詢問Opera瀏覽器內建的Aria(AI)：請用WindowsFormsApp的C#程式，提供一個可辨識圖檔上的英文、數字功能範例

Answer：好的，我將為您提供一個使用 C# Windows Forms 應用程式，辨識圖檔中英文和數字的功能範例。這個範例將使用 **Tesseract OCR 引擎** 進行文字辨識。

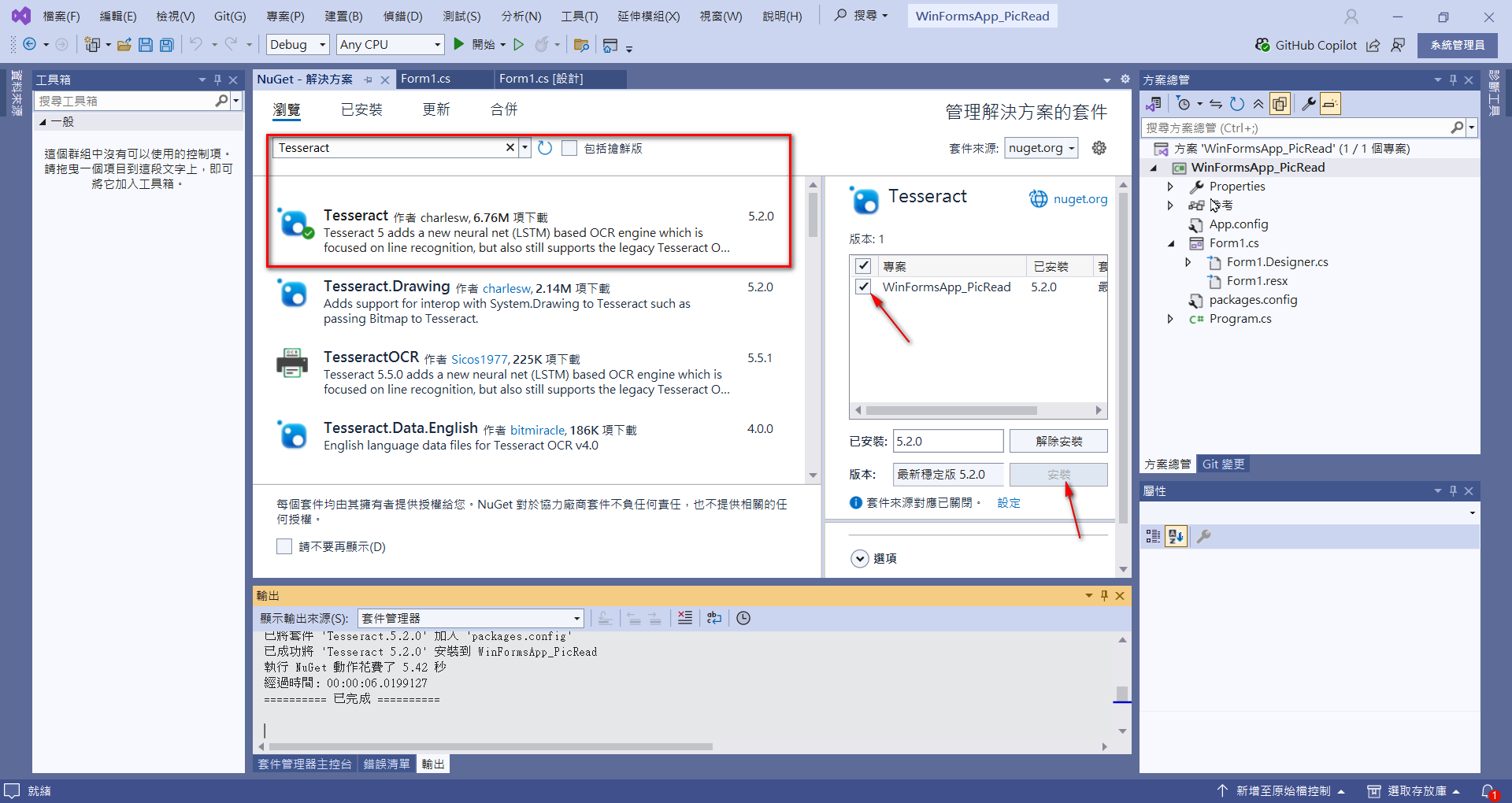
以下是程式碼範例，以及逐步說明：

1. 建立 Windows Forms 應用程式專案

* 在 Visual Studio 中，建立一個新的專案。
* 選擇 "Windows Forms 應用程式 (.NET Framework)" 或 ".NET"。
* 設定專案名稱，例如 "ImageTextRecognizer"。

