TALLER 2

Juan Daniel Castrellón (201729285); Javier Alejandro Gómez (201217975)

# Entrenamiento del modelo de Machine Learning

1. Dimensiones del conjunto de datos

El conjunto de datos está dividido en 3 conjuntos principales: entrenamiento, prueba y validación. Cada uno con de los conjuntos se conforma por un numero de variable de imágenes (expresado a continuación) en formato ‘.jpg’ y por un archivo de etiquetas con tres variables (la ruta, la clase fina y la clase gruesa). De acuerdo con lo descrito anteriormente, las dimensiones son (en número de registros):

* 1. Train: 2640
  2. Val: 2485
  3. Test: 296

1. Clases en el conjunto de datos

Hay dos clases por cada uno de los registros:

* 1. Clases gruesas

Hay 43 clases gruesas que representan categorías generales de productos. Ejemplos incluyen:

Apple, Juice, Milk, Mango, Tomato, Potato, entre otros.

Estas clases proporcionan una visión general de los tipos de productos que están siendo clasificados y servirán como una categoría intermedia en el análisis.

* 1. Clases finas

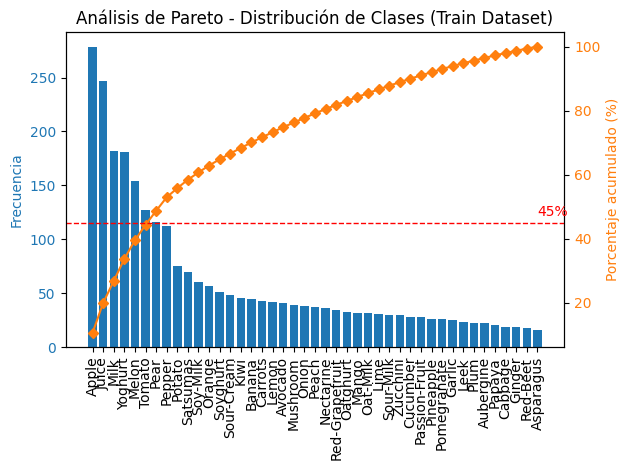
Hay 81 clases finas, que corresponden a categorías más detalladas dentro de cada clase gruesa. Estas serán fundamentales para clasificar productos específicos, como diferenciar entre "Royal Gala" y "Granny Smith" dentro de la clase gruesa "Apple".

1. Distribución de datos en entrenamiento

La distribución de los datos muestra un desbalance considerable, donde los productos más representados son:

1. Manzanas (Apple)
2. Jugos (Juice)
3. Leche (Milk)

Este desbalance puede afectar el desempeño del modelo, ya que las clases menos representadas no tienen suficiente información para un aprendizaje adecuado. Para su descripción se realizó un gráfico de Pareto, con una marca para el 45% de los datos:



1. Selección de clases

Seleccionar el 45% de los datos basándonos en el análisis de Pareto permite enfocarnos en las clases más representativas del conjunto de datos, maximizando la información contenida mientras se reduce significativamente la complejidad del problema, ya que esta tiene un impacto directo en el consumo de recursos computacionales (CPU-Memoria). Por tanto, dada la subrepresentación de muchas clases, se definió un umbral de al menos 120 imágenes por clase con el fin de ser considerada como una clase para ser clasificada, por tanto, las clases gruesas con las que se trabajará para la modelación son (clase, número de registros):

* Apple 278
* Juice 247
* Milk 182
* Yoghurt 181
* Melon 154
* Tomato 127

1. Balance de clases

Para las clases escogidas, se hizo un mapeo a un nuevo número de clase, de modo tal que se verán de esta forma:

|  |  |
| --- | --- |
| **Coarse Class str** | **Coarse Class ID** |
| Apple | 0 |
| Juice | 1 |
| Milk | 2 |
| Yoghurt | 3 |
| Melon | 4 |
| Tomato | 5 |

