**Explicación Completa del Proyecto de Detección de Fuego usando Roboflow (con integración en C#)**

**Resumen del Proyecto**

Este proyecto tiene como objetivo la detección automática de fuego en imágenes usando un modelo de detección de objetos entrenado en Roboflow. A través de un conjunto de datos etiquetado, se entrena un modelo que luego puede usarse para predecir la presencia de fuego en nuevas imágenes desde una aplicación en C#.

**1. Subida y Anotación de Datos**

En Roboflow:

* Se subieron 955 imágenes.
* Se dividieron en 3 subconjuntos:
  + 75% para entrenamiento (712 imágenes).
  + 15% para validación (142 imágenes).
  + 10% para prueba (101 imágenes).
* Las imágenes fueron anotadas manualmente marcando las zonas donde aparece fuego.

**¿Por qué es importante?**  
La calidad de las anotaciones y la variedad en los datos son claves para entrenar un modelo preciso.

**2. Entrenamiento del Modelo**

Una vez cargadas y anotadas las imágenes:

* Se entrenó el modelo directamente en la plataforma Roboflow usando la versión 1 del conjunto de datos.
* No se aplicaron pasos de preprocesamiento ni aumentos de datos.

**¿Qué hace Roboflow?**  
Entrena un modelo de detección de objetos (por ejemplo, basado en YOLO) que aprende a identificar las áreas donde hay fuego.

**3. Despliegue del Modelo**

Tras el entrenamiento:

* Se puede desplegar el modelo para hacer inferencias.
* Si no se entrena un modelo en Roboflow, también se puede cargar uno personalizado mediante archivos de pesos.

**4. Uso del Modelo desde Código (C#)**

En una aplicación en C# puedes usar HttpClient para enviar imágenes al endpoint del modelo de Roboflow y obtener las predicciones.

**Ejemplo en C# para enviar una imagen al modelo**

using System.Net.Http;

using System.Net.Http.Headers;

using System.Threading.Tasks;

using System.IO;

var filePath = "ruta/a/tu/imagen.jpg";

var apiKey = "TU\_API\_KEY";

var project = "nombre-del-proyecto";

var version = "1";

using var client = new HttpClient();

var request = new HttpRequestMessage(HttpMethod.Post, $"https://detect.roboflow.com/{project}/{version}?api\_key={apiKey}");

using var stream = File.OpenRead(filePath);

var content = new MultipartFormDataContent

{

{ new StreamContent(stream), "file", Path.GetFileName(filePath) },

{ new StringContent("test-image"), "name" }

};

request.Content = content;

var response = await client.SendAsync(request);

var jsonResponse = await response.Content.ReadAsStringAsync();

Console.WriteLine(jsonResponse);

Este código carga una imagen, la envía al modelo hospedado en Roboflow y muestra las predicciones en consola en formato JSON.

**5. Visualización de Resultados (opcional)**

Puedes usar librerías como EmguCV o System.Drawing en C# para mostrar las detecciones dibujando rectángulos sobre la imagen con los datos de predicción que devuelve Roboflow (coordenadas y tamaño de las cajas).

**Conclusión**

Este proyecto demuestra cómo se puede utilizar Roboflow para entrenar, probar y desplegar un modelo de detección de objetos, en este caso, para detectar fuego, y cómo se puede integrar fácilmente con una aplicación en C#.