pi-04-1

October 14, 2024

[ ]:
```
!git clone https://github.com/Negatix092/DM1.git
```

1 1. Dado el conjunto de datos “dog & cats” con 25k y 12.5k imágenes (gatos y perros) de training and test, el cual se puede descargar aquí dogs-vs-cats.zip). Se desea:

- Aplicar la tarea de normalización a los datos para que cumplan con los requerimientos de entrada de los modelos de clasificación. Se sugiere hacer un resize(image) para la dimensión más pequeña entre ambos conjuntos.

[1]:
```
train_folder = '/content/DM1/CNN/dogs-vs-cats/train'
test_folder = '/content/DM1/CNN/dogs-vs-cats/test1'
```

1.1 General

[3]:
```
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
import random
from concurrent.futures import ThreadPoolExecutor

apply_canny = True  # Ajustar a False si no quieres usar la detección de bordes

# Función para contar imágenes en una carpeta
def count_images_in_folder(folder):
    try:
        count = len(os.listdir(folder))
        return count
    except FileNotFoundError:
        print(f"Error: La carpeta {folder} no se encontró.")
        return 0

# Función para cargar imágenes originales (sin ninguna transformación)
def load_original_images(folder):
    images = []
```

```python
    img_paths = [os.path.join(folder, img_name) for img_name in os.listdir(folder)]

    for img_path in img_paths:
        img = cv2.imread(img_path)  # Cargar la imagen tal cual
        if img is not None:
            images.append(img)

    return images

# Función para cargar y preprocesar una imagen
def load_and_preprocess_single_image(img_path, apply_canny=False, for_naive_bayes=False):
    img = cv2.imread(img_path)

    if img is not None:
        original_size = img.shape[:2]  # Almacenar tamaño original
        if for_naive_bayes:
            img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  # Convertir a escala de grises para Naive Bayes

        img = cv2.resize(img, (64, 64))  # Redimensionar a 64x64

        if apply_canny:
          img = cv2.Canny(img, 100, 200)

        img = img / 255.0  # Normalizar los valores de los píxeles a un rango entre 0 y 1
        return img, original_size
    else:
        return None, None

# Función para cargar y preprocesar imágenes usando hilos
def load_and_preprocess_images_threaded(folder, apply_canny=False, for_naive_bayes=False):
    images = []
    original_sizes = []

    img_paths = [os.path.join(folder, img_name) for img_name in os.listdir(folder)]

    # Usar hilos para cargar y preprocesar imágenes en paralelo
    with ThreadPoolExecutor(max_workers=8) as executor:
        results = executor.map(lambda path: load_and_preprocess_single_image(path, apply_canny, for_naive_bayes), img_paths)
```

```python
    for img, original_size in results:
        if img is not None:
            images.append(img)
            original_sizes.append(original_size)

    return np.array(images), np.array(original_sizes)

# Preprocesamiento para ambos conjuntos (entrenamiento y prueba)
def preprocess_images_for_both_models(train_folder, test_folder, apply_canny=False, for_naive_bayes=False):
    # Cargar y preprocesar las imágenes de entrenamiento y prueba usando hilos
    train_images, train_original_sizes = load_and_preprocess_images_threaded(train_folder, apply_canny, for_naive_bayes)
    test_images, test_original_sizes = load_and_preprocess_images_threaded(test_folder, apply_canny, for_naive_bayes)

    return train_images, test_images, train_original_sizes, test_original_sizes

# Función para graficar comparativa de tamaños originales vs redimensionados
def plot_image_sizes(original_sizes, resized_shape):
    original_heights = [size[0] for size in original_sizes]
    original_widths = [size[1] for size in original_sizes]

    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    plt.hist(original_heights, bins=30, alpha=0.7, color='b', label='Original Heights')
    plt.hist(original_widths, bins=30, alpha=0.7, color='r', label='Original Widths')
    plt.title('Tamaños Originales')
    plt.legend()

    plt.subplot(1, 2, 2)
    plt.bar(['Resized Height', 'Resized Width'], [resized_shape[0], resized_shape[1]], color=['b', 'r'])
    plt.title(f'Tamaños Redimensionados ({resized_shape[0]}x{resized_shape[1]})')
    plt.show()

# Función para seleccionar una muestra de las imágenes
def sample_image_sizes(original_sizes, sample_fraction=0.05):
    sample_size = int(len(original_sizes) * sample_fraction)
    return random.sample(list(original_sizes), sample_size)
```

```
# Función para mostrar imágenes
def show_random_images(images, title, n=4):
    indices = random.sample(range(len(images)), n)
    plt.figure(figsize=(10, 10))
    for i, idx in enumerate(indices):
        plt.subplot(2, 2, i + 1)
        plt.imshow(images[idx], cmap='gray' if len(images[idx].shape) == 2 else␣
↪None)
        plt.title(f"{title} - {i+1}")
        plt.axis('off')
    plt.show()

# Verificar el número de imágenes procesadas
def verify_image_count(train_folder, test_folder):
    print("Verificación de número de imágenes:")
    train_count = count_images_in_folder(train_folder)
    test_count = count_images_in_folder(test_folder)
    print(f"Total imágenes en entrenamiento: {train_count}")
    print(f"Total imágenes en test: {test_count}")  # Verificar número de␣
↪imágenes en cada conjunto
```

[4]:
```
train_images, test_images, train_original_sizes, test_original_sizes =␣
↪preprocess_images_for_both_models(train_folder, test_folder, apply_canny)
```

1.1.1 Visualización

Número de imágenes en cada conjunto

[5]:
```
# Verificar número de imágenes en cada conjunto
verify_image_count(train_folder, test_folder)
```

```
Verificación de número de imágenes:
Total imágenes en entrenamiento: 25000
Total imágenes en test: 12500
```

Muestra imágenes originales

[6]:
```
# Cargar imágenes originales tal como están
original_images = load_original_images(train_folder)

# Mostrar 4 imágenes originales
show_random_images(original_images, "Imágenes Originales")
```

Imágenes Originales - 1

Imágenes Originales - 2

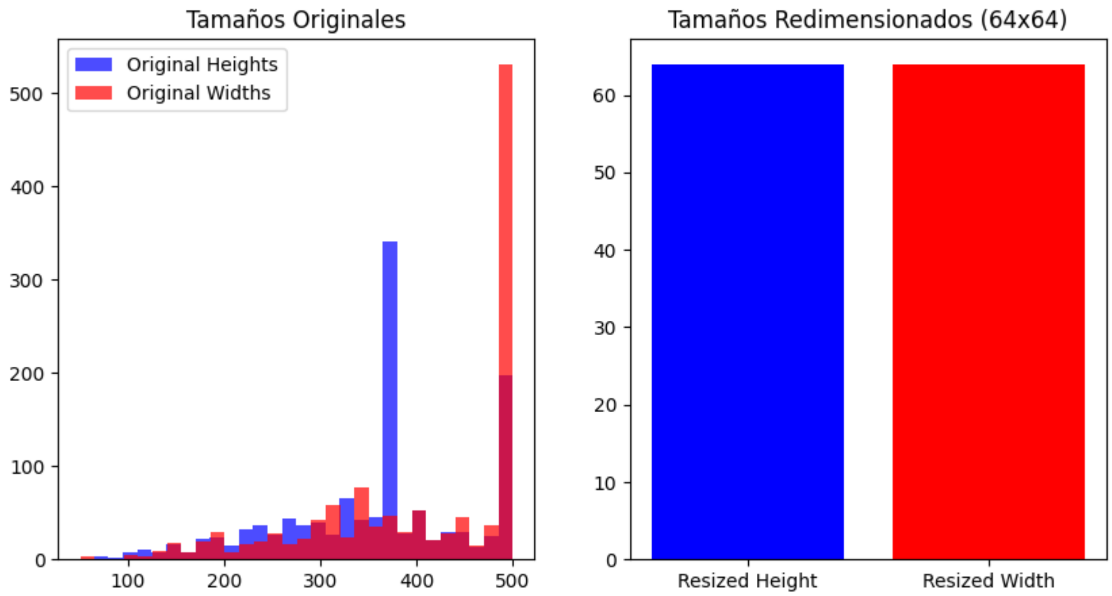
Imágenes Originales - 3

Imágenes Originales - 4

Mostrar imágenes con Preprocesamiento General

[7]:
```
# Mostrar 4 imágenes con preprocesamiento básico (sin Canny y sin Naive Bayes)
train_images_general, _ = load_and_preprocess_images_threaded(train_folder,␣
  ↪apply_canny=False, for_naive_bayes=False)
show_random_images(train_images_general, "Preprocesamiento General")
```

[8]:
```
# Tomar una muestra del 5% de las imágenes y mostrar tamaños originales para General
sampled_train_sizes_general = sample_image_sizes(train_original_sizes, sample_fraction=0.05)
plot_image_sizes(sampled_train_sizes_general, (64, 64))
```



1.2 Preprocesamiento Naive Bayes

[9]:
```
import numpy as np
from skimage.feature import hog
import cv2
from sklearn.decomposition import PCA

# Aplanar las imágenes (necesario para Naive Bayes)
def flatten_images(images):
    return images.reshape(images.shape[0], -1)

# Función para extraer características HOG
def hog_features(img, for_naive_bayes=False):
    if for_naive_bayes:
        # Si Naive Bayes, asegurarse de que la imagen esté en escala de grises
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) if len(img.shape) == 3 else␣
↪img
        features = hog(img, pixels_per_cell=(8,8), cells_per_block=(2, 2),␣
↪visualize=False)
    else:
        # Para CNN o imágenes multicanal, manejar diferentes canales de imagen
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) if len(img.shape) == 3 else␣
↪img
        features = hog(img, pixels_per_cell=(8,8), cells_per_block=(2, 2),␣
↪visualize=False)
```

```python
    return features

# Función para aplicar pca
def preprocess_images_with_pca(train_nb, test_nb):
    pca = PCA(n_components=100)  # Adjust the number of components to explain␣
 ↪enough variance
    train_nb_pca = pca.fit_transform(train_nb)
    test_nb_pca = pca.transform(test_nb)
    return train_nb_pca, test_nb_pca

# Función para aplicar HOG a todas las imágenes de un conjunto
def preprocess_naive_bayes(train_images):
    hog_train = [hog_features(img, for_naive_bayes=True) for img in␣
 ↪train_images]
    return np.array(hog_train)

# Para Naive Bayes, preprocesar en escala de grises sin detección de bordes
train_images_naive_bayes, test_images_naive_bayes,␣
 ↪train_original_sizes_naive_bayes, test_original_sizes_naive_bayes =␣
 ↪preprocess_images_for_both_models(train_folder, test_folder,␣
 ↪apply_canny=False, for_naive_bayes=True)

# Aplanar las imágenes para Naive Bayes
train_images_naive_bayes_flat = flatten_images(train_images_naive_bayes)
test_images_naive_bayes_flat = flatten_images(test_images_naive_bayes)

# Extraer características HOG para Naive Bayes
train_images_naive_bayes_HOG= preprocess_naive_bayes(train_images_naive_bayes)
test_images_naive_bayes_HOG = preprocess_naive_bayes(test_images_naive_bayes)

# Aplicar PCA
train_nb, test_nb = preprocess_images_with_pca(train_images_naive_bayes_HOG,␣
 ↪test_images_naive_bayes_HOG)
```

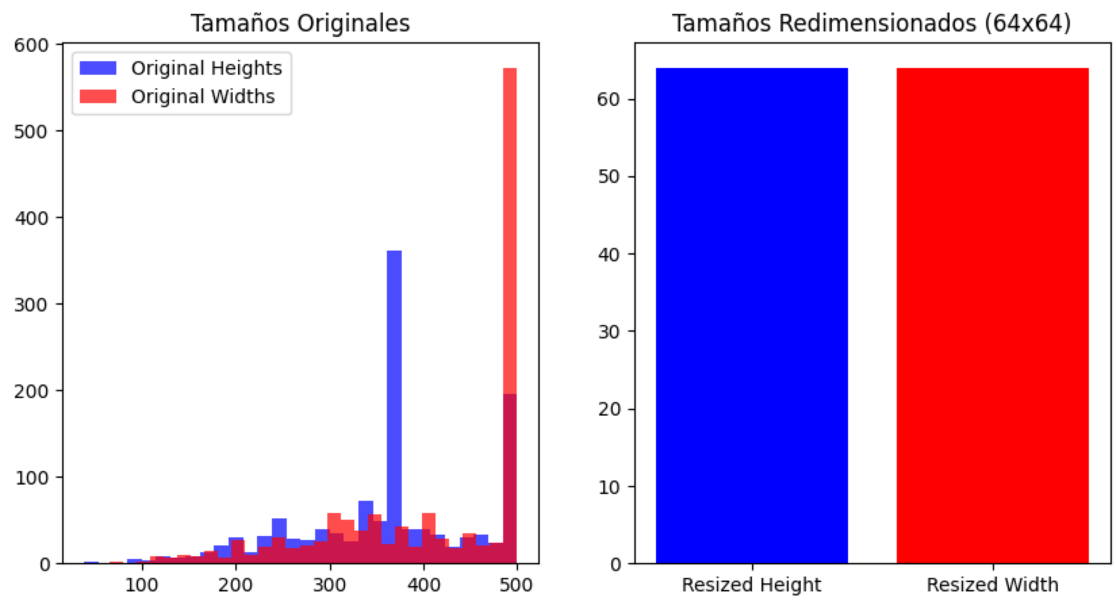
1.2.1 Visualización

[10]:
```python
# Mostrar las dimensiones de las imagenes aplanadas
print(f"Imágenes aplanadas para Naive Bayes - Train:␣
 ↪{train_images_naive_bayes_flat.shape}")
print(f"Imágenes aplanadas para Naive Bayes - Test:␣
 ↪{test_images_naive_bayes_flat.shape}")

# Mostrar las dimensiones después de aplicar HOG
print(f"Imágenes preprocesadas para Naive Bayes (HOG) - Train:␣
 ↪{train_images_naive_bayes_HOG.shape}")
```

```
print(f"Imágenes preprocesadas para Naive Bayes (HOG) - Test:␣
↪{test_images_naive_bayes_HOG.shape}")

# Mostrar las dimensiones después de aplicar PCA
print(f"Imágenes preprocesadas para Naive Bayes (PCA) - Train: {train_nb.
↪shape}")
print(f"Imágenes preprocesadas para Naive Bayes (PCA) - Test: {test_nb.shape}")
```

```
Imágenes aplanadas para Naive Bayes - Train: (25000, 4096)
Imágenes aplanadas para Naive Bayes - Test: (12500, 4096)
Imágenes preprocesadas para Naive Bayes (HOG) - Train: (25000, 1764)
Imágenes preprocesadas para Naive Bayes (HOG) - Test: (12500, 1764)
Imágenes preprocesadas para Naive Bayes (PCA) - Train: (25000, 100)
Imágenes preprocesadas para Naive Bayes (PCA) - Test: (12500, 100)
```

[11]:
```
# Tomar una muestra del 5% de las imágenes y mostrar tamaños originales para␣
↪Naive Bayes
sampled_train_sizes_nb = sample_image_sizes(train_original_sizes_naive_bayes,␣
↪sample_fraction=0.05)
plot_image_sizes(sampled_train_sizes_nb, (64, 64))
```



1.3 Preprocesamiento CNN

[12]:
```
# Preprocesamiento específico para CNN
def preprocess_cnn(train_folder, test_folder, apply_canny=False):
```

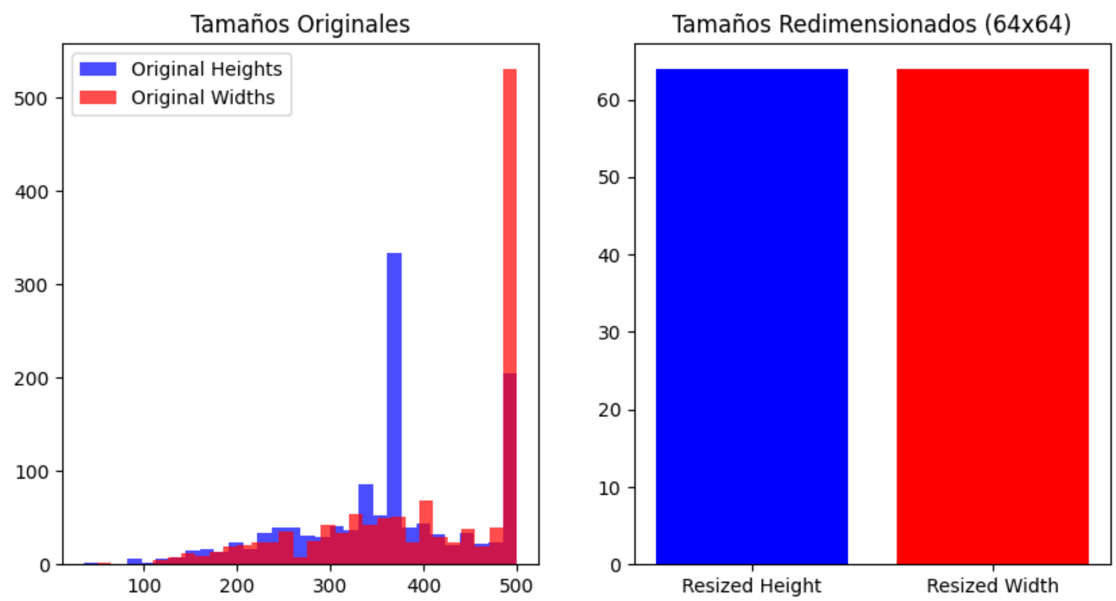
```
    train_cnn, train_original_sizes_cnn =␣
↪load_and_preprocess_images_threaded(train_folder, apply_canny=apply_canny,␣
↪for_naive_bayes=False)
    test_cnn, test_original_sizes_cnn =␣
↪load_and_preprocess_images_threaded(test_folder, apply_canny=apply_canny,␣
↪for_naive_bayes=False)

    return train_cnn, test_cnn, train_original_sizes_cnn,␣
↪test_original_sizes_cnn
```

1.3.1 Visualización

[13]:
```
train_cnn, test_cnn, train_original_sizes_cnn, test_original_sizes_cnn =␣
↪preprocess_cnn(train_folder, test_folder, apply_canny=False)

# Mostrar las dimensiones de los conjuntos preprocesados
print(f"Conjunto de entrenamiento para CNN - Train: {train_cnn.shape}")
print(f"Conjunto de test para CNN - Test: {test_cnn.shape}")
```

```
Conjunto de entrenamiento para CNN - Train: (25000, 64, 64, 3)
Conjunto de test para CNN - Test: (12500, 64, 64, 3)
```

[14]:
```
# Tomar una muestra del 5% de las imágenes y mostrar tamaños
sampled_train_sizes_cnn = sample_image_sizes(train_original_sizes_cnn,␣
↪sample_fraction=0.05)
plot_image_sizes(sampled_train_sizes_cnn, (64, 64))
```



2 Utilizar la técnica, stratified 10-fold cross-validation (CV) para mostrar los resultados a nivel de promedio y desviación estándar de cada métrica supervisada considerada para esta tarea (AUC, Precision, Recall, F1-score). Investigar la técnica k-fold Cross-Validation y las métricas solicitadas para aplicarla correctamente.

