Grupo 5
Integrantes del equipo: Eduardo Mussini, Mathías Pérez, Diego García

Clasificación de imágenes (detección de empleo de mascarilla facial)

6 de agosto 2025

1. Resumen del problema:

El problema consiste en lograr clasificar dentro de un conjunto de imágenes, tres formas diferentes posibles del uso de la mascarilla facial (clases objetivo: usa mascarilla, no usa mascarilla, usa mascarilla incorrectamente).

2. Transformación del problema

En este caso, para transformar el problema en un formato adecuado para entrenar un MLP, se debió realizar una exploración previa de los datos, dimensiones, formato, tamaño, número de muestras obtenidas, etc. En virtud que estamos trabajando con imágenes, nuestras variables de entrada (features) serán los píxeles de cada una de las imágenes, mientras que la variable de salida (target) será cada una de las 3 posibles clases objetivo.

3. Descripción del dataset:

Para abordar este problema, se utiliza un **dataset** denominado **Face Mask Detection**, que incluye:

- Una carpeta con 853 imágenes a color en formato .png, de distintos tamaños. Las imágenes muestran personas en diversas situaciones, algunas usando el tapabocas correctamente, otras usándolo de forma incorrecta, y otras sin tapabocas.
- Una carpeta con 853 archivos en formato .xml, que contienen las etiquetas de los objetos detectados en cada imagen. Cada archivo especifica las coordenadas y la clase correspondiente de los objetos, categorizados en una de las tres clases posibles.

Los ***objetivos*** de la práctica son:

1- Reafirmar los conceptos teóricos de la asignatura deep learning, mediante la aplicación práctica de una red neuronal multicapa (MLP) al problema planteado.

4. Preparación y división de los datos.

Preparación de los datos:

Imágenes: Dado que el objetivo inicial es resolver el problema con una arquitectura MLP, fue necesario adaptar el dataset, originalmente diseñado para detección de objetos. Utilizando las etiquetas disponibles, se recortaron las imágenes para obtener un nuevo conjunto con una única categoría por imagen. Adicionalmente, fue necesario cargar las imágenes, estandarizar el tamaño de todas ellas y normalizarlas de modo que el modelo pueda procesarlas con mayor facilidad. Esto se realizó mediante la creación de una función que consolida estos pasos (ver notebook).

Etiquetas: Los archivos .xml fueron procesados para extraer las etiquetas asociadas a cada recorte. Las imágenes resultantes se almacenaron en carpetas según su clase, lo que permitió recuperar la etiqueta de cada imagen directamente desde el nombre de la carpeta. Luego, se procedió a codificar cada una de las clases aplicando onehot, quedando las clases identificadas de la siguiente manera:

Código	Clase
0	Usa mascarilla incorrectamente
1	Usa mascarilla
2	No usa mascarilla

Una vez realizado esto se procedió a visualizar mediante el empleo de una gráfica de barras la distribución de las clases objetivo. En este punto se identificó un claro desbalance hacia la clase 1 (usa mascarilla): 3232 observaciones (79%). Este desbalance puede generar sesgo en los resultados del modelo, por ello se optó por utilizar undersampling (detallado en la sección 6).

División del dataset en entrenamiento y test.

Una vez procesados los datos se procedió a dividir el conjunto de datos en train (80%) y test (20%), de esta forma contamos con suficientes datos para entrenar el modelo y reservar dentro del mismo un subconjunto de validación. Mientras que el 20% de datos (814 imágenes) se estima una cantidad adecuada para evaluar el modelo.

5. Arquitectura del modelo MLP

El modelo utilizado fue el de Perceptrón Multicapa (MLP), mediante el uso de Keras. La arquitectura del modelo seleccionado fue la siguiente:

- Capa entrada: Pixeles de cada una de las imágenes aplanadas del dataset (128x128x3 = 49152)
- Capa oculta 1: con 512 neuronas y función de activación ‘relu’
- Capa oculta 2: con 256 neuronas y función de activación ‘relu’
- Capa Salida: 3 neuronas (1 por cada clase) y función de activación ‘softmax’

La función de pérdida empleada fue categorical cross entrophy, mientras que el optimizador seleccionado fue el descenso de gradiente estocástico (SGD) ejecutándose en 30 épocas de entrenamiento para 32 batches de datos. Además se reservaron 20% de los datos de entrenamiento para validación.

6. Técnica adicional aplicada

Dado el fuerte desbalance en la distribución de clases (especialmente la predominancia de la clase “mascarilla bien puesta”), se aplicó una técnica de undersampling sobre el conjunto de entrenamiento. En concreto, se redujo la cantidad de muestras de la clase mayoritaria (clase 1) a un 40 % de su tamaño original, manteniendo las clases minoritarias sin modificación.

Este recorte tuvo como objetivo presentar al modelo una distribución más equilibrada durante el entrenamiento, favoreciendo que aprenda a reconocer también las clases menos representadas, en particular “sin mascarilla” y “mascarilla mal puesta”.

La técnica se aplicó sólo sobre los datos de entrenamiento, conservando intacta la distribución real del conjunto de prueba para garantizar una evaluación fiel del rendimiento del modelo.

