# Documentación Detallada de modelo\_2.1.py

**Propósito:** Este script implementa un modelo de clasificación de imágenes usando Transfer Learning con MobileNetV2, incluyendo data augmentation, fine-tuning, guardado de modelos y visualización de métricas de entrenamiento.

## 1️⃣ Configuración Inicial

* IMG\_SIZE = (224, 224): tamaño al que se redimensionarán todas las imágenes.
* BATCH\_SIZE = 32: número de imágenes procesadas por lote.
* EPOCHS = 20: número de veces que el modelo verá todo el dataset inicialmente.
* NUM\_CLASSES = 3: clases a clasificar (nuevo, usado, mal\_estado).
* LEARNING\_RATE = 0.0001: tasa de aprendizaje para Adam.
* BASE\_DIR y DATASET\_DIR: rutas al script y dataset.
* Verificación de existencia del dataset.

## 2️⃣ Generadores de Imágenes (Data Augmentation)

* Normalización de píxeles a [0,1].
* División de 80% entrenamiento / 20% validación.
* Data augmentation: rotación, desplazamiento, zoom, corte y volteo horizontal.
* train\_generator y val\_generator: para entrenamiento y validación.

## 3️⃣ Modelo Transfer Learning (MobileNetV2)

* Uso de MobileNetV2 preentrenada en ImageNet.
* include\_top=False: se eliminan capas finales.
* Capas iniciales congeladas (trainable=False).
* Nuevas capas:
  + GlobalAveragePooling2D.
  + Dense(128, relu).
  + Dense(NUM\_CLASSES, softmax).

## 4️⃣ Compilación

* Optimizer: Adam.
* Loss: categorical\_crossentropy.
* Métrica: accuracy.

## 5️⃣ Callbacks

* ModelCheckpoint: guarda el mejor modelo según val\_accuracy.
* EarlyStopping: detiene entrenamiento si val\_loss no mejora por 5 épocas.

## 6️⃣ Entrenamiento Inicial

* Entrenamiento con capas congeladas.
* Guarda historial de accuracy y loss.

## 7️⃣ Fine-Tuning

* Descongela capas superiores del modelo base (capas iniciales congeladas).
* Se usa learning rate más bajo.
* Continúa entrenamiento desde última época.

## 8️⃣ Guardado del Modelo Final

* modelo\_final.h5: modelo final completo.
* modelo\_mejor.h5: checkpoint del mejor modelo.

## 9️⃣ Graficar Historial de Entrenamiento

* Combina métricas de entrenamiento inicial y fine-tuning.
* Gráficas de Accuracy y Loss para entrenamiento y validación.

## ✅ Resumen de Mejoras y Funcionalidades

1. Data Augmentation.
2. Transfer Learning con MobileNetV2.
3. Congelación de capas iniciales.
4. Callbacks: EarlyStopping y ModelCheckpoint.
5. Fine-tuning con learning rate reducido.
6. Guardado automático de modelos.
7. Visualización de métricas de entrenamiento.