2.0.1 Stratified 10-Fold CV usando NB

2.0.2 Improved version (multinomialNB)

[15]:
```
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import precision_score, roc_auc_score, recall_score,␣
 ↪f1_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import PolynomialFeatures
# Aplanar las imágenes para usar en Naive Bayes
# X_flat = train_images.reshape(train_images.shape[0], -1)

# Crear las etiquetas (1 para perro, 0 para gato)
y = np.array([1 if 'dog' in img else 0 for img in os.listdir(train_folder)])

# Escalar los valores para que estén en el rango [0, 1] para MultinomialNB
scaler = MinMaxScaler()
X_flat = scaler.fit_transform(train_nb)  # Normalizamos los datos entre 0 y 1

# Polynomial Features
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X_flat)

# Model
nb_model = MultinomialNB()

# Definir el Stratified K-Fold CV
skf = StratifiedKFold(n_splits=10)

# Listas para almacenar los resultados de cada métrica
auc_scores = []
precision_scores = []
recall_scores = []
f1_scores = []

# Cross-validation
for train_index, test_index in skf.split(X_poly, y):
    X_train, X_test = X_poly[train_index], X_poly[test_index]
```

```
    y_train, y_test = y[train_index], y[test_index]

    # Train model
    nb_model.fit(X_train, y_train)

    # Predict
    y_pred = nb_model.predict(X_test)
    y_prob = nb_model.predict_proba(X_test)[:, 1]

    # Evaluate
    auc_scores.append(roc_auc_score(y_test, y_prob))
    precision_scores.append(precision_score(y_test, y_pred))
    recall_scores.append(recall_score(y_test, y_pred))
    f1_scores.append(f1_score(y_test, y_pred))

# Mostrar los resultados promedio y desviación estándar
print(f"AUC: {np.mean(auc_scores):.3f} ± {np.std(auc_scores):.3f}")
print(f"Precision: {np.mean(precision_scores):.3f} ± {np.std(precision_scores):.
 ↪3f}")
print(f"Recall: {np.mean(recall_scores):.3f} ± {np.std(recall_scores):.3f}")
print(f"F1-score: {np.mean(f1_scores):.3f} ± {np.std(f1_scores):.3f}")
```

```
AUC: 0.778 ± 0.009
Precision: 0.700 ± 0.011
Recall: 0.744 ± 0.011
F1-score: 0.721 ± 0.006
```

2.0.3 gaussianNB

[16]:
```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import roc_auc_score, precision_score, recall_score,␣
 ↪f1_score
from sklearn.preprocessing import MinMaxScaler

# Escalar los valores entre 0 y 1 para garantizar un rango adecuado para␣
 ↪GaussianNB
scaler = MinMaxScaler()
train_nb_scaled = scaler.fit_transform(train_nb)
test_nb_scaled = scaler.transform(test_nb)

# Etiquetas: 1 para perros, 0 para gatos
y = np.array([1 if 'dog' in img else 0 for img in os.listdir(train_folder)])

# Crear el modelo Naive Bayes (Gaussian)
nb_model = GaussianNB()
```

```python
# Definir el Stratified K-Fold CV
skf = StratifiedKFold(n_splits=10)

# Listas para almacenar los resultados de cada métrica
auc_scores = []
precision_scores = []
recall_scores = []
f1_scores = []

# Aplicar 10-fold cross-validation
for train_index, test_index in skf.split(train_nb_scaled, y):
    X_train, X_test = train_nb_scaled[train_index], train_nb_scaled[test_index]
    y_train, y_test = y[train_index], y[test_index]

    # Entrenar el modelo
    nb_model.fit(X_train, y_train)

    # Hacer predicciones
    y_pred = nb_model.predict(X_test)
    y_prob = nb_model.predict_proba(X_test)[:, 1]

    # Calcular métricas
    auc_scores.append(roc_auc_score(y_test, y_prob))
    precision_scores.append(precision_score(y_test, y_pred))
    recall_scores.append(recall_score(y_test, y_pred))
    f1_scores.append(f1_score(y_test, y_pred))

# Mostrar los resultados promedio y desviación estándar
print(f"AUC: {np.mean(auc_scores):.3f} ± {np.std(auc_scores):.3f}")
print(f"Precision: {np.mean(precision_scores):.3f} ± {np.std(precision_scores):.
 ↪3f}")
print(f"Recall: {np.mean(recall_scores):.3f} ± {np.std(recall_scores):.3f}")
print(f"F1-score: {np.mean(f1_scores):.3f} ± {np.std(f1_scores):.3f}")
```

```
AUC: 0.724 ± 0.012
Precision: 0.654 ± 0.010
Recall: 0.690 ± 0.013
F1-score: 0.671 ± 0.008
```

2.0.4 Tres arquitecturas CNN

[19]:
```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,␣
 ↪Dropout

# Arquitectura 1: Modelo básico ajustado para 1 canal (escala de grises)
```

```python
def create_cnn_model_1():
    model = Sequential([
        Conv2D(32, (3,3), activation='relu', input_shape=(64, 64, 3)),
        MaxPooling2D(pool_size=(2,2)),
        Conv2D(64, (3,3), activation='relu'),
        MaxPooling2D(pool_size=(2,2)),
        Flatten(),
        Dense(128, activation='relu'),
        Dense(2, activation='softmax')
    ])
    return model

# Arquitectura 2: Modelo con más capas ajustado para 1 canal (escala de grises)
def create_cnn_model_2():
    model = Sequential([
        Conv2D(32, (3,3), activation='relu', input_shape=(64, 64, 3)),
        MaxPooling2D(pool_size=(2,2)),
        Conv2D(64, (3,3), activation='relu'),
        MaxPooling2D(pool_size=(2,2)),
        Conv2D(128, (3,3), activation='relu'),
        MaxPooling2D(pool_size=(2,2)),
        Flatten(),
        Dense(256, activation='relu'),
        Dropout(0.5),
        Dense(2, activation='softmax')
    ])
    return model

# Arquitectura 3: Más densa con Dropout ajustado para 1 canal (escala de grises)
def create_cnn_model_3():
    model = Sequential([
        Conv2D(64, (3,3), activation='relu', input_shape=(64, 64, 3)),
        MaxPooling2D(pool_size=(2,2)),
        Conv2D(128, (3,3), activation='relu'),
        MaxPooling2D(pool_size=(2,2)),
        Conv2D(256, (3,3), activation='relu'),
        MaxPooling2D(pool_size=(2,2)),
        Flatten(),
        Dense(512, activation='relu'),
        Dropout(0.5),
        Dense(2, activation='softmax')
    ])
    return model
```

2.0.5 Entrenar modelos CNN con Stratified 10-Fold CV

[20]:
```
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import roc_auc_score, precision_score, recall_score,␣
↪f1_score
import matplotlib.pyplot as plt

# Convertir etiquetas a one-hot encoding para CNN
y_categorical = to_categorical(y, num_classes=2)

# Definir StratifiedKFold
skf = StratifiedKFold(n_splits=10)

# Función para entrenar y mostrar métricas sin Wandb
def train_cnn_model(model_fn):
    fold = 0
    training_losses = []
    validation_losses = []

    # Listas para almacenar las métricas de cada fold
    auc_scores = []
    precision_scores = []
    recall_scores = []
    f1_scores = []

    for train_index, test_index in skf.split(train_cnn, y):  # Usar train_cnn␣
↪en lugar de train_images
        fold += 1
        print("----------------------------------------------")
        print(f"\nTraining Fold {fold}...")

        X_train, X_test = train_cnn[train_index], train_cnn[test_index]  # Usar␣
↪train_cnn en lugar de train_images
        y_train, y_test = y_categorical[train_index], y_categorical[test_index]

        # Crear el modelo
        model = model_fn()
        model.compile(optimizer='adam', loss='categorical_crossentropy',␣
↪metrics=['accuracy'])

        # Entrenar el modelo
        history = model.fit(X_train, y_train, validation_data=(X_test, y_test),␣
↪epochs=10, batch_size=32)

        # Guardar pérdidas
        training_losses.append(history.history['loss'])
```

```
        validation_losses.append(history.history['val_loss'])

        # Hacer predicciones para las métricas
        y_pred = model.predict(X_test)
        y_pred_class = y_pred.argmax(axis=1)
        y_true = y_test.argmax(axis=1)

        # Calcular las métricas
        auc = roc_auc_score(y_true, y_pred[:, 1])
        precision = precision_score(y_true, y_pred_class)
        recall = recall_score(y_true, y_pred_class)
        f1 = f1_score(y_true, y_pred_class)

        # Guardar métricas
        auc_scores.append(auc)
        precision_scores.append(precision)
        recall_scores.append(recall)
        f1_scores.append(f1)

    # Mostrar los resultados promedio y desviación estándar
    print(f"\nResults")
    print(f"AUC: {np.mean(auc_scores):.3f} ± {np.std(auc_scores):.3f}")
    print(f"Precision: {np.mean(precision_scores):.3f} ± {np.
 ↪std(precision_scores):.3f}")
    print(f"Recall: {np.mean(recall_scores):.3f} ± {np.std(recall_scores):.3f}")
    print(f"F1-score: {np.mean(f1_scores):.3f} ± {np.std(f1_scores):.3f}")

    # Graficar las pérdidas (loss) de entrenamiento y validación
    plt.figure(figsize=(10, 5))
    plt.plot(np.mean(training_losses, axis=0), label="Training Loss")
    plt.plot(np.mean(validation_losses, axis=0), label="Validation Loss")
    plt.title("Training vs Validation Loss")
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()
    plt.show()

# Entrenar usando el primer modelo CNN
train_cnn_model(create_cnn_model_1)
```

```
---------------------------------------------

Training Fold 1…
Epoch 1/10
704/704 [==============================] - 4s 3ms/step - loss: 0.5984 -
accuracy: 0.6680 - val_loss: 0.5302 - val_accuracy: 0.7308
Epoch 2/10
```

```
704/704 [==============================] - 2s 2ms/step - loss: 0.4868 -
accuracy: 0.7643 - val_loss: 0.4507 - val_accuracy: 0.7916
Epoch 3/10
704/704 [==============================] - 2s 2ms/step - loss: 0.4270 -
accuracy: 0.8020 - val_loss: 0.4612 - val_accuracy: 0.7872
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3708 -
accuracy: 0.8320 - val_loss: 0.4577 - val_accuracy: 0.7984
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3147 -
accuracy: 0.8616 - val_loss: 0.4419 - val_accuracy: 0.8064
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2568 -
accuracy: 0.8888 - val_loss: 0.4540 - val_accuracy: 0.8064
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1969 -
accuracy: 0.9195 - val_loss: 0.5060 - val_accuracy: 0.8072
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1432 -
accuracy: 0.9434 - val_loss: 0.6085 - val_accuracy: 0.7852
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1035 -
accuracy: 0.9604 - val_loss: 0.7182 - val_accuracy: 0.7904
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0677 -
accuracy: 0.9761 - val_loss: 0.8648 - val_accuracy: 0.7820
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 2…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.5796 -
accuracy: 0.6885 - val_loss: 0.5315 - val_accuracy: 0.7352
Epoch 2/10
704/704 [==============================] - 2s 2ms/step - loss: 0.4738 -
accuracy: 0.7744 - val_loss: 0.4413 - val_accuracy: 0.7968
Epoch 3/10
704/704 [==============================] - 2s 2ms/step - loss: 0.4071 -
accuracy: 0.8130 - val_loss: 0.4132 - val_accuracy: 0.8080
Epoch 4/10
704/704 [==============================] - 2s 2ms/step - loss: 0.3475 -
accuracy: 0.8440 - val_loss: 0.4102 - val_accuracy: 0.8164
Epoch 5/10
704/704 [==============================] - 2s 2ms/step - loss: 0.2790 -
accuracy: 0.8791 - val_loss: 0.4350 - val_accuracy: 0.8140
Epoch 6/10
704/704 [==============================] - 2s 2ms/step - loss: 0.1957 -
accuracy: 0.9195 - val_loss: 0.5426 - val_accuracy: 0.7784
```

```
Epoch 7/10
704/704 [==============================] - 2s 2ms/step - loss: 0.1283 -
accuracy: 0.9508 - val_loss: 0.5441 - val_accuracy: 0.8088
Epoch 8/10
704/704 [==============================] - 2s 2ms/step - loss: 0.0757 -
accuracy: 0.9728 - val_loss: 0.6991 - val_accuracy: 0.8132
Epoch 9/10
704/704 [==============================] - 2s 2ms/step - loss: 0.0494 -
accuracy: 0.9839 - val_loss: 0.7364 - val_accuracy: 0.8124
Epoch 10/10
704/704 [==============================] - 2s 2ms/step - loss: 0.0395 -
accuracy: 0.9872 - val_loss: 0.8229 - val_accuracy: 0.7924
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 3…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.6073 -
accuracy: 0.6647 - val_loss: 0.5295 - val_accuracy: 0.7356
Epoch 2/10
704/704 [==============================] - 2s 2ms/step - loss: 0.4897 -
accuracy: 0.7655 - val_loss: 0.4666 - val_accuracy: 0.7732
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4335 -
accuracy: 0.8021 - val_loss: 0.4135 - val_accuracy: 0.7984
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3902 -
accuracy: 0.8255 - val_loss: 0.3958 - val_accuracy: 0.8184
Epoch 5/10
704/704 [==============================] - 2s 2ms/step - loss: 0.3454 -
accuracy: 0.8473 - val_loss: 0.4022 - val_accuracy: 0.8152
Epoch 6/10
704/704 [==============================] - 2s 2ms/step - loss: 0.3025 -
accuracy: 0.8691 - val_loss: 0.4134 - val_accuracy: 0.8120
Epoch 7/10
704/704 [==============================] - 2s 2ms/step - loss: 0.2624 -
accuracy: 0.8887 - val_loss: 0.4166 - val_accuracy: 0.8192
Epoch 8/10
704/704 [==============================] - 2s 2ms/step - loss: 0.2205 -
accuracy: 0.9069 - val_loss: 0.4515 - val_accuracy: 0.8140
Epoch 9/10
704/704 [==============================] - 2s 2ms/step - loss: 0.1824 -
accuracy: 0.9253 - val_loss: 0.4939 - val_accuracy: 0.8116
Epoch 10/10
704/704 [==============================] - 2s 2ms/step - loss: 0.1453 -
accuracy: 0.9414 - val_loss: 0.5550 - val_accuracy: 0.8008
79/79 [==============================] - 0s 1ms/step
---------------------------------------------
```

```
Training Fold 4…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.5973 -
accuracy: 0.6666 - val_loss: 0.5328 - val_accuracy: 0.7332
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4698 -
accuracy: 0.7795 - val_loss: 0.4756 - val_accuracy: 0.7712
Epoch 3/10
704/704 [==============================] - 2s 2ms/step - loss: 0.4073 -
accuracy: 0.8144 - val_loss: 0.4548 - val_accuracy: 0.7792
Epoch 4/10
704/704 [==============================] - 2s 2ms/step - loss: 0.3532 -
accuracy: 0.8441 - val_loss: 0.4198 - val_accuracy: 0.8032
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2938 -
accuracy: 0.8739 - val_loss: 0.4614 - val_accuracy: 0.7984
Epoch 6/10
704/704 [==============================] - 2s 2ms/step - loss: 0.2356 -
accuracy: 0.9016 - val_loss: 0.5326 - val_accuracy: 0.8008
Epoch 7/10
704/704 [==============================] - 2s 2ms/step - loss: 0.1676 -
accuracy: 0.9337 - val_loss: 0.5271 - val_accuracy: 0.8100
Epoch 8/10
704/704 [==============================] - 2s 2ms/step - loss: 0.1122 -
accuracy: 0.9589 - val_loss: 0.6233 - val_accuracy: 0.8016
Epoch 9/10
704/704 [==============================] - 2s 2ms/step - loss: 0.0696 -
accuracy: 0.9752 - val_loss: 0.7040 - val_accuracy: 0.8000
Epoch 10/10
704/704 [==============================] - 2s 2ms/step - loss: 0.0525 -
accuracy: 0.9823 - val_loss: 0.9225 - val_accuracy: 0.7856
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 5…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.5912 -
accuracy: 0.6760 - val_loss: 0.5174 - val_accuracy: 0.7428
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4763 -
accuracy: 0.7720 - val_loss: 0.4700 - val_accuracy: 0.7796
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4149 -
accuracy: 0.8104 - val_loss: 0.4423 - val_accuracy: 0.7880
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3566 -
accuracy: 0.8416 - val_loss: 0.4832 - val_accuracy: 0.7860
```

