🏀⚾ ***esferas de deportes*** - Clasificación Multiclase de Imágenes

Descripción del Dataset

Este conjunto de datos contiene más de 9.000 imágenes de balones y pelotas deportivas pertenecientes a 15 deportes diferentes:

* 🏈 Fútbol americano
* ⚾ Béisbol
* 🏀 Baloncesto
* 🎱 Bola de billar
* 🎳 Bola de bolos
* 🏏 Bola de críquet
* ⚽ Fútbol
* 🏌️‍♂️ Bola de golf
* 🏑 Bola de hockey sobre césped
* 🏒 Disco de hockey
* 🏉 Balón de rugby
* 🏸 Volante (bádminton)
* 🏓 Pelota de ping-pong
* 🎾 Pelota de tenis
* 🏐 Voleibol

**Características del dataset**

* Las imágenes fueron extraídas desde Google Images.
* Posteriormente se realizó una revisión manual para eliminar imágenes mal clasificadas (aunque puede que aún haya algunas por error).
* Los duplicados se eliminaron utilizando un algoritmo de comparación visual tipo dHash.
* La división entre entrenamiento y prueba es del 80% para entrenamiento y 20% para validación/test.

**¿Por qué es interesante este dataset?**

* Algunas imágenes pueden ser confusas o engañosas, ya que ciertos balones han sido pintados o modificados visualmente para parecerse a otros.
* Esto representa un reto adicional para los modelos de redes neuronales, ya que deben aprender a distinguir entre clases muy similares o incluso camufladas.

***Tarea:***

**Clasificación Multiclase de Imágenes Deportivas con Redes Neuronales**

**🎯 Objetivo general**

Desarrollar un modelo de red neuronal convolucional (CNN) que permita **clasificar imágenes de pelotas deportivas** pertenecientes a 15 categorías distintas, utilizando el dataset **"Sports Balls - Multiclass Image Classification"**. El modelo debe ser capaz de reconocer visualmente diferentes tipos de balones (como fútbol, baloncesto, béisbol, hockey, etc.), incluso cuando algunos han sido modificados visualmente para parecerse a otros.

**🛠️ Actividades a realizar**

1. **Entrenamiento del modelo CNN**
   * Diseñar y entrenar una red neuronal convolucional utilizando Keras/TensorFlow.
   * Adaptar la arquitectura a la clasificación multiclase (15 clases).
   * Usar el conjunto de entrenamiento proporcionado (80% de las imágenes).
2. **Evaluación del modelo**
   * Calcular la **precisión (accuracy)** y la **función de pérdida (loss)** en los conjuntos de entrenamiento y validación.
   * Generar una **matriz de confusión** para observar qué clases son más difíciles de distinguir.
3. **Visualización de predicciones**
   * Mostrar ejemplos de imágenes clasificadas correctamente e incorrectamente.
   * Comparar las predicciones del modelo con las etiquetas reales.
4. **Visualización del proceso de entrenamiento**
   * Graficar el comportamiento de loss y accuracy por época durante el entrenamiento.
   * Comentar si hubo **sobreajuste** (overfitting) o **subajuste** (underfitting).
5. **Visualización de mapas de activación (Class Activation Maps - CAM)**
   * Generar mapas de activación para identificar qué regiones de la imagen está usando el modelo para tomar decisiones.
   * Interpretar si el modelo está aprendiendo correctamente o si se enfoca en zonas irrelevantes.
6. **Aplicación de *Data Augmentation***
   * Implementar técnicas de aumento de datos (rotación, zoom, inversión horizontal, etc.) para mejorar la robustez del modelo.
   * Comparar los resultados del modelo **con y sin data augmentation**.
7. **Conclusiones**
   * ¿Qué clases se confunden más entre sí? ¿Por qué?
   * ¿Qué mejoras se obtuvieron con *data augmentation*?
   * ¿Qué dificultades trajo este dataset (por ejemplo: pelotas visualmente parecidas, imágenes modificadas)?
   * ¿Qué estrategias se podrían aplicar a futuro (como *transfer learning*) para mejorar el rendimiento?

**Consideraciones específicas del dataset**

* Algunas pelotas están **pintadas o manipuladas visualmente** para parecerse a otras (por ejemplo, una pelota de béisbol pintada como una de fútbol).
* El dataset fue **curado manualmente**, pero aún pueden existir errores de clasificación.
* Las 15 clases tienen **niveles de dificultad visual variados** y pueden requerir mayor profundidad de red o procesamiento adicional.