Informe Final - Lab 2 – Rodolfo Varela

# 1. Objetivo

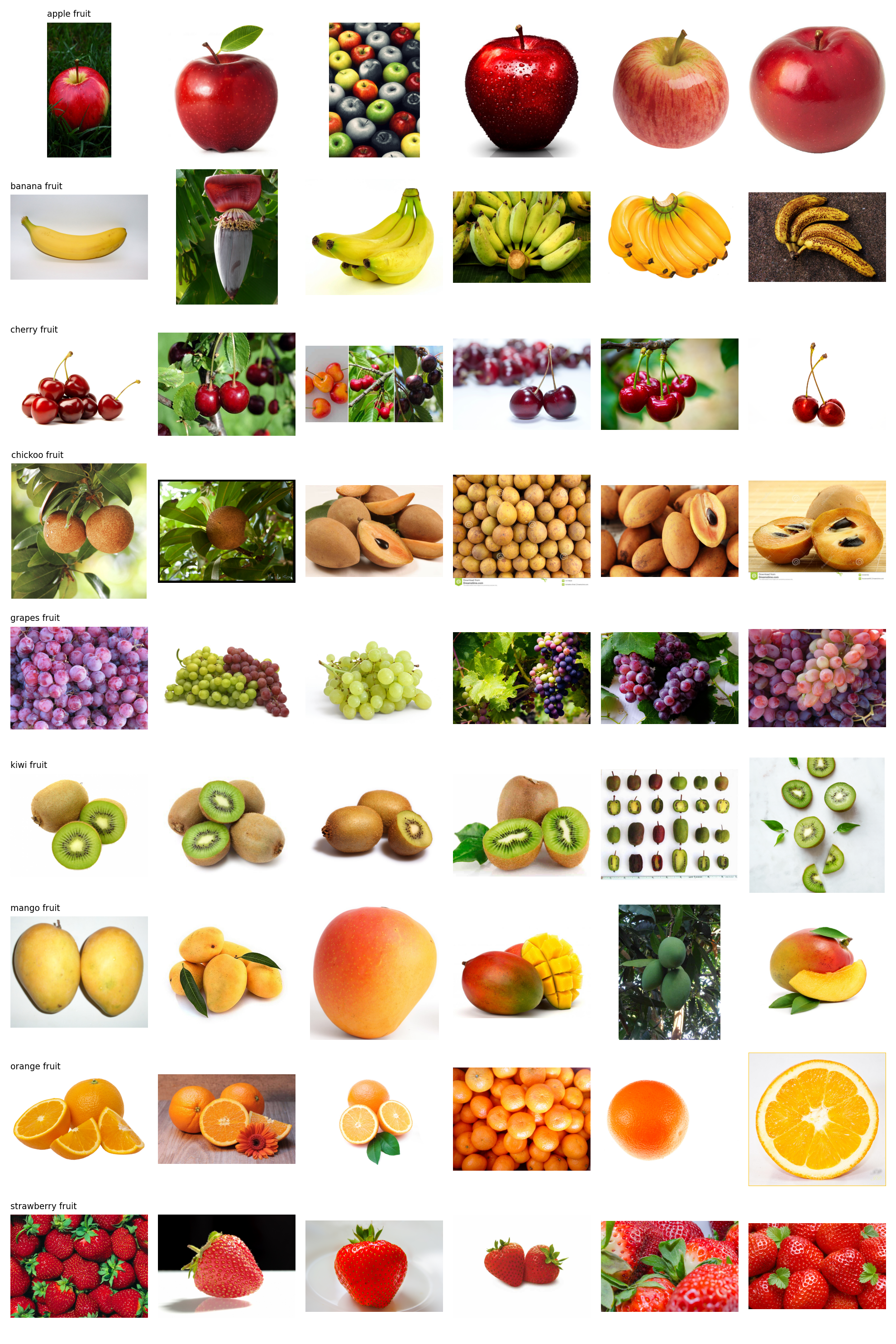
Aplicar transfer learning con VGG19 para clasificar imágenes de frutas, comparando contra baselines clásicos.

Se extrajeron deep features por etapas para entrenar SVM/KNN/RandomForest y se realizó un fine‑tuning parcial. Se reportan accuracy de validación, matrices de confusión y análisis de confianza (reliability).

# 2. Datos y preparación

• Dataset con 9 clases: apple, banana, cherry, chickoo, grapes, kiwi, mango, orange, strawberry.  
• Split 80/20 (train/val) por archivos.  
• Preprocesado: 224×224 y normalización de ImageNet (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]).

Figura 1. Mosaico de ejemplos por clase (train).



# 3. Metodología

## 3.1 Extractor de deep features

Backbone VGG19 (pre‑entrenada en ImageNet) operando como extractor congelado. Se tomaron las salidas tras MaxPool y se aplanaron (stages 1–3 por costo).

## 3.2 Modelos sobre features

SVM‑RBF, KNN‑5 y RandomForest‑200. Además, pipeline SVM + PCA (StandardScaler with\_mean=False + PCA con n\_components seguro por CV).

## 3.3 Concat de stages y fine‑tuning

Concat features de stage 2 y 3 (orden estable sin shuffle). Fine‑tuning: reemplazo de la capa final y unfreeze del último bloque conv con LR menor.

## 3.4 Métricas

Accuracy, matrices de confusión y reliability (accuracy por bin vs. confianza).

# 4. Resultados

Tabla 1. Resumen de accuracies (validación).

|  |  |  |
| --- | --- | --- |
| stage | modelo | accuracy |
| stage3 | RF-200 | 0.819 |
| stage2 | RF-200 | 0.694 |
| stage1 | RF-200 | 0.583 |
| stage2 | SVM-RBF | 0.556 |
| stage3 | SVM-RBF | 0.514 |
| stage1 | SVM-RBF | 0.417 |
| stage3 | KNN-5 | 0.389 |
| stage2 | KNN-5 | 0.194 |
| stage1 | KNN-5 | 0.181 |

Figura 2. Matriz de confusión – VGG19 fine‑tuning (mejor modelo).

Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.

Figura 3. Reliability plot – RandomForest (Stage 3).

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

# 5. Discusión

• TL vs. baselines: deep features + SVM/RF > SimpleCNN (~0.78 vs. 0.49).  
• Stages: los profundos (stage 3) son más discriminativos y menos costosos que stage 1.  
• Concat 2+3: útil marginalmente; no supera al fine‑tune.  
• Fine‑tune: mejor resultado (0.847). Con más épocas/unfreeze/LR menor suele superar 0.90.  
• Confianza: RF conservador (masa en 0.25–0.5); calibrar si se usan probabilidades.

# 6. Conclusiones

El transfer learning con VGG19 + clásicos rinde bien; el fine‑tuning domina. Se alcanzó 0.847 de accuracy en validación. Próximos pasos: aug extra, early stopping, calibración y probar ResNet18/50 con FT similar.

# 7. Reproducibilidad

Ejecutar `main.py` con entorno Anaconda configurado. Entradas: `archive/images/`; salidas: `features/\*.npz`, `outputs/\*.csv|.png|.pth`. Fijar seeds y `shuffle=False` al extraer features para concatenación reproducible.

# 8. Github

https://github.com/RodolfoVarela/Laboratorios\_MLOps/tree/main