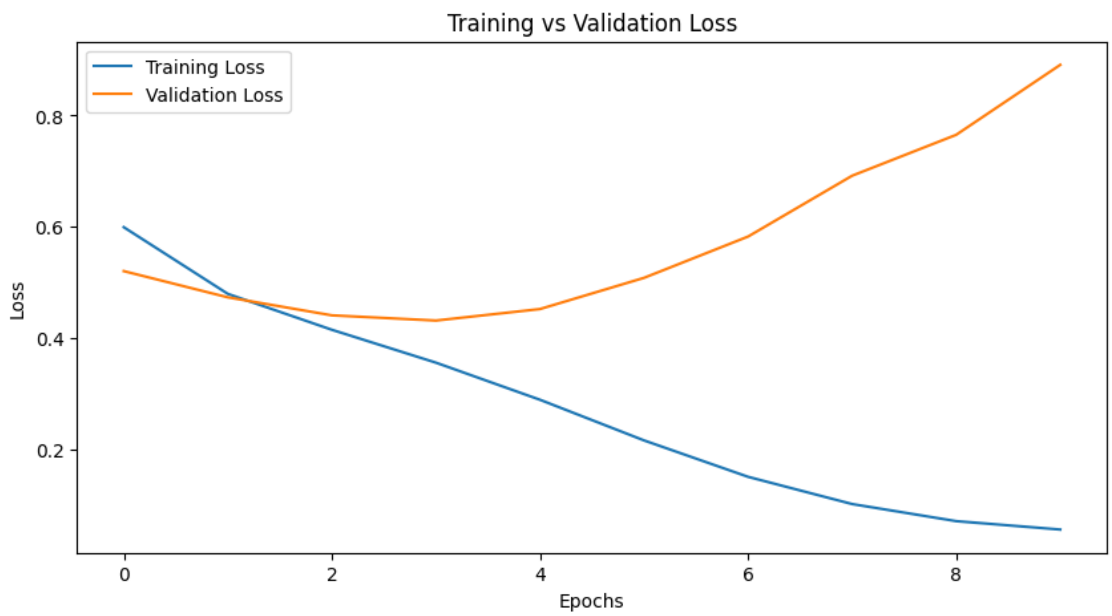
```
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2846 -
accuracy: 0.8778 - val_loss: 0.4788 - val_accuracy: 0.8048
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1985 -
accuracy: 0.9182 - val_loss: 0.5224 - val_accuracy: 0.8152
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1251 -
accuracy: 0.9528 - val_loss: 0.7351 - val_accuracy: 0.7800
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0783 -
accuracy: 0.9718 - val_loss: 0.7164 - val_accuracy: 0.8020
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0468 -
accuracy: 0.9842 - val_loss: 0.7473 - val_accuracy: 0.8100
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0432 -
accuracy: 0.9860 - val_loss: 0.9409 - val_accuracy: 0.7956
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 6…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.6499 -
accuracy: 0.6175 - val_loss: 0.5716 - val_accuracy: 0.7112
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.5184 -
accuracy: 0.7475 - val_loss: 0.4878 - val_accuracy: 0.7576
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4299 -
accuracy: 0.7988 - val_loss: 0.4716 - val_accuracy: 0.7808
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3507 -
accuracy: 0.8466 - val_loss: 0.4389 - val_accuracy: 0.7932
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2521 -
accuracy: 0.8940 - val_loss: 0.4550 - val_accuracy: 0.8112
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1422 -
accuracy: 0.9444 - val_loss: 0.5378 - val_accuracy: 0.8068
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0713 -
accuracy: 0.9754 - val_loss: 0.6819 - val_accuracy: 0.8064
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0363 -
accuracy: 0.9884 - val_loss: 0.7943 - val_accuracy: 0.7972
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0236 -
```

```
accuracy: 0.9932 - val_loss: 0.8553 - val_accuracy: 0.7928
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0299 -
accuracy: 0.9909 - val_loss: 1.0413 - val_accuracy: 0.7828
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 7…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.5952 -
accuracy: 0.6662 - val_loss: 0.4835 - val_accuracy: 0.7716
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4715 -
accuracy: 0.7750 - val_loss: 0.4388 - val_accuracy: 0.7956
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4084 -
accuracy: 0.8121 - val_loss: 0.4251 - val_accuracy: 0.8036
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3514 -
accuracy: 0.8410 - val_loss: 0.4275 - val_accuracy: 0.8044
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2905 -
accuracy: 0.8749 - val_loss: 0.4412 - val_accuracy: 0.8060
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2148 -
accuracy: 0.9084 - val_loss: 0.5060 - val_accuracy: 0.7972
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1424 -
accuracy: 0.9447 - val_loss: 0.5522 - val_accuracy: 0.8036
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0855 -
accuracy: 0.9687 - val_loss: 0.8098 - val_accuracy: 0.7908
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0530 -
accuracy: 0.9824 - val_loss: 0.8126 - val_accuracy: 0.7920
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0442 -
accuracy: 0.9858 - val_loss: 0.9574 - val_accuracy: 0.7888
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 8…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.5761 -
accuracy: 0.6899 - val_loss: 0.4766 - val_accuracy: 0.7756
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4641 -
accuracy: 0.7814 - val_loss: 0.4516 - val_accuracy: 0.7844
```

```
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4043 -
accuracy: 0.8153 - val_loss: 0.4304 - val_accuracy: 0.8008
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3439 -
accuracy: 0.8489 - val_loss: 0.4128 - val_accuracy: 0.8036
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2746 -
accuracy: 0.8820 - val_loss: 0.4409 - val_accuracy: 0.8212
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2050 -
accuracy: 0.9160 - val_loss: 0.4891 - val_accuracy: 0.8064
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1356 -
accuracy: 0.9471 - val_loss: 0.5606 - val_accuracy: 0.8076
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0775 -
accuracy: 0.9721 - val_loss: 0.6816 - val_accuracy: 0.8072
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0570 -
accuracy: 0.9810 - val_loss: 0.8747 - val_accuracy: 0.7948
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0425 -
accuracy: 0.9861 - val_loss: 0.8509 - val_accuracy: 0.8008
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 9…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.5913 -
accuracy: 0.6714 - val_loss: 0.5178 - val_accuracy: 0.7432
Epoch 2/10
704/704 [==============================] - 2s 2ms/step - loss: 0.4620 -
accuracy: 0.7793 - val_loss: 0.6009 - val_accuracy: 0.7140
Epoch 3/10
704/704 [==============================] - 2s 2ms/step - loss: 0.4033 -
accuracy: 0.8144 - val_loss: 0.4476 - val_accuracy: 0.7904
Epoch 4/10
704/704 [==============================] - 2s 2ms/step - loss: 0.3516 -
accuracy: 0.8442 - val_loss: 0.4462 - val_accuracy: 0.7956
Epoch 5/10
704/704 [==============================] - 2s 2ms/step - loss: 0.3024 -
accuracy: 0.8667 - val_loss: 0.4652 - val_accuracy: 0.7828
Epoch 6/10
704/704 [==============================] - 2s 2ms/step - loss: 0.2492 -
accuracy: 0.8948 - val_loss: 0.4769 - val_accuracy: 0.8108
Epoch 7/10
704/704 [==============================] - 2s 2ms/step - loss: 0.1960 -
```

```
accuracy: 0.9188 - val_loss: 0.5178 - val_accuracy: 0.7932
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1449 -
accuracy: 0.9425 - val_loss: 0.6161 - val_accuracy: 0.8012
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0981 -
accuracy: 0.9633 - val_loss: 0.6995 - val_accuracy: 0.7832
Epoch 10/10
704/704 [==============================] - 2s 2ms/step - loss: 0.0750 -
accuracy: 0.9732 - val_loss: 0.8105 - val_accuracy: 0.7848
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 10…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.6103 -
accuracy: 0.6543 - val_loss: 0.5174 - val_accuracy: 0.7480
Epoch 2/10
704/704 [==============================] - 2s 2ms/step - loss: 0.4874 -
accuracy: 0.7638 - val_loss: 0.4544 - val_accuracy: 0.7916
Epoch 3/10
704/704 [==============================] - 2s 2ms/step - loss: 0.4194 -
accuracy: 0.8053 - val_loss: 0.4546 - val_accuracy: 0.7904
Epoch 4/10
704/704 [==============================] - 2s 2ms/step - loss: 0.3489 -
accuracy: 0.8414 - val_loss: 0.4287 - val_accuracy: 0.8128
Epoch 5/10
704/704 [==============================] - 2s 2ms/step - loss: 0.2619 -
accuracy: 0.8876 - val_loss: 0.5055 - val_accuracy: 0.7988
Epoch 6/10
704/704 [==============================] - 2s 2ms/step - loss: 0.1688 -
accuracy: 0.9313 - val_loss: 0.6126 - val_accuracy: 0.7992
Epoch 7/10
704/704 [==============================] - 2s 2ms/step - loss: 0.0900 -
accuracy: 0.9674 - val_loss: 0.7873 - val_accuracy: 0.7868
Epoch 8/10
704/704 [==============================] - 2s 2ms/step - loss: 0.0527 -
accuracy: 0.9820 - val_loss: 0.9198 - val_accuracy: 0.7876
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.0366 -
accuracy: 0.9887 - val_loss: 1.0130 - val_accuracy: 0.7828
Epoch 10/10
704/704 [==============================] - 2s 2ms/step - loss: 0.0296 -
accuracy: 0.9905 - val_loss: 1.1448 - val_accuracy: 0.7956
79/79 [==============================] - 0s 1ms/step

Results
AUC: 0.878 ± 0.007
```

```
Precision: 0.793 ± 0.034
Recall: 0.793 ± 0.051
F1-score: 0.791 ± 0.011
```



[21]:
```
# Entrenar usando el primer modelo CNN
train_cnn_model(create_cnn_model_2)
```

```
---------------------------------------------

Training Fold 1…
Epoch 1/10
704/704 [==============================] - 4s 3ms/step - loss: 0.6203 -
accuracy: 0.6448 - val_loss: 0.5310 - val_accuracy: 0.7452
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4768 -
accuracy: 0.7719 - val_loss: 0.4297 - val_accuracy: 0.8048
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3979 -
accuracy: 0.8197 - val_loss: 0.4348 - val_accuracy: 0.7916
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3480 -
accuracy: 0.8468 - val_loss: 0.3665 - val_accuracy: 0.8388
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2956 -
accuracy: 0.8719 - val_loss: 0.3589 - val_accuracy: 0.8308
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2529 -
```

```
accuracy: 0.8934 - val_loss: 0.3431 - val_accuracy: 0.8520
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2104 -
accuracy: 0.9129 - val_loss: 0.3563 - val_accuracy: 0.8536
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1735 -
accuracy: 0.9275 - val_loss: 0.3952 - val_accuracy: 0.8456
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1414 -
accuracy: 0.9426 - val_loss: 0.4548 - val_accuracy: 0.8548
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1165 -
accuracy: 0.9532 - val_loss: 0.5003 - val_accuracy: 0.8416
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 2…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.6162 -
accuracy: 0.6541 - val_loss: 0.5512 - val_accuracy: 0.7180
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4847 -
accuracy: 0.7708 - val_loss: 0.4668 - val_accuracy: 0.7812
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4205 -
accuracy: 0.8093 - val_loss: 0.3998 - val_accuracy: 0.8228
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3738 -
accuracy: 0.8319 - val_loss: 0.3619 - val_accuracy: 0.8372
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3203 -
accuracy: 0.8617 - val_loss: 0.3665 - val_accuracy: 0.8392
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2809 -
accuracy: 0.8804 - val_loss: 0.3622 - val_accuracy: 0.8500
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2388 -
accuracy: 0.9001 - val_loss: 0.4069 - val_accuracy: 0.8328
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1938 -
accuracy: 0.9202 - val_loss: 0.3646 - val_accuracy: 0.8640
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1646 -
accuracy: 0.9334 - val_loss: 0.4133 - val_accuracy: 0.8444
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1290 -
accuracy: 0.9492 - val_loss: 0.4262 - val_accuracy: 0.8628
79/79 [==============================] - 0s 1ms/step
```

```
--------------------------------------------

Training Fold 3…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.5918 -
accuracy: 0.6745 - val_loss: 0.4895 - val_accuracy: 0.7652
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4659 -
accuracy: 0.7807 - val_loss: 0.4212 - val_accuracy: 0.8072
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3950 -
accuracy: 0.8195 - val_loss: 0.3711 - val_accuracy: 0.8308
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3423 -
accuracy: 0.8469 - val_loss: 0.3485 - val_accuracy: 0.8496
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2992 -
accuracy: 0.8710 - val_loss: 0.3368 - val_accuracy: 0.8496
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2574 -
accuracy: 0.8907 - val_loss: 0.3478 - val_accuracy: 0.8456
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2240 -
accuracy: 0.9046 - val_loss: 0.3379 - val_accuracy: 0.8592
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1879 -
accuracy: 0.9233 - val_loss: 0.3659 - val_accuracy: 0.8612
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1562 -
accuracy: 0.9370 - val_loss: 0.4186 - val_accuracy: 0.8448
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1294 -
accuracy: 0.9483 - val_loss: 0.4938 - val_accuracy: 0.8528
79/79 [==============================] - 0s 1ms/step
--------------------------------------------

Training Fold 4…
Epoch 1/10
704/704 [==============================] - 4s 3ms/step - loss: 0.6145 -
accuracy: 0.6485 - val_loss: 0.5241 - val_accuracy: 0.7340
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4849 -
accuracy: 0.7691 - val_loss: 0.4640 - val_accuracy: 0.7808
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4099 -
accuracy: 0.8150 - val_loss: 0.4393 - val_accuracy: 0.7892
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3603 -
```

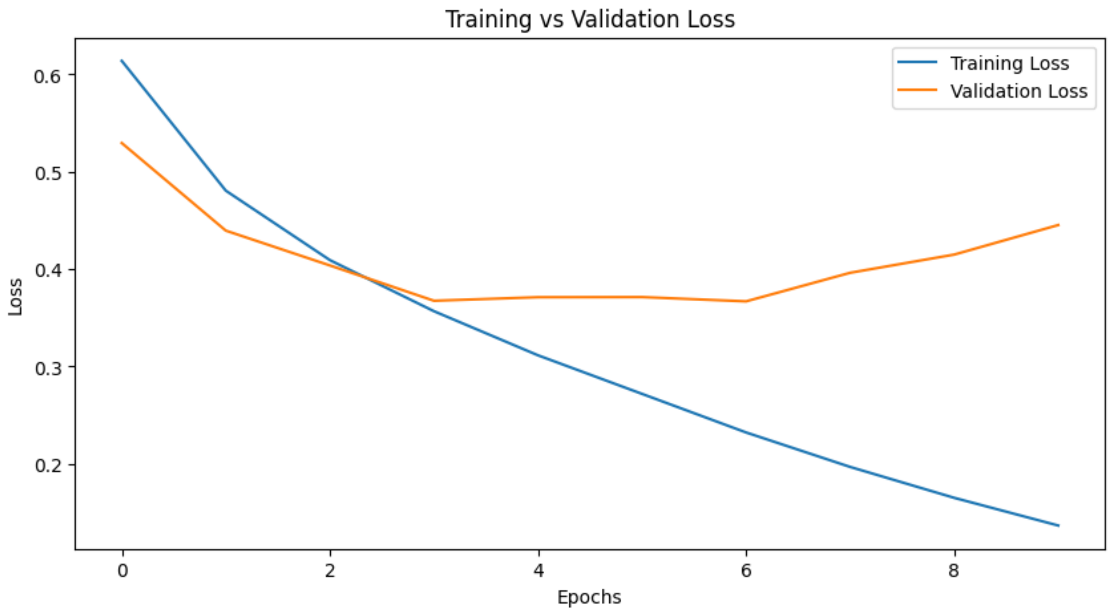
```
accuracy: 0.8420 - val_loss: 0.3857 - val_accuracy: 0.8244
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3176 -
accuracy: 0.8588 - val_loss: 0.3748 - val_accuracy: 0.8224
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2840 -
accuracy: 0.8762 - val_loss: 0.4060 - val_accuracy: 0.8176
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2421 -
accuracy: 0.8970 - val_loss: 0.4032 - val_accuracy: 0.8324
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2137 -
accuracy: 0.9113 - val_loss: 0.4440 - val_accuracy: 0.8224
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1833 -
accuracy: 0.9256 - val_loss: 0.4832 - val_accuracy: 0.8264
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1547 -
accuracy: 0.9392 - val_loss: 0.4581 - val_accuracy: 0.8452
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 5…
Epoch 1/10
704/704 [==============================] - 4s 4ms/step - loss: 0.5997 -
accuracy: 0.6687 - val_loss: 0.5263 - val_accuracy: 0.7232
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4710 -
accuracy: 0.7756 - val_loss: 0.4218 - val_accuracy: 0.8000
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4074 -
accuracy: 0.8132 - val_loss: 0.3644 - val_accuracy: 0.8324
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3569 -
accuracy: 0.8425 - val_loss: 0.3525 - val_accuracy: 0.8504
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3134 -
accuracy: 0.8611 - val_loss: 0.3329 - val_accuracy: 0.8508
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2760 -
accuracy: 0.8814 - val_loss: 0.3878 - val_accuracy: 0.8400
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2354 -
accuracy: 0.9023 - val_loss: 0.3580 - val_accuracy: 0.8564
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2074 -
accuracy: 0.9163 - val_loss: 0.3580 - val_accuracy: 0.8636
Epoch 9/10
```

```
704/704 [==============================] - 2s 3ms/step - loss: 0.1721 -
accuracy: 0.9279 - val_loss: 0.3720 - val_accuracy: 0.8640
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1443 -
accuracy: 0.9407 - val_loss: 0.4144 - val_accuracy: 0.8652
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 6…
Epoch 1/10
704/704 [==============================] - 4s 3ms/step - loss: 0.6082 -
accuracy: 0.6568 - val_loss: 0.4755 - val_accuracy: 0.7748
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4731 -
accuracy: 0.7758 - val_loss: 0.4055 - val_accuracy: 0.8132
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4091 -
accuracy: 0.8138 - val_loss: 0.3814 - val_accuracy: 0.8212
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3604 -
accuracy: 0.8386 - val_loss: 0.3429 - val_accuracy: 0.8520
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3232 -
accuracy: 0.8584 - val_loss: 0.3452 - val_accuracy: 0.8448
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2839 -
accuracy: 0.8765 - val_loss: 0.3262 - val_accuracy: 0.8528
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2461 -
accuracy: 0.8949 - val_loss: 0.3173 - val_accuracy: 0.8632
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2044 -
accuracy: 0.9160 - val_loss: 0.3242 - val_accuracy: 0.8684
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1746 -
accuracy: 0.9285 - val_loss: 0.3719 - val_accuracy: 0.8648
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1439 -
accuracy: 0.9409 - val_loss: 0.3850 - val_accuracy: 0.8652
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 7…
Epoch 1/10
704/704 [==============================] - 4s 3ms/step - loss: 0.6188 -
accuracy: 0.6412 - val_loss: 0.5517 - val_accuracy: 0.7148
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4826 -
```

```
accuracy: 0.7688 - val_loss: 0.4159 - val_accuracy: 0.8096
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4045 -
accuracy: 0.8156 - val_loss: 0.4390 - val_accuracy: 0.7948
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3457 -
accuracy: 0.8498 - val_loss: 0.3860 - val_accuracy: 0.8248
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2983 -
accuracy: 0.8718 - val_loss: 0.3860 - val_accuracy: 0.8232
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2515 -
accuracy: 0.8937 - val_loss: 0.3935 - val_accuracy: 0.8468
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2109 -
accuracy: 0.9118 - val_loss: 0.3631 - val_accuracy: 0.8548
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1716 -
accuracy: 0.9308 - val_loss: 0.4905 - val_accuracy: 0.8432
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1417 -
accuracy: 0.9432 - val_loss: 0.4348 - val_accuracy: 0.8412
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1156 -
accuracy: 0.9546 - val_loss: 0.4873 - val_accuracy: 0.8560
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 8…
Epoch 1/10
704/704 [==============================] - 4s 3ms/step - loss: 0.6484 -
accuracy: 0.6047 - val_loss: 0.5731 - val_accuracy: 0.7024
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.5080 -
accuracy: 0.7480 - val_loss: 0.4573 - val_accuracy: 0.7852
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4378 -
accuracy: 0.7978 - val_loss: 0.4163 - val_accuracy: 0.8100
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3744 -
accuracy: 0.8319 - val_loss: 0.3845 - val_accuracy: 0.8276
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3359 -
accuracy: 0.8509 - val_loss: 0.3723 - val_accuracy: 0.8352
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2941 -
accuracy: 0.8725 - val_loss: 0.3794 - val_accuracy: 0.8272
Epoch 7/10
```

```
704/704 [==============================] - 2s 3ms/step - loss: 0.2616 -
accuracy: 0.8894 - val_loss: 0.3843 - val_accuracy: 0.8336
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2227 -
accuracy: 0.9070 - val_loss: 0.3589 - val_accuracy: 0.8428
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1904 -
accuracy: 0.9209 - val_loss: 0.3624 - val_accuracy: 0.8476
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1632 -
accuracy: 0.9339 - val_loss: 0.3785 - val_accuracy: 0.8496
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 9…
Epoch 1/10
704/704 [==============================] - 3s 3ms/step - loss: 0.5958 -
accuracy: 0.6664 - val_loss: 0.5234 - val_accuracy: 0.7320
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4603 -
accuracy: 0.7828 - val_loss: 0.4744 - val_accuracy: 0.7748
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3844 -
accuracy: 0.8279 - val_loss: 0.4004 - val_accuracy: 0.8140
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3332 -
accuracy: 0.8548 - val_loss: 0.3681 - val_accuracy: 0.8404
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2895 -
accuracy: 0.8763 - val_loss: 0.3955 - val_accuracy: 0.8288
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2495 -
accuracy: 0.8948 - val_loss: 0.3880 - val_accuracy: 0.8436
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2105 -
accuracy: 0.9135 - val_loss: 0.3692 - val_accuracy: 0.8620
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1820 -
accuracy: 0.9265 - val_loss: 0.4004 - val_accuracy: 0.8512
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1524 -
accuracy: 0.9374 - val_loss: 0.4601 - val_accuracy: 0.8580
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1217 -
accuracy: 0.9509 - val_loss: 0.4630 - val_accuracy: 0.8492
79/79 [==============================] - 0s 1ms/step
---------------------------------------------
```

```
Training Fold 10…
Epoch 1/10
704/704 [==============================] - 4s 3ms/step - loss: 0.6259 -
accuracy: 0.6397 - val_loss: 0.5499 - val_accuracy: 0.7232
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4993 -
accuracy: 0.7574 - val_loss: 0.4393 - val_accuracy: 0.7908
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4269 -
accuracy: 0.8048 - val_loss: 0.3921 - val_accuracy: 0.8328
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3738 -
accuracy: 0.8324 - val_loss: 0.3792 - val_accuracy: 0.8292
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3226 -
accuracy: 0.8602 - val_loss: 0.4434 - val_accuracy: 0.8116
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2893 -
accuracy: 0.8745 - val_loss: 0.3795 - val_accuracy: 0.8488
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2435 -
accuracy: 0.8976 - val_loss: 0.3735 - val_accuracy: 0.8628
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2124 -
accuracy: 0.9118 - val_loss: 0.4612 - val_accuracy: 0.8316
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1753 -
accuracy: 0.9256 - val_loss: 0.3788 - val_accuracy: 0.8584
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1490 -
accuracy: 0.9402 - val_loss: 0.4470 - val_accuracy: 0.8536
79/79 [==============================] - 0s 1ms/step

Results
AUC: 0.932 ± 0.005
Precision: 0.855 ± 0.022
Recall: 0.855 ± 0.038
F1-score: 0.854 ± 0.011
```



