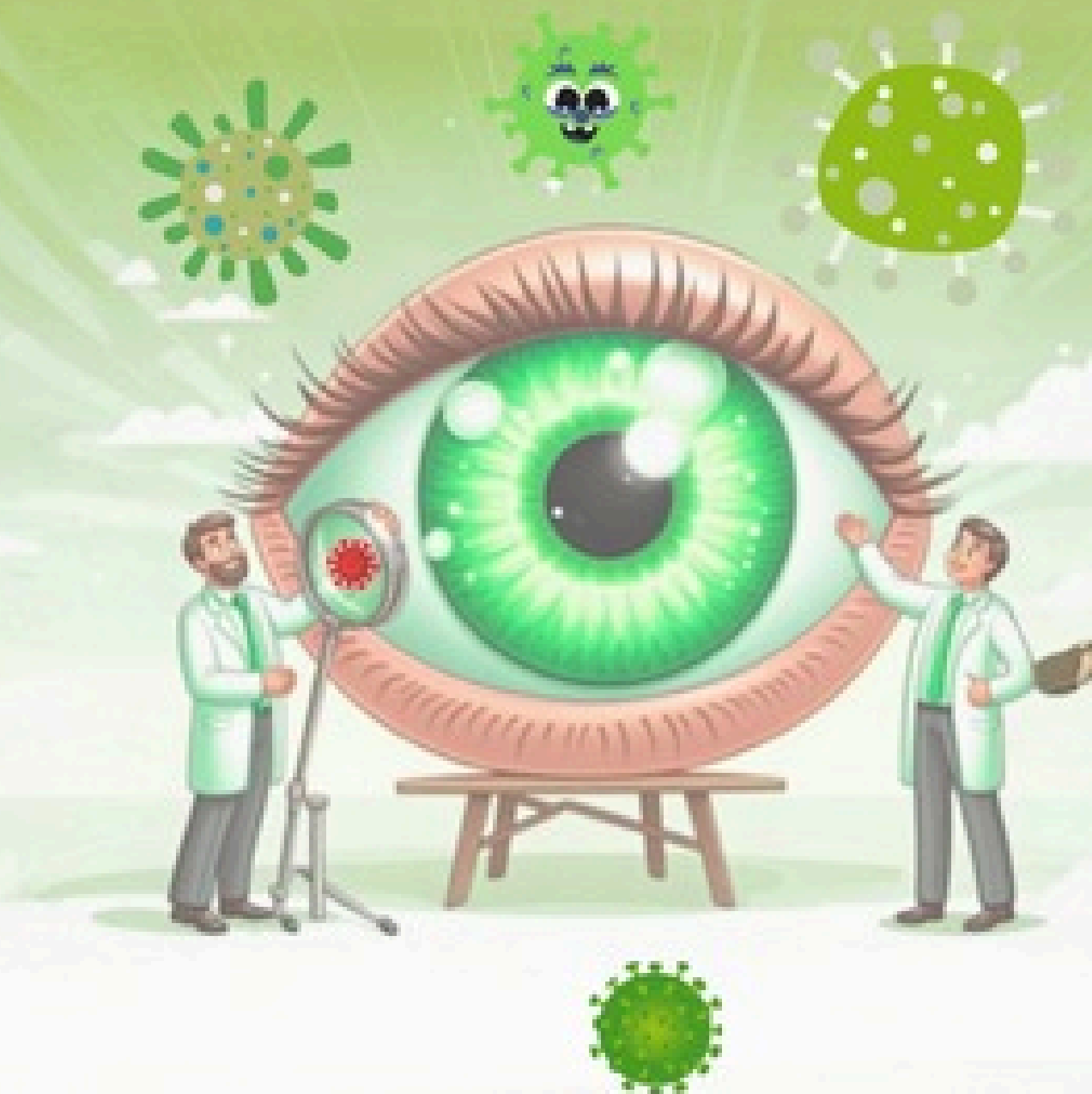


CLASIFICACIÓN ENFERMEDADES OCULARES



Universidad
Industrial de
Santander



- SANTIAGO ANDRÉS CHAÍN SANTOS
- SANTIAGO GELVEZ GONZALEZ

CONTENIDO

Contexto del Proyecto	3
Dataset Utilizado	6
Primeros Avances	9
Pipelines Finales	14
Resultados	17
Conclusiones	22

CONTEXTO DEL PROYECTO

DESCRIPCIÓN DEL PROBLEMA

La clasificación de las condiciones oculares a partir de imágenes del fondo de ojo es compleja debido a la alta variabilidad en las características visuales de las enfermedades y la dificultad para automatizar el diagnóstico mediante métodos convencionales.



OBJETIVO

Desarrollar y evaluar un modelo de red neuronal convolucional (CNN) capaz de clasificar imágenes oculares en diferentes categorías de enfermedades.



