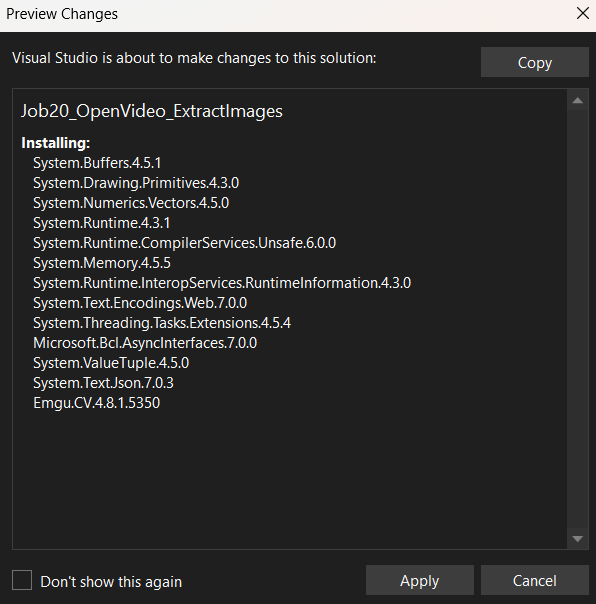
**Bước 4:** Gõ "Emgu.CV" vào ô tìm kiếm và nhấn Enter.



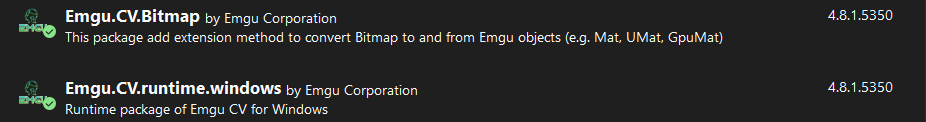
**Bước 5:** Tìm gói "Emgu.CV" trong danh sách kết quả và chọn nó.



Nhấn nút "Install" để bắt đầu quá trình cài đặt. Khi nhấn sẽ có 1 hộp thoại bật lên thì chọn **Apply** >> **I Accept**

****

Và tiếp tục với các modules tích hợp của EmguCV như dưới đây

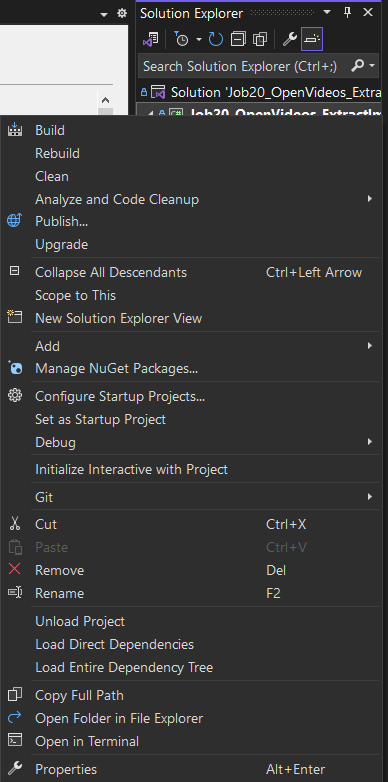
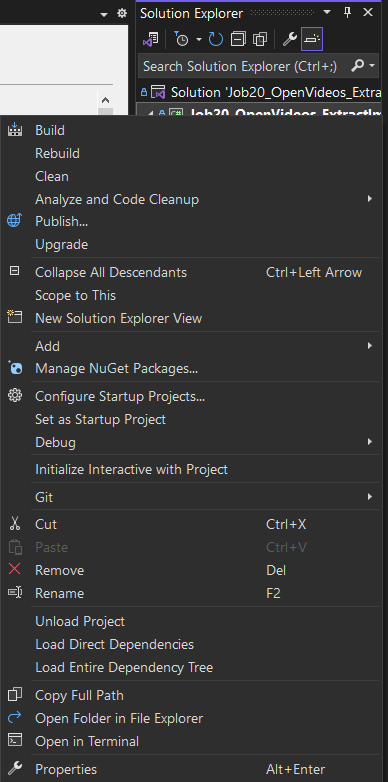


Chờ đợi quá trình cài đặt hoàn thành. Visual Studio sẽ tự động tải xuống và cài đặt gói Emgu.CV cùng với các gói phụ thuộc liên quan.

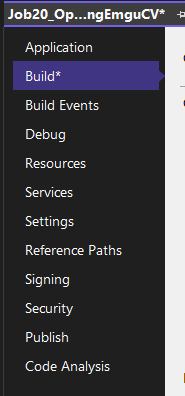
Khi quá trình cài đặt hoàn tất, EmguCV sẽ được thêm vào dự án của bạn và bạn có thể sử dụng các lớp và phương thức của EmguCV trong mã nguồn của mình.

