

# Memoria del Proyecto: Clasificación Multietiqueta con CNNs

---

Alumnos: José Delgado, Gonzalo de Antonio, Javier Orti, Joan Benlloch, Alexander Bas

## 1. Temática y características de la clasificación realizada

---

### a. ¿Qué objetos habéis decidido clasificar? ¿Por qué?

Para este proyecto, hemos decidido clasificar imágenes de rostros humanos basándonos en las siguientes características:

#### 1. Clasificación Multietiqueta:

Inicialmente trabajamos con 5 etiquetas, pero el modelo tenía dificultades para ajustarse debido a la baja representación de algunas de ellas, lo que lo llevaba a centrarse demasiado en los ceros. Por eso decidimos reducirlas a las 3 etiquetas más equilibradas y significativas, que además presentaban una mejor correlación entre sí, logrando un modelo más eficiente y preciso. Las características elegidas son las siguientes:

- Longitud del cabello (corto/largo).
- Género (masculino/femenino).
- Uso de joyería.

### b. ¿Qué tipo de clasificación se ha aplicado? ¿Por qué?

- **Multietiqueta:** Para escenarios donde una imagen puede tener múltiples atributos.

**Justificación:** Pensamos que era una opción con la que podríamos llegar a tener un buen valor de `f1_score` y participar de forma competente en el concurso de CNNs. Además que nunca habíamos hecho un modelo de clasificación similar y nos pareció una oportunidad para trabajar en algo nuevo.

### c. ¿Se han realizado modificaciones al dataset? Indica cuáles y por qué.

- **División del dataset:** Se separó en `train` (70%), `val` (15%) y `test` (15%) para garantizar la evaluación justa del modelo.
- **Redimensionamiento:** Todas las imágenes fueron ajustadas a 224x224 píxeles para un procesamiento eficiente por parte de las CNNs.
- **Normalización:** Los valores de píxeles fueron escalados al rango [0, 1] para estabilizar el entrenamiento.
- **Data Augmentation:** Se aplicaron transformaciones como rotaciones, desplazamientos, flips horizontales y zooms aleatorios para mejorar la robustez del modelo.

## 2. Decisiones en cuanto a la arquitectura de la CNN

---

### a. ¿Cuántas capas tiene vuestro modelo? ¿Por qué?

El modelo multietiqueta tiene un total de 12 capas. Tres convolucionales con sus respectivas capas de max\_pooling, una flatten y luego dos capas densas con dropouts y por último una capa densa de salida con activación sigmoid.

#### Justificación:

- Es lo que mejor resultado nos ha dado probando distintas combinaciones. Las funciones de activación son las mejores vistas en clase para este tipo de clasificación. Concretamente la activación sigmoid no da una probabilidad de que la imagen pertenezca a la etiqueta en cuestión.

### b. ¿Qué modificaciones habéis introducido? ¿Por qué?

- **Batch Normalization:** Introducida después de cada capa convolucional para acelerar la convergencia.
- **Dropout:** Aplicado en las capas densas para prevenir el sobreajuste.
- **Hiperparámetros optimizados:** Ajustados usando `Keras Tuner` para encontrar la mejor configuración.

## c. Justificación de las métricas, función de pérdida y optimizador

- **Métricas:**
- Multietiqueta: `binary_accuracy` y una métrica personalizada `F1Score`.
- **Función de pérdida:**
- Multietiqueta: `binary_crossentropy` para manejar etiquetas independientes.
- **Optimizador:** `Adam` por su capacidad de ajuste dinámico del learning rate.

## 3. Metodología y resultados

---

### a. Hiperparámetros y selección de éstos

#### Número de filtros:

- **filters\_1 = 96:** El primer conjunto de filtros tiene 96 unidades, lo que significa que la primera capa convolucional busca 96 características diferentes.
- **filters\_2 = 32:** La segunda capa tiene 32 filtros, lo que indica una reducción en la cantidad de características detectadas, posiblemente para capturar patrones más específicos después de la primera capa.
- **filters\_3 = 32:** La tercera capa también tiene 32 filtros, manteniendo la reducción de la complejidad.

#### Tamaño del kernel:

- **kernel\_size\_1 = 3:** El filtro en la primera capa es de tamaño 3x3, un valor común que permite capturar características pequeñas en la imagen.
- **kernel\_size\_2 = 3 y kernel\_size\_3 = 3:** Los filtros en las siguientes capas también son de tamaño 3x3, manteniendo la consistencia y enfocándose en patrones de tamaño similar.

#### Número de unidades en capas densas:

- **units\_1 = 192:** La primera capa densa tiene 192 unidades, lo que sugiere que el modelo tiene una capacidad intermedia para capturar relaciones no lineales.

- **units\_2 = 192:** La segunda capa densa también tiene 192 unidades, indicando un diseño de red con una capacidad similar en ambas capas.

#### **Tasa de “dropout”:**

- **dropout\_1 = 0.2:** En la primera capa densa, el 20% de las neuronas se “apagan” aleatoriamente durante el entrenamiento para prevenir el sobreajuste.
- **dropout\_2 = 0.4:** En la segunda capa densa, el 40% de las neuronas se apagan, lo que puede ayudar a controlar aún más el sobreajuste en una capa con más unidades.