1. Características de las imágenes

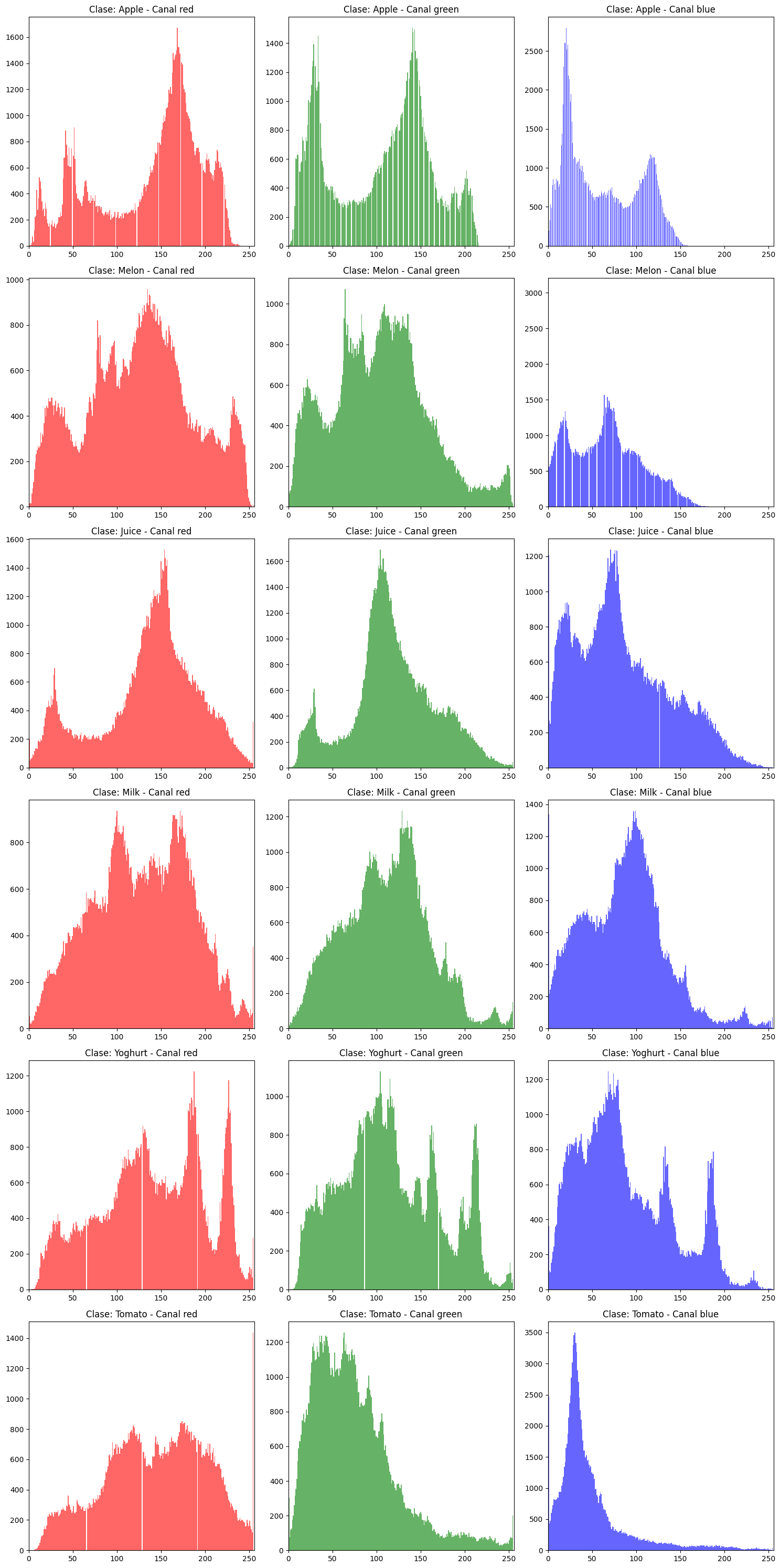
Las imágenes tienen canales RGB, y sus características se describen a continuación:

* Máxima altura: 464
* Máxima anchura: 464
* Máxima área: 161472
* Mínima altura: 348
* Mínima anchura: 348
* Mínima área: 121104

1. Histogramas de color

El gráfico que se presenta a continuación muestra histogramas de color (RGB) de diferentes clases de productos, cada uno dividido en los canales rojo, verde y azul. Los histogramas representan la distribución de intensidades de color (en un rango de 0 a 255) para las imágenes de las respectivas clases seleccionadas previamente.  
  
Ejes:

* Eje X: Valores de intensidad de color (0 a 255).
* Eje Y: Frecuencia de píxeles con esa intensidad.



Observaciones por Clase

* Apple:
  + Predominancia en el canal rojo y verde, con una menor contribución del canal azul.
  + La intensidad de color está distribuida principalmente en valores medios y altos, lo que sugiere tonos cálidos.
* Melon:
  + Similar a Apple, pero con un pico más uniforme en los canales rojo y verde.
  + El canal azul tiene una contribución baja, indicando colores cálidos dominantes.
* Juice:
  + Mayor contribución del canal rojo, con picos definidos en valores altos.
  + El canal verde tiene una distribución media, mientras que el azul es mínimo, lo que sugiere tonos rojizos.
* Milk:
  + Los tres canales tienen distribuciones más uniformes.
  + El color blanco dominante en "Milk" se refleja en la contribución balanceada de los canales RGB.
* Yoghurt:
  + Similar a Milk, pero con más variación en los canales rojo y verde.
  + La distribución uniforme en todos los canales sugiere colores claros con matices cálidos.
* Tomato:
  + Predominancia fuerte en el canal rojo, con picos altos en los valores intermedios.
  + El canal azul tiene una contribución mínima, indicando colores cálidos intensos.

