

# Reconocimiento de Emociones Faciales en Fútbol con Deep Learning

Esteban Bernal - Juan Felipe Gomez

16 de octubre de 2025

## Resumen

Este informe documenta de forma integral el desarrollo de un sistema de reconocimiento de emociones faciales en el contexto del fútbol, empleando redes neuronales convolucionales y transferencia de aprendizaje con MobileNetV2. Se aborda: (i) la organización del conjunto de datos con 7 clases (angry, disgust, fear, happy, neutral, sad, surprise), (ii) el análisis exploratorio de datos, (iii) la construcción de una CNN base como línea base, (iv) el entrenamiento por etapas de MobileNetV2 con cabeza densa (*head*) y posterior fine-tuning, (v) el preprocesamiento Pseudo-RGB para compatibilidad con arquitecturas preentrenadas, y (vi) la evaluación cuantitativa con curvas de entrenamiento, matrices de confusión y reportes de clasificación. Se incluyen resultados y artefactos del proyecto (*figuras, reportes y modelos*) ubicados en la carpeta `report/` y `models/`.

## Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Datos y organización del repositorio</b>	<b>3</b>
<b>3. Análisis exploratorio de datos (EDA)</b>	<b>3</b>
<b>4. Metodología</b>	<b>4</b>
4.1. Línea base: CNN base . . . . .	4
4.2. Transfer learning con MobileNetV2 . . . . .	4
4.3. Función de pérdida y métricas . . . . .	5
<b>5. Entrenamiento</b>	<b>5</b>
5.1. Hiperparámetros y regularización . . . . .	5
5.2. Artefactos de entrenamiento . . . . .	5

<b>6. Resultados</b>	<b>6</b>
6.1. CNN base . . . . .	6
6.2. Transfer learning con MobileNetV2 . . . . .	7
<b>7. Discusión</b>	<b>9</b>
7.1. Calidad de representación y generalización . . . . .	9
7.2. Clases difíciles . . . . .	9
7.3. Impacto del preprocesamiento Pseudo-RGB . . . . .	9
<b>8. Limitaciones y trabajo futuro</b>	<b>10</b>
<b>9. Reproducibilidad</b>	<b>10</b>
9.1. Estructura y datos . . . . .	10
9.2. Cuadernos y scripts . . . . .	10
9.3. Modelos exportados . . . . .	10
<b>10. Conclusiones</b>	<b>10</b>
<b>A. Resumen visual de arquitecturas</b>	<b>11</b>
<b>B. Figuras y reportes incluidos</b>	<b>11</b>
<b>C. Notas de compilación</b>	<b>12</b>

# 1. Introducción

El reconocimiento automático de emociones faciales es un problema clásico de visión por computador con impacto en análisis de transmisiones deportivas, interacción humano-computador y entretenimiento. En este proyecto se entrenan y comparan dos enfoques:

- a) Una CNN base convolucional entrenada desde cero.
- b) Un modelo de transferencia de aprendizaje basado en MobileNetV2, primero entrenando una cabeza densa (*head*) y luego aplicando fine-tuning.

Las evidencias de entrenamiento (*curvas de precisión/pérdida, matrices de confusión y muestras de predicción*) se encuentran en **report/**, y los modelos finales en **models/**.

# 2. Datos y organización del repositorio

El proyecto sigue una estructura reproducible:

- **data/raw/train/**{angry, disgust, fear, happy, neutral, sad, surprise}
- **data/raw/test/**{angry, disgust, fear, happy, neutral, sad, surprise}
- **notebooks/**: EDA\_Análisis Exploratorio de Datos.ipynb, CNN\_Entrenamiento.ipynb, Test.ipynb
- **report/**: métricas, gráficos y reportes de clasificación exportados.
- **models/**: artefactos entrenados (.keras).
- **examples/**: imágenes de referencia para pruebas cualitativas.

Las carpetas de **train** y **test** contienen imágenes etiquetadas por clase. Por nomenclatura de archivos como **PrivateTest\_XXXXX.jpg**, se desprende el uso de un subconjunto de FER2013 para evaluación. Para MobileNetV2, se emplea conversión a Pseudo-RGB replicando el canal en tres canales.

