another-copy-of-autoencoders

November 11, 2024

1 Tarea 6: Autoencoders Sparse y Denoising

Objetivo: Implementar arquitecturas autoencoder eficientes para analizar datos de imágenes y maximizar la clasificación de individuos.

1.1 Importar Librerías

[ ]:
```
import os  # Para manejo de directorios
import numpy as np  # Para manejo de arrays
import matplotlib.pyplot as plt  # Para manejo de gráficos
import tensorflow as tf  # Para manejo de redes neuronales
from tensorflow.keras.preprocessing.image import load_img, img_to_array  # Para␣
↪manejo de imágenes
from tensorflow.keras.preprocessing.image import ImageDataGenerator  # Para␣
↪aumento de datos
import random  # Para manejo de números aleatorios
import pandas as pd  # Para manejo de DataFrames
from PIL import Image  # Para manejo de imágenes
from sklearn.model_selection import train_test_split  # Para dividir el dataset
from tensorflow.keras.layers import Input, Dense, GaussianNoise  # Para capas␣
↪de redes neuronales
from tensorflow.keras.models import Model  # Para crear modelos
from tensorflow.keras.regularizers import l1  # Para regularización L1
from tensorflow.keras.callbacks import EarlyStopping  # Para Early Stopping
```

1.2 Carga de Datos

[ ]:
```
# Reemplaza el enlace con el enlace directo obtenido
!wget -O face_dataset.zip "https://github.com/Negatix092/DM1/raw/main/
↪Autoencoders/face_dataset.zip"
```

```
--2024-11-10 22:59:43--
https://github.com/Negatix092/DM1/raw/main/Autoencoders/face_dataset.zip
Resolving github.com (github.com)… 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443… connected.
HTTP request sent, awaiting response… 302 Found
Location: https://raw.githubusercontent.com/Negatix092/DM1/main/Autoencoders/fac
```

```
e_dataset.zip [following]
--2024-11-10 22:59:44--  https://raw.githubusercontent.com/Negatix092/DM1/main/A
utoencoders/face_dataset.zip
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.109.133, 185.199.108.133, 185.199.110.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.109.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 2666608 (2.5M) [application/zip]
Saving to: 'face_dataset.zip'

face_dataset.zip    100%[===================>]   2.54M  --.-KB/s    in 0.1s

2024-11-10 22:59:44 (25.0 MB/s) - 'face_dataset.zip' saved [2666608/2666608]
```

[ ]:
```
# Descomprime el archivo en el directorio actual
!unzip face_dataset.zip -d face_dataset
```

```
Archive:  face_dataset.zip
replace face_dataset/__MACOSX/._face_dataset? [y]es, [n]o, [A]ll, [N]one,
[r]ename:
```

[ ]:
```
# Listar el contenido de la carpeta descomprimida para asegurarte de que todo␣
 ↪esté correcto
!ls face_dataset
```

```
face_dataset  __MACOSX
```

[ ]:
```
# Listar el contenido de la carpeta para verificar la existencia de las␣
 ↪carpetad de train & test
!ls face_dataset/face_dataset/
```

```
test  train
```

---

[23]:
```
import os  # Para manejo de directorios
import numpy as np  # Para manejo de arrays
import matplotlib.pyplot as plt  # Para manejo de gráficos
import tensorflow as tf  # Para manejo de redes neuronales
from tensorflow.keras.preprocessing.image import load_img, img_to_array  # Para␣
 ↪manejo de imágenes
from tensorflow.keras.preprocessing.image import ImageDataGenerator  # Para␣
 ↪aumento de datos
import random  # Para manejo de números aleatorios
import pandas as pd  # Para manejo de DataFrames
from PIL import Image  # Para manejo de imágenes
```

```
from sklearn.model_selection import train_test_split  # Para dividir el dataset
from tensorflow.keras.layers import Input, Dense, GaussianNoise  # Para capas␣
 ↪de redes neuronales
from tensorflow.keras.models import Model  # Para crear modelos
from tensorflow.keras.regularizers import l1  # Para regularización L1
from tensorflow.keras.callbacks import EarlyStopping  # Para Early Stopping
```