Por tanto, frente a estas descripciones, de las clases escogidas se puede decir que las clases como Apple, Juice y Tomato tienen histogramas dominados por los canales rojo y verde, lo que es consistente con colores cálidos como rojos y anaranjados. Las clases como Milk y Yoghurt presentan distribuciones más equilibradas en los tres canales, lo que refleja su naturaleza de colores claros (blancos o cremas). Las distribuciones en el canal azul son menores en general, excepto para clases más claras como Milk.

1. Aumentación de datos y balance de clases

* Redimensionamiento de datos
  + Las imágenes fueron redimensionadas a un tamaño fijo de 256x256 píxeles utilizando interpolación bicúbica con cv2.resize.
  + Posteriormente, las imágenes fueron normalizadas dividiendo los valores de los píxeles entre 255, escalando así los valores entre 0 y 1. Esto facilitó el entrenamiento de la red neuronal y aceleró la convergencia.
* Balance de Clases con Aumentación de Datos
  + Se determinó que la clase con el mayor número de imágenes serviría como referencia, y todas las demás clases generaron nuevas muestras hasta alcanzar este máximo.
  + Se utilizó ImageDataGenerator para aplicar transformaciones a las imágenes existentes:
    - Rotaciones hasta 45 grados
    - Desplazamientos horizontales y verticales
    - Transformaciones de perspectiva y zoom aleatorio
    - Generación de imágenes reflejadas
    - Los Píxeles introducidos por transformaciones fueron llenados de manera cercana al color de los bordes.
* Preparación del Conjunto de Entrenamiento Balanceado
  + Se creó un conjunto balanceado combinando las imágenes originales con las generadas hasta alcanzar el número máximo de imágenes en cada clase.
* Preparación de los Conjuntos de Validación y Prueba
  + Los conjuntos de validación y prueba no requirieron aumentación. Sin embargo:
    - Las imágenes fueron redimensionadas al mismo tamaño (256x256 píxeles) para mantener consistencia con el conjunto de entrenamiento.
    - Las imágenes fueron normalizadas dividiendo los valores de los píxeles entre 255.
* Extracción de Estadísticas de Color
  + Se calcularon estadísticas detalladas de los canales RGB de cada imagen para análisis adicionales o como posibles características complementarias para modelos.
  + Estadisticas extraidas:
    - Básicas: Media, desviación estándar, mediana, percentiles (25 y 75).
    - Avanzadas:
      * Asimetría (skew): Indicó la simetría de la distribución de los valores del canal.
      * Curtosis: Evaluó la concentración de valores en los extremos.
    - Adicionales: Rango (max-min) y varianza.