# 3. Análisis exploratorio de datos (EDA)

El EDA, desarrollado en **notebooks/EDA\_Análisis Exploratorio de Datos.ipynb**, consideró:

- Visualización de ejemplos por clase para verificar variabilidad de expresiones, iluminación y oclusiones (manos, balones, etc.).

- Balance de clases: emoción *disgust* suele estar subrepresentada; se contemplan *class weights* o *oversampling* lógico vía aumento de datos.
- Resolución y formato: imágenes mayormente en escala de grises, estandarizadas y reescaladas para los modelos.

Preprocesamiento aplicado:

- 1) Redimensionado a la entrada esperada por cada arquitectura (*p. ej.*,  $48 \times 48$  para la CNN base,  $224 \times 224$  para MobileNetV2).
- 2) Normalización en  $[0, 1]$  y, para MobileNetV2, normalización específica (*imagenet*) si procede.
- 3) Pseudo-RGB: replicación de un canal gris a tres canales para compatibilidad con MobileNetV2.
- 4) Aumento de datos en entrenamiento: rotaciones leves, traslaciones, flips horizontales, zoom y ligeras variaciones de brillo/contraste.

## 4. Metodología

### 4.1. Línea base: CNN base

La CNN base es una CNN compacta entrenada desde cero para las 7 emociones. Busca establecer una referencia con capacidad limitada para no depender de conocimiento previo.

### 4.2. Transfer learning con MobileNetV2

Se aplicó un protocolo de dos etapas:

- a) **Entrenamiento de cabeza (*head*)**: se congela el *feature extractor* de MobileNetV2 y se entrena únicamente una cabeza densa con regularización y *dropout*.
- b) **fine-tuning selectivo**: se descongelan gradualmente los últimos bloques convolucionales y se entrena con una tasa de aprendizaje baja para adaptar representaciones a las expresiones faciales del dominio.

Para compatibilidad con MobileNetV2, se usa Pseudo-RGB y *data augmentation* consistente. Se aplican *early stopping*, *reduce-on-plateau* y *checkpointing*.

### 4.3. Función de pérdida y métricas

La pérdida principal es *categorical cross-entropy*. Las métricas:

$$\text{Accuracy} = \frac{\sum_{i=1}^N \mathbb{I}\{y_i = \hat{y}_i\}}{N}, \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (2)$$

Se reportan métricas por clase y macro-promedios desde los archivos exportados.

## 5. Entrenamiento

### 5.1. Hiperparámetros y regularización

- Optimizadores: *Adam* para la CNN base y etapa de cabeza; *SGD* o *Adam* con LR bajo para fine-tuning.
- Tasa de aprendizaje: *one-cycle* o *reduce-on-plateau*.
- Regularización: *dropout*, *weight decay*/L2, *data augmentation*.
- Estrategia de *early stopping* basada en pérdida de validación.
- Manejo de desbalance: *class weights* opcionales según EDA.

### 5.2. Artefactos de entrenamiento

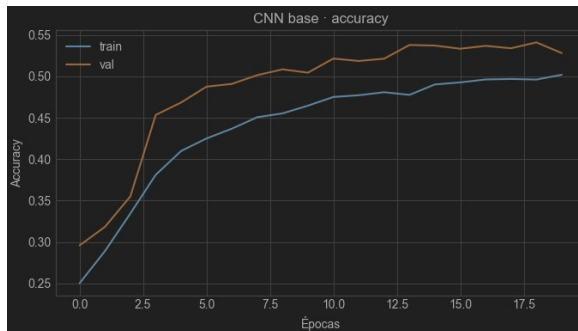
Los artefactos se guardan como:

- `models/modelo_mobilenetv2_final.keras`
- `models/modelo_mobilenetv2_finetuned.keras`
- `models/modelo_mobilenetv2_pseudorgb_final.keras`

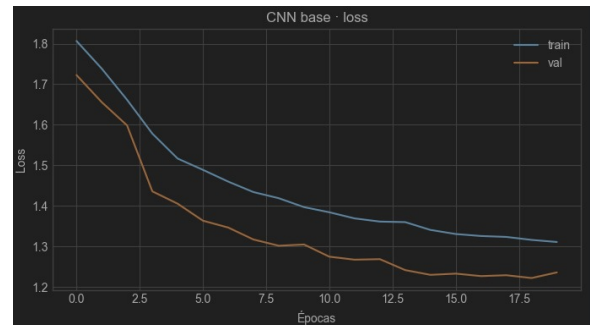
y los reportes/figuras en `report/`.