[24]:
```
# Definir las rutas y cargar las imágenes
train_dir = '/content/face_dataset/face_dataset/train'
test_dir = '/content/face_dataset/face_dataset/test'

# Definir el tamaño de las imágenes
img_size = (128, 128)

# Función para cargar las imágenes
def cargar_imagenes(ruta, img_size=(128, 128)):
    imagenes = []
    for img_name in os.listdir(ruta):
        img_path = os.path.join(ruta, img_name)
        img = load_img(img_path, target_size=img_size)
        img_array = img_to_array(img)
        imagenes.append(img_array)
    return np.array(imagenes)

# Cargar imágenes de entrenamiento y prueba
X_train = cargar_imagenes(train_dir, img_size)
X_test = cargar_imagenes(test_dir, img_size)

# Imprimir estadísticas antes de la normalización
print("---------------Antes de la Normalización---------------")
print("\nX_train")
print("Valor mínimo: ", np.min(X_train))
print("Valor máximo: ", np.max(X_train))
print("Valor medio:  ", np.mean(X_train))

print("\nX_test")
print("Valor mínimo: ", np.min(X_test))
print("Valor máximo: ", np.max(X_test))
print("Valor medio:  ", np.mean(X_test))

# Normalizar los valores de los píxeles a [0, 1]
X_train = X_train / 255.0
X_test = X_test / 255.0

# Dividir el dataset de entrenamiento en entrenamiento y validación (80%␣
 ↪entrenamiento, 20% validación)
X_train, X_val = train_test_split(X_train, test_size=0.2, random_state=26)
```

```python
# Imprimir tamaño de cada conjunto después de la división
print("\nTamaño del conjunto de entrenamiento final: ", X_train.shape)
print("Tamaño del conjunto de validación:           ", X_val.shape)
print("Tamaño del conjunto de prueba:               ", X_test.shape)

# Imprimir estadísticas después de la normalización
print("\n---------------Después de la Normalización---------------")
print("\nX_train")
print("Valor mínimo: ", np.min(X_train))
print("Valor máximo: ", np.max(X_train))
print("Valor medio:  ", np.mean(X_train))

print("\nX_val")
print("Valor mínimo: ", np.min(X_val))
print("Valor máximo: ", np.max(X_val))
print("Valor medio:  ", np.mean(X_val))

print("\nX_test")
print("Valor mínimo:", np.min(X_test))
print("Valor máximo:", np.max(X_test))
print("Valor medio: ", np.mean(X_test))
```

```
---------------Antes de la Normalización---------------

X_train
Valor mínimo:  0.0
Valor máximo:  255.0
Valor medio:   111.66865

X_test
Valor mínimo:  0.0
Valor máximo:  255.0
Valor medio:   106.77395

Tamaño del conjunto de entrenamiento final:  (160, 128, 128, 3)
Tamaño del conjunto de validación:            (40, 128, 128, 3)
Tamaño del conjunto de prueba:                (12, 128, 128, 3)

---------------Después de la Normalización---------------

X_train
Valor mínimo:  0.0
Valor máximo:  1.0
Valor medio:   0.43824565

X_val
```

```
Valor mínimo:  0.0
Valor máximo:  1.0
Valor medio:   0.43659905

X_test
Valor mínimo: 0.0
Valor máximo: 1.0
Valor medio:  0.4187213
```

