

Proyecto 3 — Transfer Learning para Clasificación de Frutas y Verduras en PyTorch

INFO1185: Inteligencia Artificial
Prof. Dr. Ricardo Soto Catalán

19 de noviembre de 2025

1. Contexto general

El procesamiento de imágenes mediante redes neuronales convolucionales (CNN) ha demostrado un rendimiento sobresaliente en tareas de clasificación visual. Sin embargo, entrenar modelos complejos desde cero requiere grandes volúmenes de datos y recursos computacionales significativos. Para abordar estas limitaciones, el *Transfer Learning* permite reutilizar modelos preentrenados en grandes bases de datos (como ImageNet) y adaptarlos a nuevas tareas específicas con un menor costo computacional.

En este proyecto, los estudiantes deberán aplicar esta técnica para clasificar diferentes tipos de frutas o verduras utilizando un subconjunto del *Fruits and Vegetables Image Recognition Dataset*, disponible en: Kaggle.

2. Objetivo del proyecto

El objetivo principal es que el/la estudiante sea capaz de adaptar y ajustar un modelo preentrenado utilizando *transfer learning*, construyendo dos variantes del clasificador final y comparando su desempeño en un problema realista de clasificación de imágenes.

3. Descripción del trabajo

El presente trabajo se realizará en grupos de 2 integrantes.

Cada equipo deberá seleccionar uno de los siguientes modelos de `torchvision.models`: VGG-16, ResNet18, ResNet50, DenseNet121, MobileNetV3, EfficientNetV2, GoogLeNet, InceptionV3, SqueezeNet o ShuffleNet. El modelo seleccionado deberá ser adaptado a uno de los problemas de clasificación propuestos.

Para seleccionar el modelo y el problema de clasificación, cada grupo debe inscribirse en el siguiente enlace: Formulario de inscripción.

3.1 Construcción de dos variantes del modelo

A) Versión 1: Clasificador simple

- Reemplazar la capa final por una única capa *Fully Connected* (FC) con salida igual al número de clases.
- No utilizar *Batch Normalization* ni *Dropout*.

B) Versión 2: Clasificador extendido tipo embudo

- Construir un clasificador con al menos dos capas ocultas.
- Utilizar una arquitectura tipo embudo (por ejemplo: $512 \rightarrow 256 \rightarrow 128 \rightarrow N$).
- Incluir *Batch Normalization* en cada capa oculta.
- Incluir *Dropout* con probabilidad entre 0.2 y 0.5.
- Utilizar funciones de activación no lineales (por ejemplo, ReLU).

Además, para la Versión 2, el equipo deberá realizar al menos dos entrenamientos:

- Uno sin *Batch Normalization* ni *Dropout*.
- Otro aplicando ambas técnicas de regularización.

3.2 Entrenamiento y validación

- Aplicar transformaciones apropiadas: redimensionado, normalización y *data augmentation* en el conjunto de entrenamiento.
- Entrenar ambas variantes utilizando *early stopping*.
- Registrar la pérdida de entrenamiento y de validación.
- Generar gráficos que muestren las curvas de pérdida (*loss curves*) durante el entrenamiento.

3.3 Evaluación final

- Evaluar cada modelo en el conjunto de prueba.
- Reportar la matriz de confusión.
- Incluir métricas de desempeño: exactitud (*accuracy*), precisión (*precision*) y sensibilidad (*recall*) por clase.

3.4 Análisis y discusión

El informe debe incluir una comparación detallada entre ambas variantes, analizando:

- El impacto del uso o ausencia de *Batch Normalization* y *Dropout*.
- La estabilidad del entrenamiento.
- Las diferencias de desempeño entre variantes.
- Limitaciones observadas al trabajar con los recursos disponibles en Google Colab.

4. Entregables

- **Notebook de Google Colab** completamente ejecutable, con el pipeline completo: carga de datos, entrenamiento, evaluación y generación de gráficos.
- **Presentación PowerPoint** (o equivalente), que incluya:
 - Descripción del problema de clasificación seleccionado.
 - Arquitectura del modelo base y del clasificador adaptado.
 - Estrategia de entrenamiento (tasa de aprendizaje, optimizador, función de pérdida, *data augmentation*, técnicas de regularización).
 - Resultados obtenidos.
 - Comparación entre las variantes entrenadas.
 - Conclusiones del equipo.
- **Presentación oral:** cada grupo deberá exponer su trabajo al curso en un tiempo máximo de 8 minutos.

Fecha de entrega (código + presentación): Miércoles 03 de diciembre de 2025 a las 13:00 hrs.

Las presentaciones se realizarán en el horario de clases, iniciando el día 03 de diciembre a las 13.50 hrs.

La calificación final del proyecto se obtiene a partir del puntaje logrado en las Rúbricas 1 (Código) y 2 (Presentación), considerando la escala de exigencia del 60%.

Rúbrica 1: Evaluación del Código (Notebook Colab)

Categoría	Excelente (1)	Bien (0.66)	Suficiente (0.33)	Deficiente (0)
1. Implementación del modelo preentrenado (10 pts.)	Modelo preentrenado bien cargado y adaptado. Ambas variantes construidas correctamente (FC simple / embudo con BN y Dropout).	Implementación correcta con detalles menores faltantes o simplificados.	Variantes incompletas o con errores conceptuales moderados.	Implementación incorrecta o no funcional; una o ambas variantes ausentes.
2. Preparación del dataset y DataLoaders (10 pts.)	Dataset organizado; separación clara train/val/test; transformaciones correctas (augmentación solo en train).	Conjuntos definidos y transformaciones adecuadas, con pequeños detalles faltantes.	Separación de datos mínima o con inconsistencias en transformaciones.	No existe separación train/val/test o el pipeline es incorrecto.
3. Entrenamientos requeridos y early stopping (10 pts.)	Ejecuta todos los entrenamientos: V1, V2-sin BN/Dropout, V2-con BN/Dropout. Early stopping bien implementado. Parámetros documentados.	Entrenamientos realizados, aunque con uso básico o poco documentado de early stopping.	Solo parte de los entrenamientos requeridos; early stopping mal aplicado.	Entrenamientos incompletos, incorrectos o no ejecutables.
4. Métricas, gráficos y evaluación (10 pts.)	Incluye curvas de pérdida train/val, matriz de confusión y métricas por clase (accuracy, precision, recall). Gráficos claros y comentados.	Incluye métricas y gráficos principales, con algunas ausencias menores.	Solo métricas básicas o gráficos parciales; evaluación incompleta.	Sin gráficos, sin métricas relevantes o evaluación insuficiente.
5. Claridad, estructura y calidad del código (10 pts.)	Código limpio, comentado, organizado por secciones. Notebook ejecuta de inicio a fin sin errores.	Código claro en general, con algunos desórdenes o pocos comentarios.	Código funcional pero difícil de seguir o pobremente comentado.	Código desordenado, con errores o imposible de ejecutar completamente.

Rúbrica 2: Evaluación de la Presentación

Categoría	Excelente (1)	Bien (0.66)	Suficiente (0.33)	Deficiente (0)
1. Descripción del problema (5 pts.)	Problema claramente explicado; clases, motivación y objetivo bien definidos.	Descripción clara pero con menos detalle.	Explicación superficial; faltan elementos clave.	Problema poco claro o no descrito.
2. Descripción del dataset (5 pts.)	Cantidad de imágenes, clases, división train/val/test explicada con claridad y precisión.	Dataset descrito correctamente, aunque con detalle limitado.	Descripción parcial o imprecisa.	Dataset no explicado o explicado incorrectamente.
3. Explicación del modelo y variantes (10 pts.)	Modelo preentrenado explicado correctamente. Diferencias entre las dos variantes claras. Justificación técnica bien argumentada.	Explicación adecuada, con algunos elementos menos desarrollados.	Explicación incompleta o con confusiones.	Explicación incorrecta o ausente.
4. Resultados, métricas y gráficos (10 pts.)	Resultados bien presentados: curvas, matrices, métricas. Comparación profunda entre variantes. Análisis crítico.	Resultados claros con análisis básico. Comparación suficiente.	Resultados mínimos y análisis superficial.	Resultados incompletos o sin análisis.
5. Conclusiones y reflexión (10 pts.)	Conclusiones sólidas, conectadas con los resultados. Discusión de errores, limitaciones y mejoras futuras.	Conclusiones correctas pero sin profundidad.	Conclusiones muy generales o poco vinculadas al trabajo.	Conclusiones ausentes o incorrectas.
6. Comunicación oral y uso del tiempo (10 pts.)	Presentación fluida, bien estructurada. Uso adecuado del tiempo (≤ 8 min). Participación equilibrada del equipo. Slides claras y profesionales.	Buena comunicación y uso aceptable del tiempo. Slides correctas.	Comunicación entendible pero con problemas de ritmo o coordinación. Slides sobrecargadas o con falta de información.	Presentación confusa, mal organizada o excesivamente larga/corta. Falta participación del equipo.
7. Respuesta a preguntas (10 pts.)	Responde con claridad, precisión y seguridad. Evidencia dominio técnico. Distribución equilibrada entre integrantes.	Responde correctamente la mayoría de las preguntas con seguridad moderada.	Respuestas superficiales o incompletas. Evidencia dominio parcial.	No logra responder preguntas clave; evidencia desconocimiento del trabajo.