#### **Tasa de aprendizaje:**

- **learning\_rate = 0.004:** Este valor es una tasa de aprendizaje moderada, lo que significa que los pasos del optimizador para actualizar los pesos no son ni demasiado grandes ni demasiado pequeños, permitiendo un buen equilibrio entre convergencia rápida y estabilidad del entrenamiento.

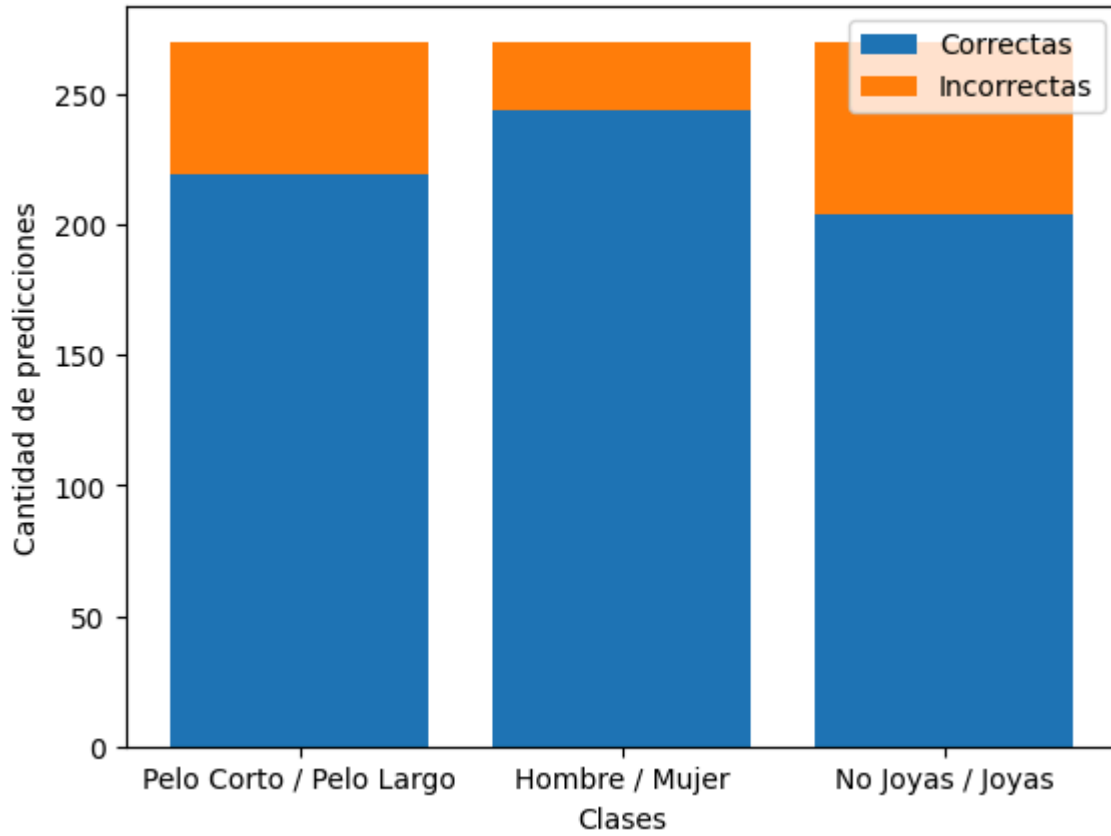
### **b. Resultado de las métricas seleccionadas**