1. 安裝必要的 NuGet 套件

* 在專案中，透過 NuGet 套件管理員安裝以下套件：
  + Tesseract (用於 OCR 文字辨識)
  + System.Drawing.Common (用於處理圖片)



1. 設計 UI 介面

* 在 Form1.cs 的設計檢視中，添加以下控件：
  + 一個 PictureBox：用於顯示要辨識的圖片。
  + 一個 Button：用於觸發圖片載入。
  + 一個 Button：用於觸發文字辨識。
  + 一個 TextBox：用於顯示辨識出的文字。

1. 程式碼 (Form1.cs)

csharp

using System;

using System.Drawing;

using System.Windows.Forms;

using Tesseract;

namespace ImageTextRecognizer

{

public partial class Form1 : Form

{

private Bitmap \_selectedImage;

public Form1()

{

InitializeComponent();

}

private void buttonLoadImage\_Click(object sender, EventArgs e)

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "Image Files (\*.jpg; \*.jpeg; \*.png; \*.bmp)|\*.jpg; \*.jpeg; \*.png; \*.bmp";

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

try

{

\_selectedImage = new Bitmap(openFileDialog.FileName);

pictureBoxImage.Image = \_selectedImage;

}

catch (Exception ex)

{

MessageBox.Show("Error loading image: " + ex.Message);

}

}

}

private void buttonRecognizeText\_Click(object sender, EventArgs e)

{

if (\_selectedImage == null)

{

MessageBox.Show("Please load an image first.");

return;

}

try

{

using (var engine = new TesseractEngine(@"./tessdata", "eng", EngineMode.Default)) // 確保 tessdata 資料夾存在，並包含 "eng.traineddata"

{

// 建立一個 TesseractEngine 實例，指定 tessdata 資料夾的路徑和語言

// "eng" 代表英文。 你可以根據需要更改語言 (例如 "chi\_tra" 代表繁體中文)

// 請確保你的 tessdata 資料夾中包含對應的語言檔 (.traineddata)

var page = engine.Process(\_selectedImage);

string text = page.GetText();

textBoxResult.Text = text;

}

}

catch (Exception ex)

{

MessageBox.Show("Error recognizing text: " + ex.Message);

}

}

}

}

1. 程式碼說明

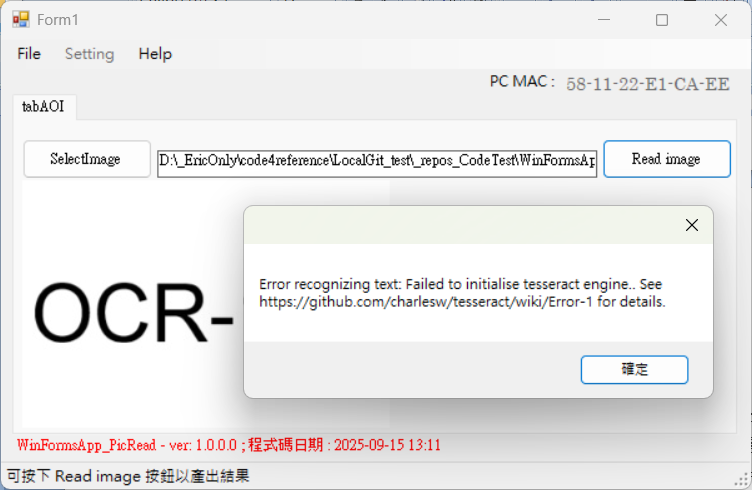
* **引用命名空間:** 程式碼中引用了必要的命名空間，包括 System.Drawing、System.Windows.Forms 和 Tesseract。
* **UI 元素:** 程式碼處理了兩個按鈕的點擊事件：
  + buttonLoadImage\_Click: 打開一個檔案選擇對話框，讓使用者選擇圖片，並將圖片顯示在 pictureBoxImage 中。
  + buttonRecognizeText\_Click: 使用 Tesseract OCR 引擎辨識圖片中的文字，並將結果顯示在 textBoxResult 中。
* Tesseract 設定:
  + TesseractEngine(@"./tessdata", "eng", EngineMode.Default): 建立 TesseractEngine 實例。
* @"./tessdata": 指定 tessdata 資料夾的路徑。 **重要:** 您需要在專案的執行目錄下 (通常是 bin\Debug 或 bin\Release) 建立一個名為 "tessdata" 的資料夾，並將您需要的語言檔 (例如 "eng.traineddata" - 英文) 放入其中。 您可以從 Tesseract 的官方網站或其他來源下載這些語言檔。
* "eng": 指定要使用的語言。 在這個例子中，我們使用英文。
* EngineMode.Default: 設定引擎模式。
  + page = engine.Process(\_selectedImage): 使用引擎處理圖片，進行 OCR。
  + string text = page.GetText(): 從 OCR 結果中獲取文字。
* **錯誤處理:** 程式碼包含基本的錯誤處理，以處理圖片載入和文字辨識過程中可能發生的異常。