[22]:
```
# Entrenar usando el primer modelo CNN
train_cnn_model(create_cnn_model_3)
```

```
---------------------------------------------

Training Fold 1…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 0.6393 -
accuracy: 0.6169 - val_loss: 0.5429 - val_accuracy: 0.7152
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4943 -
accuracy: 0.7620 - val_loss: 0.4468 - val_accuracy: 0.7812
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4172 -
accuracy: 0.8104 - val_loss: 0.3885 - val_accuracy: 0.8216
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3524 -
accuracy: 0.8428 - val_loss: 0.3757 - val_accuracy: 0.8344
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3090 -
accuracy: 0.8663 - val_loss: 0.3816 - val_accuracy: 0.8328
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2662 -
accuracy: 0.8857 - val_loss: 0.4133 - val_accuracy: 0.8436
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2224 -
accuracy: 0.9049 - val_loss: 0.3852 - val_accuracy: 0.8452
```

```
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1847 -
accuracy: 0.9234 - val_loss: 0.4076 - val_accuracy: 0.8496
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1548 -
accuracy: 0.9362 - val_loss: 0.4095 - val_accuracy: 0.8624
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1169 -
accuracy: 0.9539 - val_loss: 0.5199 - val_accuracy: 0.8432
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 2…
Epoch 1/10
704/704 [==============================] - 4s 4ms/step - loss: 0.6089 -
accuracy: 0.6548 - val_loss: 0.5048 - val_accuracy: 0.7612
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4769 -
accuracy: 0.7691 - val_loss: 0.4230 - val_accuracy: 0.7968
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3977 -
accuracy: 0.8176 - val_loss: 0.3878 - val_accuracy: 0.8256
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3494 -
accuracy: 0.8439 - val_loss: 0.3389 - val_accuracy: 0.8516
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3004 -
accuracy: 0.8683 - val_loss: 0.3292 - val_accuracy: 0.8528
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2517 -
accuracy: 0.8920 - val_loss: 0.3361 - val_accuracy: 0.8584
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2089 -
accuracy: 0.9110 - val_loss: 0.3502 - val_accuracy: 0.8648
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1747 -
accuracy: 0.9283 - val_loss: 0.3815 - val_accuracy: 0.8576
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1430 -
accuracy: 0.9415 - val_loss: 0.3972 - val_accuracy: 0.8604
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1102 -
accuracy: 0.9564 - val_loss: 0.4618 - val_accuracy: 0.8552
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 3…
Epoch 1/10
```

```
704/704 [==============================] - 4s 4ms/step - loss: 0.6186 -
accuracy: 0.6471 - val_loss: 0.5199 - val_accuracy: 0.7464
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4770 -
accuracy: 0.7725 - val_loss: 0.4170 - val_accuracy: 0.8044
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4021 -
accuracy: 0.8172 - val_loss: 0.3539 - val_accuracy: 0.8344
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3453 -
accuracy: 0.8477 - val_loss: 0.3353 - val_accuracy: 0.8528
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2983 -
accuracy: 0.8728 - val_loss: 0.3404 - val_accuracy: 0.8456
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2496 -
accuracy: 0.8932 - val_loss: 0.3349 - val_accuracy: 0.8592
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2053 -
accuracy: 0.9151 - val_loss: 0.3629 - val_accuracy: 0.8588
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1692 -
accuracy: 0.9306 - val_loss: 0.3758 - val_accuracy: 0.8572
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1321 -
accuracy: 0.9476 - val_loss: 0.3961 - val_accuracy: 0.8544
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1036 -
accuracy: 0.9604 - val_loss: 0.4704 - val_accuracy: 0.8616
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 4…
Epoch 1/10
704/704 [==============================] - 4s 4ms/step - loss: 0.6286 -
accuracy: 0.6370 - val_loss: 0.5359 - val_accuracy: 0.7292
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4836 -
accuracy: 0.7679 - val_loss: 0.4674 - val_accuracy: 0.7732
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4073 -
accuracy: 0.8144 - val_loss: 0.4038 - val_accuracy: 0.8120
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3470 -
accuracy: 0.8478 - val_loss: 0.4142 - val_accuracy: 0.8048
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2990 -
accuracy: 0.8724 - val_loss: 0.3566 - val_accuracy: 0.8344
```

```
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2568 -
accuracy: 0.8918 - val_loss: 0.3588 - val_accuracy: 0.8484
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2061 -
accuracy: 0.9140 - val_loss: 0.3870 - val_accuracy: 0.8508
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1649 -
accuracy: 0.9329 - val_loss: 0.4407 - val_accuracy: 0.8436
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1308 -
accuracy: 0.9479 - val_loss: 0.4886 - val_accuracy: 0.8348
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1063 -
accuracy: 0.9588 - val_loss: 0.5119 - val_accuracy: 0.8284
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 5…
Epoch 1/10
704/704 [==============================] - 4s 4ms/step - loss: 0.6820 -
accuracy: 0.5521 - val_loss: 0.6184 - val_accuracy: 0.6612
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.5332 -
accuracy: 0.7357 - val_loss: 0.4433 - val_accuracy: 0.7920
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4217 -
accuracy: 0.8051 - val_loss: 0.4587 - val_accuracy: 0.7860
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3596 -
accuracy: 0.8378 - val_loss: 0.3635 - val_accuracy: 0.8344
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3110 -
accuracy: 0.8639 - val_loss: 0.3759 - val_accuracy: 0.8312
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2657 -
accuracy: 0.8883 - val_loss: 0.3308 - val_accuracy: 0.8616
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2234 -
accuracy: 0.9062 - val_loss: 0.3609 - val_accuracy: 0.8572
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1832 -
accuracy: 0.9253 - val_loss: 0.4308 - val_accuracy: 0.8408
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1544 -
accuracy: 0.9368 - val_loss: 0.4468 - val_accuracy: 0.8444
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1166 -
```

```
accuracy: 0.9536 - val_loss: 0.4439 - val_accuracy: 0.8540
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 6…
Epoch 1/10
704/704 [==============================] - 4s 4ms/step - loss: 0.6834 -
accuracy: 0.5485 - val_loss: 0.6086 - val_accuracy: 0.6688
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.5677 -
accuracy: 0.7048 - val_loss: 0.4560 - val_accuracy: 0.7824
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4542 -
accuracy: 0.7869 - val_loss: 0.4063 - val_accuracy: 0.8148
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3928 -
accuracy: 0.8220 - val_loss: 0.3933 - val_accuracy: 0.8196
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3469 -
accuracy: 0.8458 - val_loss: 0.3226 - val_accuracy: 0.8632
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3016 -
accuracy: 0.8679 - val_loss: 0.3119 - val_accuracy: 0.8752
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2626 -
accuracy: 0.8847 - val_loss: 0.3194 - val_accuracy: 0.8772
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2308 -
accuracy: 0.9013 - val_loss: 0.3285 - val_accuracy: 0.8592
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1950 -
accuracy: 0.9207 - val_loss: 0.4127 - val_accuracy: 0.8540
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1646 -
accuracy: 0.9337 - val_loss: 0.3953 - val_accuracy: 0.8664
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 7…
Epoch 1/10
704/704 [==============================] - 4s 4ms/step - loss: 0.6100 -
accuracy: 0.6550 - val_loss: 0.5470 - val_accuracy: 0.7184
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4802 -
accuracy: 0.7680 - val_loss: 0.4840 - val_accuracy: 0.7752
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4076 -
accuracy: 0.8160 - val_loss: 0.4063 - val_accuracy: 0.8172
```

```
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3534 -
accuracy: 0.8440 - val_loss: 0.3857 - val_accuracy: 0.8276
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3115 -
accuracy: 0.8626 - val_loss: 0.3933 - val_accuracy: 0.8268
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2704 -
accuracy: 0.8824 - val_loss: 0.3444 - val_accuracy: 0.8576
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2295 -
accuracy: 0.9040 - val_loss: 0.3739 - val_accuracy: 0.8372
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1943 -
accuracy: 0.9192 - val_loss: 0.3926 - val_accuracy: 0.8484
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1577 -
accuracy: 0.9351 - val_loss: 0.3793 - val_accuracy: 0.8528
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1384 -
accuracy: 0.9445 - val_loss: 0.4602 - val_accuracy: 0.8496
79/79 [==============================] - 0s 1ms/step
---------------------------------------------

Training Fold 8…
Epoch 1/10
704/704 [==============================] - 4s 4ms/step - loss: 0.6092 -
accuracy: 0.6571 - val_loss: 0.4932 - val_accuracy: 0.7616
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4732 -
accuracy: 0.7760 - val_loss: 0.4777 - val_accuracy: 0.7676
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3980 -
accuracy: 0.8220 - val_loss: 0.3979 - val_accuracy: 0.8088
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3450 -
accuracy: 0.8470 - val_loss: 0.3801 - val_accuracy: 0.8276
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2999 -
accuracy: 0.8701 - val_loss: 0.3835 - val_accuracy: 0.8368
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2540 -
accuracy: 0.8926 - val_loss: 0.3759 - val_accuracy: 0.8344
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2133 -
accuracy: 0.9110 - val_loss: 0.3731 - val_accuracy: 0.8508
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1756 -
```

```
accuracy: 0.9279 - val_loss: 0.3878 - val_accuracy: 0.8600
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1419 -
accuracy: 0.9432 - val_loss: 0.4307 - val_accuracy: 0.8540
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1173 -
accuracy: 0.9535 - val_loss: 0.4468 - val_accuracy: 0.8492
79/79 [==============================] - 0s 1ms/step
--------------------------------------------

Training Fold 9…
Epoch 1/10
704/704 [==============================] - 4s 4ms/step - loss: 0.6229 -
accuracy: 0.6405 - val_loss: 0.5753 - val_accuracy: 0.7176
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4902 -
accuracy: 0.7634 - val_loss: 0.4713 - val_accuracy: 0.7804
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4196 -
accuracy: 0.8094 - val_loss: 0.4312 - val_accuracy: 0.7980
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3613 -
accuracy: 0.8412 - val_loss: 0.4002 - val_accuracy: 0.8232
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3087 -
accuracy: 0.8660 - val_loss: 0.4414 - val_accuracy: 0.8160
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2677 -
accuracy: 0.8853 - val_loss: 0.3548 - val_accuracy: 0.8556
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2314 -
accuracy: 0.9035 - val_loss: 0.3761 - val_accuracy: 0.8512
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1928 -
accuracy: 0.9220 - val_loss: 0.4007 - val_accuracy: 0.8496
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1600 -
accuracy: 0.9349 - val_loss: 0.3725 - val_accuracy: 0.8640
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1330 -
accuracy: 0.9476 - val_loss: 0.4274 - val_accuracy: 0.8600
79/79 [==============================] - 0s 1ms/step
--------------------------------------------

Training Fold 10…
Epoch 1/10
704/704 [==============================] - 4s 4ms/step - loss: 0.6554 -
accuracy: 0.6064 - val_loss: 0.5840 - val_accuracy: 0.6836
```

```
Epoch 2/10
704/704 [==============================] - 2s 3ms/step - loss: 0.5160 -
accuracy: 0.7479 - val_loss: 0.4579 - val_accuracy: 0.7836
Epoch 3/10
704/704 [==============================] - 2s 3ms/step - loss: 0.4215 -
accuracy: 0.8047 - val_loss: 0.3912 - val_accuracy: 0.8296
Epoch 4/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3617 -
accuracy: 0.8359 - val_loss: 0.3544 - val_accuracy: 0.8372
Epoch 5/10
704/704 [==============================] - 2s 3ms/step - loss: 0.3063 -
accuracy: 0.8655 - val_loss: 0.3473 - val_accuracy: 0.8580
Epoch 6/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2643 -
accuracy: 0.8870 - val_loss: 0.3852 - val_accuracy: 0.8368
Epoch 7/10
704/704 [==============================] - 2s 3ms/step - loss: 0.2195 -
accuracy: 0.9092 - val_loss: 0.3786 - val_accuracy: 0.8564
Epoch 8/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1860 -
accuracy: 0.9234 - val_loss: 0.3614 - val_accuracy: 0.8556
Epoch 9/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1468 -
accuracy: 0.9424 - val_loss: 0.4202 - val_accuracy: 0.8624
Epoch 10/10
704/704 [==============================] - 2s 3ms/step - loss: 0.1231 -
accuracy: 0.9507 - val_loss: 0.4602 - val_accuracy: 0.8716
79/79 [==============================] - 0s 1ms/step

Results
AUC: 0.933 ± 0.007
Precision: 0.852 ± 0.028
Recall: 0.859 ± 0.048
F1-score: 0.854 ± 0.016
```

2.0.6 Incluyendo Callbacks y Data Augmentation

[23]:
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,␣
↪Dropout, BatchNormalization, Input
from tensorflow.keras.regularizers import l2
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import roc_auc_score, precision_score, recall_score,␣
↪f1_score
import matplotlib.pyplot as plt
import numpy as np
import os

# Crear las etiquetas (1 para perro, 0 para gato)
y = np.array([1 if 'dog' in img else 0 for img in os.listdir(train_folder)])

# Crear un generador de datos con data augmentation
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    zoom_range=0.2,
```

```
    horizontal_flip=True
)

# Callback para reducir el learning rate cuando la métrica no mejora
lr_scheduler = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3,␣
 ↪min_lr=1e-5)

# Arquitectura 1: Modelo básico ajustado para 3 canales (RGB) usando Input
def create_cnn_model_1():
    model = Sequential([
        Input(shape=(64, 64, 3)),  # Definir la entrada explícitamente con Input
        Conv2D(32, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),
        Conv2D(64, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),
        Flatten(),
        Dense(128, activation='relu', kernel_regularizer=l2(0.001)),
        Dense(2, activation='softmax')
    ])
    return model

# Arquitectura 2: Modelo con más capas ajustado para 3 canales (RGB) usando␣
 ↪Input
def create_cnn_model_2():
    model = Sequential([
        Input(shape=(64, 64, 3)),  # Definir la entrada explícitamente con Input
        Conv2D(32, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),
        Conv2D(64, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),
        Conv2D(128, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),
        Flatten(),
        Dense(256, activation='relu', kernel_regularizer=l2(0.001)),
        Dropout(0.5),
        Dense(2, activation='softmax')
    ])
    return model

# Arquitectura 3: Más densa con Dropout ajustado para 3 canales (RGB) usando␣
 ↪Input
def create_cnn_model_3():
```

```python
    model = Sequential([
        Input(shape=(64, 64, 3)),  # Definir la entrada explícitamente con Input
        Conv2D(64, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),
        Conv2D(128, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),
        Conv2D(256, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),
        Flatten(),
        Dense(512, activation='relu', kernel_regularizer=l2(0.001)),
        Dropout(0.6),  # Aumentado el dropout para mayor regularización
        Dense(2, activation='softmax')
    ])
    return model

# Convertir etiquetas a one-hot encoding para CNN
y_categorical = to_categorical(y, num_classes=2)

# Definir StratifiedKFold
skf = StratifiedKFold(n_splits=10)

# Función para entrenar y mostrar métricas
def train_cnn_model(model_fn):
    fold = 0
    training_losses = []
    validation_losses = []

    # Listas para almacenar las métricas de cada fold
    auc_scores = []
    precision_scores = []
    recall_scores = []
    f1_scores = []

    for train_index, test_index in skf.split(train_cnn, y):
        fold += 1
        print(f"Training Fold {fold}...")

        X_train, X_test = train_cnn[train_index], train_cnn[test_index]
        y_train, y_test = y_categorical[train_index], y_categorical[test_index]

        # Crear el modelo
        model = model_fn()
        model.compile(optimizer='adam', loss='categorical_crossentropy',␣
↪metrics=['accuracy'])
```

```python
        # Aplicar data augmentation solo al conjunto de entrenamiento
        datagen.fit(X_train)
        history = model.fit(datagen.flow(X_train, y_train, batch_size=32),
                            validation_data=(X_test, y_test), epochs=10,␣
↪callbacks=[lr_scheduler])

        # Guardar pérdidas
        training_losses.append(history.history['loss'])
        validation_losses.append(history.history['val_loss'])

        # Hacer predicciones para las métricas
        y_pred = model.predict(X_test)
        y_pred_class = y_pred.argmax(axis=1)
        y_true = y_test.argmax(axis=1)

        # Calcular las métricas
        auc = roc_auc_score(y_true, y_pred[:, 1])
        precision = precision_score(y_true, y_pred_class)
        recall = recall_score(y_true, y_pred_class)
        f1 = f1_score(y_true, y_pred_class)

        # Guardar métricas
        auc_scores.append(auc)
        precision_scores.append(precision)
        recall_scores.append(recall)
        f1_scores.append(f1)

    # Mostrar los resultados promedio y desviación estándar
    print(f"\nResults")
    print(f"AUC: {np.mean(auc_scores):.3f} ± {np.std(auc_scores):.3f}")
    print(f"Precision: {np.mean(precision_scores):.3f} ± {np.
↪std(precision_scores):.3f}")
    print(f"Recall: {np.mean(recall_scores):.3f} ± {np.std(recall_scores):.3f}")
    print(f"F1-score: {np.mean(f1_scores):.3f} ± {np.std(f1_scores):.3f}")

    # Graficar las pérdidas (loss) de entrenamiento y validación
    plt.figure(figsize=(10, 5))
    plt.plot(np.mean(training_losses, axis=0), label="Training Loss")
    plt.plot(np.mean(validation_losses, axis=0), label="Validation Loss")
    plt.title("Training vs Validation Loss")
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()
    plt.show()

# Entrenar usando el primer modelo CNN con data augmentation
```

```
train_cnn_model(create_cnn_model_1)
```

```
Training Fold 1…
Epoch 1/10
704/704 [==============================] - 26s 34ms/step - loss: 0.9396 -
accuracy: 0.6294 - val_loss: 0.7293 - val_accuracy: 0.7036 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 25s 35ms/step - loss: 0.7275 -
accuracy: 0.6823 - val_loss: 0.8603 - val_accuracy: 0.5912 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 25s 36ms/step - loss: 0.6945 -
accuracy: 0.7061 - val_loss: 0.6796 - val_accuracy: 0.7280 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 25s 36ms/step - loss: 0.6439 -
accuracy: 0.7334 - val_loss: 0.7464 - val_accuracy: 0.6820 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 26s 37ms/step - loss: 0.6056 -
accuracy: 0.7428 - val_loss: 0.5713 - val_accuracy: 0.7776 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 25s 36ms/step - loss: 0.6067 -
accuracy: 0.7560 - val_loss: 0.6358 - val_accuracy: 0.7148 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 26s 37ms/step - loss: 0.5710 -
accuracy: 0.7666 - val_loss: 0.6420 - val_accuracy: 0.7040 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 25s 36ms/step - loss: 0.5528 -
accuracy: 0.7698 - val_loss: 0.5434 - val_accuracy: 0.7864 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 25s 35ms/step - loss: 0.5403 -
accuracy: 0.7781 - val_loss: 0.8480 - val_accuracy: 0.5948 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 26s 36ms/step - loss: 0.5289 -
accuracy: 0.7836 - val_loss: 0.6097 - val_accuracy: 0.7544 - lr: 0.0010
79/79 [==============================] - 0s 2ms/step
Training Fold 2…
Epoch 1/10
704/704 [==============================] - 27s 36ms/step - loss: 0.8988 -
accuracy: 0.6536 - val_loss: 0.8198 - val_accuracy: 0.6460 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 26s 36ms/step - loss: 0.7280 -
accuracy: 0.6971 - val_loss: 0.9104 - val_accuracy: 0.6576 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 26s 37ms/step - loss: 0.6678 -
accuracy: 0.7248 - val_loss: 0.6460 - val_accuracy: 0.7092 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 25s 36ms/step - loss: 0.6326 -
accuracy: 0.7393 - val_loss: 0.7439 - val_accuracy: 0.7300 - lr: 0.0010
```