# 3. Thiết lập một số chức năng

Nhấp chuột phải vào dự án ở **Solution explore** và chọn vào **Properties**

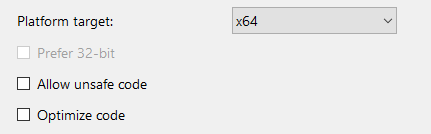


Chọn Build



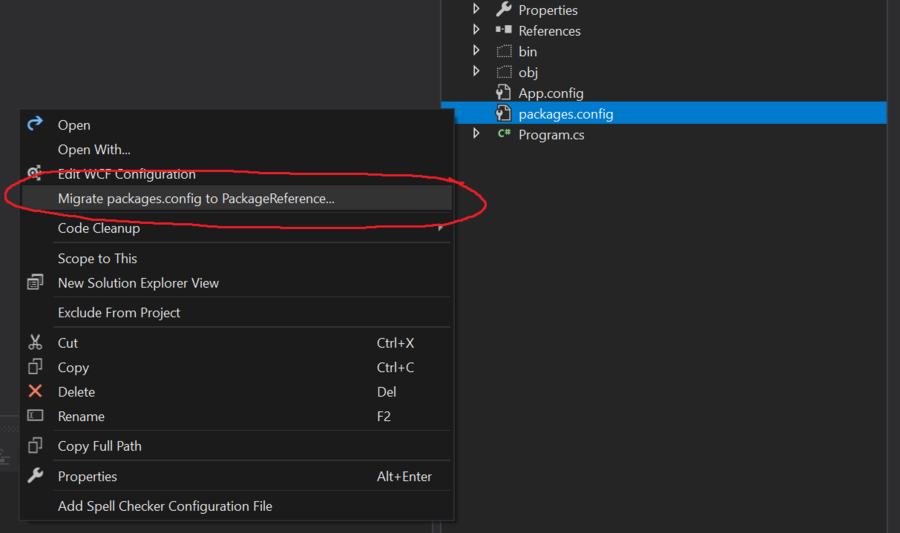
Hoặc thay vì thực hiện các bước trên, bạn cũng có thể nhấp chuột trái vào dự án và nhất tổ hợp **Alt + Enter** để bật lên.

Tiếp tục tại mục **Platform target** chọn x64 như hình dưới:



Nhấn **Ctrl + S** để lưu lại

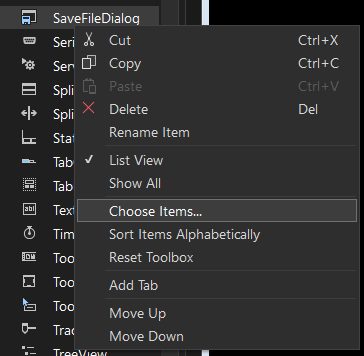
Thiết lập tham chiếu gói như sau



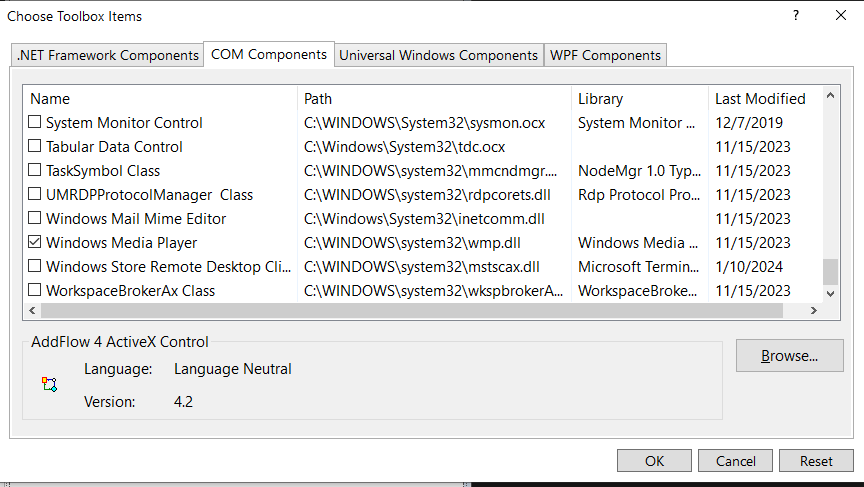
# 4. Tạo giao diện ứng dụng

Trong giao diện có sử dụng WindowsMediaPlayer để phát video nhưng do Hộp công cụ (Toolbox) chưa có nên ta cần thêm công cụ vào như các bước dưới đây:

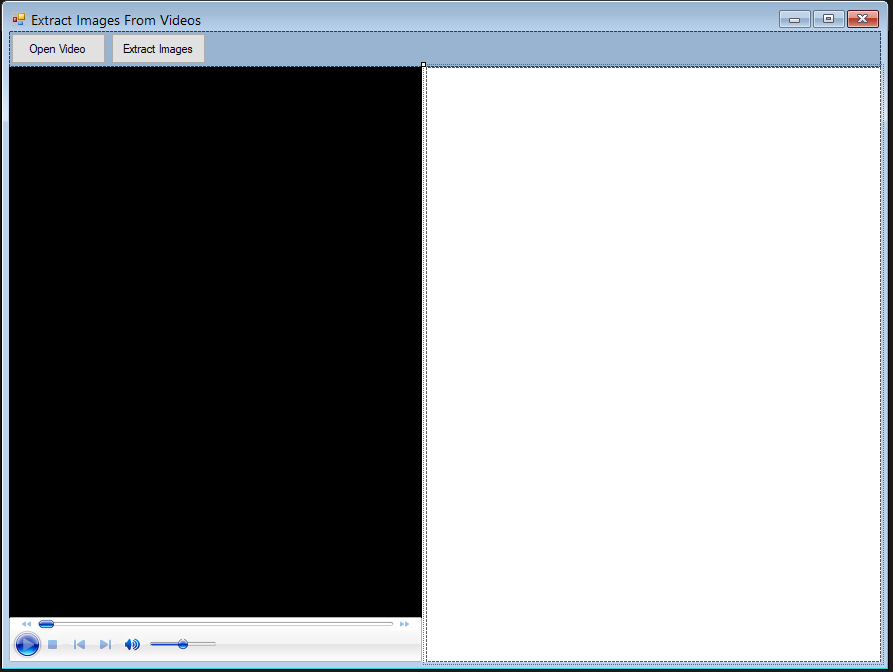
**Bước 1:** Nhấp chuột phải vào công cụ bất kỳ trong hộp công cụ và chọn **Choose Items**



Bước 2: Chọn vào tab **COM Components**, lướt xuống cuối và tích chọn vào **Windows Media Player**. Cuối cùng nhấn **OK**

****

Sau khi hoàn thành các bước kể trên, tiến hành làm giao diện theo sở thích của bạn. Ví dụ dưới đây tôi có sử dụng 2 button, 1 Windows Media Player, 1 PictureBox để hiển thị các frame được trích xuất.



# 5. Hướng dẫn chi tiết

## 5.1. Import các thư viện cần thiết

Ngoài các thư viện được import sẵn bởi hệ thống, ta cần có thêm 1 số thư viện sau:

| using AxWMPLib;  using Emgu.CV;  using Emgu.CV.CvEnum;  using Emgu.CV.Structure; |
| --- |

Trong đó:

* **AxWMPLib**: Namespace chứa các lớp liên quan đến việc tích hợp Windows Media Player.
* **Emgu.CV**: Namespace chứa thư viện EmguCV, thư viện trình điều khiển OpenCV cho C#.

## 5.2. Khai báo các thuộc tính và biến

| private VideoCapture capture;  private int frameIndex = 0;  private bool capturing = false;  private Mat frame = new Mat(); |
| --- |

Trong đó:

* **capture**: Đối tượng VideoCapture để mở video.
* **frameIndex**: Chỉ số của khung hình hiện tại.
* **capturing**: Biến kiểm tra xem quá trình trích xuất đang diễn ra hay không.
* **frame**: Đối tượng Mat để lưu trữ khung hình từ video.

## 5.3. Định nghĩa các hàm sử dụng

### 5.3.1. Hàm ExtractImages

Sử dụng một vòng lặp để đọc từng khung hình từ video và gọi hàm ProcessFrameAndSave để xử lý và lưu ảnh.

| private void ExtractImages()  {  // Lặp qua từng khung hình của video và gọi hàm xử lý và lưu ảnh  while (capture != null && capture.Ptr != IntPtr.Zero && capture.Read(frame))  {  ProcessFrameAndSave(frame);  frameIndex++;  }  // Đã đọc hết video, dừng quá trình trích xuất  capture.Stop();  capture.Dispose();  } |
| --- |

### 5.3.2. Hàm ProcessFrameAndSave

Thực hiện các bước sau đối với mỗi khung hình:

* Tính toán và thay đổi kích thước khung hình để hiển thị trên PictureBox.
* Lưu khung hình vào một thư mục được xác định.
* Hiển thị khung hình trên PictureBox.