1. 步驟
   1. 建立專案並安裝 NuGet 套件。
   2. 設計 UI 介面。
   3. 複製程式碼到 Form1.cs 中。
   4. 建立 "tessdata" 資料夾，並放入語言檔。 確保您的專案的執行目錄下有 "tessdata" 資料夾，並且資料夾中包含您要辨識的語言的 .traineddata 檔案 (例如 "eng.traineddata" - 英文)。
   5. 執行程式。 點擊 "載入圖片" 按鈕，選擇一個包含英文和數字的圖片。 然後點擊 "辨識文字" 按鈕，結果將顯示在文字框中。
2. 額外提示

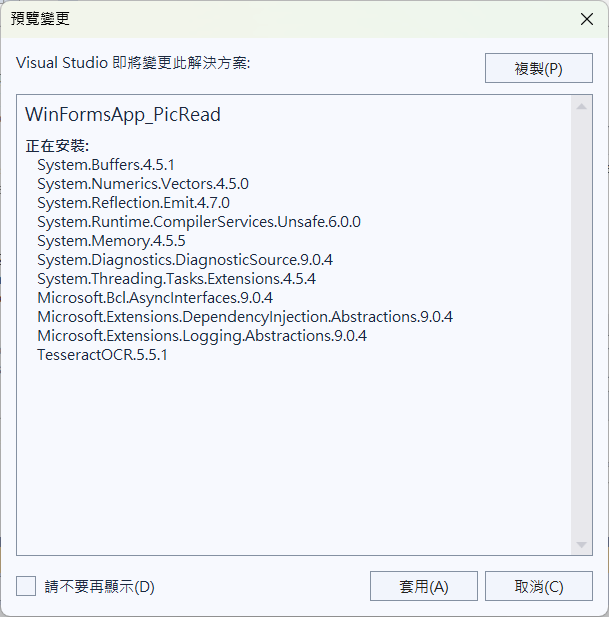
* **語言支援:** Tesseract 支援多種語言。 您可以在 Tesseract 引擎初始化時，更改語言參數 (例如，使用 "chi\_tra" 進行繁體中文辨識)。 請確保您已將相應的語言檔 (.traineddata) 放入 "tessdata" 資料夾中。
* **圖片預處理:** 在某些情況下，您可能需要對圖片進行預處理，以提高辨識準確性。 例如，您可以調整圖片的對比度、亮度，或進行去噪處理。 您可以使用 System.Drawing 命名空間中的類別來實現這些操作。
* **效能:** 對於大型圖片，OCR 過程可能需要一些時間。 您可以在後台執行 OCR 操作，以避免阻塞 UI 執行緒。
* **Tesseract 版本:** 不同的 Tesseract 版本可能會有細微的差異。 請參閱 Tesseract 的官方文檔，以獲取更多資訊。