```
Epoch 5/10
704/704 [==============================] - 26s 37ms/step - loss: 0.6193 -
accuracy: 0.7492 - val_loss: 0.5869 - val_accuracy: 0.7800 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 26s 36ms/step - loss: 0.5997 -
accuracy: 0.7545 - val_loss: 0.7241 - val_accuracy: 0.6596 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 24s 34ms/step - loss: 0.5849 -
accuracy: 0.7678 - val_loss: 0.6172 - val_accuracy: 0.7436 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 26s 37ms/step - loss: 0.5647 -
accuracy: 0.7718 - val_loss: 0.8315 - val_accuracy: 0.6536 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 25s 36ms/step - loss: 0.5083 -
accuracy: 0.7942 - val_loss: 0.4831 - val_accuracy: 0.8164 - lr: 5.0000e-04
Epoch 10/10
704/704 [==============================] - 26s 36ms/step - loss: 0.4965 -
accuracy: 0.7958 - val_loss: 0.5028 - val_accuracy: 0.7944 - lr: 5.0000e-04
79/79 [==============================] - 0s 2ms/step
Training Fold 3…
Epoch 1/10
704/704 [==============================] - 27s 36ms/step - loss: 0.9179 -
accuracy: 0.6332 - val_loss: 1.0822 - val_accuracy: 0.5688 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 26s 37ms/step - loss: 0.7216 -
accuracy: 0.6908 - val_loss: 0.9352 - val_accuracy: 0.6188 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 25s 35ms/step - loss: 0.6706 -
accuracy: 0.7171 - val_loss: 0.6242 - val_accuracy: 0.7488 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 26s 37ms/step - loss: 0.6418 -
accuracy: 0.7345 - val_loss: 0.6024 - val_accuracy: 0.7636 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 26s 37ms/step - loss: 0.6138 -
accuracy: 0.7467 - val_loss: 0.7140 - val_accuracy: 0.7176 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 26s 37ms/step - loss: 0.5896 -
accuracy: 0.7571 - val_loss: 0.5792 - val_accuracy: 0.7672 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 25s 36ms/step - loss: 0.5702 -
accuracy: 0.7655 - val_loss: 0.5451 - val_accuracy: 0.7720 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 26s 37ms/step - loss: 0.5606 -
accuracy: 0.7711 - val_loss: 0.7212 - val_accuracy: 0.6884 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 26s 37ms/step - loss: 0.5466 -
accuracy: 0.7781 - val_loss: 0.7223 - val_accuracy: 0.7500 - lr: 0.0010
Epoch 10/10
```

```
704/704 [==============================] - 25s 35ms/step - loss: 0.5320 -
accuracy: 0.7816 - val_loss: 0.6263 - val_accuracy: 0.7316 - lr: 0.0010
79/79 [==============================] - 0s 2ms/step
Training Fold 4…
Epoch 1/10
704/704 [==============================] - 27s 36ms/step - loss: 0.9212 -
accuracy: 0.6363 - val_loss: 0.8968 - val_accuracy: 0.6084 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 24s 35ms/step - loss: 0.7377 -
accuracy: 0.6758 - val_loss: 0.6424 - val_accuracy: 0.7380 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 24s 34ms/step - loss: 0.6681 -
accuracy: 0.7131 - val_loss: 0.6685 - val_accuracy: 0.7252 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 20s 29ms/step - loss: 0.6684 -
accuracy: 0.7323 - val_loss: 0.6341 - val_accuracy: 0.7448 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 25s 35ms/step - loss: 0.6236 -
accuracy: 0.7516 - val_loss: 0.8079 - val_accuracy: 0.6880 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 24s 34ms/step - loss: 0.6037 -
accuracy: 0.7591 - val_loss: 0.7877 - val_accuracy: 0.6560 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 25s 36ms/step - loss: 0.5901 -
accuracy: 0.7666 - val_loss: 0.5937 - val_accuracy: 0.7608 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 24s 34ms/step - loss: 0.5735 -
accuracy: 0.7727 - val_loss: 0.6222 - val_accuracy: 0.7660 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 25s 35ms/step - loss: 0.5515 -
accuracy: 0.7820 - val_loss: 0.5665 - val_accuracy: 0.7696 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 26s 37ms/step - loss: 0.5408 -
accuracy: 0.7856 - val_loss: 0.5373 - val_accuracy: 0.7916 - lr: 0.0010
79/79 [==============================] - 0s 2ms/step
Training Fold 5…
Epoch 1/10
704/704 [==============================] - 29s 38ms/step - loss: 0.9279 -
accuracy: 0.6285 - val_loss: 0.7996 - val_accuracy: 0.6424 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 25s 36ms/step - loss: 0.7202 -
accuracy: 0.6900 - val_loss: 0.7867 - val_accuracy: 0.6300 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 26s 37ms/step - loss: 0.6794 -
accuracy: 0.7116 - val_loss: 0.6549 - val_accuracy: 0.7180 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 25s 35ms/step - loss: 0.6177 -
accuracy: 0.7390 - val_loss: 0.6863 - val_accuracy: 0.7232 - lr: 0.0010
```

```
Epoch 5/10
704/704 [==============================] - 25s 36ms/step - loss: 0.6030 -
accuracy: 0.7480 - val_loss: 0.6698 - val_accuracy: 0.7520 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 24s 34ms/step - loss: 0.6029 -
accuracy: 0.7633 - val_loss: 0.5720 - val_accuracy: 0.7872 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 24s 34ms/step - loss: 0.5708 -
accuracy: 0.7676 - val_loss: 0.5569 - val_accuracy: 0.7940 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 26s 37ms/step - loss: 0.5526 -
accuracy: 0.7780 - val_loss: 0.5034 - val_accuracy: 0.8008 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 25s 36ms/step - loss: 0.5395 -
accuracy: 0.7835 - val_loss: 0.5726 - val_accuracy: 0.7660 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 25s 36ms/step - loss: 0.5371 -
accuracy: 0.7816 - val_loss: 0.5397 - val_accuracy: 0.7808 - lr: 0.0010
79/79 [==============================] - 0s 2ms/step
Training Fold 6…
Epoch 1/10
704/704 [==============================] - 22s 29ms/step - loss: 0.8939 -
accuracy: 0.6421 - val_loss: 0.7168 - val_accuracy: 0.6996 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7355 -
accuracy: 0.6825 - val_loss: 0.9427 - val_accuracy: 0.6504 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6642 -
accuracy: 0.7126 - val_loss: 0.6545 - val_accuracy: 0.7308 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6126 -
accuracy: 0.7403 - val_loss: 0.8416 - val_accuracy: 0.5884 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6024 -
accuracy: 0.7502 - val_loss: 0.7956 - val_accuracy: 0.6036 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.5904 -
accuracy: 0.7529 - val_loss: 0.5959 - val_accuracy: 0.7696 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5668 -
accuracy: 0.7645 - val_loss: 0.5132 - val_accuracy: 0.7864 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5552 -
accuracy: 0.7632 - val_loss: 0.5184 - val_accuracy: 0.7904 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5418 -
accuracy: 0.7720 - val_loss: 0.4855 - val_accuracy: 0.8120 - lr: 0.0010
Epoch 10/10
```

```
704/704 [==============================] - 17s 25ms/step - loss: 0.5297 -
accuracy: 0.7803 - val_loss: 0.5358 - val_accuracy: 0.7900 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 7…
Epoch 1/10
704/704 [==============================] - 20s 27ms/step - loss: 0.9733 -
accuracy: 0.6076 - val_loss: 0.8123 - val_accuracy: 0.6480 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7442 -
accuracy: 0.6744 - val_loss: 0.6542 - val_accuracy: 0.7272 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 24ms/step - loss: 0.6694 -
accuracy: 0.6996 - val_loss: 0.9856 - val_accuracy: 0.6232 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6914 -
accuracy: 0.7139 - val_loss: 0.6447 - val_accuracy: 0.7456 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6271 -
accuracy: 0.7389 - val_loss: 0.5492 - val_accuracy: 0.7732 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5983 -
accuracy: 0.7517 - val_loss: 0.5845 - val_accuracy: 0.7592 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5839 -
accuracy: 0.7608 - val_loss: 0.5601 - val_accuracy: 0.7904 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5778 -
accuracy: 0.7644 - val_loss: 0.6123 - val_accuracy: 0.7572 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5281 -
accuracy: 0.7824 - val_loss: 0.4703 - val_accuracy: 0.8140 - lr: 5.0000e-04
Epoch 10/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5042 -
accuracy: 0.7946 - val_loss: 0.6893 - val_accuracy: 0.7168 - lr: 5.0000e-04
79/79 [==============================] - 0s 1ms/step
Training Fold 8…
Epoch 1/10
704/704 [==============================] - 21s 27ms/step - loss: 0.9428 -
accuracy: 0.6215 - val_loss: 0.7503 - val_accuracy: 0.6796 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7375 -
accuracy: 0.6819 - val_loss: 0.6536 - val_accuracy: 0.7192 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6734 -
accuracy: 0.7111 - val_loss: 0.6509 - val_accuracy: 0.7376 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6578 -
accuracy: 0.7320 - val_loss: 1.3765 - val_accuracy: 0.5468 - lr: 0.0010
```

```
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6314 -
accuracy: 0.7422 - val_loss: 0.7054 - val_accuracy: 0.6744 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6002 -
accuracy: 0.7556 - val_loss: 0.5100 - val_accuracy: 0.7988 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5748 -
accuracy: 0.7657 - val_loss: 0.5019 - val_accuracy: 0.8108 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5549 -
accuracy: 0.7732 - val_loss: 0.5284 - val_accuracy: 0.7928 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5423 -
accuracy: 0.7749 - val_loss: 0.8374 - val_accuracy: 0.6972 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5270 -
accuracy: 0.7814 - val_loss: 0.4885 - val_accuracy: 0.8140 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 9…
Epoch 1/10
704/704 [==============================] - 21s 27ms/step - loss: 0.9238 -
accuracy: 0.6340 - val_loss: 0.7440 - val_accuracy: 0.6800 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7111 -
accuracy: 0.6879 - val_loss: 0.7922 - val_accuracy: 0.5972 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 24ms/step - loss: 0.6732 -
accuracy: 0.7132 - val_loss: 0.6328 - val_accuracy: 0.7408 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6420 -
accuracy: 0.7399 - val_loss: 0.6545 - val_accuracy: 0.7720 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6128 -
accuracy: 0.7513 - val_loss: 0.6193 - val_accuracy: 0.7708 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.5911 -
accuracy: 0.7588 - val_loss: 0.6377 - val_accuracy: 0.7464 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5737 -
accuracy: 0.7698 - val_loss: 1.1236 - val_accuracy: 0.5836 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 24ms/step - loss: 0.5526 -
accuracy: 0.7786 - val_loss: 0.5844 - val_accuracy: 0.7752 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5423 -
accuracy: 0.7857 - val_loss: 0.5721 - val_accuracy: 0.7752 - lr: 0.0010
Epoch 10/10
```

```
704/704 [==============================] - 17s 25ms/step - loss: 0.5299 -
accuracy: 0.7897 - val_loss: 0.8529 - val_accuracy: 0.7144 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 10…
Epoch 1/10
704/704 [==============================] - 20s 27ms/step - loss: 0.8854 -
accuracy: 0.6388 - val_loss: 0.8474 - val_accuracy: 0.6192 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7111 -
accuracy: 0.6871 - val_loss: 0.6306 - val_accuracy: 0.7428 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6954 -
accuracy: 0.7138 - val_loss: 0.6857 - val_accuracy: 0.7116 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6138 -
accuracy: 0.7392 - val_loss: 0.6523 - val_accuracy: 0.7020 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6230 -
accuracy: 0.7482 - val_loss: 0.6313 - val_accuracy: 0.7424 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.5493 -
accuracy: 0.7774 - val_loss: 0.5299 - val_accuracy: 0.7804 - lr: 5.0000e-04
Epoch 7/10
704/704 [==============================] - 18s 26ms/step - loss: 0.5185 -
accuracy: 0.7881 - val_loss: 0.7230 - val_accuracy: 0.7040 - lr: 5.0000e-04
Epoch 8/10
704/704 [==============================] - 18s 25ms/step - loss: 0.5067 -
accuracy: 0.7924 - val_loss: 0.4892 - val_accuracy: 0.8024 - lr: 5.0000e-04
Epoch 9/10
704/704 [==============================] - 18s 26ms/step - loss: 0.4992 -
accuracy: 0.7968 - val_loss: 0.4810 - val_accuracy: 0.8140 - lr: 5.0000e-04
Epoch 10/10
704/704 [==============================] - 18s 26ms/step - loss: 0.4871 -
accuracy: 0.8013 - val_loss: 0.4902 - val_accuracy: 0.8024 - lr: 5.0000e-04
79/79 [==============================] - 0s 2ms/step

Results
AUC: 0.886 ± 0.007
Precision: 0.757 ± 0.091
Recall: 0.841 ± 0.107
F1-score: 0.784 ± 0.018
```

[24]:
```
# Entrenar usando el primer modelo CNN con data augmentation
train_cnn_model(create_cnn_model_2)
```

```
Training Fold 1…
Epoch 1/10
704/704 [==============================] - 22s 28ms/step - loss: 1.2194 -
accuracy: 0.6283 - val_loss: 0.9224 - val_accuracy: 0.7096 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8695 -
accuracy: 0.6904 - val_loss: 0.7755 - val_accuracy: 0.7132 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 24ms/step - loss: 0.7454 -
accuracy: 0.7288 - val_loss: 0.8785 - val_accuracy: 0.5928 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7148 -
accuracy: 0.7488 - val_loss: 0.7287 - val_accuracy: 0.7400 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 18s 26ms/step - loss: 0.6954 -
accuracy: 0.7666 - val_loss: 0.7066 - val_accuracy: 0.7612 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6727 -
accuracy: 0.7826 - val_loss: 0.8488 - val_accuracy: 0.7508 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6482 -
accuracy: 0.7916 - val_loss: 0.8699 - val_accuracy: 0.7288 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 24ms/step - loss: 0.6265 -
```

```
accuracy: 0.7993 - val_loss: 0.6007 - val_accuracy: 0.8248 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6156 -
accuracy: 0.8102 - val_loss: 0.7532 - val_accuracy: 0.7388 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6016 -
accuracy: 0.8095 - val_loss: 0.6409 - val_accuracy: 0.7936 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 2…
Epoch 1/10
704/704 [==============================] - 21s 28ms/step - loss: 1.2114 -
accuracy: 0.6296 - val_loss: 0.9658 - val_accuracy: 0.6716 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8607 -
accuracy: 0.6888 - val_loss: 0.9979 - val_accuracy: 0.6296 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7732 -
accuracy: 0.7204 - val_loss: 1.0671 - val_accuracy: 0.5880 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7131 -
accuracy: 0.7482 - val_loss: 0.7822 - val_accuracy: 0.7420 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6996 -
accuracy: 0.7689 - val_loss: 0.8739 - val_accuracy: 0.7184 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6709 -
accuracy: 0.7832 - val_loss: 0.6174 - val_accuracy: 0.8200 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6384 -
accuracy: 0.7968 - val_loss: 0.6418 - val_accuracy: 0.7816 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6315 -
accuracy: 0.8024 - val_loss: 0.6752 - val_accuracy: 0.8176 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6070 -
accuracy: 0.8117 - val_loss: 0.6866 - val_accuracy: 0.7612 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5378 -
accuracy: 0.8364 - val_loss: 0.4730 - val_accuracy: 0.8592 - lr: 5.0000e-04
79/79 [==============================] - 0s 1ms/step
Training Fold 3…
Epoch 1/10
704/704 [==============================] - 22s 28ms/step - loss: 1.1561 -
accuracy: 0.6193 - val_loss: 0.9373 - val_accuracy: 0.6668 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 24ms/step - loss: 0.8747 -
accuracy: 0.6796 - val_loss: 0.9354 - val_accuracy: 0.6188 - lr: 0.0010
Epoch 3/10
```

```
704/704 [==============================] - 18s 25ms/step - loss: 0.7525 -
accuracy: 0.7198 - val_loss: 0.7820 - val_accuracy: 0.6532 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7220 -
accuracy: 0.7494 - val_loss: 1.5507 - val_accuracy: 0.5440 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 23s 33ms/step - loss: 0.6829 -
accuracy: 0.7713 - val_loss: 0.7420 - val_accuracy: 0.7456 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 24s 35ms/step - loss: 0.6578 -
accuracy: 0.7878 - val_loss: 0.9861 - val_accuracy: 0.6484 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 25s 36ms/step - loss: 0.6407 -
accuracy: 0.7976 - val_loss: 0.6002 - val_accuracy: 0.8108 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 25s 35ms/step - loss: 0.6185 -
accuracy: 0.8048 - val_loss: 0.6950 - val_accuracy: 0.7356 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 25s 35ms/step - loss: 0.5919 -
accuracy: 0.8156 - val_loss: 0.5981 - val_accuracy: 0.8096 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 23s 33ms/step - loss: 0.5805 -
accuracy: 0.8173 - val_loss: 0.8688 - val_accuracy: 0.6976 - lr: 0.0010
79/79 [==============================] - 0s 2ms/step
Training Fold 4…
Epoch 1/10
704/704 [==============================] - 24s 31ms/step - loss: 1.1911 -
accuracy: 0.6292 - val_loss: 1.0122 - val_accuracy: 0.6504 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8689 -
accuracy: 0.6881 - val_loss: 1.0648 - val_accuracy: 0.5908 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 18s 26ms/step - loss: 0.7546 -
accuracy: 0.7180 - val_loss: 0.7764 - val_accuracy: 0.7200 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7121 -
accuracy: 0.7464 - val_loss: 0.6937 - val_accuracy: 0.7512 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 18s 26ms/step - loss: 0.6894 -
accuracy: 0.7677 - val_loss: 0.7485 - val_accuracy: 0.7208 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6715 -
accuracy: 0.7795 - val_loss: 0.6275 - val_accuracy: 0.7956 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 18s 26ms/step - loss: 0.6451 -
accuracy: 0.7928 - val_loss: 0.6448 - val_accuracy: 0.7960 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6363 -
```

```
accuracy: 0.7988 - val_loss: 0.6138 - val_accuracy: 0.8124 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 18s 26ms/step - loss: 0.6169 -
accuracy: 0.8079 - val_loss: 0.5738 - val_accuracy: 0.8240 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6012 -
accuracy: 0.8082 - val_loss: 0.8503 - val_accuracy: 0.7248 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 5…
Epoch 1/10
704/704 [==============================] - 21s 27ms/step - loss: 1.2009 -
accuracy: 0.6234 - val_loss: 1.0202 - val_accuracy: 0.6320 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8705 -
accuracy: 0.6822 - val_loss: 0.7630 - val_accuracy: 0.7212 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 24ms/step - loss: 0.7507 -
accuracy: 0.7184 - val_loss: 0.7411 - val_accuracy: 0.6932 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7066 -
accuracy: 0.7403 - val_loss: 0.8033 - val_accuracy: 0.7396 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 24ms/step - loss: 0.7022 -
accuracy: 0.7574 - val_loss: 1.0118 - val_accuracy: 0.7056 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 17s 24ms/step - loss: 0.6752 -
accuracy: 0.7768 - val_loss: 0.7338 - val_accuracy: 0.7336 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6542 -
accuracy: 0.7928 - val_loss: 0.6042 - val_accuracy: 0.8148 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 24ms/step - loss: 0.6369 -
accuracy: 0.7981 - val_loss: 0.6633 - val_accuracy: 0.7912 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6136 -
accuracy: 0.8037 - val_loss: 0.8450 - val_accuracy: 0.6868 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6021 -
accuracy: 0.8104 - val_loss: 0.5548 - val_accuracy: 0.8384 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 6…
Epoch 1/10
704/704 [==============================] - 22s 28ms/step - loss: 1.1920 -
accuracy: 0.6266 - val_loss: 0.9359 - val_accuracy: 0.6780 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8747 -
accuracy: 0.6825 - val_loss: 0.7431 - val_accuracy: 0.7440 - lr: 0.0010
Epoch 3/10
```