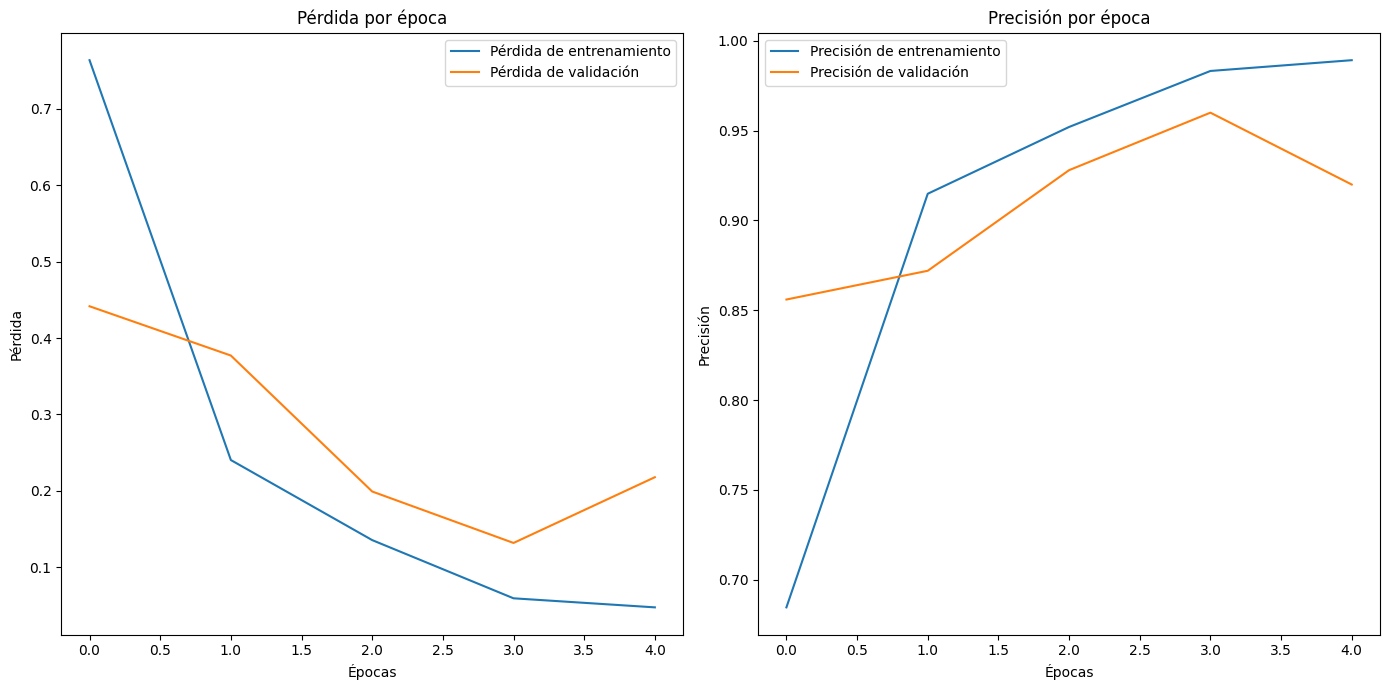
1. Almacenamiento de datos para la modelación

Para su posterior análisis, las clases se almacenaron en un archivo pickle denominado ‘data.pickle’

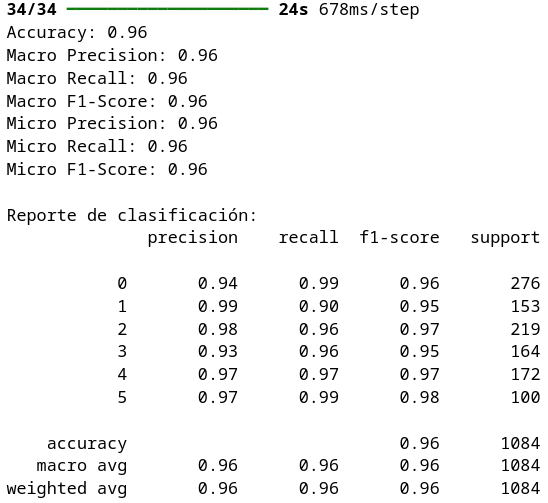
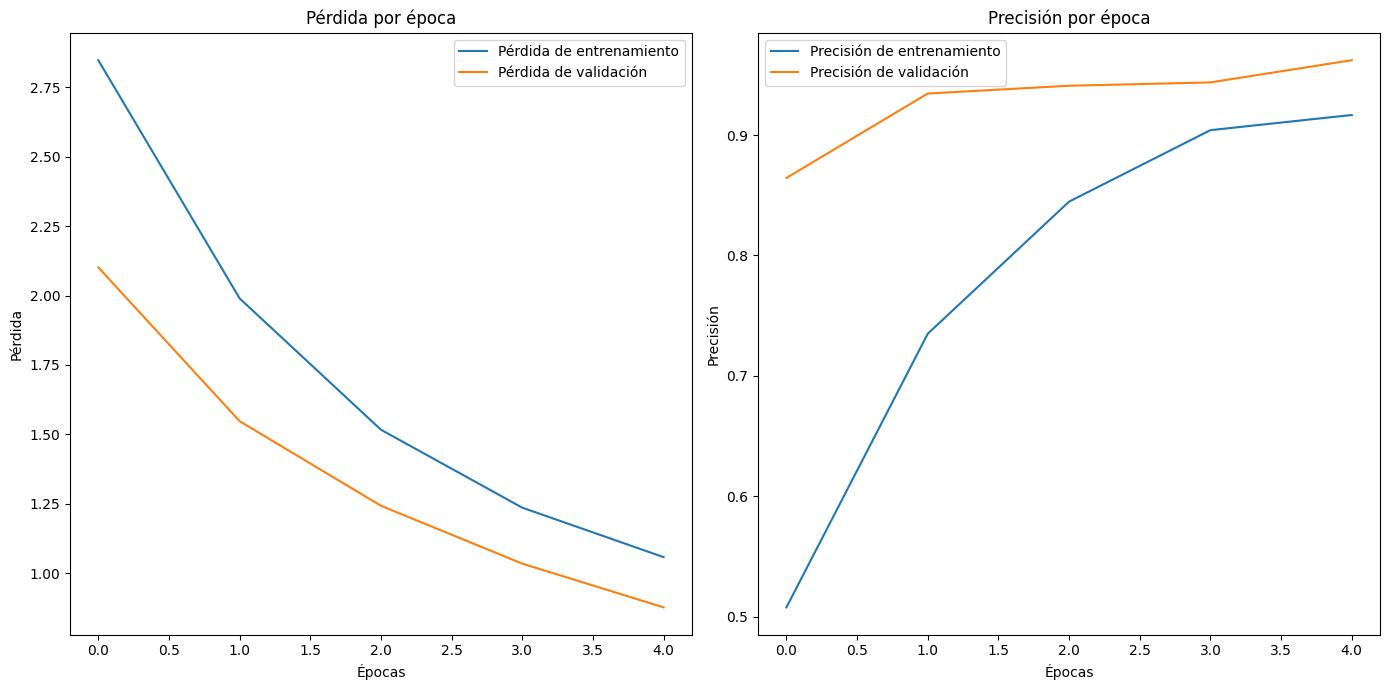
# Análisis y resultados del modelo