希望這個範例對您有所幫助！ 如果您有任何問題，請隨時提出。

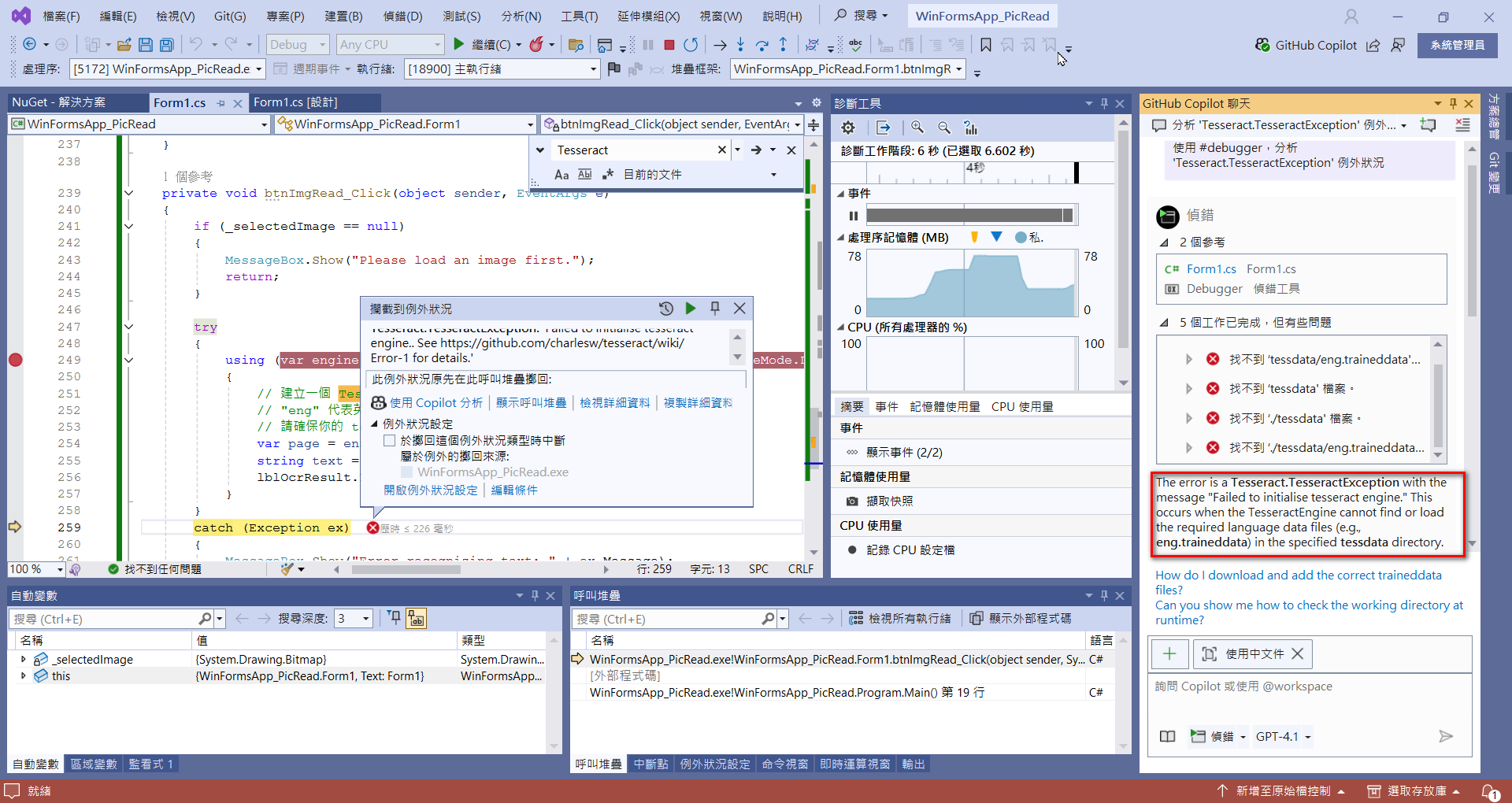
實作應用結果：

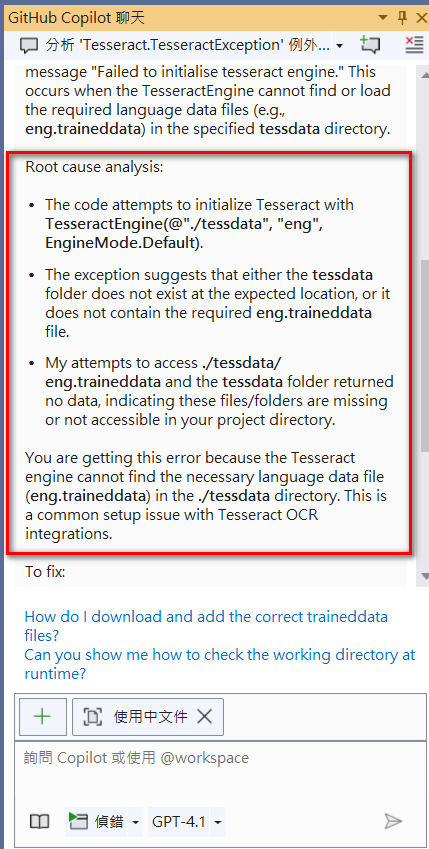
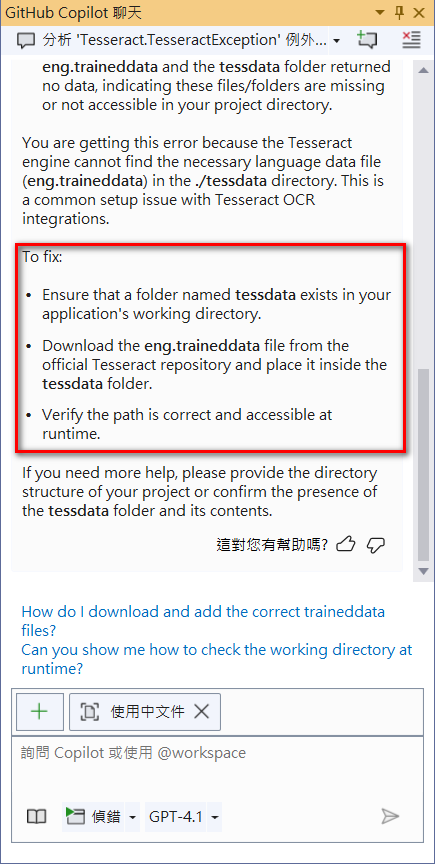


Error recognizing text: Failed to initialise tesseract engine .. See https://github.com/charlesw/tesseract/wiki/Error-1 for details



Add NuGet TesseractOCR5.5.1🡺仍然執行失敗



How do I download and add the correct traineddata files?

您可以依照以下步驟下載並加入正確的 **traineddata** 檔案：

1. 下載語言檔案

* 前往 Tesseract [官方語言檔案下載頁](https://github.com/tesseract-ocr/tessdata)。
* 找到您需要的語言（例如英文：**eng.traineddata**，繁體中文：**chi\_tra.traineddata**）。