```
704/704 [==============================] - 17s 25ms/step - loss: 0.7605 -
accuracy: 0.7197 - val_loss: 1.8351 - val_accuracy: 0.5176 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 24ms/step - loss: 0.7078 -
accuracy: 0.7484 - val_loss: 0.9111 - val_accuracy: 0.6856 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6906 -
accuracy: 0.7714 - val_loss: 0.6933 - val_accuracy: 0.7220 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6560 -
accuracy: 0.7879 - val_loss: 0.6714 - val_accuracy: 0.7668 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6425 -
accuracy: 0.7956 - val_loss: 0.6525 - val_accuracy: 0.7872 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6245 -
accuracy: 0.8011 - val_loss: 0.6263 - val_accuracy: 0.7728 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6037 -
accuracy: 0.8106 - val_loss: 0.5650 - val_accuracy: 0.8248 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 17s 24ms/step - loss: 0.5970 -
accuracy: 0.8149 - val_loss: 0.6407 - val_accuracy: 0.8016 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 7…
Epoch 1/10
704/704 [==============================] - 21s 27ms/step - loss: 1.2284 -
accuracy: 0.6237 - val_loss: 0.9485 - val_accuracy: 0.6928 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8773 -
accuracy: 0.6814 - val_loss: 0.7870 - val_accuracy: 0.7188 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7697 -
accuracy: 0.7067 - val_loss: 0.7623 - val_accuracy: 0.6716 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7077 -
accuracy: 0.7316 - val_loss: 0.9082 - val_accuracy: 0.6132 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6888 -
accuracy: 0.7564 - val_loss: 0.6160 - val_accuracy: 0.8048 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6617 -
accuracy: 0.7765 - val_loss: 0.6110 - val_accuracy: 0.8124 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6442 -
accuracy: 0.7856 - val_loss: 0.6721 - val_accuracy: 0.7840 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6249 -
```

```
accuracy: 0.7975 - val_loss: 0.6170 - val_accuracy: 0.8012 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6025 -
accuracy: 0.8111 - val_loss: 0.5528 - val_accuracy: 0.8272 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 17s 24ms/step - loss: 0.5942 -
accuracy: 0.8108 - val_loss: 0.6117 - val_accuracy: 0.7924 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 8…
Epoch 1/10
704/704 [==============================] - 20s 26ms/step - loss: 1.2219 -
accuracy: 0.6220 - val_loss: 0.9634 - val_accuracy: 0.6600 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8736 -
accuracy: 0.6780 - val_loss: 1.0266 - val_accuracy: 0.6224 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7738 -
accuracy: 0.7131 - val_loss: 0.6548 - val_accuracy: 0.7884 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7271 -
accuracy: 0.7428 - val_loss: 0.8023 - val_accuracy: 0.7108 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6979 -
accuracy: 0.7651 - val_loss: 0.6840 - val_accuracy: 0.7844 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 17s 24ms/step - loss: 0.6647 -
accuracy: 0.7802 - val_loss: 0.7317 - val_accuracy: 0.7424 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5870 -
accuracy: 0.8157 - val_loss: 0.5566 - val_accuracy: 0.8276 - lr: 5.0000e-04
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5413 -
accuracy: 0.8273 - val_loss: 0.4911 - val_accuracy: 0.8584 - lr: 5.0000e-04
Epoch 9/10
704/704 [==============================] - 18s 25ms/step - loss: 0.5240 -
accuracy: 0.8323 - val_loss: 0.5066 - val_accuracy: 0.8480 - lr: 5.0000e-04
Epoch 10/10
704/704 [==============================] - 17s 25ms/step - loss: 0.5144 -
accuracy: 0.8377 - val_loss: 0.4875 - val_accuracy: 0.8496 - lr: 5.0000e-04
79/79 [==============================] - 0s 1ms/step
Training Fold 9…
Epoch 1/10
704/704 [==============================] - 21s 26ms/step - loss: 1.1991 -
accuracy: 0.6222 - val_loss: 1.0366 - val_accuracy: 0.6144 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8732 -
accuracy: 0.6839 - val_loss: 0.7658 - val_accuracy: 0.7336 - lr: 0.0010
Epoch 3/10
```

```
704/704 [==============================] - 18s 25ms/step - loss: 0.7682 -
accuracy: 0.7125 - val_loss: 0.7516 - val_accuracy: 0.6968 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7062 -
accuracy: 0.7402 - val_loss: 0.7651 - val_accuracy: 0.6588 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6878 -
accuracy: 0.7610 - val_loss: 0.7083 - val_accuracy: 0.7408 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6658 -
accuracy: 0.7763 - val_loss: 0.6137 - val_accuracy: 0.7920 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6385 -
accuracy: 0.7892 - val_loss: 0.7176 - val_accuracy: 0.7256 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6243 -
accuracy: 0.7964 - val_loss: 0.6319 - val_accuracy: 0.7916 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6102 -
accuracy: 0.8079 - val_loss: 0.5541 - val_accuracy: 0.8316 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6039 -
accuracy: 0.8073 - val_loss: 0.5225 - val_accuracy: 0.8460 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 10…
Epoch 1/10
704/704 [==============================] - 21s 27ms/step - loss: 1.2100 -
accuracy: 0.6105 - val_loss: 0.9549 - val_accuracy: 0.6556 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8804 -
accuracy: 0.6701 - val_loss: 0.7890 - val_accuracy: 0.7264 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7675 -
accuracy: 0.7093 - val_loss: 0.8071 - val_accuracy: 0.6748 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7296 -
accuracy: 0.7355 - val_loss: 0.7052 - val_accuracy: 0.7456 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6926 -
accuracy: 0.7588 - val_loss: 0.7836 - val_accuracy: 0.7152 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6733 -
accuracy: 0.7736 - val_loss: 0.6073 - val_accuracy: 0.8152 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6563 -
accuracy: 0.7840 - val_loss: 0.6478 - val_accuracy: 0.7984 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6342 -
```

```
accuracy: 0.7948 - val_loss: 0.6851 - val_accuracy: 0.7848 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6253 -
accuracy: 0.7977 - val_loss: 0.6494 - val_accuracy: 0.7884 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 18s 25ms/step - loss: 0.5470 -
accuracy: 0.8277 - val_loss: 0.5305 - val_accuracy: 0.8088 - lr: 5.0000e-04
79/79 [==============================] - 0s 1ms/step

Results
AUC: 0.918 ± 0.021
Precision: 0.839 ± 0.081
Recall: 0.780 ± 0.187
F1-score: 0.785 ± 0.093
```

[25]:
```python
# Entrenar usando el primer modelo CNN con data augmentation
train_cnn_model(create_cnn_model_3)
```

```
Training Fold 1…
Epoch 1/10
704/704 [==============================] - 21s 26ms/step - loss: 1.8652 -
accuracy: 0.5988 - val_loss: 1.2968 - val_accuracy: 0.6104 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 1.0912 -
accuracy: 0.6384 - val_loss: 1.0012 - val_accuracy: 0.5984 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8998 -
```

```
accuracy: 0.6679 - val_loss: 0.9674 - val_accuracy: 0.6228 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8768 -
accuracy: 0.6990 - val_loss: 0.8189 - val_accuracy: 0.7272 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8265 -
accuracy: 0.7260 - val_loss: 0.8240 - val_accuracy: 0.7400 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7984 -
accuracy: 0.7514 - val_loss: 0.8740 - val_accuracy: 0.7392 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7795 -
accuracy: 0.7627 - val_loss: 0.8537 - val_accuracy: 0.7500 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6611 -
accuracy: 0.8036 - val_loss: 0.7004 - val_accuracy: 0.7812 - lr: 5.0000e-04
Epoch 9/10
704/704 [==============================] - 18s 25ms/step - loss: 0.5922 -
accuracy: 0.8210 - val_loss: 1.1573 - val_accuracy: 0.6540 - lr: 5.0000e-04
Epoch 10/10
704/704 [==============================] - 18s 25ms/step - loss: 0.5727 -
accuracy: 0.8251 - val_loss: 0.5272 - val_accuracy: 0.8572 - lr: 5.0000e-04
79/79 [==============================] - 0s 1ms/step
Training Fold 2…
Epoch 1/10
704/704 [==============================] - 20s 26ms/step - loss: 1.8219 -
accuracy: 0.5986 - val_loss: 1.2073 - val_accuracy: 0.6548 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 1.0721 -
accuracy: 0.6405 - val_loss: 0.9344 - val_accuracy: 0.6704 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 25ms/step - loss: 0.9118 -
accuracy: 0.6728 - val_loss: 0.8799 - val_accuracy: 0.6728 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8685 -
accuracy: 0.6918 - val_loss: 1.9138 - val_accuracy: 0.5140 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8485 -
accuracy: 0.7204 - val_loss: 0.7987 - val_accuracy: 0.7356 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7987 -
accuracy: 0.7445 - val_loss: 1.0398 - val_accuracy: 0.6264 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7816 -
accuracy: 0.7645 - val_loss: 0.7414 - val_accuracy: 0.7652 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7524 -
accuracy: 0.7764 - val_loss: 0.7453 - val_accuracy: 0.7696 - lr: 0.0010
```

```
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7152 -
accuracy: 0.7891 - val_loss: 0.6443 - val_accuracy: 0.8172 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6763 -
accuracy: 0.7988 - val_loss: 0.7051 - val_accuracy: 0.7624 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 3…
Epoch 1/10
704/704 [==============================] - 21s 27ms/step - loss: 1.8951 -
accuracy: 0.6107 - val_loss: 1.2394 - val_accuracy: 0.6532 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 1.0972 -
accuracy: 0.6603 - val_loss: 1.0008 - val_accuracy: 0.6700 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 18s 25ms/step - loss: 0.9131 -
accuracy: 0.6849 - val_loss: 0.9404 - val_accuracy: 0.6732 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8928 -
accuracy: 0.7119 - val_loss: 0.8746 - val_accuracy: 0.7156 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8421 -
accuracy: 0.7439 - val_loss: 0.9378 - val_accuracy: 0.7116 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8137 -
accuracy: 0.7589 - val_loss: 0.8035 - val_accuracy: 0.7988 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8063 -
accuracy: 0.7745 - val_loss: 0.8219 - val_accuracy: 0.7576 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7732 -
accuracy: 0.7891 - val_loss: 0.7647 - val_accuracy: 0.8092 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7468 -
accuracy: 0.7977 - val_loss: 0.6844 - val_accuracy: 0.8324 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7183 -
accuracy: 0.8030 - val_loss: 0.6731 - val_accuracy: 0.8276 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 4…
Epoch 1/10
704/704 [==============================] - 21s 26ms/step - loss: 2.0310 -
accuracy: 0.5961 - val_loss: 1.3239 - val_accuracy: 0.5900 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 1.1103 -
accuracy: 0.6507 - val_loss: 0.9788 - val_accuracy: 0.6940 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 18s 25ms/step - loss: 0.9085 -
```

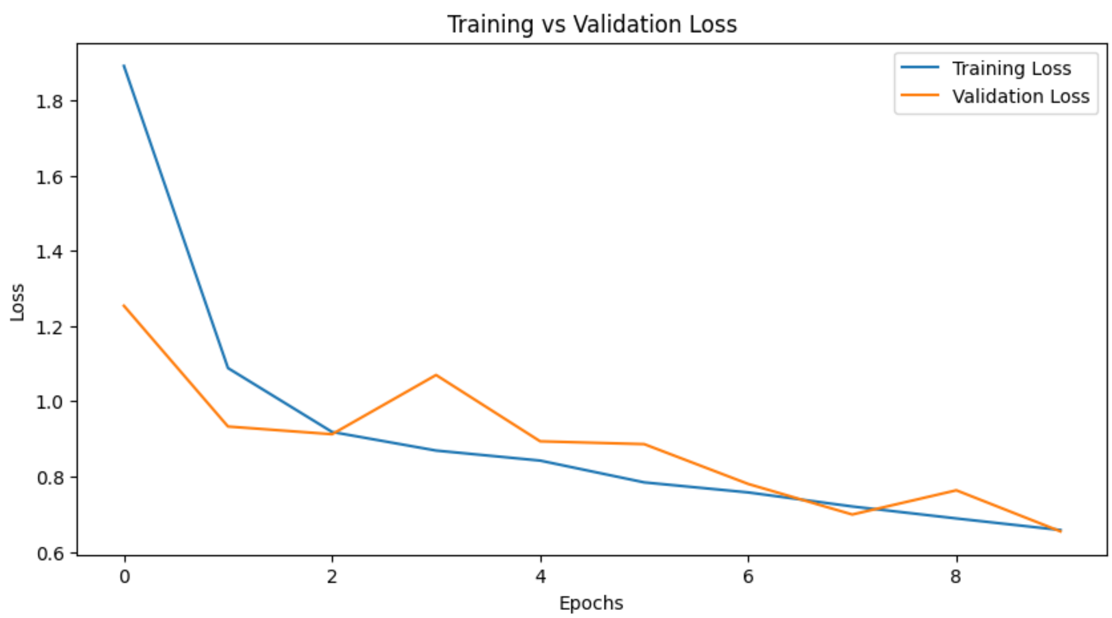
```
accuracy: 0.6820 - val_loss: 0.9761 - val_accuracy: 0.5856 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8388 -
accuracy: 0.7132 - val_loss: 0.9416 - val_accuracy: 0.6728 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8630 -
accuracy: 0.7324 - val_loss: 0.8910 - val_accuracy: 0.6888 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 22s 31ms/step - loss: 0.8165 -
accuracy: 0.7570 - val_loss: 1.0178 - val_accuracy: 0.6456 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 24s 34ms/step - loss: 0.8231 -
accuracy: 0.7705 - val_loss: 0.8025 - val_accuracy: 0.8088 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 23s 32ms/step - loss: 0.7879 -
accuracy: 0.7800 - val_loss: 0.7099 - val_accuracy: 0.8216 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7673 -
accuracy: 0.7853 - val_loss: 0.8729 - val_accuracy: 0.7332 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7121 -
accuracy: 0.7981 - val_loss: 0.7492 - val_accuracy: 0.7664 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 5…
Epoch 1/10
704/704 [==============================] - 22s 29ms/step - loss: 1.9445 -
accuracy: 0.5984 - val_loss: 1.2871 - val_accuracy: 0.6048 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 26s 37ms/step - loss: 1.0971 -
accuracy: 0.6432 - val_loss: 0.8974 - val_accuracy: 0.6992 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 25s 35ms/step - loss: 0.9364 -
accuracy: 0.6633 - val_loss: 0.9017 - val_accuracy: 0.6868 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 20s 29ms/step - loss: 0.8348 -
accuracy: 0.7032 - val_loss: 1.3716 - val_accuracy: 0.6244 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8498 -
accuracy: 0.7198 - val_loss: 0.7572 - val_accuracy: 0.7596 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8016 -
accuracy: 0.7496 - val_loss: 1.5468 - val_accuracy: 0.5272 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7849 -
accuracy: 0.7646 - val_loss: 0.8937 - val_accuracy: 0.7016 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7712 -
accuracy: 0.7826 - val_loss: 0.7372 - val_accuracy: 0.7696 - lr: 0.0010
```

```
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7188 -
accuracy: 0.7942 - val_loss: 0.6858 - val_accuracy: 0.7952 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6818 -
accuracy: 0.7994 - val_loss: 0.7617 - val_accuracy: 0.7720 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 6…
Epoch 1/10
704/704 [==============================] - 21s 27ms/step - loss: 1.8136 -
accuracy: 0.5959 - val_loss: 1.2282 - val_accuracy: 0.6332 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 17s 25ms/step - loss: 1.0531 -
accuracy: 0.6355 - val_loss: 0.8403 - val_accuracy: 0.7136 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 25ms/step - loss: 0.9308 -
accuracy: 0.6559 - val_loss: 0.8895 - val_accuracy: 0.7092 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8715 -
accuracy: 0.6948 - val_loss: 0.8520 - val_accuracy: 0.7504 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 17s 24ms/step - loss: 0.8281 -
accuracy: 0.7284 - val_loss: 1.0654 - val_accuracy: 0.6696 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6912 -
accuracy: 0.7748 - val_loss: 0.5795 - val_accuracy: 0.8296 - lr: 5.0000e-04
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6321 -
accuracy: 0.7941 - val_loss: 0.6228 - val_accuracy: 0.7964 - lr: 5.0000e-04
Epoch 8/10
704/704 [==============================] - 17s 25ms/step - loss: 0.6105 -
accuracy: 0.8118 - val_loss: 0.5131 - val_accuracy: 0.8648 - lr: 5.0000e-04
Epoch 9/10
704/704 [==============================] - 19s 27ms/step - loss: 0.5909 -
accuracy: 0.8184 - val_loss: 0.5290 - val_accuracy: 0.8576 - lr: 5.0000e-04
Epoch 10/10
704/704 [==============================] - 26s 36ms/step - loss: 0.5777 -
accuracy: 0.8261 - val_loss: 0.5042 - val_accuracy: 0.8672 - lr: 5.0000e-04
79/79 [==============================] - 0s 2ms/step
Training Fold 7…
Epoch 1/10
704/704 [==============================] - 28s 36ms/step - loss: 1.9160 -
accuracy: 0.6090 - val_loss: 1.2547 - val_accuracy: 0.6948 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 24s 34ms/step - loss: 1.0970 -
accuracy: 0.6605 - val_loss: 0.8736 - val_accuracy: 0.7160 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 25s 36ms/step - loss: 0.9641 -
```

```
accuracy: 0.6885 - val_loss: 0.8980 - val_accuracy: 0.6944 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 20s 28ms/step - loss: 0.8720 -
accuracy: 0.7132 - val_loss: 1.0087 - val_accuracy: 0.6288 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 19s 28ms/step - loss: 0.8582 -
accuracy: 0.7414 - val_loss: 0.9809 - val_accuracy: 0.6732 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 22s 31ms/step - loss: 0.7272 -
accuracy: 0.7850 - val_loss: 0.6238 - val_accuracy: 0.8176 - lr: 5.0000e-04
Epoch 7/10
704/704 [==============================] - 24s 34ms/step - loss: 0.6422 -
accuracy: 0.8000 - val_loss: 0.6127 - val_accuracy: 0.8108 - lr: 5.0000e-04
Epoch 8/10
704/704 [==============================] - 25s 36ms/step - loss: 0.6116 -
accuracy: 0.8152 - val_loss: 0.6254 - val_accuracy: 0.8268 - lr: 5.0000e-04
Epoch 9/10
704/704 [==============================] - 25s 35ms/step - loss: 0.5990 -
accuracy: 0.8197 - val_loss: 0.5721 - val_accuracy: 0.8324 - lr: 5.0000e-04
Epoch 10/10
704/704 [==============================] - 25s 35ms/step - loss: 0.5876 -
accuracy: 0.8297 - val_loss: 0.5424 - val_accuracy: 0.8408 - lr: 5.0000e-04
79/79 [==============================] - 0s 2ms/step
Training Fold 8…
Epoch 1/10
704/704 [==============================] - 28s 36ms/step - loss: 1.8433 -
accuracy: 0.6110 - val_loss: 1.2770 - val_accuracy: 0.6312 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 24s 34ms/step - loss: 1.0496 -
accuracy: 0.6615 - val_loss: 0.9352 - val_accuracy: 0.6608 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 18s 26ms/step - loss: 0.9372 -
accuracy: 0.6715 - val_loss: 0.9341 - val_accuracy: 0.6728 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 18s 26ms/step - loss: 0.8541 -
accuracy: 0.7106 - val_loss: 0.8131 - val_accuracy: 0.7320 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8374 -
accuracy: 0.7372 - val_loss: 1.1274 - val_accuracy: 0.5820 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8256 -
accuracy: 0.7502 - val_loss: 0.7747 - val_accuracy: 0.7752 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8042 -
accuracy: 0.7637 - val_loss: 0.8522 - val_accuracy: 0.7328 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7721 -
accuracy: 0.7780 - val_loss: 0.7425 - val_accuracy: 0.8112 - lr: 0.0010
```