DATASET

ODIR-5K

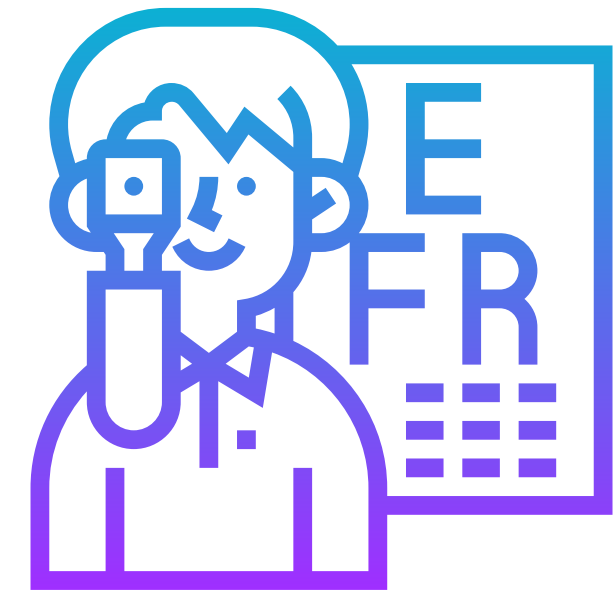
Colección de imágenes de fondo de ojo acompañadas de información clínica.



- LEFT FUNDUS: IMAGEN DEL OJO IZQUIERDO.
- RIGHT FUNDUS: IMAGEN DEL OJO DERECHO.
- DIAGNÓSTICOS: LEFT DIAGNOSIS, RIGHT DIAGNOSIS.
- ETIQUETAS: CLASIFICACIÓN DE ENFERMEDAD

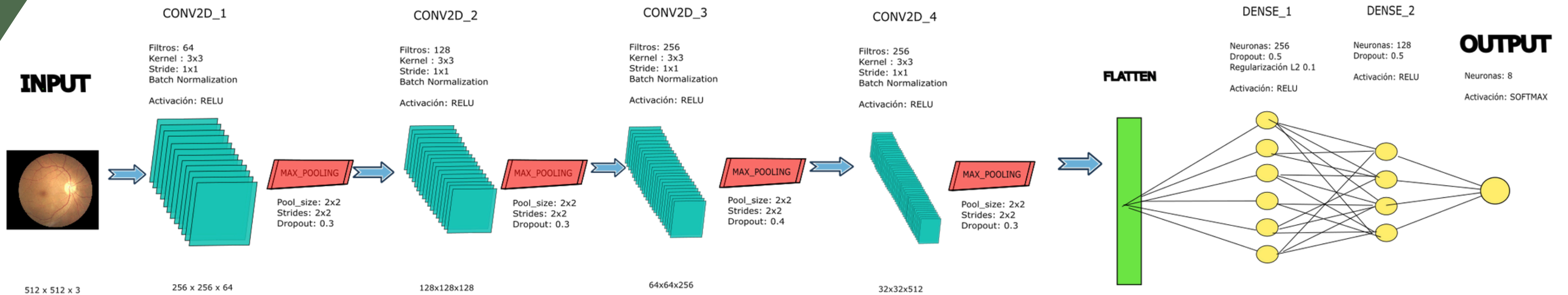
ETIQUETAS

- **N:** Normal.
- **D:** Diabetes.
- **G:** Glaucoma.
- **C:** Catarata.
- **A:** Degeneración macular.
- **H:** Hipertensión.
- **M:** Miopía patológica.
- **O:** Otras enfermedades oculares.

