* + Image rotation (fixed and auto-alignment).
  + Brightness and contrast adjustment.
  + Cropping or perspective correction.

**Preprocessing Edge detection**

OpenCVSharp path: /Users/khushalsingh/Downloads/ocr-techtitans/src/OCR/OCRPreprocessor/Tesseract/ConsoleApp1/ConsoleApp1/bin/Debug/net9.0/osx-arm64/OpenCvSharp.dll

Tesseract Path:

/Users/khushalsingh/Downloads/ocr-techtitans/src/OCR/OCRPreprocessor/Tesseract/ConsoleApp1/ConsoleApp1/bin/Debug/net9.0/Tesseract.dll

CLI Commands:

cd /path/to/your/project

/reference:/Users/khushalsingh/Downloads/ocr-techtitans/src/OCR/OCRPreprocessor/Tesseract/ConsoleApp1/ConsoleApp1/bin/Debug/net9.0/Tesseract.dll /Users/khushalsingh/Downloads/ocr-techtitans/src/OCR/OCRPreprocessor/Tesseract/ConsoleApp1/ConsoleApp1/bin/Debug/net9.0/osx-arm64/OpenCvSharp.dll

**Preprocessign FixedRotation**

**using System;**

**using System.Drawing; *// For Bitmap***

**using AForge.Imaging.Filters; *// For RotateBilinear filter***

**namespace FixedRotationTest**

**{**

**class Program**

**{**

**static void Main()**

**{**

**try**

**{**

**Console.WriteLine("Loading image...");**

**string imagePath = @"/Users/khushalsingh/Downloads/ocr-techtitans/Input/Test\_2.jpg"; *// Update path to your image***

**string outputPath = @"/Users/khushalsingh/Downloads/ocr-techtitans/Output/rotated\_image.jpg"; *// Output path for the rotated image***

***// Ensure the file exists***

**if (!System.IO.File.Exists(imagePath))**

**{**

**Console.WriteLine("Image file not found!");**

**return;**

**}**

***// Load the image***

**Bitmap originalImage = new Bitmap(imagePath);**

***// Rotate the image by 90 degrees clockwise***

**Console.WriteLine("Rotating image by 90 degrees...");**

**RotateBilinear rotationFilter = new RotateBilinear(90);**

**Bitmap rotatedImage = rotationFilter.Apply(originalImage);**

***// Save the rotated image***

**rotatedImage.Save(outputPath);**

**Console.WriteLine($"Rotated image saved at: {outputPath}");**

**}**

**catch (Exception ex)**

**{**

**Console.WriteLine("An error occurred:");**

**Console.WriteLine(ex.ToString());**

**}**

**}**

**}**

**}**

To fix the warning below   
  
khushalsingh@MacBookAir FixedRotation1 % dotnet build

Restore complete (0.2s)

FixedRotation1 **succeeded with 2 warning(s)** (0.3s) → bin/Debug/net9.0/FixedRotation1.dll

/Users/khushalsingh/Downloads/ocr-techtitans/src/OCR/Preprocessing/FixedRotation1/FixedRotation1/**Program.cs**(33,17): **warning** **CA1416**: This call site is reachable on all platforms. 'Image.Save(string)' is only supported on: 'windows'. (https://learn.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1416)

/Users/khushalsingh/Downloads/ocr-techtitans/src/OCR/Preprocessing/FixedRotation1/FixedRotation1/**Program.cs**(25,40): **warning** **CA1416**: This call site is reachable on all platforms. 'Bitmap' is only supported on: 'windows'. (https://learn.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1416)

Build **succeeded with 2 warning(s)** in 0.8s

khushalsingh@MacBookAir FixedRotation1 % dotnet build

Restore complete (0.3s)

FixedRotation1 **succeeded** (0.3s) → bin/Debug/net9.0/FixedRotation1.dll

Build **succeeded** in 0.9s

khushalsingh@MacBookAir FixedRotation1 %

Added

<NoWarn>$(NoWarn);CA1416</NoWarn>

To suppress the warnings, in .csproj file inside the <PropertyGroup> section

**SixLabors.ImageSharp** nugget added

dotnet build

dotnet run

Working code below  
using System;

using SixLabors.ImageSharp; *// For ImageSharp*

using SixLabors.ImageSharp.Processing; *// For image processing (rotate)*

using SixLabors.ImageSharp.Formats.Jpeg; *// For saving as JPEG*

namespace FixedRotationTest

{

class Program

{

static void Main()

{

try

{

Console.WriteLine("Loading image...");

string imagePath = @"/Users/khushalsingh/Downloads/ocr-techtitans/Input/Test\_2.jpg"; *// Update path to your image*

string outputPath = @"/Users/khushalsingh/Downloads/ocr-techtitans/Output/rotated\_image.jpg"; *// Output path for the rotated image*

*// Ensure the file exists*

if (!System.IO.File.Exists(imagePath))

{

Console.WriteLine("Image file not found!");

return;

}

*// Load the image using ImageSharp*

using (Image image = Image.Load(imagePath))

{

*// Rotate the image by 90 degrees clockwise*

Console.WriteLine("Rotating image by 90 degrees...");

image.Mutate(x => x.Rotate(90));

*// Save the rotated image*

image.Save(outputPath, new JpegEncoder());

Console.WriteLine($"Rotated image saved at: {outputPath}");

}

}

catch (Exception ex)

{

Console.WriteLine("An error occurred:");

Console.WriteLine(ex.ToString());

}

}

}

}