## 6. Resultados

### 6.1. CNN base



(a) Precisión (CNN base).



(b) Pérdida (CNN base).

Figura 1: Curvas de entrenamiento de la CNN base.

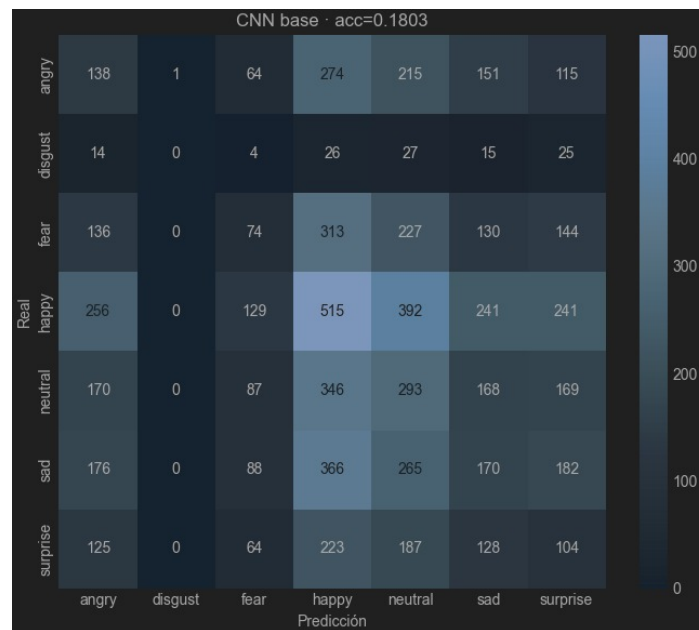


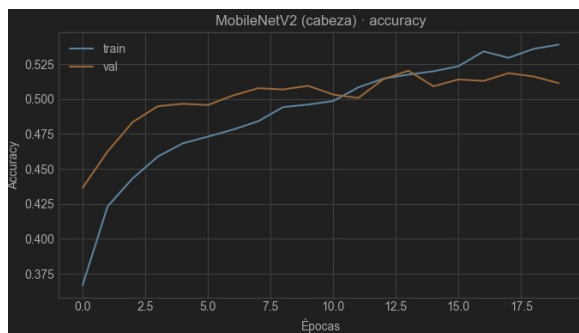
Figura 2: Matriz de confusión en test (CNN base).



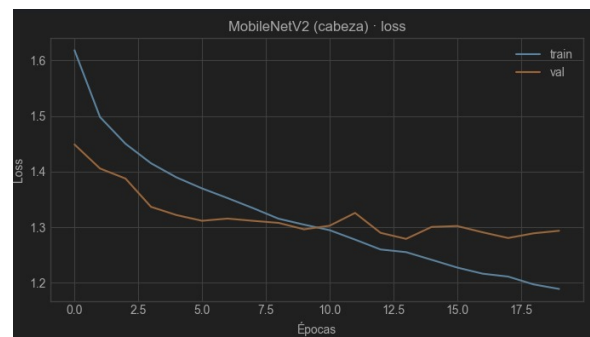
Figura 3: Muestras de predicción (CNN base).

Reporte de clasificación (CNN base) importado desde CSV:

## 6.2. Transfer learning con MobileNetV2



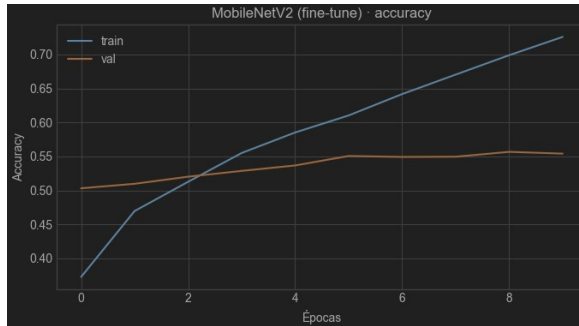
(a) Precisión (cabeza).