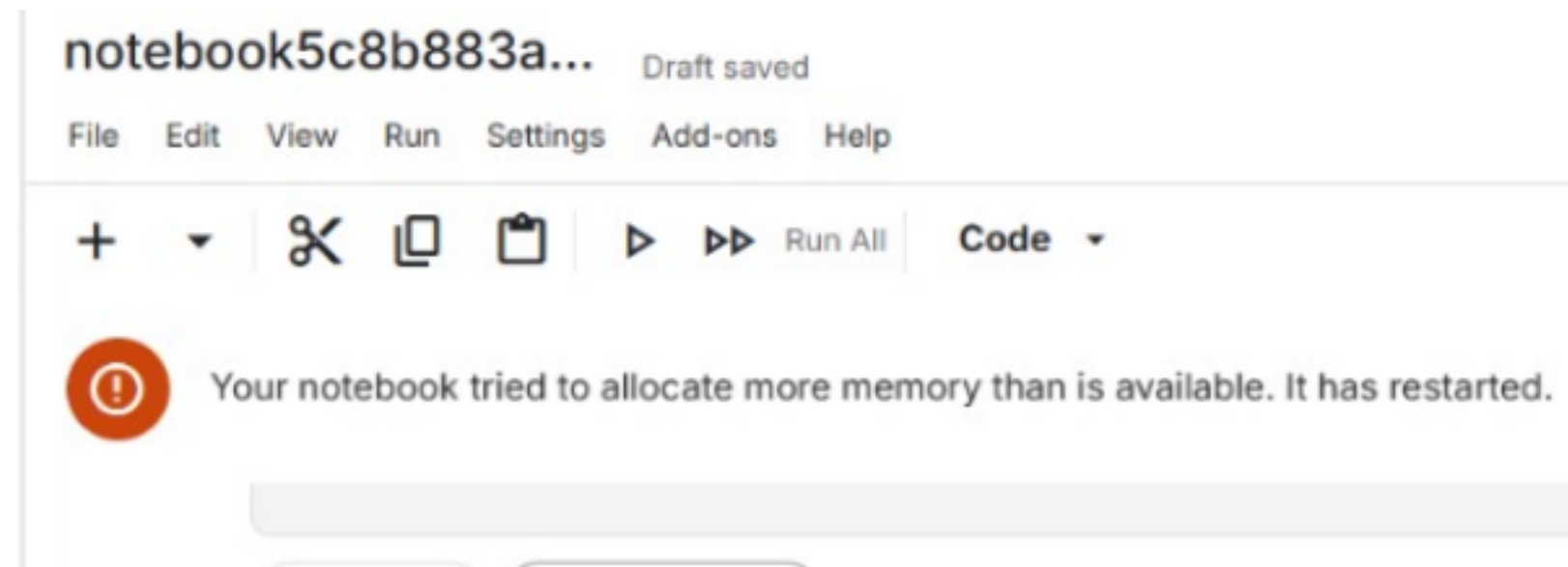


PRIMEROS AVANCES

PIPELINE PRELIMINAR



RESULTADOS PRELIMINARES



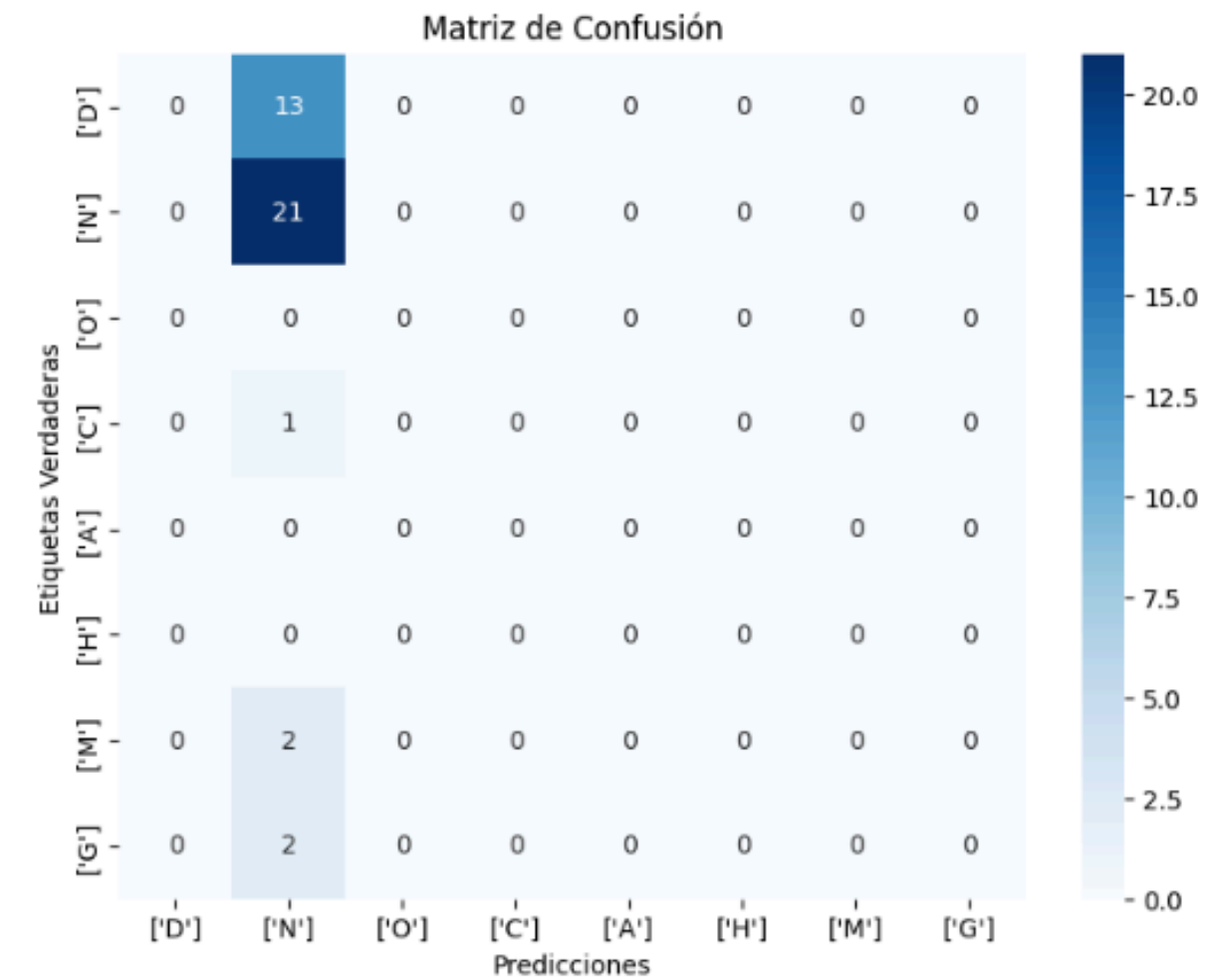
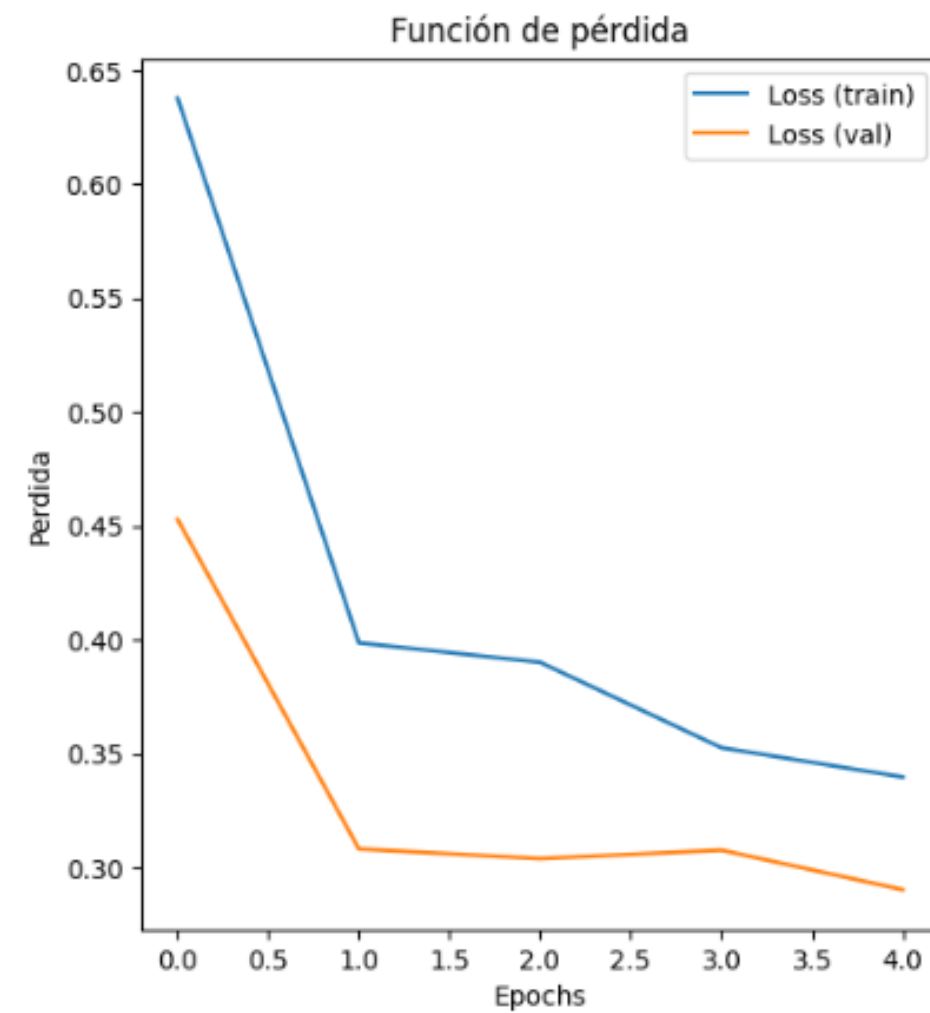
```
# Carga de imágenes de entrenamiento y prueba
X_train_sample = np.array([load_image(f) for f in train_df_sample['filename']])
X_test_sample = np.array([load_image(f) for f in test_df_sample['filename']])

# Arquitectura
model = Sequential([
    Conv2D(64, (3, 3), activation='relu', input_shape=(IMG_SIZE, IMG_SIZE, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(256, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(256, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(128, activation='relu'),