```
Epoch 9/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7340 -
accuracy: 0.7951 - val_loss: 0.9611 - val_accuracy: 0.7504 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7042 -
accuracy: 0.7960 - val_loss: 0.6877 - val_accuracy: 0.7844 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 9…
Epoch 1/10
704/704 [==============================] - 21s 27ms/step - loss: 1.9231 -
accuracy: 0.6056 - val_loss: 1.1957 - val_accuracy: 0.6952 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 18s 25ms/step - loss: 1.1661 -
accuracy: 0.6418 - val_loss: 0.9719 - val_accuracy: 0.6396 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 17s 25ms/step - loss: 0.8967 -
accuracy: 0.6670 - val_loss: 0.9149 - val_accuracy: 0.6108 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8996 -
accuracy: 0.6933 - val_loss: 1.2293 - val_accuracy: 0.5700 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8332 -
accuracy: 0.7228 - val_loss: 0.8515 - val_accuracy: 0.7256 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 18s 25ms/step - loss: 0.8073 -
accuracy: 0.7436 - val_loss: 0.8549 - val_accuracy: 0.6460 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7661 -
accuracy: 0.7588 - val_loss: 0.9046 - val_accuracy: 0.6992 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 18s 26ms/step - loss: 0.7411 -
accuracy: 0.7770 - val_loss: 0.7076 - val_accuracy: 0.7948 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7138 -
accuracy: 0.7849 - val_loss: 0.6495 - val_accuracy: 0.8212 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6751 -
accuracy: 0.7966 - val_loss: 0.8109 - val_accuracy: 0.7344 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step
Training Fold 10…
Epoch 1/10
704/704 [==============================] - 21s 27ms/step - loss: 1.8537 -
accuracy: 0.5896 - val_loss: 1.2251 - val_accuracy: 0.6212 - lr: 0.0010
Epoch 2/10
704/704 [==============================] - 22s 31ms/step - loss: 1.0473 -
accuracy: 0.6330 - val_loss: 0.8948 - val_accuracy: 0.6476 - lr: 0.0010
Epoch 3/10
704/704 [==============================] - 24s 35ms/step - loss: 0.8859 -
```

```
accuracy: 0.6509 - val_loss: 0.8213 - val_accuracy: 0.7288 - lr: 0.0010
Epoch 4/10
704/704 [==============================] - 25s 35ms/step - loss: 0.8805 -
accuracy: 0.6729 - val_loss: 0.8732 - val_accuracy: 0.6904 - lr: 0.0010
Epoch 5/10
704/704 [==============================] - 26s 36ms/step - loss: 0.8362 -
accuracy: 0.7096 - val_loss: 0.7014 - val_accuracy: 0.7756 - lr: 0.0010
Epoch 6/10
704/704 [==============================] - 25s 36ms/step - loss: 0.7639 -
accuracy: 0.7433 - val_loss: 0.7454 - val_accuracy: 0.7260 - lr: 0.0010
Epoch 7/10
704/704 [==============================] - 18s 26ms/step - loss: 0.7569 -
accuracy: 0.7577 - val_loss: 0.6966 - val_accuracy: 0.7916 - lr: 0.0010
Epoch 8/10
704/704 [==============================] - 18s 25ms/step - loss: 0.7230 -
accuracy: 0.7771 - val_loss: 0.7439 - val_accuracy: 0.7640 - lr: 0.0010
Epoch 9/10
704/704 [==============================] - 17s 25ms/step - loss: 0.7078 -
accuracy: 0.7856 - val_loss: 0.8774 - val_accuracy: 0.6772 - lr: 0.0010
Epoch 10/10
704/704 [==============================] - 18s 25ms/step - loss: 0.6727 -
accuracy: 0.7964 - val_loss: 0.5806 - val_accuracy: 0.8360 - lr: 0.0010
79/79 [==============================] - 0s 1ms/step

Results
AUC: 0.915 ± 0.020
Precision: 0.831 ± 0.077
Recall: 0.794 ± 0.150
F1-score: 0.797 ± 0.066
```



2.1 +1

(mejorado)

[37]:
```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,␣
↪Dropout, BatchNormalization
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import roc_auc_score, precision_score, recall_score,␣
↪f1_score
import numpy as np
import matplotlib.pyplot as plt

# Deep CNN Architecture
def create_deep_cnn_model():
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(64, (3, 3), activation='relu'),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(128, (3, 3), activation='relu'),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(256, (3, 3), activation='relu'),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),

        Flatten(),
        Dense(512, activation='relu'),
        Dropout(0.5),

        Dense(256, activation='relu'),
        Dropout(0.5),

        Dense(2, activation='softmax')
    ])

    return model
```

```python
# Function to train and evaluate the deep CNN model
def train_deep_cnn_model():
    skf = StratifiedKFold(n_splits=10)
    auc_scores = []
    precision_scores = []
    recall_scores = []
    f1_scores = []

    # Para guardar las pérdidas de entrenamiento y validación
    training_losses = []
    validation_losses = []

    fold = 0
    for train_index, test_index in skf.split(train_cnn, y):
        fold += 1
        print(f"\nTraining Fold {fold}...")

        X_train, X_test = train_cnn[train_index], train_cnn[test_index]
        y_train, y_test = y_categorical[train_index], y_categorical[test_index]

        model = create_deep_cnn_model()
        model.compile(optimizer='adam', loss='categorical_crossentropy',␣
↪metrics=['accuracy'])

        history = model.fit(X_train, y_train, validation_data=(X_test, y_test),␣
↪epochs=10, batch_size=32)

        # Almacenar las pérdidas de entrenamiento y validación para cada época
        training_losses.append(history.history['loss'])
        validation_losses.append(history.history['val_loss'])

        y_pred = model.predict(X_test)
        y_pred_class = y_pred.argmax(axis=1)
        y_true = y_test.argmax(axis=1)

        auc = roc_auc_score(y_true, y_pred[:, 1])
        precision = precision_score(y_true, y_pred_class)
        recall = recall_score(y_true, y_pred_class)
        f1 = f1_score(y_true, y_pred_class)

        auc_scores.append(auc)
        precision_scores.append(precision)
        recall_scores.append(recall)
        f1_scores.append(f1)

    # Mostrar resultados promedio
    print("\nResults")
```

```
    print(f"AUC: {np.mean(auc_scores):.3f} ± {np.std(auc_scores):.3f}")
    print(f"Precision: {np.mean(precision_scores):.3f} ± {np.
 ↪std(precision_scores):.3f}")
    print(f"Recall: {np.mean(recall_scores):.3f} ± {np.std(recall_scores):.3f}")
    print(f"F1-score: {np.mean(f1_scores):.3f} ± {np.std(f1_scores):.3f}")

    # Graficar pérdidas de entrenamiento y validación
    plt.figure(figsize=(10, 5))
    plt.plot(np.mean(training_losses, axis=0), label="Training Loss")
    plt.plot(np.mean(validation_losses, axis=0), label="Validation Loss")
    plt.title("Training vs Validation Loss")
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()
    plt.show()

# Train the deep CNN model
train_deep_cnn_model()
```

```
Training Fold 1…
Epoch 1/10
704/704 [==============================] - 6s 5ms/step - loss: 0.6824 -
accuracy: 0.6453 - val_loss: 0.6691 - val_accuracy: 0.5884
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4905 -
accuracy: 0.7679 - val_loss: 0.4942 - val_accuracy: 0.7732
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3907 -
accuracy: 0.8252 - val_loss: 0.4512 - val_accuracy: 0.7876
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3321 -
accuracy: 0.8565 - val_loss: 0.4624 - val_accuracy: 0.7632
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2900 -
accuracy: 0.8804 - val_loss: 0.3603 - val_accuracy: 0.8528
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2456 -
accuracy: 0.8985 - val_loss: 0.3197 - val_accuracy: 0.8728
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2128 -
accuracy: 0.9149 - val_loss: 0.3499 - val_accuracy: 0.8440
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1859 -
accuracy: 0.9279 - val_loss: 0.2964 - val_accuracy: 0.8808
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1579 -
```

```
accuracy: 0.9369 - val_loss: 0.3905 - val_accuracy: 0.8548
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1584 -
accuracy: 0.9393 - val_loss: 0.4578 - val_accuracy: 0.8440
79/79 [==============================] - 0s 1ms/step

Training Fold 2…
Epoch 1/10
704/704 [==============================] - 8s 5ms/step - loss: 0.6870 -
accuracy: 0.6488 - val_loss: 0.7559 - val_accuracy: 0.5836
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4800 -
accuracy: 0.7718 - val_loss: 0.4992 - val_accuracy: 0.7480
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3830 -
accuracy: 0.8328 - val_loss: 0.4303 - val_accuracy: 0.7768
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3224 -
accuracy: 0.8624 - val_loss: 0.3423 - val_accuracy: 0.8404
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2842 -
accuracy: 0.8834 - val_loss: 0.3465 - val_accuracy: 0.8488
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2372 -
accuracy: 0.9043 - val_loss: 0.4196 - val_accuracy: 0.8104
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2102 -
accuracy: 0.9184 - val_loss: 0.3191 - val_accuracy: 0.8572
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1742 -
accuracy: 0.9325 - val_loss: 0.3306 - val_accuracy: 0.8580
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1435 -
accuracy: 0.9448 - val_loss: 0.3544 - val_accuracy: 0.8496
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1366 -
accuracy: 0.9501 - val_loss: 0.3701 - val_accuracy: 0.8608
79/79 [==============================] - 0s 2ms/step

Training Fold 3…
Epoch 1/10
704/704 [==============================] - 6s 5ms/step - loss: 0.6682 -
accuracy: 0.6586 - val_loss: 0.5033 - val_accuracy: 0.7568
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4680 -
accuracy: 0.7801 - val_loss: 0.4734 - val_accuracy: 0.7672
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3783 -
```

```
accuracy: 0.8348 - val_loss: 0.3920 - val_accuracy: 0.8240
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3117 -
accuracy: 0.8679 - val_loss: 0.4255 - val_accuracy: 0.7872
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2741 -
accuracy: 0.8858 - val_loss: 0.4468 - val_accuracy: 0.7808
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2308 -
accuracy: 0.9071 - val_loss: 0.3725 - val_accuracy: 0.8448
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1965 -
accuracy: 0.9218 - val_loss: 0.3041 - val_accuracy: 0.8728
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1618 -
accuracy: 0.9362 - val_loss: 0.3297 - val_accuracy: 0.8536
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1529 -
accuracy: 0.9402 - val_loss: 0.3438 - val_accuracy: 0.8700
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1239 -
accuracy: 0.9528 - val_loss: 0.3850 - val_accuracy: 0.8580
79/79 [==============================] - 0s 1ms/step

Training Fold 4…
Epoch 1/10
704/704 [==============================] - 6s 5ms/step - loss: 0.6939 -
accuracy: 0.6427 - val_loss: 0.5776 - val_accuracy: 0.6756
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4725 -
accuracy: 0.7788 - val_loss: 0.5292 - val_accuracy: 0.7308
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3830 -
accuracy: 0.8292 - val_loss: 0.4137 - val_accuracy: 0.8080
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3144 -
accuracy: 0.8678 - val_loss: 0.4085 - val_accuracy: 0.8116
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2729 -
accuracy: 0.8873 - val_loss: 0.3587 - val_accuracy: 0.8388
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2333 -
accuracy: 0.9030 - val_loss: 0.4873 - val_accuracy: 0.8104
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2030 -
accuracy: 0.9195 - val_loss: 0.4023 - val_accuracy: 0.8496
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1767 -
```

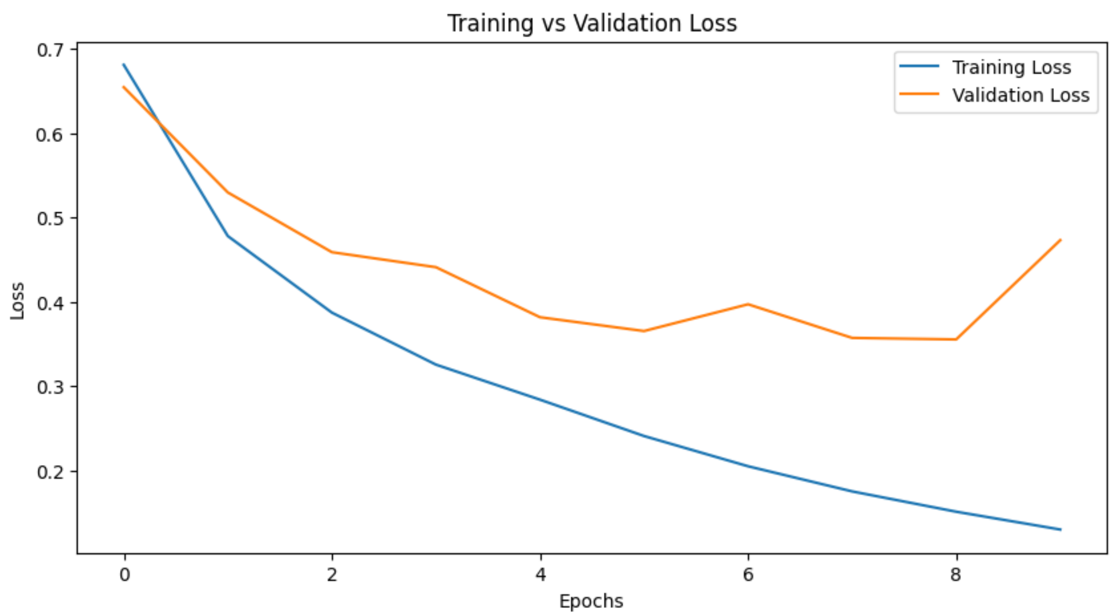
```
accuracy: 0.9307 - val_loss: 0.3871 - val_accuracy: 0.8296
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1570 -
accuracy: 0.9401 - val_loss: 0.3372 - val_accuracy: 0.8528
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1234 -
accuracy: 0.9531 - val_loss: 0.6363 - val_accuracy: 0.8252
79/79 [==============================] - 0s 2ms/step

Training Fold 5…
Epoch 1/10
704/704 [==============================] - 6s 5ms/step - loss: 0.6826 -
accuracy: 0.6464 - val_loss: 0.6234 - val_accuracy: 0.6232
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4925 -
accuracy: 0.7692 - val_loss: 0.7016 - val_accuracy: 0.5876
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3999 -
accuracy: 0.8224 - val_loss: 0.4468 - val_accuracy: 0.7808
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3370 -
accuracy: 0.8572 - val_loss: 0.6282 - val_accuracy: 0.7024
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2904 -
accuracy: 0.8764 - val_loss: 0.3084 - val_accuracy: 0.8656
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2511 -
accuracy: 0.8964 - val_loss: 0.3149 - val_accuracy: 0.8668
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2144 -
accuracy: 0.9135 - val_loss: 0.2840 - val_accuracy: 0.8812
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1883 -
accuracy: 0.9268 - val_loss: 0.3499 - val_accuracy: 0.8540
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1598 -
accuracy: 0.9381 - val_loss: 0.3680 - val_accuracy: 0.8448
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1344 -
accuracy: 0.9478 - val_loss: 0.3652 - val_accuracy: 0.8664
79/79 [==============================] - 0s 1ms/step

Training Fold 6…
Epoch 1/10
704/704 [==============================] - 6s 5ms/step - loss: 0.6742 -
accuracy: 0.6592 - val_loss: 0.6134 - val_accuracy: 0.6596
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4764 -
```

```
accuracy: 0.7747 - val_loss: 0.6319 - val_accuracy: 0.6276
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3898 -
accuracy: 0.8283 - val_loss: 0.3741 - val_accuracy: 0.8292
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3314 -
accuracy: 0.8585 - val_loss: 0.3740 - val_accuracy: 0.8284
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2809 -
accuracy: 0.8836 - val_loss: 0.4943 - val_accuracy: 0.8096
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2430 -
accuracy: 0.9000 - val_loss: 0.3086 - val_accuracy: 0.8604
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2037 -
accuracy: 0.9205 - val_loss: 0.4817 - val_accuracy: 0.7944
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1695 -
accuracy: 0.9340 - val_loss: 0.3117 - val_accuracy: 0.8920
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1390 -
accuracy: 0.9464 - val_loss: 0.3278 - val_accuracy: 0.8884
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1188 -
accuracy: 0.9544 - val_loss: 0.6728 - val_accuracy: 0.8136
79/79 [==============================] - 0s 1ms/step

Training Fold 7…
Epoch 1/10
704/704 [==============================] - 6s 5ms/step - loss: 0.6896 -
accuracy: 0.6521 - val_loss: 0.5850 - val_accuracy: 0.6756
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4825 -
accuracy: 0.7749 - val_loss: 0.4169 - val_accuracy: 0.8100
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3996 -
accuracy: 0.8224 - val_loss: 0.5213 - val_accuracy: 0.7420
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3297 -
accuracy: 0.8586 - val_loss: 0.5712 - val_accuracy: 0.7296
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2881 -
accuracy: 0.8827 - val_loss: 0.3692 - val_accuracy: 0.8304
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2486 -
accuracy: 0.8954 - val_loss: 0.3364 - val_accuracy: 0.8552
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2050 -
```

```
accuracy: 0.9176 - val_loss: 0.4792 - val_accuracy: 0.8088
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1740 -
accuracy: 0.9328 - val_loss: 0.3700 - val_accuracy: 0.8524
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1570 -
accuracy: 0.9393 - val_loss: 0.3622 - val_accuracy: 0.8420
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1258 -
accuracy: 0.9516 - val_loss: 0.3691 - val_accuracy: 0.8800
79/79 [==============================] - 0s 2ms/step

Training Fold 8…
Epoch 1/10
704/704 [==============================] - 6s 5ms/step - loss: 0.6939 -
accuracy: 0.6491 - val_loss: 0.8504 - val_accuracy: 0.5504
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4799 -
accuracy: 0.7749 - val_loss: 0.4926 - val_accuracy: 0.7612
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3842 -
accuracy: 0.8270 - val_loss: 0.5012 - val_accuracy: 0.7612
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3225 -
accuracy: 0.8644 - val_loss: 0.4019 - val_accuracy: 0.8180
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3049 -
accuracy: 0.8722 - val_loss: 0.4277 - val_accuracy: 0.8072
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2385 -
accuracy: 0.9028 - val_loss: 0.4029 - val_accuracy: 0.8276
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1964 -
accuracy: 0.9183 - val_loss: 0.3111 - val_accuracy: 0.8700
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1715 -
accuracy: 0.9345 - val_loss: 0.4360 - val_accuracy: 0.7984
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1492 -
accuracy: 0.9462 - val_loss: 0.3014 - val_accuracy: 0.8784
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1221 -
accuracy: 0.9528 - val_loss: 0.5613 - val_accuracy: 0.8264
79/79 [==============================] - 0s 1ms/step

Training Fold 9…
Epoch 1/10
704/704 [==============================] - 6s 5ms/step - loss: 0.6611 -
```

```
accuracy: 0.6625 - val_loss: 0.5339 - val_accuracy: 0.7328
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4661 -
accuracy: 0.7829 - val_loss: 0.4953 - val_accuracy: 0.7588
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3761 -
accuracy: 0.8350 - val_loss: 0.5902 - val_accuracy: 0.7060
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3270 -
accuracy: 0.8644 - val_loss: 0.4398 - val_accuracy: 0.8104
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2792 -
accuracy: 0.8837 - val_loss: 0.3534 - val_accuracy: 0.8392
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2346 -
accuracy: 0.9051 - val_loss: 0.3127 - val_accuracy: 0.8580
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1991 -
accuracy: 0.9179 - val_loss: 0.3108 - val_accuracy: 0.8676
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1760 -
accuracy: 0.9324 - val_loss: 0.4270 - val_accuracy: 0.8044
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1465 -
accuracy: 0.9442 - val_loss: 0.3859 - val_accuracy: 0.8400
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1185 -
accuracy: 0.9532 - val_loss: 0.4268 - val_accuracy: 0.8504
79/79 [==============================] - 0s 1ms/step

