

---

## Grupo 5

Integrantes del equipo: Eduardo Mussini, Mathías Pérez, Diego García

# Clasificación de imágenes (detección de empleo de mascarilla facial)

15 de agosto 2025

## 1. Selección y Justificación del tipo de arquitectura a usar.

La arquitectura seleccionada para la segunda fase del proyecto, fue la aplicación de redes neuronales convolucionales (CNN). El motivo del mismo es que este tipo de arquitectura está especialmente diseñada para trabajar con imágenes, ya que la información que estas contienen están en forma de cuadrículas de píxeles, sobre los cuales se les aplican diferentes operaciones de convolución (filtros o kernels), logrando a través de estas extraer las características principales de los datos (formas, texturas, bordes, etc).

## 2. Preparación del Dataset.

A diferencia de la práctica 1, se ajusta el procesamiento que se había realizado previamente para que las imágenes sean con un margen mayor a las coordenadas clasificadas, ya que las clasificaciones en algunos casos recortaron información útil. Por otra parte, en función que las CNN captan información de los 3 canales de las imágenes, se omite en este caso el procedimiento de aplanar las mismas.

Dicho esto, se procedió a cargar y redimensionar las imágenes en formato 128x128 para no perder detalles importantes y relevantes para la tarea asignada, obteniendo un total de 4072 imágenes. Luego, se dividió el conjunto de datos en train y test, observándose como fue detectado previamente un claro desbalance hacía la clase 1. En este sentido, la información inicial del conjunto de train y test fue la siguiente:

TAMAÑO DE LOS DATOS DE ENTRENAMIENTO : 3257 IMÁGENES	
Distribución de clases en datos de entrenamiento	
Clase 0: Usa mascarilla incorrectamente	98 imágenes
Clase 1: Usa mascarilla	2585 imágenes

Clase 2: No usa mascarilla	574 imágenes
----------------------------	--------------

TAMAÑO DE LOS DATOS DE TEST: 815 IMÁGENES

### 3. Implementación y Entrenamiento del Modelo.

Modelo CNN:

El modelo implementado consiste en una capa de entrada cuyas imágenes se establecieron en una dimensión de 128 x 128 para poder captar mayor detalle.

Posteriormente se añadieron 2 capas convolucionales: la primera consistente en 32 filtros de dimensiones 3 x 3 píxeles y una función de activación (en este caso Relu) para introducir no linealidad y un paso (stride) de 1. A continuación se procedió a agregar una capa pooling de 2 x 2 tomando el valor máximo. La segunda capa convolucional agrega 64 filtros, de modo de poder detectar más características de las 32 previas en la primera convolución, manteniendo los mismos ajustes de dimensión del kernel, paso, función de activación y padding. Al igual que en el caso anterior se agregó también una capa pooling de 2x2 pero aplicado a los mapas de características más profundas.

Más tarde se procedió al aplanamiento de los datos previamente obtenidos, convirtiéndolos los mapas 2D a un vector de una sola dimensión. Aquí se procede a colocar una capa de dropout de 0,3 durante el entrenamiento (deja afuera aleatoriamente el 30% de las neuronas) para evitar el sobreajuste.

Por último se agrega una capa de salida que determina las predicciones, se establece el número de clases ( que en nuestro caso son 3) y seleccionando la función de activación softmax, que permite convertir los logits finales a probabilidades.

Al momento de compilar el modelo, se optó por mantener la función de pérdida (categorical cross entropy) pero agregando el parámetro label\_smoothing al 0.05, lo cual reduce el sobreajuste y mejora la generalización del modelo. Finalmente se mantiene el mismo optimizador (SGD) y la misma métrica de evaluación (accuracy) que había sido empleada previamente en el MLP.

Entrenamiento del modelo CNN:

Una vez definida la arquitectura del modelo, se procedió a su entrenamiento durante 30 épocas, utilizando un conjunto de validación separado para evaluar el rendimiento. En los resultados obtenidos, las métricas de pérdida y precisión en entrenamiento y validación se mantuvieron muy cercanas a lo largo de las épocas, indicando un bajo nivel de overfitting. Al finalizar, se alcanzó aproximadamente un 95 % de accuracy tanto en entrenamiento como en validación, con una pérdida del 30 % en entrenamiento y del 32 % en validación.

---

## 4. Evaluación del Modelo.

El rendimiento del modelo en test fue de 94% de accuracy y una pérdida del entorno del 33%. Pese a obtener un desempeño similar al MLP, el modelo CNN logró obtener un resultado levemente mejor al visualizar el puntaje F1 para cada una de las clases. De este modo logró clasificar mejor la clase 0 (clase minoritaria) logrando alcanzar un f1 de 0.32, frente al 0.24 del modelo MLP.

## 5. Comparación con MLP.

Comparando el desempeño de ambas arquitecturas en términos de F1-score por clase:

Clase 0 (minoritaria):

CNN: 0.32

MLP: 0.24

El CNN logra una mejora notable en esta clase, aumentando 0.19 puntos en F1-score respecto al MLP, lo que indica una mejor capacidad para identificar correctamente ejemplos minoritarios.

Clase 1 (mayoritaria):

CNN: 0.97

MLP: 0.96

El rendimiento es prácticamente equivalente, con una ligera ventaja para el CNN (+0.01).

Clase 2:

CNN: 0.88

MLP: 0.87

Al igual que la Clase 1, se observa una ligera ventaja para el CNN (+0.01)..