**Note**: Để giải quyết vấn đề đường dẫn, bạn cần chắc chắn rằng thư mục "ExtractImages" đã tồn tại trong thư mục hiện tại của ứng dụng hoặc điều chỉnh đường dẫn relativePath sao cho phản ánh cấu trúc thư mục của dự án của bạn.

| private void ProcessFrameAndSave(Mat frame)  {  // Tính toán chiều rộng và chiều cao mới của frame dựa trên tỷ lệ khung hình  double aspectRatio = (double)frame.Width / frame.Height;  // Lấy kích thước hiện tại của PictureBox  int picBoxWidth = picBox.Width;  int picBoxHeight = picBox.Height;  // Tính toán chiều rộng và chiều cao mới của frame dựa trên tỷ lệ khung hình và kích thước của PictureBox  int newWidth = picBoxWidth;  int newHeight = (int)(newWidth / aspectRatio);  // Nếu chiều cao mới vượt quá chiều cao của PictureBox, thì tính lại kích thước dựa trên chiều cao  if (newHeight > picBoxHeight)  {  newHeight = picBoxHeight;  newWidth = (int)(newHeight \* aspectRatio);  }  // Resize frame theo chiều rộng và chiều cao mới  CvInvoke.Resize(frame, frame, new Size(newWidth, newHeight));  // Chuyển đổi frame thành đối tượng Image  Image<Bgr, byte> img = frame.ToImage<Bgr, byte>();  Bitmap bitmap = img.ToBitmap();  // Hiển thị frame trên PictureBox (cần sử dụng Invoke vì đang chạy trong một thread khác)  picBox.Invoke((MethodInvoker)delegate  {  picBox.Image = bitmap;  picBox.Invalidate();  });  // Lấy đường dẫn thư mục hiện tại của ứng dụng  string currentDirectory = AppDomain.CurrentDomain.BaseDirectory;  // Kết hợp đường dẫn tương đối với thư mục hiện tại để có đường dẫn đầy đủ  string relativePath = @"..\..\..\ExtractImages"; // Đường dẫn tương đối từ thư mục hiện tại  string extractPath = Path.Combine(currentDirectory, relativePath);  // Đảm bảo rằng thư mục đích tồn tại và tạo nó nếu chưa có  if (!Directory.Exists(extractPath))  {  Directory.CreateDirectory(extractPath);  }  // Lưu frame thành ảnh  string videoFilePath = WinMediaPlayer.URL;  string videoFileName = Path.GetFileNameWithoutExtension(videoFilePath);  string fileName = $"{videoFileName}\_frame\_{frameIndex + 1}.png";  string filePath = Path.Combine(extractPath, fileName);  img.Save(filePath);  } |
| --- |

### 5.3.3. Hàm btnOpenVideo\_Click

Hàm bắt sự kiện nút Mở Video

Mở một hộp thoại để chọn video và khởi tạo VideoCapture từ đường dẫn được chọn.

| private void btnOpenVideo\_Click(object sender, EventArgs e)  {  // Tạo một hộp thoại mở tệp để chọn video  OpenFileDialog ofd = new OpenFileDialog();  ofd.Title = "Chọn video";  ofd.Filter = "Tất cả các file|\*.\*|Video Files|\*.avi;\*.mp4;\*.wmv|Tệp AVI|\*.avi|Tệp MP4|\*.mp4|Tệp WMV|\*.wmv";  // Nếu người dùng chọn một tệp, thì tiếp tục xử lý  if (ofd.ShowDialog() == DialogResult.OK)  {  // Lấy đường dẫn tệp được chọn  string selectedFilePath = ofd.FileName;  // Sử dụng Windows Media Player để phát video  WinMediaPlayer.URL = selectedFilePath;  WinMediaPlayer.Ctlcontrols.play();  // Khởi tạo VideoCapture từ đường dẫn video  capture = new VideoCapture(selectedFilePath);  }  } |
| --- |

### 5.3.4. btnExtractImage\_Click

Hàm bắt sự kiện Trích xuất ảnh

Khi nút "Trích xuất ảnh" được nhấn, hàm này sẽ gọi hàm ExtractImages thông qua một Task để chạy quá trình trích xuất ảnh từ video.

| private async void btnExtractImage\_Click(object sender, EventArgs e)  {  // Kiểm tra xem VideoCapture đã được khởi tạo và không trong quá trình trích xuất  if (capture != null && !capturing)  {  capturing = true; // Đánh dấu bắt đầu quá trình trích xuất  // Tạo một Task chạy quá trình trích xuất ảnh từ video  await Task.Run(() => ExtractImages());  // Hiển thị thông báo khi quá trình trích xuất hoàn tất  MessageBox.Show("Hình ảnh đã được trích xuất và lưu lại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  capturing = false; // Đánh dấu kết thúc quá trình trích xuất  }  else  {  // Thông báo nếu chưa chọn video  MessageBox.Show("Vui lòng chọn một tệp video.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  } |
| --- |