* 點擊檔案名稱，然後按下「Download」或「Raw」並另存新檔。

2. 建立 **tessdata** 資料夾

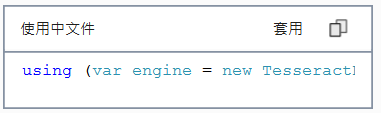
* 在您的專案根目錄（與 **.exe** 執行檔同層）建立一個名為 **tessdata** 的資料夾。

3. 複製檔案

* 將剛剛下載的 **eng.traineddata**（或其他語言檔案）複製到 **tessdata** 資料夾內。

4. 確認路徑

確認程式碼中的路徑與資料夾名稱一致：



🡪using (var engine = new TesseractEngine(@"./tessdata", "eng", EngineMode.Default))

* ./tessdata 必須存在，且裡面有 eng.traineddata。

5. 重新執行程式

* 執行您的程式，Tesseract 應該可以正確載入語言檔案。

---

補充：

* 若要辨識其他語言，請下載對應的 **.traineddata** 檔案並放入同一資料夾，然後在程式中指定語言代碼（如 **"chi\_tra"**）。
* 若專案部署到其他機器，請確保 **tessdata** 資料夾與語言檔案一併複製。

以上步驟完成後，即可正常執行，但OCR有的字不太正確。