Training Fold 10…
Epoch 1/10
704/704 [==============================] - 6s 5ms/step - loss: 0.6757 -
accuracy: 0.6567 - val_loss: 0.8315 - val_accuracy: 0.6380
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 0.4706 -
accuracy: 0.7789 - val_loss: 0.5616 - val_accuracy: 0.6848
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3870 -
accuracy: 0.8276 - val_loss: 0.4670 - val_accuracy: 0.7872
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 0.3269 -
accuracy: 0.8624 - val_loss: 0.3561 - val_accuracy: 0.8428
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2749 -
accuracy: 0.8864 - val_loss: 0.3516 - val_accuracy: 0.8488
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2460 -
```

```
accuracy: 0.8997 - val_loss: 0.3785 - val_accuracy: 0.8292
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.2090 -
accuracy: 0.9138 - val_loss: 0.7279 - val_accuracy: 0.7160
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1743 -
accuracy: 0.9302 - val_loss: 0.3336 - val_accuracy: 0.8560
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1486 -
accuracy: 0.9430 - val_loss: 0.3826 - val_accuracy: 0.8292
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.1384 -
accuracy: 0.9459 - val_loss: 0.4855 - val_accuracy: 0.8476
79/79 [==============================] - 0s 1ms/step

Results
AUC: 0.942 ± 0.010
Precision: 0.898 ± 0.039
Recall: 0.789 ± 0.078
F1-score: 0.836 ± 0.030
```



[33]:
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,␣
 ↪Dropout, BatchNormalization
from tensorflow.keras.regularizers import l2
from sklearn.model_selection import StratifiedKFold
```

```
from sklearn.metrics import roc_auc_score, precision_score, recall_score,␣
↪f1_score
import numpy as np
import matplotlib.pyplot as plt

# Arquitectura optimizada del modelo Deep CNN
def create_optimized_deep_cnn_model():
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3),␣
↪kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(64, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(128, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
        BatchNormalization(),
        MaxPooling2D(pool_size=(2, 2)),

        Flatten(),
        Dense(256, activation='relu', kernel_regularizer=l2(0.001)),
        Dropout(0.5),

        Dense(128, activation='relu', kernel_regularizer=l2(0.001)),
        Dropout(0.5),

        Dense(2, activation='softmax')  # Salida para clasificación binaria
    ])
    return model

# Función para entrenar y evaluar el modelo Deep CNN optimizado
def train_optimized_deep_cnn_model():
    skf = StratifiedKFold(n_splits=10)
    auc_scores = []
    precision_scores = []
    recall_scores = []
    f1_scores = []

    training_losses = []
    validation_losses = []

    fold = 0
    for train_index, test_index in skf.split(train_cnn, y):
        fold += 1
        print(f"\nTraining Fold {fold}...")
```

```
        X_train, X_test = train_cnn[train_index], train_cnn[test_index]
        y_train, y_test = y_categorical[train_index], y_categorical[test_index]

        model = create_optimized_deep_cnn_model()
        model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
                      loss='categorical_crossentropy', metrics=['accuracy'])

        history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=10, batch_size=32)

        # Guardar pérdidas para graficar
        training_losses.append(history.history['loss'])
        validation_losses.append(history.history['val_loss'])

        y_pred = model.predict(X_test)
        y_pred_class = y_pred.argmax(axis=1)
        y_true = y_test.argmax(axis=1)

        auc = roc_auc_score(y_true, y_pred[:, 1])
        precision = precision_score(y_true, y_pred_class)
        recall = recall_score(y_true, y_pred_class)
        f1 = f1_score(y_true, y_pred_class)

        auc_scores.append(auc)
        precision_scores.append(precision)
        recall_scores.append(recall)
        f1_scores.append(f1)

    # Mostrar resultados promedio
    print("\nResults")
    print(f"AUC: {np.mean(auc_scores):.3f} ± {np.std(auc_scores):.3f}")
    print(f"Precision: {np.mean(precision_scores):.3f} ± {np.std(precision_scores):.3f}")
    print(f"Recall: {np.mean(recall_scores):.3f} ± {np.std(recall_scores):.3f}")
    print(f"F1-score: {np.mean(f1_scores):.3f} ± {np.std(f1_scores):.3f}")

    # Graficar pérdidas de entrenamiento y validación
    plt.figure(figsize=(10, 5))
    plt.plot(np.mean(training_losses, axis=0), label="Training Loss")
    plt.plot(np.mean(validation_losses, axis=0), label="Validation Loss")
    plt.title("Training vs Validation Loss")
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()
    plt.show()
```

```python
# Entrenar el modelo Deep CNN optimizado
train_optimized_deep_cnn_model()
```

```
Training Fold 1…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 1.5924 -
accuracy: 0.5907 - val_loss: 1.3550 - val_accuracy: 0.6908
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3491 -
accuracy: 0.6761 - val_loss: 1.2621 - val_accuracy: 0.7264
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2568 -
accuracy: 0.7199 - val_loss: 1.2144 - val_accuracy: 0.7288
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1611 -
accuracy: 0.7581 - val_loss: 1.1027 - val_accuracy: 0.7720
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 1.0715 -
accuracy: 0.7860 - val_loss: 1.0331 - val_accuracy: 0.7864
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.9800 -
accuracy: 0.8060 - val_loss: 0.9574 - val_accuracy: 0.7952
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8953 -
accuracy: 0.8238 - val_loss: 0.9037 - val_accuracy: 0.8024
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8124 -
accuracy: 0.8381 - val_loss: 0.8438 - val_accuracy: 0.8084
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7385 -
accuracy: 0.8564 - val_loss: 0.7851 - val_accuracy: 0.8204
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6640 -
accuracy: 0.8724 - val_loss: 0.7479 - val_accuracy: 0.8212
79/79 [==============================] - 0s 1ms/step

Training Fold 2…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 1.5606 -
accuracy: 0.5929 - val_loss: 1.3606 - val_accuracy: 0.6684
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3399 -
accuracy: 0.6778 - val_loss: 1.2530 - val_accuracy: 0.7492
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2384 -
accuracy: 0.7230 - val_loss: 1.1494 - val_accuracy: 0.7820
```

```
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1403 -
accuracy: 0.7614 - val_loss: 1.0882 - val_accuracy: 0.7852
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 1.0410 -
accuracy: 0.7889 - val_loss: 1.0038 - val_accuracy: 0.7980
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.9480 -
accuracy: 0.8100 - val_loss: 0.9180 - val_accuracy: 0.8152
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8567 -
accuracy: 0.8313 - val_loss: 0.8494 - val_accuracy: 0.8200
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7694 -
accuracy: 0.8497 - val_loss: 0.7974 - val_accuracy: 0.8196
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6948 -
accuracy: 0.8648 - val_loss: 0.7507 - val_accuracy: 0.8236
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6228 -
accuracy: 0.8824 - val_loss: 0.7397 - val_accuracy: 0.8228
79/79 [==============================] - 0s 1ms/step

Training Fold 3…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 1.5738 -
accuracy: 0.5974 - val_loss: 1.3554 - val_accuracy: 0.6860
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3475 -
accuracy: 0.6736 - val_loss: 1.2650 - val_accuracy: 0.7336
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2550 -
accuracy: 0.7219 - val_loss: 1.1925 - val_accuracy: 0.7396
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1566 -
accuracy: 0.7527 - val_loss: 1.1025 - val_accuracy: 0.7800
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 1.0638 -
accuracy: 0.7847 - val_loss: 1.0622 - val_accuracy: 0.7576
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.9710 -
accuracy: 0.8061 - val_loss: 0.9474 - val_accuracy: 0.8052
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8773 -
accuracy: 0.8310 - val_loss: 0.8949 - val_accuracy: 0.8096
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7896 -
accuracy: 0.8491 - val_loss: 0.8165 - val_accuracy: 0.8184
```

```
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7144 -
accuracy: 0.8623 - val_loss: 0.7897 - val_accuracy: 0.8184
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6423 -
accuracy: 0.8757 - val_loss: 0.7475 - val_accuracy: 0.8164
79/79 [==============================] - 0s 1ms/step

Training Fold 4…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 1.5471 -
accuracy: 0.6019 - val_loss: 1.3691 - val_accuracy: 0.6712
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3344 -
accuracy: 0.6866 - val_loss: 1.2614 - val_accuracy: 0.7312
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2283 -
accuracy: 0.7373 - val_loss: 1.2015 - val_accuracy: 0.7348
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1257 -
accuracy: 0.7736 - val_loss: 1.0826 - val_accuracy: 0.7848
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 1.0327 -
accuracy: 0.7960 - val_loss: 1.0219 - val_accuracy: 0.7832
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.9380 -
accuracy: 0.8184 - val_loss: 0.9297 - val_accuracy: 0.8044
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8438 -
accuracy: 0.8415 - val_loss: 0.8610 - val_accuracy: 0.8040
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7581 -
accuracy: 0.8540 - val_loss: 0.8164 - val_accuracy: 0.8100
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6823 -
accuracy: 0.8702 - val_loss: 0.7718 - val_accuracy: 0.8140
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6154 -
accuracy: 0.8846 - val_loss: 0.7766 - val_accuracy: 0.8116
79/79 [==============================] - 0s 1ms/step

Training Fold 5…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 1.5643 -
accuracy: 0.6004 - val_loss: 1.3650 - val_accuracy: 0.6768
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3413 -
accuracy: 0.6754 - val_loss: 1.2504 - val_accuracy: 0.7428
```

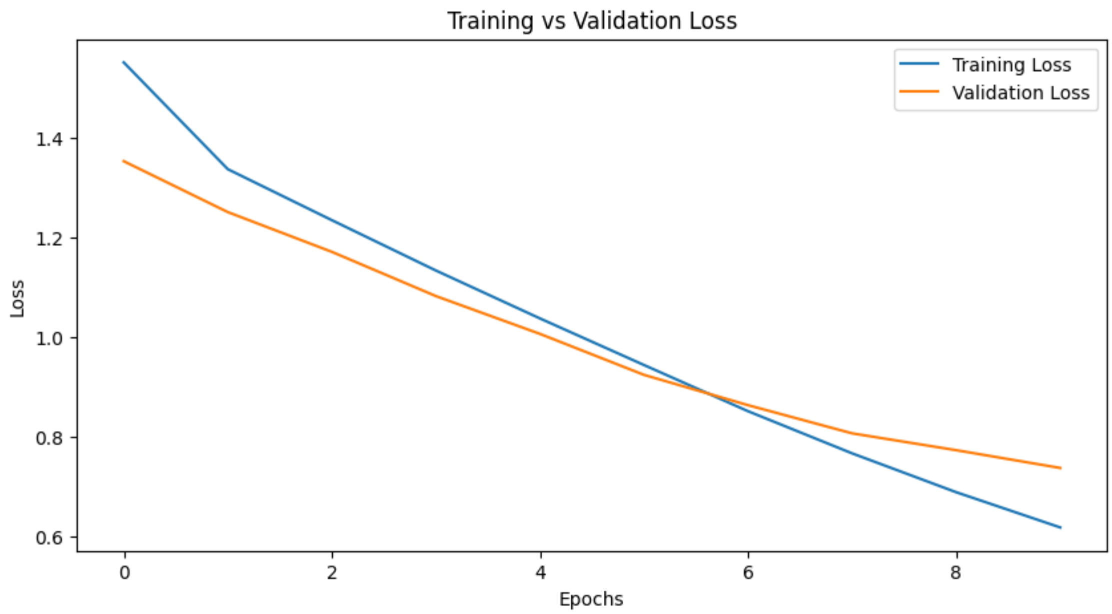
```
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2323 -
accuracy: 0.7313 - val_loss: 1.1727 - val_accuracy: 0.7584
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1298 -
accuracy: 0.7670 - val_loss: 1.0680 - val_accuracy: 0.7980
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 1.0369 -
accuracy: 0.7936 - val_loss: 0.9922 - val_accuracy: 0.8144
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.9445 -
accuracy: 0.8172 - val_loss: 0.9330 - val_accuracy: 0.7996
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8499 -
accuracy: 0.8344 - val_loss: 0.8491 - val_accuracy: 0.8272
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7681 -
accuracy: 0.8530 - val_loss: 0.7812 - val_accuracy: 0.8400
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6901 -
accuracy: 0.8696 - val_loss: 0.7592 - val_accuracy: 0.8388
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6245 -
accuracy: 0.8864 - val_loss: 0.7483 - val_accuracy: 0.8304
79/79 [==============================] - 0s 1ms/step

Training Fold 6…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 1.5418 -
accuracy: 0.6004 - val_loss: 1.3489 - val_accuracy: 0.6976
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3382 -
accuracy: 0.6792 - val_loss: 1.2351 - val_accuracy: 0.7536
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2293 -
accuracy: 0.7277 - val_loss: 1.1417 - val_accuracy: 0.7768
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1270 -
accuracy: 0.7694 - val_loss: 1.0482 - val_accuracy: 0.7936
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 1.0244 -
accuracy: 0.7962 - val_loss: 0.9609 - val_accuracy: 0.8092
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.9261 -
accuracy: 0.8210 - val_loss: 0.8859 - val_accuracy: 0.8244
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8326 -
accuracy: 0.8400 - val_loss: 0.8121 - val_accuracy: 0.8376
```

```
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7488 -
accuracy: 0.8546 - val_loss: 0.7568 - val_accuracy: 0.8400
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6644 -
accuracy: 0.8785 - val_loss: 0.7264 - val_accuracy: 0.8312
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.5926 -
accuracy: 0.8911 - val_loss: 0.6810 - val_accuracy: 0.8432
79/79 [==============================] - 0s 1ms/step

Training Fold 7…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 1.5419 -
accuracy: 0.6004 - val_loss: 1.3500 - val_accuracy: 0.6956
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3303 -
accuracy: 0.6871 - val_loss: 1.2415 - val_accuracy: 0.7396
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2236 -
accuracy: 0.7352 - val_loss: 1.1524 - val_accuracy: 0.7584
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1269 -
accuracy: 0.7677 - val_loss: 1.0628 - val_accuracy: 0.7884
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 1.0252 -
accuracy: 0.7934 - val_loss: 0.9829 - val_accuracy: 0.7992
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.9298 -
accuracy: 0.8149 - val_loss: 0.9057 - val_accuracy: 0.8100
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8294 -
accuracy: 0.8398 - val_loss: 0.8513 - val_accuracy: 0.8172
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7429 -
accuracy: 0.8552 - val_loss: 0.7867 - val_accuracy: 0.8260
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6652 -
accuracy: 0.8744 - val_loss: 0.7452 - val_accuracy: 0.8228
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.5893 -
accuracy: 0.8901 - val_loss: 0.7175 - val_accuracy: 0.8224
79/79 [==============================] - 0s 1ms/step

Training Fold 8…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 1.5331 -
accuracy: 0.6011 - val_loss: 1.3339 - val_accuracy: 0.6896
```

```
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3295 -
accuracy: 0.6851 - val_loss: 1.2390 - val_accuracy: 0.7336
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2243 -
accuracy: 0.7404 - val_loss: 1.1514 - val_accuracy: 0.7708
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1253 -
accuracy: 0.7705 - val_loss: 1.0859 - val_accuracy: 0.7636
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 1.0290 -
accuracy: 0.7950 - val_loss: 0.9889 - val_accuracy: 0.7932
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.9367 -
accuracy: 0.8163 - val_loss: 0.9167 - val_accuracy: 0.8020
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8397 -
accuracy: 0.8361 - val_loss: 0.8570 - val_accuracy: 0.8112
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7572 -
accuracy: 0.8509 - val_loss: 0.8510 - val_accuracy: 0.7920
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6773 -
accuracy: 0.8698 - val_loss: 0.7689 - val_accuracy: 0.8160
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6024 -
accuracy: 0.8857 - val_loss: 0.7160 - val_accuracy: 0.8288
79/79 [==============================] - 0s 1ms/step

Training Fold 9…
Epoch 1/10
704/704 [==============================] - 6s 4ms/step - loss: 1.5319 -
accuracy: 0.6000 - val_loss: 1.3488 - val_accuracy: 0.6840
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3312 -
accuracy: 0.6860 - val_loss: 1.2594 - val_accuracy: 0.7288
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2310 -
accuracy: 0.7253 - val_loss: 1.1763 - val_accuracy: 0.7508
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1241 -
accuracy: 0.7681 - val_loss: 1.1093 - val_accuracy: 0.7568
Epoch 5/10
704/704 [==============================] - 3s 4ms/step - loss: 1.0248 -
accuracy: 0.7967 - val_loss: 1.0189 - val_accuracy: 0.7824
Epoch 6/10
704/704 [==============================] - 3s 4ms/step - loss: 0.9312 -
accuracy: 0.8176 - val_loss: 0.9239 - val_accuracy: 0.7984
```

```
Epoch 7/10
704/704 [==============================] - 3s 4ms/step - loss: 0.8392 -
accuracy: 0.8340 - val_loss: 0.8993 - val_accuracy: 0.7840
Epoch 8/10
704/704 [==============================] - 3s 4ms/step - loss: 0.7540 -
accuracy: 0.8565 - val_loss: 0.8222 - val_accuracy: 0.8100
Epoch 9/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6774 -
accuracy: 0.8726 - val_loss: 0.7588 - val_accuracy: 0.8268
Epoch 10/10
704/704 [==============================] - 3s 4ms/step - loss: 0.6098 -
accuracy: 0.8888 - val_loss: 0.7544 - val_accuracy: 0.8164
79/79 [==============================] - 0s 1ms/step

Training Fold 10…
Epoch 1/10
704/704 [==============================] - 5s 4ms/step - loss: 1.5300 -
accuracy: 0.6028 - val_loss: 1.3444 - val_accuracy: 0.7028
Epoch 2/10
704/704 [==============================] - 3s 4ms/step - loss: 1.3292 -
accuracy: 0.6872 - val_loss: 1.2413 - val_accuracy: 0.7504
Epoch 3/10
704/704 [==============================] - 3s 4ms/step - loss: 1.2269 -
accuracy: 0.7314 - val_loss: 1.1571 - val_accuracy: 0.7636
Epoch 4/10
704/704 [==============================] - 3s 4ms/step - loss: 1.1172 -
accuracy: 0.7735 - val_loss: 1.0702 - val_accuracy: 0.7872
Epoch 5/10
704/704 [==============================] - 4s 5ms/step - loss: 1.0219 -
accuracy: 0.7985 - val_loss: 0.9960 - val_accuracy: 0.8008
Epoch 6/10
704/704 [==============================] - 4s 5ms/step - loss: 0.9295 -
accuracy: 0.8216 - val_loss: 0.9193 - val_accuracy: 0.8088
Epoch 7/10
704/704 [==============================] - 4s 6ms/step - loss: 0.8424 -
accuracy: 0.8394 - val_loss: 0.8501 - val_accuracy: 0.8212
Epoch 8/10
704/704 [==============================] - 4s 6ms/step - loss: 0.7589 -
accuracy: 0.8583 - val_loss: 0.7896 - val_accuracy: 0.8208
Epoch 9/10
704/704 [==============================] - 4s 5ms/step - loss: 0.6743 -
accuracy: 0.8769 - val_loss: 0.8673 - val_accuracy: 0.7836
Epoch 10/10
704/704 [==============================] - 4s 5ms/step - loss: 0.6072 -
accuracy: 0.8921 - val_loss: 0.7371 - val_accuracy: 0.8224
79/79 [==============================] - 0s 2ms/step

Results
```

```
AUC: 0.911 ± 0.007
Precision: 0.816 ± 0.033
Recall: 0.841 ± 0.042
F1-score: 0.826 ± 0.009
```



[ ]: