Enfoque seleccionado para el reconocimiento de imágenes

Clasificación mediante Transfer Learning

- **Aprovecha conocimiento preexistente**: Utiliza modelos pre-entrenados (como MobileNet o ResNet) que ya capturan características generales de imágenes, reduciendo tiempo y recursos.
- **Eficacia con pocos datos**: El reentrenamiento de capas superiores y el aumento de datos compensan la escasez de imágenes por ítem.
- **Alta precisión**: Ideal si los ítems son similares a las categorías del modelo base, logrando buenos resultados incluso con pocas muestras.

Justicación de la elección

Seleccinamos Transfer Learning por 6 puntos a detallar:

1. **Eficiencia con datos limitados:**
   Al tener solo 1-5 imágenes por ítem, entrenar un modelo desde cero sería inviable. Transfer Learning aprovecha modelos pre-entrenados (como MobileNet o ResNet) que ya capturan características generales de imágenes (bordes, texturas, formas), permitiendo adaptarlos a tus 100 ítems con pocos datos.
2. **Ahorro de tiempo y recursos:**
   Reentrenar solo las capas superiores del modelo reduce drásticamente el tiempo de entrenamiento y el costo computacional, ideal para proyectos con plazos ajustados o hardware limitado.
3. **Alta precisión:**
   Los modelos pre-entrenados están optimizados para tareas genéricas de clasificación. Al ajustarlos con técnicas de aumento de datos (rotaciones, cambios de brillo, recortes), se compensa la escasez de imágenes y se mejora la generalización.
4. **Integración con TensorFlow y OpenCV:**
   - **TensorFlow** ofrece APIs sencillas para aplicar Transfer Learning (ej: `tf.keras.applications`).
   - **OpenCV** facilita el preprocesamiento (redimensionar, normalizar) y el aumento de datos, crucial para maximizar el rendimiento con pocas muestras.
5. **Adaptabilidad al dominio específico:**
   Si los ítems a clasificar comparten características visuales con las categorías del modelo base (ej: objetos comunes, animales), el ajuste fino

(*fine-tuning*) logrará alta precisión. Incluso si son distintos, las capas iniciales del modelo ya extraen patrones útiles.

6. **Escalabilidad:**
   Si el proyecto crece (más ítems o imágenes), el enfoque se mantiene viable. Puedes seguir usando Transfer Learning o combinar con otras técnicas sin reestructurar todo el sistema.

Consideraciones básicas para implementar Transfer Learning con MobileNet:

Preprocesamiento de imágenes

Tamaño de entrada:
MobileNet requiere imágenes de **224x224 píxeles**. Usa OpenCV para redimensionar

Normalización:
Asegúrate de escalar los valores de píxeles al rango **[-1, 1]** o **[0, 1]**, según el modelo base.

Congelamiento de capas base

Evitar sobreajuste:
Congela las capas convolucionales de MobileNet para preservar las características aprendidas en ImageNet

Aumento de datos

Generar variedad artificial
Usa técnicas como:

- **Rotaciones (±20°), volteos horizontales.**
- **Ajustes de brillo/contraste.**
- **Recortes aleatorios.**

Compilación y entrenamiento

Optimizador y función de pérdida

Integración con OpenCV

Conversión de color:
OpenCV carga imágenes en formato BGR, pero MobileNet espera RGB

7. Evaluación y ajustes

Métricas clave:

- Precisión en el conjunto de validación.
- Pérdida de entrenamiento vs. validación (para detectar sobreajuste).

Si la precisión es baja:

- Aumenta el número de imágenes sintéticas.
- Descongela capas intermedias de MobileNet para *fine-tuning*.

Despliegue

Exportar el modelo:

Optimización para producción:

Usa TensorFlow Lite si el sistema se ejecuta en dispositivos móviles