    Dense(8, activation='softmax') # 8 clases para las enfermedades
])

# Compilar el modelo
model.compile(optimizer=SGD(learning_rate=0.001, momentum=0.9),
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

RESULTADOS PRELIMINARES (10%)

Test Loss: 0.2903998792171478, Test Accuracy: 0.5384615659713745



BALANCEO DE ETIQUETAS

Para las clases con más de 1200 imágenes se redujo el número de muestras.

Para las clases con menos de 1200 imágenes, se generaron muestras adicionales mediante transformaciones como rotaciones, cambios de brillo, contrastes, flips horizontales y verticales, y ajustes en el tamaño de las imágenes (usando augmentations).

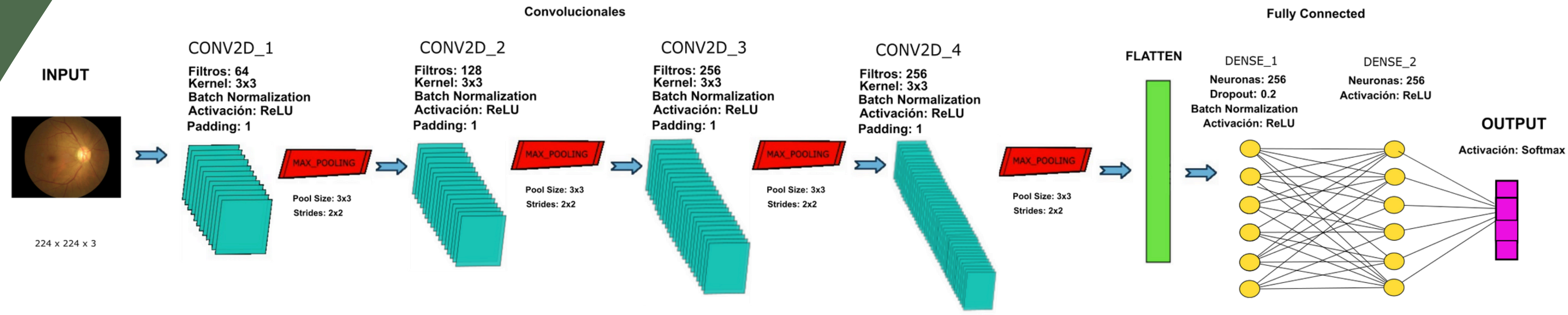
Nueva distribución de etiquetas:
labels

N	1200
D	1200
O	1200
M	1200
H	1200
C	1200
A	1200
G	1200

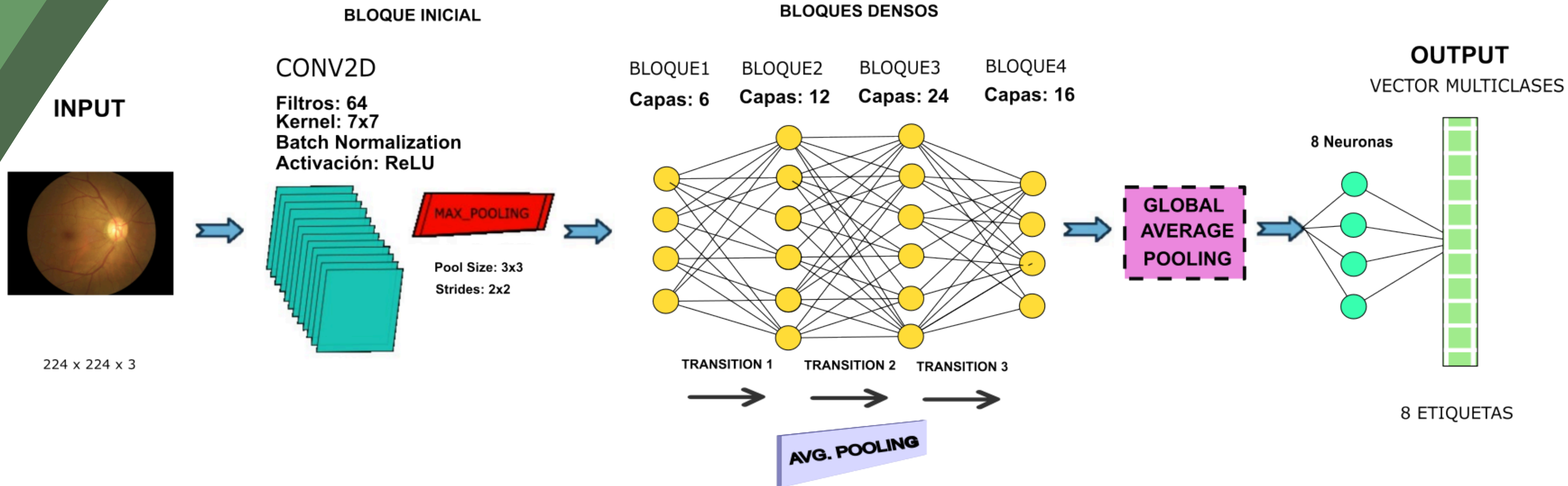
PIPELINES

FINALES

MODELO PROPUESTO



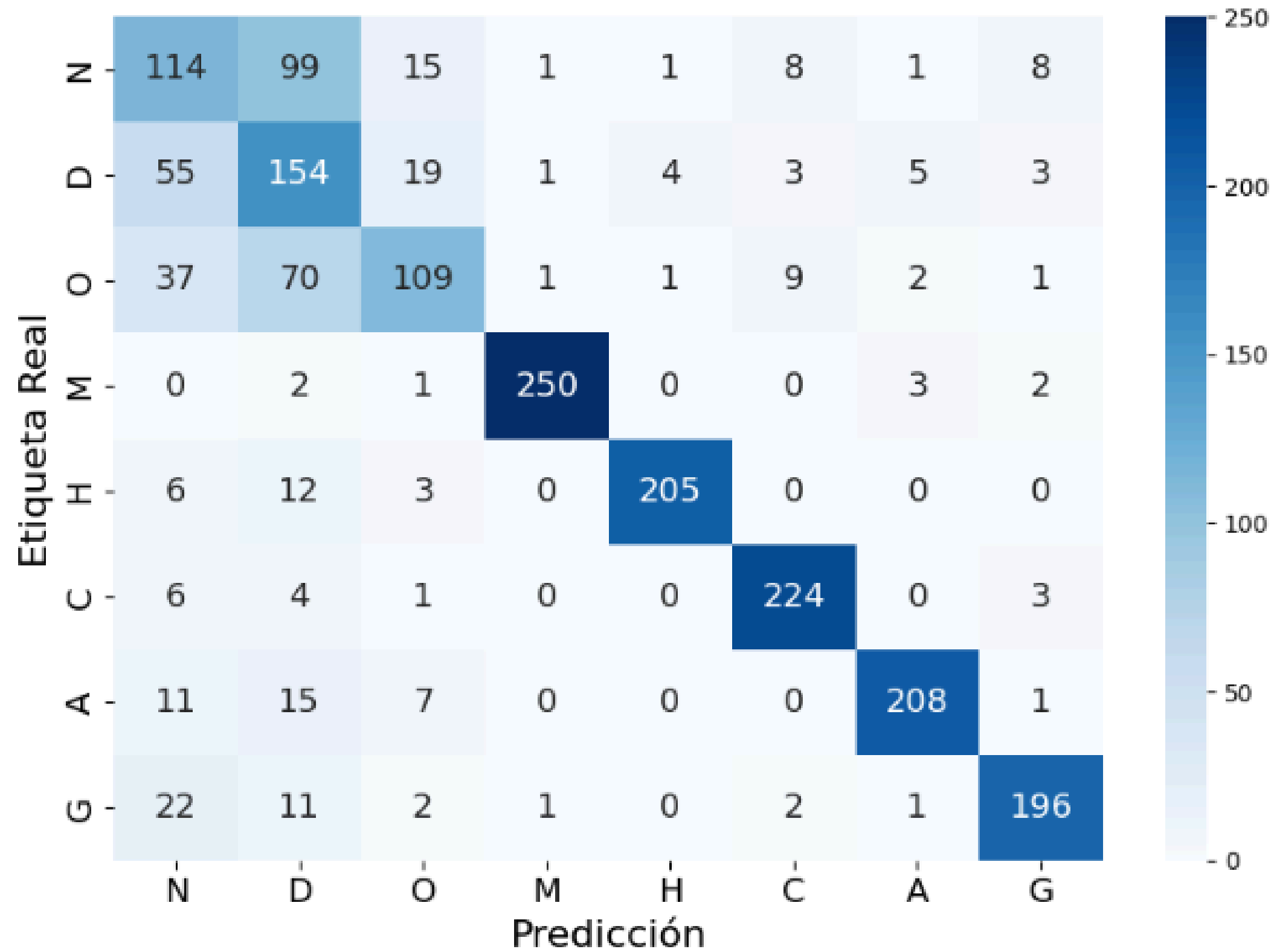
DENSENET



RESULTADOS

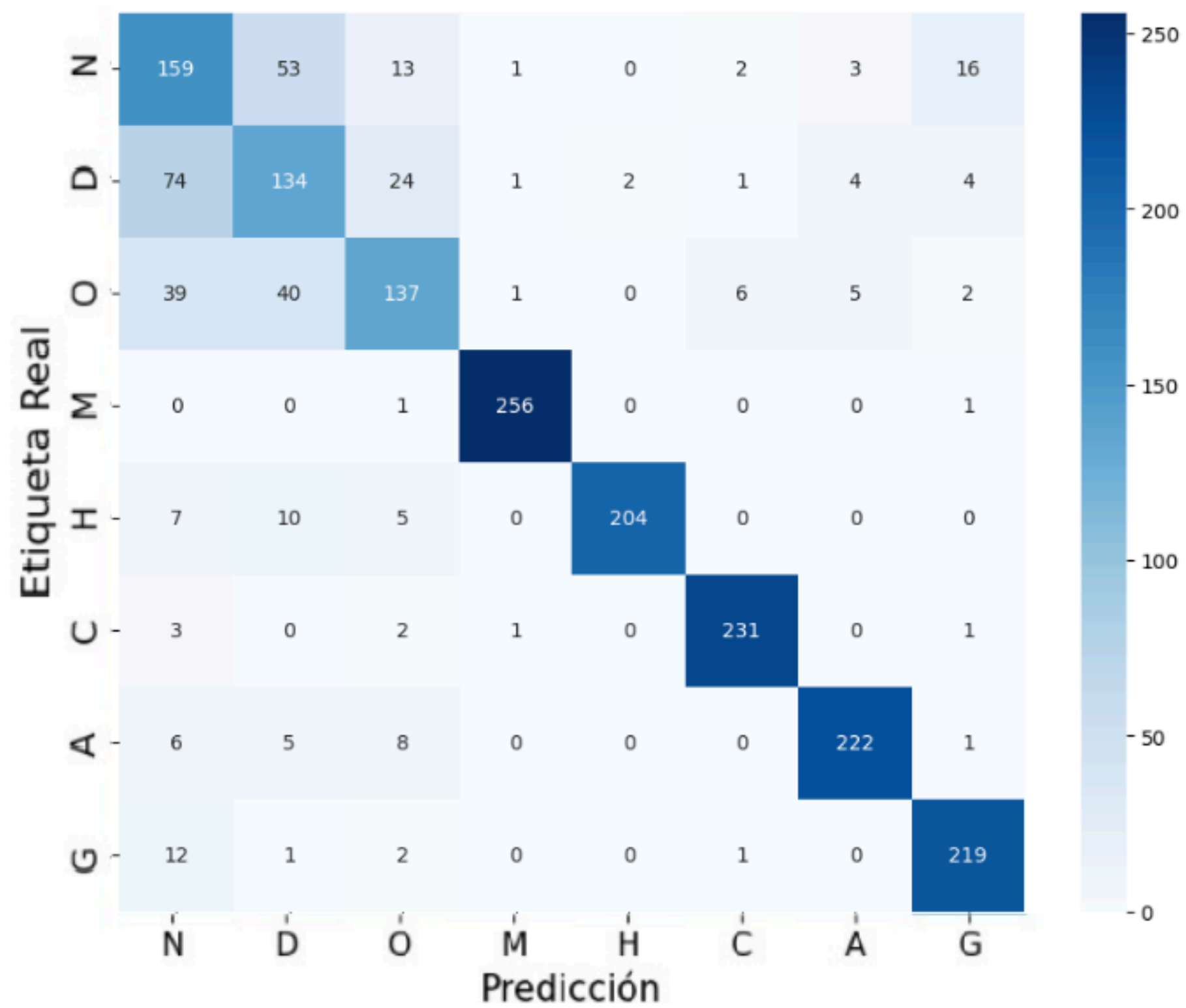
MATRIZ DE CONFUSIÓN

Propuesto



MATRIZ DE CONFUSIÓN

DenseNET



MÉTRICAS EVALUADAS

Propuesto

Classification Report:

	precision	recall	f1-score	support
N	0.4542	0.4615	0.4578	247
D	0.4196	0.6311	0.5041	244
O	0.6943	0.4739	0.5633	230
M	0.9843	0.9690	0.9766	258
H	0.9716	0.9071	0.9382	226
C	0.9106	0.9412	0.9256	238
A	0.9455	0.8595	0.9004	242
G	0.9159	0.8340	0.8731	235
accuracy			0.7604	1920
macro avg	0.7870	0.7597	0.7674	1920
weighted avg	0.7857	0.7604	0.7672	1920

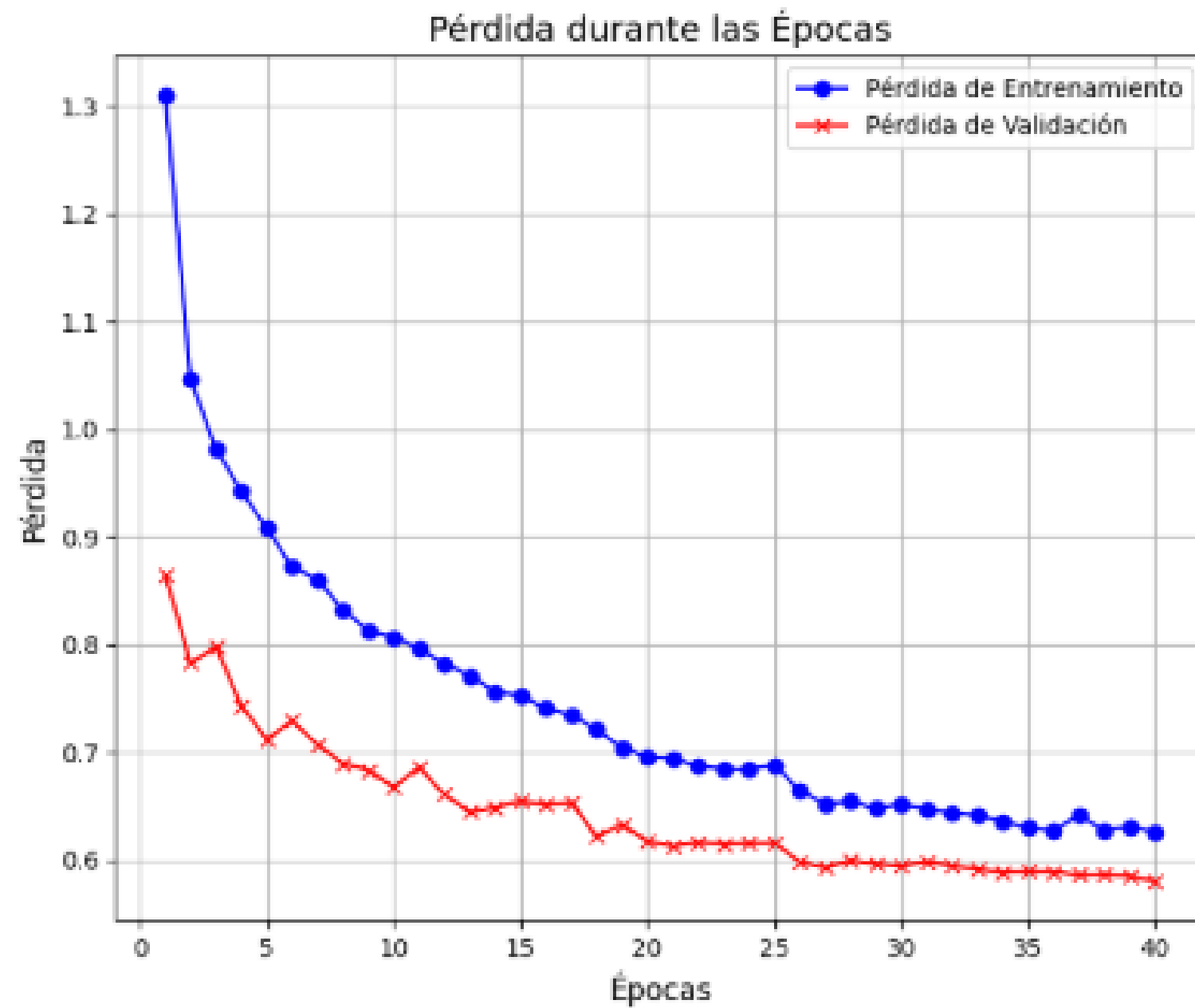
DenseNET

Classification Report:

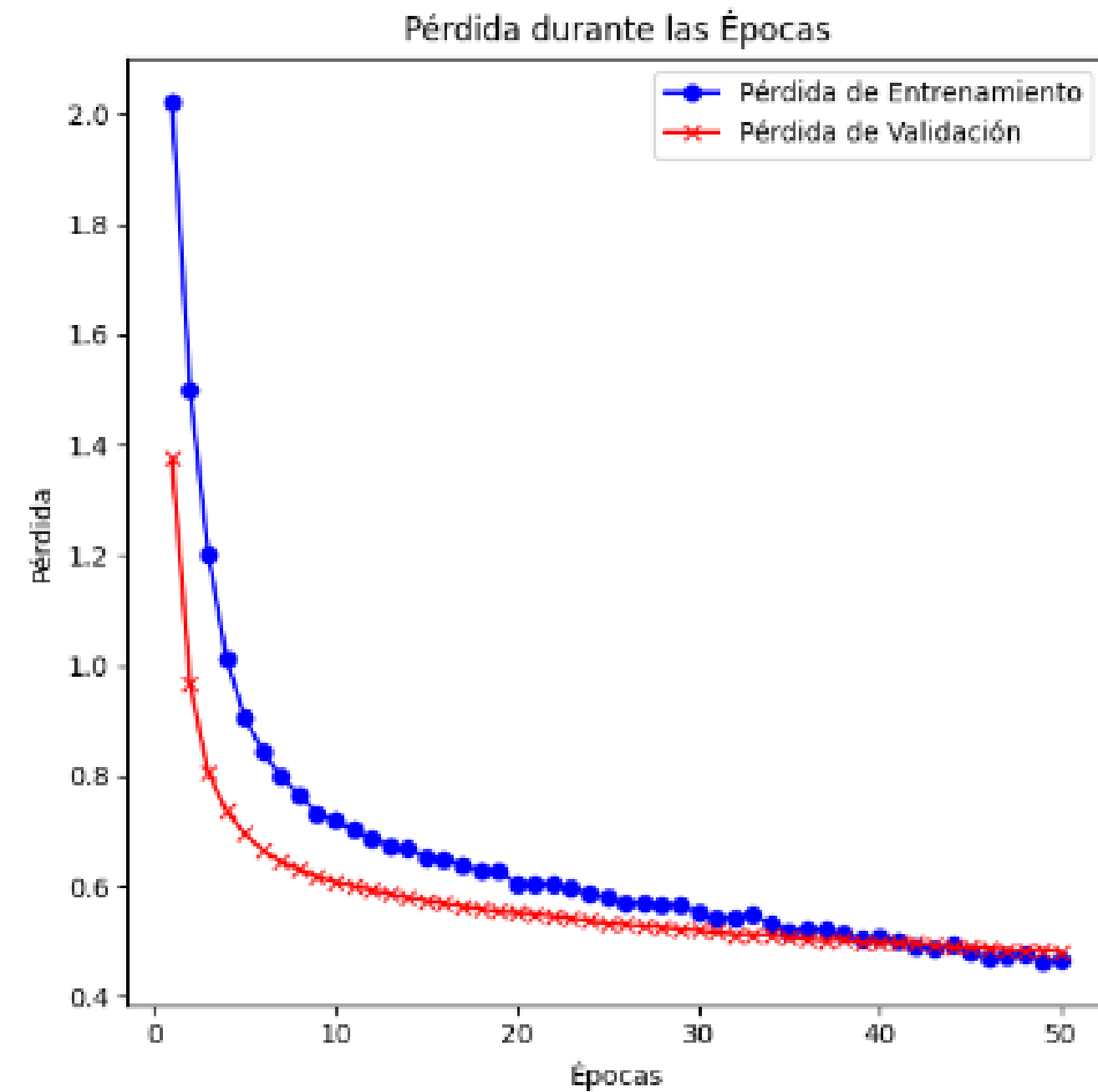
	precision	recall	f1-score	support
N	0.53	0.64	0.58	247
D	0.55	0.55	0.55	244
O	0.71	0.60	0.65	230
M	0.98	0.99	0.99	258
H	0.99	0.90	0.94	226
C	0.96	0.97	0.96	238
A	0.95	0.92	0.93	242
G	0.90	0.93	0.91	235
accuracy			0.81	1920
macro avg	0.82	0.81	0.82	1920
weighted avg	0.82	0.81	0.82	1920

GRÁFICAS DE PERDIDA

Propuesto



DenseNET



CONCLUSIONES

CONCLUSIONES

- La DenseNet, al ser un modelo pre-entrenado, mostró un rendimiento significativamente mejor en términos de pérdida y precisión en comparación con el modelo propuesto.
- Las clases con frecuencia inicial alta experimentaron una disminución en precisión notable. Se redujo la cantidad de información disponible para estas clases, limitando la capacidad del modelo.
- Aquellas clases con menor cantidad de muestras fueron muy precisamente generalizadas por ambos modelos (+90% precisión).
- Tanto en entrenamiento como en validación el modelo progresa adecuadamente en los datos y no se encuentran signos de sobreajuste

TRABAJO FUTURO

- Optimización del Balanceo de Clases: Refinar las técnicas de data augmentation y re-muestreo, profundizando en las primeras clases donde el balanceo fue insuficiente, para mejorar la distribución de clases y evitar sesgos en el entrenamiento.
- Ajuste de Hiperparámetros: Profundizar en el ajuste de optimizadores y tasas de aprendizaje para minimizar el loss.
- Reducir el Loss y Mejorar el Modelo: Explorar el uso de regularización y arquitecturas más complejas para mejorar la precisión del modelo.
- Aumentar el Accuracy: Trabajar en la optimización continua del modelo para lograr un accuracy superior en la validación.

¡GRACIAS!