# 6. Chi tiết mã nguồn

| using System;  using System.Drawing;  using System.IO;  using System.Threading.Tasks;  using System.Windows.Forms;  using AxWMPLib;  using Emgu.CV;  using Emgu.CV.CvEnum;  using Emgu.CV.Structure;  namespace Job20\_OpenVideos\_ExtractImages\_UsingEmguCV  {  public partial class FormMain : Form  {  private VideoCapture capture;  private int frameIndex = 0;  private bool capturing = false;  private Mat frame = new Mat();  public FormMain()  {  InitializeComponent();  }  private void btnOpenVideo\_Click(object sender, EventArgs e)  {  OpenFileDialog ofd = new OpenFileDialog();  ofd.Title = "Chọn video";  ofd.Filter = "Tất cả các file|\*.\*|Video Files|\*.avi;\*.mp4;\*.wmv|Tệp AVI|\*.avi|Tệp MP4|\*.mp4|Tệp WMV|\*.wmv";  if (ofd.ShowDialog() == DialogResult.OK)  {  string selectedFilePath = ofd.FileName;  WinMediaPlayer.URL = selectedFilePath;  WinMediaPlayer.Ctlcontrols.play();  // Khởi tạo VideoCapture từ đường dẫn video  capture = new VideoCapture(selectedFilePath);  }  }  private async void btnExtractImage\_Click(object sender, EventArgs e)  {  if (capture != null && !capturing)  {  capturing = true;  // Tạo một Task chạy quá trình trích xuất ảnh từ video  await Task.Run(() => ExtractImages());  MessageBox.Show("Hình ảnh đã được trích xuất và lưu lại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  // Đặt capturing về false sau khi quá trình trích xuất hoàn tất  capturing = false;  }  else  {  MessageBox.Show("Vui lòng chọn một tệp video.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  private void ExtractImages()  {  while (capture != null && capture.Ptr != IntPtr.Zero && capture.Read(frame))  {  ProcessFrameAndSave(frame);  frameIndex++;  }  // Đã đọc hết video, dừng quá trình trích xuất  capture.Stop();  capture.Dispose();  }  private void ProcessFrameAndSave(Mat frame)  {  // Tính toán chiều rộng và chiều cao mới của frame dựa trên tỷ lệ khung hình  double aspectRatio = (double)frame.Width / frame.Height;  // Lấy kích thước hiện tại của PictureBox  int picBoxWidth = picBox.Width;  int picBoxHeight = picBox.Height;  // Tính toán chiều rộng và chiều cao mới của frame dựa trên tỷ lệ khung hình và kích thước của PictureBox  int newWidth = picBoxWidth;  int newHeight = (int)(newWidth / aspectRatio);  // Nếu chiều cao mới vượt quá chiều cao của PictureBox, thì tính lại kích thước dựa trên chiều cao  if (newHeight > picBoxHeight)  {  newHeight = picBoxHeight;  newWidth = (int)(newHeight \* aspectRatio);  }  // Resize frame theo chiều rộng và chiều cao mới  CvInvoke.Resize(frame, frame, new Size(newWidth, newHeight));  Image<Bgr, byte> img = frame.ToImage<Bgr, byte>();  Bitmap bitmap = img.ToBitmap();  // Hiển thị frame trên PictureBox  picBox.Invoke((MethodInvoker)delegate  {  picBox.Image = bitmap;  picBox.Invalidate();  });  // Lấy đường dẫn thư mục hiện tại của ứng dụng  string currentDirectory = AppDomain.CurrentDomain.BaseDirectory;  // Kết hợp đường dẫn tương đối với thư mục hiện tại để có đường dẫn đầy đủ  string relativePath = @"..\..\..\ExtractImages"; // Đường dẫn tương đối từ thư mục hiện tại  string extractPath = Path.Combine(currentDirectory, relativePath);  // Đảm bảo rằng thư mục đích tồn tại và tạo nó nếu chưa có  if (!Directory.Exists(extractPath))  {  Directory.CreateDirectory(extractPath);  }  // Lưu frame thành ảnh  string videoFilePath = WinMediaPlayer.URL;  string videoFileName = Path.GetFileNameWithoutExtension(videoFilePath);  string fileName = $"{videoFileName}\_frame\_{frameIndex + 1}.png";  string filePath = Path.Combine(extractPath, fileName);  img.Save(filePath);  }  }  } |
| --- |