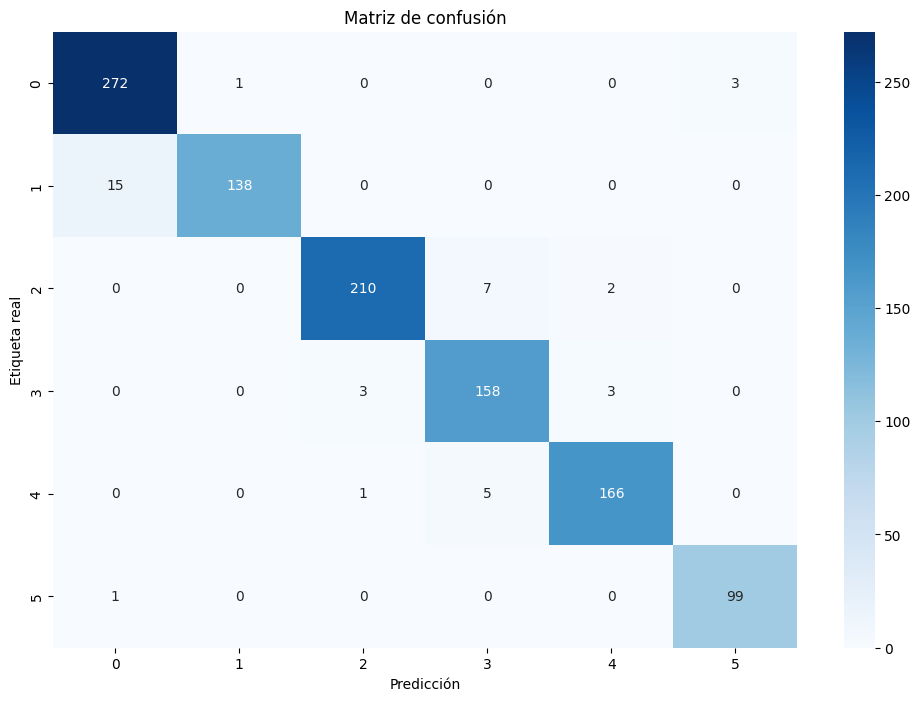
Modelo 1

El modelo 1 base utilizó la arquitectura MobileNetV2 preentrenada en ImageNet como extractor de características visuales, configurada con entradas de tamaño (256, 256, 3) y sin incluir sus capas superiores (include\_top=False). La base del modelo se congeló para preservar los pesos preentrenados, aprovechando su capacidad para extraer representaciones útiles de las imágenes sin modificar sus parámetros.  
  
Sobre esta base, se añadieron capas personalizadas: una capa de global average pooling para reducir las dimensiones de las características, seguida de dos capas densas de 64 unidades con activación ReLU y dropout del 20% para mitigar el sobreajuste. Finalmente, se incluyó una capa de salida con activación softmax, configurada para clasificar las imágenes en un número de clases equivalente a las etiquetas únicas del conjunto de entrenamiento.

El modelo se compiló con el optimizador Adam (tasa de aprendizaje de 0.001), la función de pérdida sparse\_categorical\_crossentropy y la métrica de precisión. Se entrenó durante 5 épocas, con un tamaño de batch de 16, utilizando un conjunto de entrenamiento balanceado y un conjunto de validación para evaluar el desempeño. Esta arquitectura permitió aprovechar la robustez de MobileNetV2 combinada con capas densas ajustadas a las necesidades de la tarea de clasificación.  
  
Con base a la gráfica anterior, se identificar que:

* La pérdida de validación no disminuye consistentemente después de unas pocas épocas, lo que podría indicar cierto sobreajuste.
* La precisión de entrenamiento es consistentemente más alta que la precisión de validación, lo que también es un signo potencial de sobreajuste.
* El modelo converge rápidamente, ya que la pérdida de entrenamiento disminuye significativamente en las primeras épocas.

Por tanto, para la búsqueda de hiperparámetros y reducir el sobreajuste, se entrenó un modelo igual al descrito, pero se aumentaron los valores de la regularización y se agregó un early stopping, conservando el número de épocas.  
  




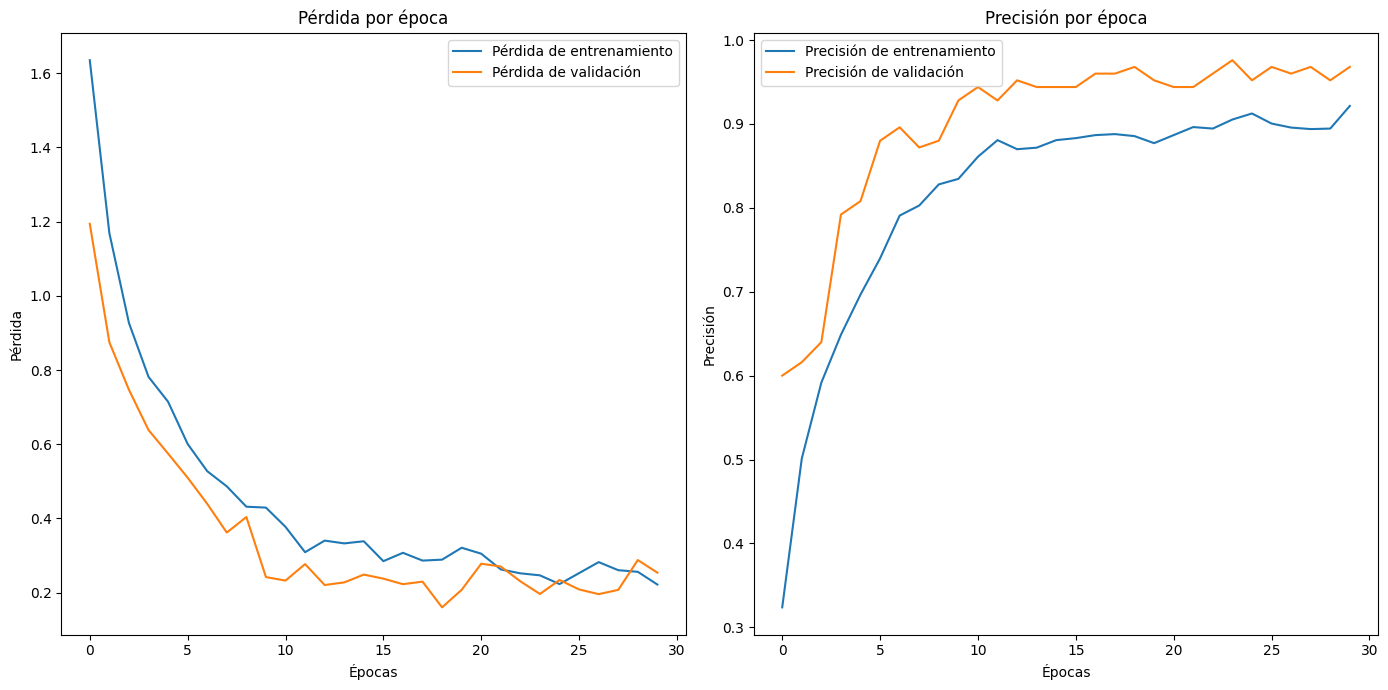
La precisión de validación comienza en 0.8644 (86.44%) en la primera época y alcanza un valor final de 0.9622 (96.22%) en la quinta época. Esto indica que el modelo es capaz de generalizar bien incluso desde las primeras etapas del entrenamiento y mejora constantemente con cada época. La pérdida de validación disminuye consistentemente de 2.1016 en la primera época a 0.8772 en la quinta época. Esta tendencia es un signo de que el modelo está ajustándose bien a los datos sin sobre ajustarse, ya que la pérdida sigue disminuyendo sin signos de estancamiento o aumento. La regularización implementada (como dropout) está ayudando a evitar que el modelo memorice los datos de entrenamiento, lo que mejora la generalización.  
  
En conclusión, sobre el modelo 1 optimizado, se ve un desempeño sobresaliente con un 96% de precisión general, métricas balanceadas entre las clases y los errores están distribuidos uniformemente.

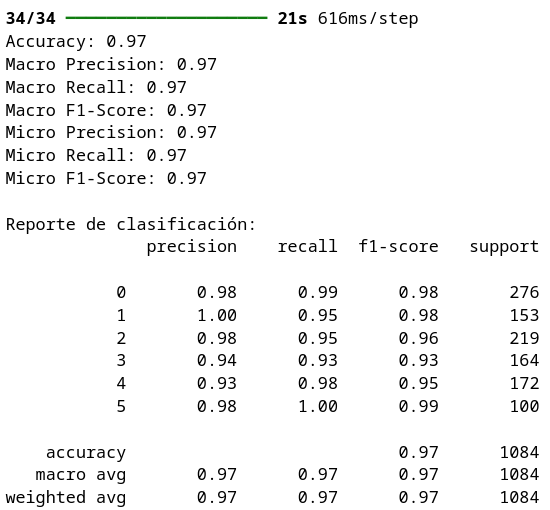
Modelo 2

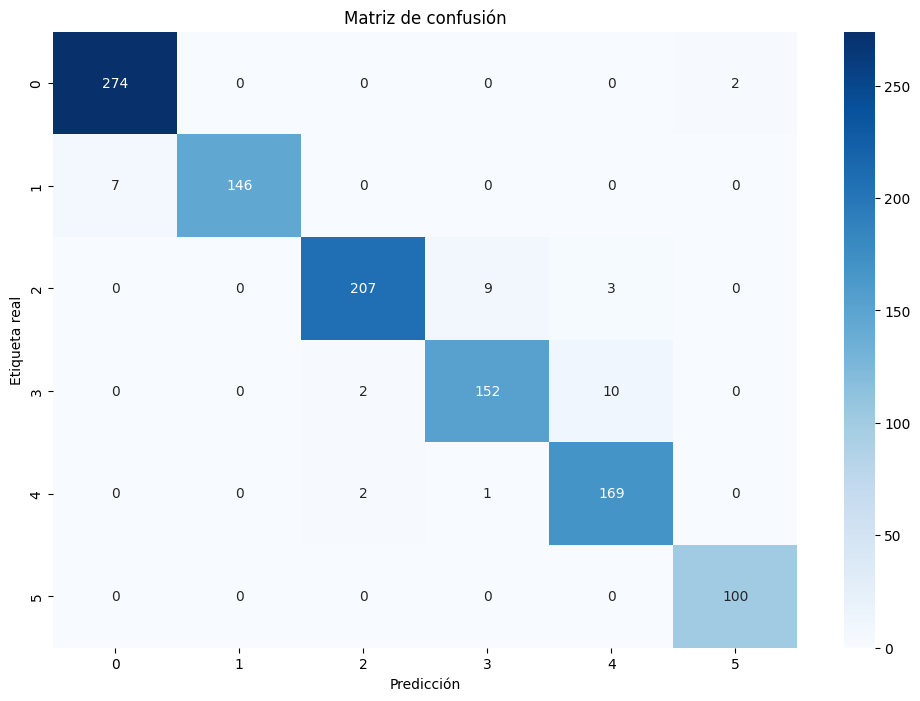
Este modelo combinó información visual y estadística para realizar la clasificación. Se utilizó MobileNetV2, preentrenado en ImageNet, como extractor de características visuales a partir de imágenes de entrada con tamaño (256, 256, 3). La arquitectura base de MobileNetV2 se congeló para mantener los pesos preentrenados, y se añadió una capa de global average pooling, seguida de una capa densa de 64 unidades con activación ReLU y una regularización mediante dropout del 50%. Esta rama de características visuales procesó las imágenes.

Además, se integró una segunda rama que procesó las características estadísticas derivadas de las imágenes (como histogramas de color u otras métricas). Esta rama utilizó un vector de entrada con tantas dimensiones como columnas tenía el conjunto estadístico, seguido de una capa densa de 64 unidades con activación ReLU y dropout del 50%.

Las dos ramas se combinaron mediante concatenación, creando una representación enriquecida de las características visuales y estadísticas. Sobre esta combinación se añadieron dos capas densas adicionales, cada una con 64 unidades, activación ReLU y dropout del 50% para mitigar el sobreajuste. Finalmente, se incluyó una capa de salida con activación softmax para realizar la clasificación multiclase, adaptándose al número de clases en el conjunto de datos.