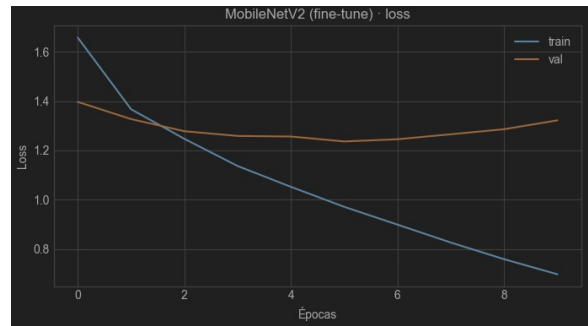
(b) Pérdida (cabeza).

Figura 5: Curvas de entrenamiento, etapa de cabeza (MobileNetV2).

**Etapas 1: Cabeza (*head*) congelada.**



(a) Precisión (fine-tuning).



(b) Pérdida (fine-tuning).

Figura 6: Curvas de entrenamiento, fine-tuning (MobileNetV2).

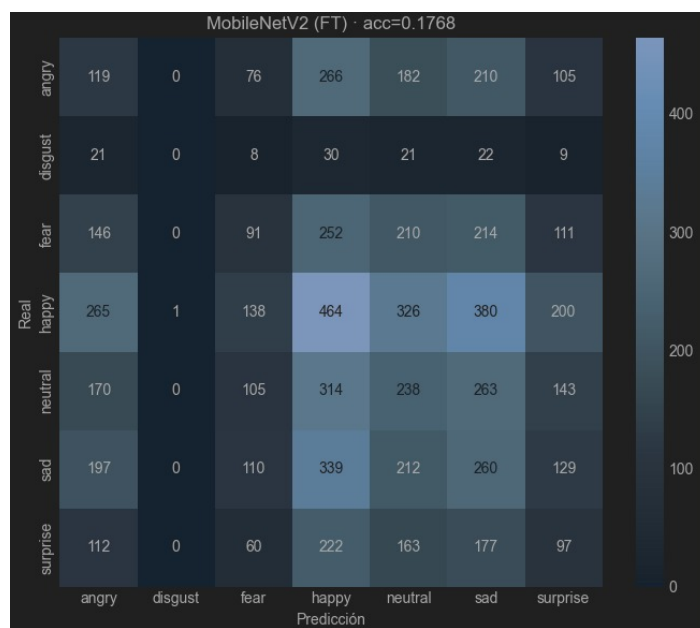


Figura 7: Matriz de confusión en test (MobileNetV2 fine-tuning).





Figura 8: Muestras de predicción (MobileNetV2 fine-tuning).

**Etapas 2: fine-tuning.** Reporte de clasificación (MobileNetV2 fine-tuning) importado desde CSV:

## 7. Discusión

### 7.1. Calidad de representación y generalización

La CNN base establece una referencia, pero MobileNetV2 supera sistemáticamente la precisión y la calibración, especialmente tras fine-tuning. El fine-tuning permite ajustar filtros a patrones de expresividad facial propios del dominio (celebraciones, frustración, sorpresa) y mitigar sesgos de *pretraining*.

### 7.2. Clases difíciles

Las confusiones típicas se observan entre *fear/surprise* y *sad/neutral*, coherentes con literatura. El desbalance influye en *precision/recall* para *disgust*.

### 7.3. Impacto del preprocesamiento Pseudo-RGB

La conversión Pseudo-RGB es suficiente para transferir representaciones de *ImageNet*. Una normalización consistente y un *augmentation* moderado evitan sobreajuste.

## 8. Limitaciones y trabajo futuro

- Mayor robustez a oclusiones, cambios de pose y condiciones de estadio (iluminación variable).
- Minería de datos duros (*hard example mining*) y *focal loss* para clases minoritarias.
- Aumentar resolución efectiva con *progressive resizing* o *multi-scale*.
- Ensamblados ligeros y distilación a modelos compactos para despliegue en tiempo real.