- **Modelo Multietiqueta:**
- F1-Score ponderado: 0.8365979495231353

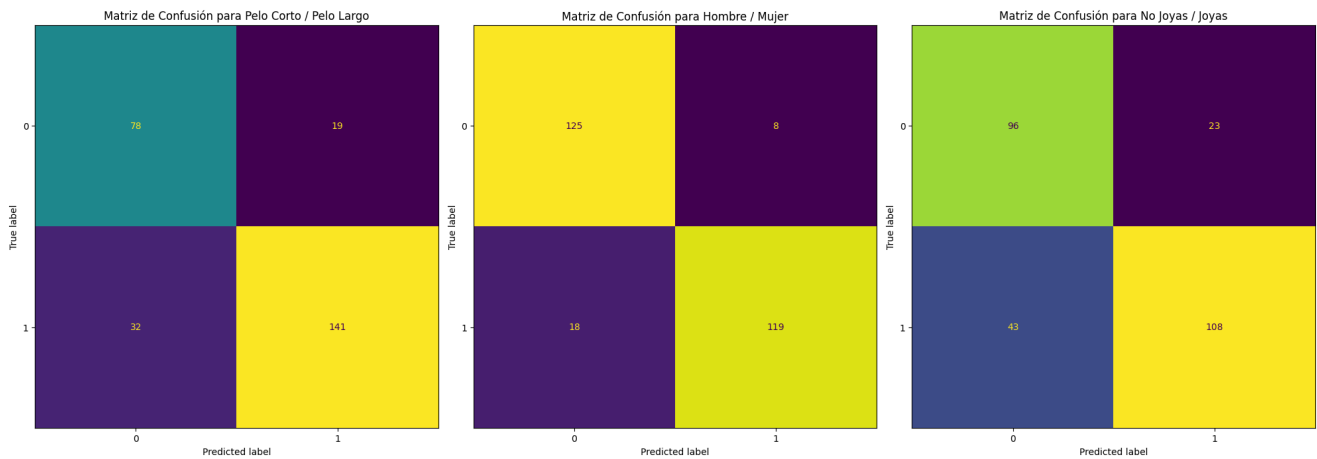
### **c. Gráficas de resultados**

#### **1. Matrices de confusión para etiquetas**

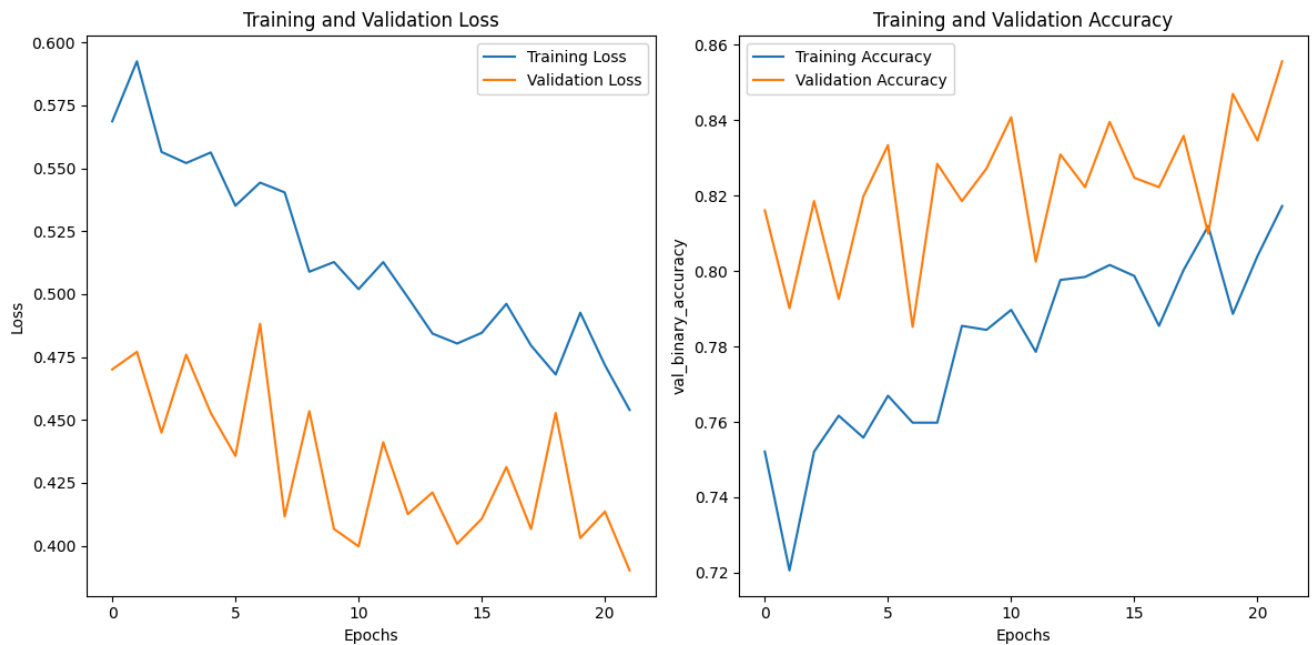
## Comparación de predicciones con etiquetas originales



## 2. Comparación de predicciones con las originales



## 3. Curvas de pérdidas



## 4. Visualización de predicciones

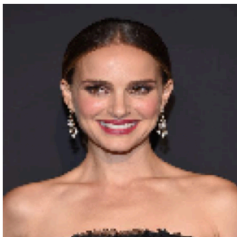
True: 0, Pred: [0.46853676 0.5086469 0.48330128]



True: 0, Pred: [0.4526052 0.50807357 0.4880671 ]



True: 0, Pred: [0.47059605 0.5111947 0.48574808]



True: 0, Pred: [0.4666856 0.49571395 0.4922864 ]



True: 0, Pred: [0.46752957 0.506444 0.4691966 ]



True: 0, Pred: [0.46945548 0.51470244 0.48870024]



True: 0, Pred: [0.4893516 0.5027776 0.5030225]



True: 0, Pred: [0.4817234 0.51755816 0.48221117]



True: 2, Pred: [0.47915584 0.5013023 0.4877515 ]



## Conclusión

La implementación de CNNs para clasificación multiclase y multietiqueta ha demostrado ser eficaz. Ambos enfoques presentan altos F1-Scores en los conjuntos de prueba. Las técnicas utilizadas, como data augmentation y optimización de hiperparámetros, fueron clave para el

éxito del modelo. La memoria documenta en detalle cada paso del proceso, asegurando reproducibilidad y claridad.