El modelo se compiló utilizando el optimizador Adam con una tasa de aprendizaje de 0.001, la función de pérdida sparse\_categorical\_crossentropy, y se evaluó con la métrica de precisión. El entrenamiento se realizó durante 30 épocas con un tamaño de batch de 16, utilizando un conjunto de entrenamiento balanceado que incluía imágenes y características estadísticas, y un conjunto de validación con la misma estructura. Esta arquitectura permitió aprovechar tanto las características visuales extraídas de las imágenes como las métricas estadísticas derivadas, logrando un modelo más robusto y adaptado a las particularidades del problema. 





Los dos modelos lograron un rendimiento similar, por lo que basándonos en las métricas de precisión y su capacidad de generalizar sobre el conjunto de pruebas, concluimos que el modelo a usar (Modelo 2) tiene una precisión de 97% (sobre datos de prueba) y generaliza adecuadamente. el modelo es altamente confiable y generaliza bien, con un equilibrio óptimo entre precisión y recall, lo que lo hace adecuado para tareas de clasificación en un entorno real.

# Generación de valor

El análisis incluye suposiciones y cálculos estimados para determinar el ahorro generado por el modelo de clasificación y el punto de equilibrio (break-even point) tomando en cuenta los costos y beneficios. Todos los valores se expresan en pesos colombianos (COP) y se basan en datos y supuestos razonables. Los datos se estimaron registrando el promedio de las actividades, muestreando

Los costos de tiempo asociado al registro de productos.

El proceso manual de registro de productos en un supermercado implica que el cajero escanee o ingrese manualmente cada artículo. Según lo medido *in situ* en un supermercado en la localidad de las nieves (Tienda ARA, City U), el tiempo promedio para registrar un producto es de aproximadamente 3 segundos. Considerando una jornada laboral de 8 horas, un cajero puede procesar alrededor de 9,600 productos al día.

El ahorro de tiempo teórico de su modelo.

El modelo de reconocimiento de productos automatiza el proceso de registro, reduciendo el tiempo necesario por producto. Si el modelo permite registrar un producto en 1 segundo, se ahorran 2 segundos por producto en comparación con el método manual.

El costo asociado a los errores de su modelo.

Los errores en el reconocimiento pueden llevar a registros incorrectos, estimamos que un reproceso por un reproceso tiene un costo asociado de 120 segundos, según lo visto en la tienda. Basándonos en el salario mínimo diario en Colombia para 2023 ($38,667 COP diarios, equivalente a $1.34 COP por segundo), el costo asociado al reproceso de cada error sería:

1.34 $/segundos \* 120 segundos = $ 161 COP.  
  
Por tanto, el costo por error es:

Costo por error = 0.03×$161=$4.83 COP por producto

El ahorro real por predicción acertada.

Ahorro por predicción acertada: Estimamos que el modelo reduce el tiempo de registro manual en 2 segundos por producto. Considerando que el salario mínimo mensual en Colombia en 2023 es de 1,160,000 COP (Fuente: Decreto 2613 de 2022), y asumiendo un mes laboral de 160 horas, el costo por segundo de un cajero es:

1,160,000 COP / (160 horas × 3600 segundos) = 2.01 COP/segundo.

Ahorro por producto registrado mensual es de: 57600 peos diarios, y de $1.728.000 COP menuslaes

El ROI tomando en cuenta los costos de desarrollo y despliegue de su modelo.

Tiempo para alcanzar el ROI de 0: El tiempo necesario para que los ahorros acumulados igualen el costo de desarrollo es:  
  
- Considerando que el modelo tardo 16 horas en construirse y desplegarse, a 0 costo, ya que se busca usar las mismas maquinas de registro, y el salario por hora para un cientifico de datos es de $28.700 COP/hora, el costo total del modelo es de $459.200 COP.

Por tanto, este modelo tendra un ROI de 0 despues de: $ 459.200 COP / $1.728.000 = 0.26 Meses o 8 días.

# Insights

- Con un ahorro estimado de 1,728,000 COP al mes y un punto de equilibrio de solo 8 días, el modelo ofrece una solución rentable y escalable para supermercados que buscan automatizar procesos. Aunque se usaron solo 6 clases, esto indica que podría recolectarse un numero de fotos por clase de 300 y alcanzar un rendimiento similar.  
  
- La combinación de MobileNetV2 para características visuales y una rama para histogramas de color demuestra ser efectiva. Este enfoque permite capturar información complementaria de los datos, mejorando el desempeño del modelo combinado.

* Al reducir 2 segundos por producto registrado, el modelo no solo ahorra tiempo, sino que también mejora la experiencia del cliente al disminuir los tiempos de espera en caja.
* El enfoque podría adaptarse a otras clases no incluidas aqui, expandiendo así su impacto.