Red Neuronal Convolucional – Detección de Frutas

Juan Carlos Faz Leal

22 mayo 2023

---

1. Introducción.

Actualmente, el desarrollo de la tecnología de inteligencia artificial ha traído cambios revolucionarios en varios campos, y la visión artificial no es una excepción. La detección de objetos en imágenes es un gran desafío en este campo, especialmente la detección de frutas tiene un gran potencial en agricultura, alimentación y salud. Este informe describe el desarrollo de una red neuronal convolucional (CNN) para la detección de frutas utilizando esta arquitectura para el análisis y la clasificación de imágenes.

2. Objetivo.

El objetivo principal de este trabajo es desarrollar una red neuronal convolucional precisa y eficiente para la detección automática de frutas en imágenes. Se busca entrenar un modelo capaz de reconocer y clasificar diferentes tipos de frutas con alto grado de precisión.

3. Conjunto de datos.

El conjunto de datos utilizado en este trabajo proviene de Kaggle y se denomina "Fruits 360". Este conjunto de datos consiste en una amplia variedad de imágenes de frutas pertenecientes a 120 categorías diferentes. Cada imagen está etiquetada con la clase correspondiente, lo que permite entrenar y evaluar el modelo de detección de frutas.

Las imágenes del conjunto de datos "Fruits 360" presentan variaciones en cuanto a la calidad, el tamaño, la iluminación, el fondo y la orientación de las frutas. Esto proporciona un desafío realista para el modelo de detección de frutas, ya que se asemeja a las condiciones reales en las que se aplicaría la detección automática de frutas.

4. Preprocesamiento de los datos.

Antes de utilizar el conjunto de datos, se realizó un preprocesamiento para garantizar la calidad y la consistencia de las imágenes. Esto incluyó la normalización y el reescalamiento de las imágenes.

5. Arquitectura de la red neuronal.

La arquitectura de la red neuronal convolucional utilizada en este trabajo consta de varias capas convolucionales y de pooling, seguidas de capas completamente conectadas.

Para este trabajo, se optó por una arquitectura sencilla pero efectiva. La arquitectura consta de capas convolucionales seguidas de capas de pooling, lo que permite extraer características relevantes de las imágenes de frutas.

6. Hiperparámetros.

En cuanto a los hiperparámetros, se utilizó una arquitectura que consta de tres capas convolucionales, cada una con 16, 32 y 16 filtros respectivamente. Se aplicó una función de activación ReLU después de cada capa convolucional. Además, se utilizaron capas de pooling MaxPooling2D después de cada capa convolucional para reducir la dimensionalidad de las características extraídas.

A continuación, se agregó una capa Flatten para aplanar las características y se conectó a una capa Dense con 128 neuronas y una función de activación ReLU. Finalmente, se agregó una capa de salida Dense con 6 neuronas y una función de activación softmax para realizar la clasificación de las frutas en las 6 categorías objetivo.

7. Entrenamiento de la red.

Para el entrenamiento de la red neuronal, se utilizó el conjunto de datos descargado de Kaggle: enlace del dataset. En este caso, se tomó la carpeta "Train" y se renombró como "data". Dentro de esta carpeta, se encuentran las imágenes de las diferentes frutas divididas en carpetas, como "Manzana", "Banana", "Kiwi", "Mango", "Naranja", "Papaya", "Pepino", "Fresa" y "Sandia".

Para el entrenamiento de la red neuronal convolucional, se utilizó la función de pérdida de entropía cruzada escasa (SparseCategoricalCrossentropy) como criterio de optimización, junto con el optimizador Adam. Esto permitió que el modelo aprendiera a clasificar las imágenes de frutas en las categorías correspondientes.
Además, se configuró un registro de TensorBoard utilizando la biblioteca TensorBoard de TensorFlow. Esto permitió visualizar y analizar métricas y gráficos relacionados con el rendimiento del modelo durante el entrenamiento.

El entrenamiento de la red se realizó mediante la función fit, utilizando el conjunto de entrenamiento (train) y especificando el número de épocas, que en este caso fue de 10. También se utilizó un conjunto de validación (val) para monitorear el rendimiento del modelo en datos no vistos durante el entrenamiento.

Durante el entrenamiento, se utilizó el callback TensorBoard para registrar métricas y guardar los registros en un directorio llamado "logs". Esto permitió un seguimiento detallado del proceso de entrenamiento y la capacidad de visualizar métricas como la precisión (accuracy) y la pérdida (loss) a lo largo del tiempo.

8. Aplicación HTML.

Además del entrenamiento y evaluación del modelo de detección de frutas, se llevó a cabo la exportación del modelo entrenado a formato JSON utilizando la biblioteca TensorFlow.js. Esta exportación permitió la integración del modelo en una aplicación web basada en HTML para realizar inferencias en tiempo real.

La exportación a JSON con TensorFlow.js brindó la ventaja de poder aprovechar las capacidades de la tecnología de inteligencia artificial en el navegador, evitando la necesidad de una infraestructura adicional para realizar inferencias.

En la aplicación web, se utilizó la interfaz HTML y JavaScript para cargar el modelo exportado y realizar predicciones sobre imágenes de frutas proporcionadas por el usuario. Esto facilitó la implementación de una solución práctica y accesible, donde los usuarios pueden obtener clasificaciones de frutas con solo cargar una imagen en el navegador.

Esta integración de la detección de frutas entrenada en TensorFlow en una aplicación web amplía aún más las posibilidades de uso y facilita su implementación en diferentes entornos y plataformas.

Ejemplos:

9. Conclusiones.

En conclusión, este informe ha presentado el desarrollo de una red neuronal convolucional para la detección automática de frutas en imágenes. El modelo ha demostrado una precisión promedio del 90% en la clasificación de diferentes tipos de frutas.

El uso del conjunto de datos "Fruits 360" y la arquitectura de la red neuronal han sido fundamentales para el éxito del modelo. Sin embargo, se han identificado desafíos en la detección de frutas con formas irregulares, tamaños pequeños o imágenes con mala iluminación, lo que sugiere la necesidad de investigar técnicas de procesamiento de imágenes más avanzadas para mejorar el rendimiento en estos casos difíciles.

Además, se ha destacado la exportación del modelo a formato JSON utilizando TensorFlow.js, lo que ha permitido su integración en una aplicación web basada en HTML. Esta aplicación ofrece la capacidad de realizar inferencias en tiempo real y brinda una solución práctica y accesible para los usuarios.

En general, el desarrollo de sistemas de detección automática de frutas tiene un gran potencial en diversos campos, como la agricultura, la alimentación y la salud. Los resultados obtenidos en este trabajo sientan las bases para futuras investigaciones y mejoras en la precisión y robustez de los sistemas de detección de frutas.