7. Evaluación del modelo

El modelo alcanzó una accuracy del 93 % sobre el conjunto de prueba. Sin embargo, al analizar las métricas por clase se observan importantes diferencias en el rendimiento:

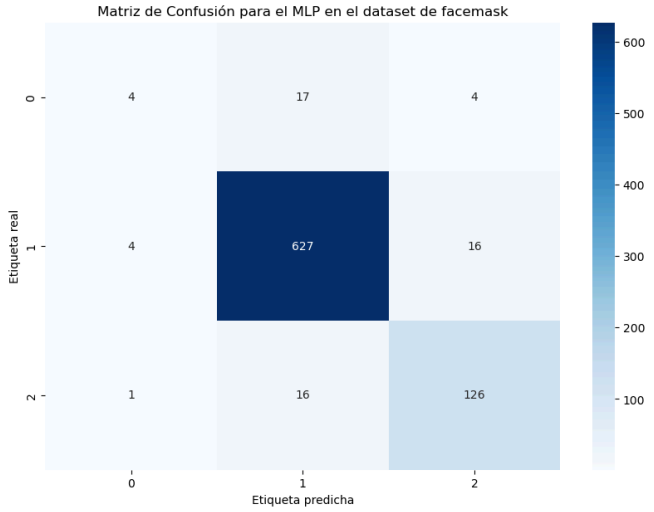
Clase 1 (mascarilla bien puesta) fue la mejor clasificada, con una precisión del 0.95, recall del 0.97 y f1-score del 0.96.

Clase 2 (mascarilla mal puesta) también obtuvo buenos resultados, destacando un recall alto (0.88) y un f1-score de 0.87.

Clase 0 (sin mascarilla) fue la más afectada, con un recall muy bajo (0.16) y un f1-score de apenas 0.24, reflejando la dificultad del modelo para identificar esta categoría.

La matriz de confusión confirma esta tendencia: de los 25 ejemplos reales de clase 0, solo 4 fueron correctamente clasificados, mientras que los errores más comunes fueron confundir clase 0 con clase 1. En contraste, las clases 1 y 2 muestran una cantidad elevada de aciertos.

A pesar del buen desempeño global, las métricas macro (f1 = 0.69, recall = 0.67) evidencian un desequilibrio importante en el rendimiento entre clases.



Reporte de Clasificación para el MLP de facemask:

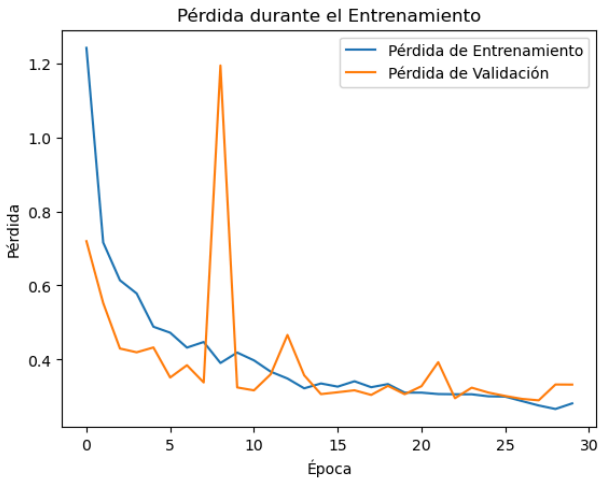
	precision	recall	f1-score	support
0	0.44	0.16	0.24	25
1	0.95	0.97	0.96	647
2	0.86	0.88	0.87	143
accuracy			0.93	815
macro avg	0.75	0.67	0.69	815
weighted avg	0.92	0.93	0.92	815

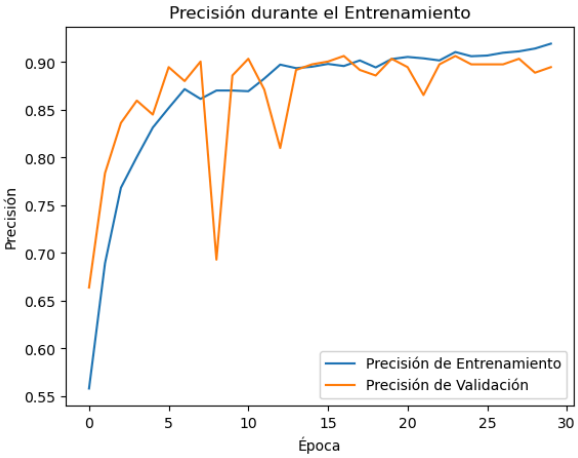
8. Visualizaciones

Las curvas de pérdida muestran una disminución constante tanto en entrenamiento como en validación. La pérdida de entrenamiento desciende de forma progresiva y estable, mientras que la de validación se mantiene ligeramente más baja, con algunas oscilaciones en las primeras épocas. Esto sugiere una buena capacidad de generalización del modelo, sin señales claras de sobreajuste.

En cuanto a la precisión, ambas curvas (entrenamiento y validación) aumentan rápidamente durante las primeras épocas, estabilizándose alrededor de 0.90. La precisión de validación se mantiene cercana a la de entrenamiento a lo largo de todo el proceso, con pequeñas variaciones esperables.

En conjunto, ambas gráficas indican un entrenamiento exitoso: el modelo converge de forma estable, con buen rendimiento y sin una brecha significativa entre entrenamiento y validación.





9. Conclusiones y próximos pasos

El modelo alcanzó un desempeño general alto (93% de accuracy), con buena capacidad para distinguir entre imágenes con mascarilla correctamente e incorrectamente colocada. Sin embargo, presentó serias dificultades para identificar la clase minoritaria (“sin mascarilla”), con un recall muy bajo (0,16) y un f1-score de solo 0,24.

Esto evidencia que, aunque el modelo generaliza bien para las clases mayoritarias, su desempeño está fuertemente condicionado por el desbalance en los datos. El undersampling aplicado durante el entrenamiento ayudó, pero no fue suficiente para lograr una detección efectiva de todas las clases.

10. Referencias

- AndrewMVD. (subido c. 2020). *Face Mask Detection* [Dataset]. Kaggle. Recuperado el 31 de julio de 2025, de https://www.kaggle.com/datasets/andrewmvd/face-mask-detection