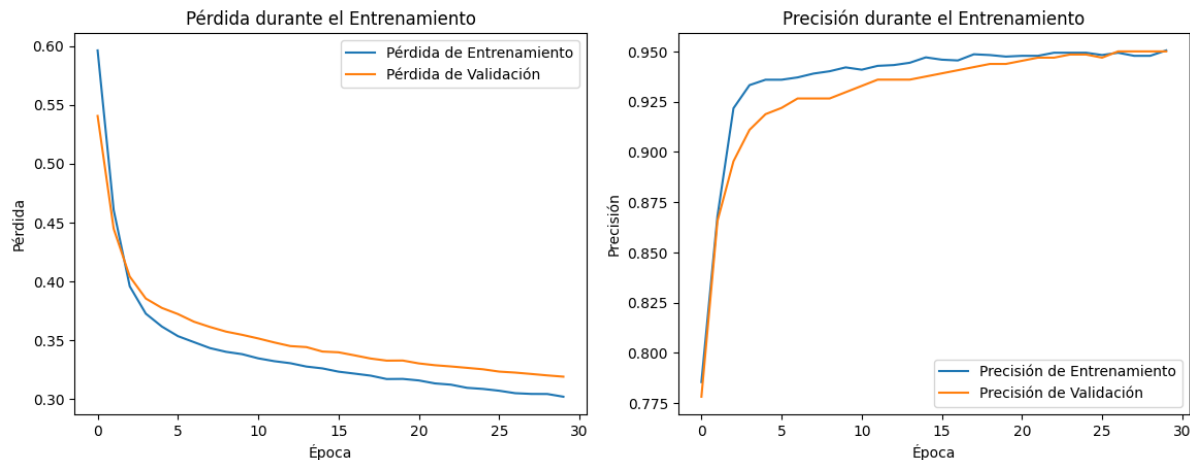
Si bien ambas redes muestran un rendimiento global similar (accuracy  $\approx$  0.93-0.94), el CNN destaca por su mejor desempeño en la clase minoritaria, lo que sugiere una mayor robustez frente al desbalance.

## 6. Visualizaciones.

Las curvas de pérdida muestran una disminución sostenida y paralela en entrenamiento y validación, estabilizándose en torno al 30 % y 32 % respectivamente, sin indicios relevantes de sobreajuste.

En cuanto a la precisión, ambas aumentan rápidamente y se mantienen cercanas, alcanzando un 95 % al final del entrenamiento.

En conjunto, las gráficas evidencian una buena convergencia y una adecuada capacidad de generalización del modelo.



El modelo CNN evaluado sobre el conjunto de test mostró un rendimiento general sólido, alcanzando un accuracy del 94%. El f1-score promedio ponderado fue de 0.93, lo que indica una consistencia global en el desempeño entre clases.

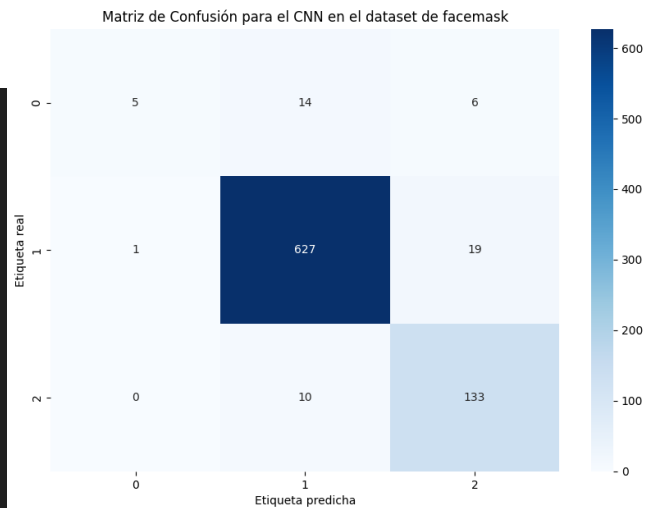
En términos de desempeño por clase:

- La clase 1 (la mayoritaria) fue correctamente clasificada con alta precisión y recall (0.96 y 0.97, respectivamente), lo cual era esperable por su fuerte representación en los datos.
- La clase 2 alcanzó un f1-score de 0.88 y un recall de 0.93, evidenciando una capacidad razonable del modelo para capturar patrones incluso con menos muestras.
- La clase 0, que es la más minoritaria, continúa siendo la más desafiante, con un f1-score de 0.32 y un recall de apenas 0.20, lo que implica que el modelo aún tiene dificultades para identificar correctamente instancias de esta clase.

En conjunto, si bien persisten limitaciones en la detección de la clase minoritaria, los resultados reflejan una mejora respecto al modelo MLP, con mayor robustez en generalización y un mejor equilibrio entre precisión y cobertura en las clases menos representadas.

Reporte de Clasificación para el CNN de facemask:

	precision	recall	f1-score	support
0	0.83	0.20	0.32	25
1	0.96	0.97	0.97	647
2	0.84	0.93	0.88	143
accuracy			0.94	815
macro avg	0.88	0.70	0.72	815
weighted avg	0.94	0.94	0.93	815



## 7. Conclusiones.

En conclusión, ambos modelos, MLP y CNN, alcanzan un rendimiento global muy similar, pero difieren en cómo distribuyen su capacidad predictiva entre clases.

El CNN logra una mejora en la clase minoritaria (0), elevando su F1-score de 0.24 a 0.32, lo que indica una mayor sensibilidad y equilibrio frente al desbalance del dataset. En cuanto a las clases 1 y 2 muestra un rendimiento superior pero muy similar al MLP.

El MLP, por su parte, mantiene un alto rendimiento en las clases mayoritarias, pero a costa de un desempeño más débil en la clase minoritaria, lo que podría limitar su utilidad en escenarios donde esa clase sea crítica.

En síntesis, CNN alcanzó un mejor desempeño en todas las clases, principalmente en la minoritaria, el CNN ofrece una mejor capacidad de generalización, obteniendo resultados muy similares al MLP en las clases mayoritarias.