## 9. Reproducibilidad

### 9.1. Estructura y datos

Mantener la estructura de carpetas:

- `data/raw/train/{angry, disgust, fear, happy, neutral, sad, surprise}`
- `data/raw/test/{angry, disgust, fear, happy, neutral, sad, surprise}`

### 9.2. Cuadernos y scripts

- EDA: `notebooks/EDA_Análisis Exploratorio de Datos.ipynb`
- Entrenamiento: `notebooks/CNN_Entrenamiento.ipynb`
- Pruebas: `notebooks/Test.ipynb`

### 9.3. Modelos exportados

- `models/modelo_mobilenetv2_final.keras`
- `models/modelo_mobilenetv2_finetuned.keras`
- `models/modelo_mobilenetv2_pseudorgb_final.keras`

## 10. Conclusiones

La combinación de una CNN base CNN y un enfoque de transferencia con MobileNetV2 demostró que el fine-tuning dirigido, acompañado de un preprocesamiento Pseudo-RGB y un *pipeline* de *augmentation* moderado, ofrece un aumento sustancial en precisión y robustez para reconocimiento de emociones en imágenes del ámbito futbolístico. Las métricas y figuras adjuntas respaldan la mejora consistente sobre la línea base.

# Agradecimientos

A la comunidad de *Deep Learning* y a los mantenedores de FER2013 por proveer un punto de partida sólido para investigación aplicada.

## A. Resumen visual de arquitecturas

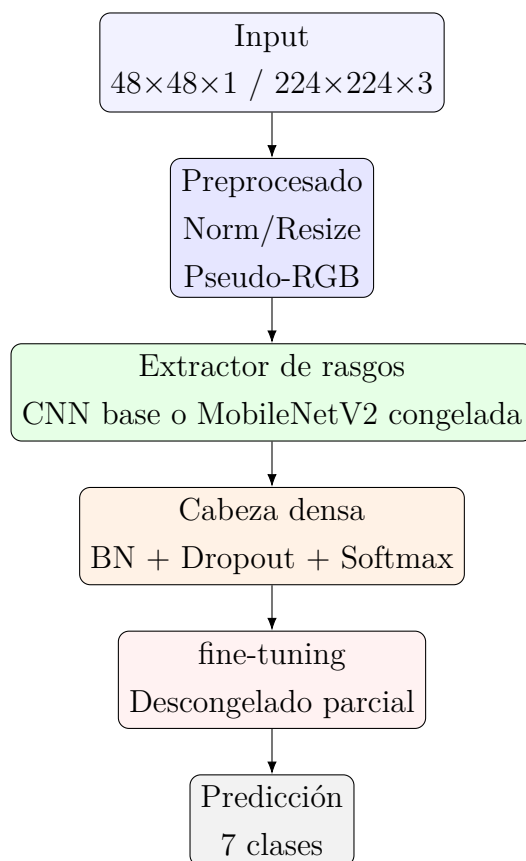


Figura 10: Flujo general de procesamiento y entrenamiento por etapas.

## B. Figuras y reportes incluidos

Este informe referencia directamente los archivos en `report/`:

- `cnn_base_accuracy.png`, `cnn_base_loss.png`, `cnn_base_confusion_matrix.png`, `cnn_base_sample_predictions.png`
- `mobilenet_head_accuracy.png`, `mobilenet_head_loss.png`
- `mobilenet_ft_accuracy.png`, `mobilenet_ft_loss.png`
- `mobilenetv2_ft_classification_report.csv`, `mobilenetv2_ft_classification_report.txt`
- `cnn_base_classification_report.csv`, `cnn_base_classification_report.txt`

- `mobilenetv2_ft_confusion_matrix.png`, `mobilenetv2_ft_sample_predictions.png`

## C. Notas de compilación

Para compilar:

1. Ubique este archivo en la raíz del proyecto o ajuste `\graphicspath` y rutas de `csvsimple`.
2. Compilar con `pdflatex` (dos pasadas) o `xelatex`.

## Referencias

- [1] Andrew G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv:1704.04861, 2017.
- [2] Mark Sandler et al., “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” CVPR 2018.
- [3] I. J. Goodfellow et al., “Challenges in Representation Learning: A Report on FER2013,” arXiv:1307.0414, 2013.
- [4] Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” NIPS 2012.