Modelos de Autoencoders Convolucionales

Denoising

[51]:
```python
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Conv2D, MaxPooling2D, UpSampling2D, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.optimizers import Adam
from skimage.metrics import peak_signal_noise_ratio as psnr
from skimage.metrics import structural_similarity as ssim

# EarlyStopping para monitorear la pérdida de validación
early_stopping = EarlyStopping(monitor='val_loss', patience=10,␣
 ↪restore_best_weights=True)

# Configuraciones de Autoencoder Convolucional Denoising
denoising_configs = [
    {'encoding_dim': 32, 'activation': 'relu', 'noise_factor': 0.1},
    {'encoding_dim': 64, 'activation': 'relu', 'noise_factor': 0.2},
    {'encoding_dim': 128, 'activation': 'relu', 'noise_factor': 0.3},
]

# Redimensionar los datos de entrada
X_train = X_train.reshape(-1, 128, 128, 3)
X_val = X_val.reshape(-1, 128, 128, 3)

# Entrenamiento del Autoencoder Convolucional Denoising con diferentes␣
 ↪configuraciones
denoising_histories = []
# Lista para guardar los modelos Denoising Autoencoder
denoising_models = []

for config in denoising_configs:
    encoding_dim = config['encoding_dim']
    activation_fn = config['activation']
    noise_factor = config['noise_factor']
```

```
    # Añadir ruido a los datos de entrenamiento y validación para denoising
    X_train_noisy = X_train + noise_factor * np.random.normal(loc=0.0, scale=1.
↪0, size=X_train.shape)
    X_val_noisy = X_val + noise_factor * np.random.normal(loc=0.0, scale=1.0,␣
↪size=X_val.shape)
    X_train_noisy = np.clip(X_train_noisy, 0., 1.)
    X_val_noisy = np.clip(X_val_noisy, 0., 1.)

    # Definir el modelo de Autoencoder Convolucional
    input_img = Input(shape=(128, 128, 3))  # Respetar la forma redimensionada

    # Encoder
    x = Conv2D(64, (3, 3), activation=activation_fn, padding='same')(input_img)
    x = MaxPooling2D((2, 2), padding='same')(x)
    x = Dropout(0.1)(x)
    x = Conv2D(128, (3, 3), activation=activation_fn, padding='same')(x)
    x = MaxPooling2D((2, 2), padding='same')(x)
    x = Dropout(0.1)(x)
    encoded = Conv2D(encoding_dim, (3, 3), activation=activation_fn,␣
↪padding='same')(x)

    # Decoder
    x = Conv2D(encoding_dim, (3, 3), activation=activation_fn,␣
↪padding='same')(encoded)
    x = UpSampling2D((2, 2))(x)
    x = Conv2D(128, (3, 3), activation=activation_fn, padding='same')(x)
    x = UpSampling2D((2, 2))(x)
    x = Conv2D(64, (3, 3), activation=activation_fn, padding='same')(x)
    decoded = Conv2D(3, (3, 3), activation='sigmoid', padding='same')(x)

    autoencoder = Model(input_img, decoded)
    autoencoder.compile(optimizer=Adam(learning_rate=1e-4),␣
↪loss='mean_absolute_error')

    # Entrenar el Autoencoder Convolucional de Denoising
    print(f"Entrenando Denoising AE con Dim: {encoding_dim}, Noise Factor:␣
↪{noise_factor}")
    history = autoencoder.fit(
        X_train_noisy, X_train,  # Usar entrada ruidosa y salida limpia␣
↪redimensionada
        epochs=70,
        batch_size=16,
        shuffle=True,
        validation_data=(X_val_noisy, X_val),  # Usar datos de validación␣
↪ruidosos y salida limpia redimensionada
```

```
        callbacks=[early_stopping],
        verbose=1
    )
    denoising_histories.append((config, history))
    denoising_models.append(autoencoder)  # Guardar el modelo específico para
↪cada configuración
```

```
Entrenando Denoising AE con Dim: 32, Noise Factor: 0.1
Epoch 1/70
10/10 [==============================] - 5s 396ms/step - loss: 0.1990 -
val_loss: 0.1905
Epoch 2/70
10/10 [==============================] - 0s 29ms/step - loss: 0.1937 - val_loss:
0.1840
Epoch 3/70
10/10 [==============================] - 0s 29ms/step - loss: 0.1829 - val_loss:
0.1659
Epoch 4/70
10/10 [==============================] - 0s 27ms/step - loss: 0.1612 - val_loss:
0.1476
Epoch 5/70
10/10 [==============================] - 0s 27ms/step - loss: 0.1269 - val_loss:
0.1273
Epoch 6/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0991 - val_loss:
0.0981
Epoch 7/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0858 - val_loss:
0.0933
Epoch 8/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0783 - val_loss:
0.0737
Epoch 9/70
10/10 [==============================] - 0s 26ms/step - loss: 0.0719 - val_loss:
0.0766
Epoch 10/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0671 - val_loss:
0.0707
Epoch 11/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0633 - val_loss:
0.0597
Epoch 12/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0623 - val_loss:
0.0523
Epoch 13/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0612 - val_loss:
0.0605
```

```
Epoch 14/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0591 - val_loss:
0.0669
Epoch 15/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0571 - val_loss:
0.0525
Epoch 16/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0555 - val_loss:
0.0491
Epoch 17/70
10/10 [==============================] - 0s 26ms/step - loss: 0.0541 - val_loss:
0.0522
Epoch 18/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0533 - val_loss:
0.0489
Epoch 19/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0523 - val_loss:
0.0457
Epoch 20/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0523 - val_loss:
0.0455
Epoch 21/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0514 - val_loss:
0.0453
Epoch 22/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0505 - val_loss:
0.0452
Epoch 23/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0498 - val_loss:
0.0437
Epoch 24/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0493 - val_loss:
0.0436
Epoch 25/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0490 - val_loss:
0.0432
Epoch 26/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0488 - val_loss:
0.0433
Epoch 27/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0478 - val_loss:
0.0432
Epoch 28/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0472 - val_loss:
0.0435
Epoch 29/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0470 - val_loss:
0.0438
```

```
Epoch 30/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0471 - val_loss:
0.0427
Epoch 31/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0470 - val_loss:
0.0448
Epoch 32/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0457 - val_loss:
0.0465
Epoch 33/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0455 - val_loss:
0.0477
Epoch 34/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0450 - val_loss:
0.0458
Epoch 35/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0445 - val_loss:
0.0449
Epoch 36/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0442 - val_loss:
0.0485
Epoch 37/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0440 - val_loss:
0.0456
Epoch 38/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0436 - val_loss:
0.0464
Epoch 39/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0432 - val_loss:
0.0445
Epoch 40/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0433 - val_loss:
0.0439
Entrenando Denoising AE con Dim: 64, Noise Factor: 0.2
Epoch 1/70
10/10 [==============================] - 5s 140ms/step - loss: 0.1969 -
val_loss: 0.1851
Epoch 2/70
10/10 [==============================] - 0s 25ms/step - loss: 0.1818 - val_loss:
0.1610
Epoch 3/70
10/10 [==============================] - 0s 28ms/step - loss: 0.1413 - val_loss:
0.1151
Epoch 4/70
10/10 [==============================] - 0s 27ms/step - loss: 0.1037 - val_loss:
0.0951
Epoch 5/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0866 - val_loss:
```

```
0.0987
Epoch 6/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0787 - val_loss:
0.0915
Epoch 7/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0735 - val_loss:
0.0815
Epoch 8/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0693 - val_loss:
0.0759
Epoch 9/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0677 - val_loss:
0.0677
Epoch 10/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0648 - val_loss:
0.0775
Epoch 11/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0621 - val_loss:
0.0754
Epoch 12/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0600 - val_loss:
0.0667
Epoch 13/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0582 - val_loss:
0.0595
Epoch 14/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0569 - val_loss:
0.0619
Epoch 15/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0557 - val_loss:
0.0650
Epoch 16/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0552 - val_loss:
0.0633
Epoch 17/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0551 - val_loss:
0.0621
Epoch 18/70
10/10 [==============================] - 0s 27ms/step - loss: 0.0538 - val_loss:
0.0602
Epoch 19/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0532 - val_loss:
0.0526
Epoch 20/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0522 - val_loss:
0.0526
Epoch 21/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0515 - val_loss:
```

```
0.0514
Epoch 22/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0514 - val_loss:
0.0485
Epoch 23/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0510 - val_loss:
0.0458
Epoch 24/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0503 - val_loss:
0.0489
Epoch 25/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0495 - val_loss:
0.0479
Epoch 26/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0491 - val_loss:
0.0456
Epoch 27/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0487 - val_loss:
0.0444
Epoch 28/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0484 - val_loss:
0.0450
Epoch 29/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0479 - val_loss:
0.0442
Epoch 30/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0476 - val_loss:
0.0432
Epoch 31/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0474 - val_loss:
0.0441
Epoch 32/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0472 - val_loss:
0.0436
Epoch 33/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0464 - val_loss:
0.0427
Epoch 34/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0465 - val_loss:
0.0438
Epoch 35/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0461 - val_loss:
0.0424
Epoch 36/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0457 - val_loss:
0.0420
Epoch 37/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0452 - val_loss:
```

```
0.0416
Epoch 38/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0448 - val_loss:
0.0414
Epoch 39/70
10/10 [==============================] - 0s 36ms/step - loss: 0.0445 - val_loss:
0.0414
Epoch 40/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0442 - val_loss:
0.0414
Epoch 41/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0440 - val_loss:
0.0412
Epoch 42/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0438 - val_loss:
0.0413
Epoch 43/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0433 - val_loss:
0.0414
Epoch 44/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0431 - val_loss:
0.0410
Epoch 45/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0428 - val_loss:
0.0408
Epoch 46/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0424 - val_loss:
0.0414
Epoch 47/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0423 - val_loss:
0.0409
Epoch 48/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0422 - val_loss:
0.0428
Epoch 49/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0418 - val_loss:
0.0415
Epoch 50/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0416 - val_loss:
0.0425
Epoch 51/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0415 - val_loss:
0.0406
Epoch 52/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0411 - val_loss:
0.0411
Epoch 53/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0408 - val_loss:
```

```
0.0415
Epoch 54/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0406 - val_loss:
0.0418
Epoch 55/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0404 - val_loss:
0.0421
Epoch 56/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0402 - val_loss:
0.0408
Epoch 57/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0401 - val_loss:
0.0407
Epoch 58/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0400 - val_loss:
0.0409
Epoch 59/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0397 - val_loss:
0.0432
Epoch 60/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0396 - val_loss:
0.0431
Epoch 61/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0395 - val_loss:
0.0401
Epoch 62/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0394 - val_loss:
0.0406
Epoch 63/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0392 - val_loss:
0.0404
Epoch 64/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0389 - val_loss:
0.0409
Epoch 65/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0388 - val_loss:
0.0428
Epoch 66/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0386 - val_loss:
0.0425
Epoch 67/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0385 - val_loss:
0.0416
Epoch 68/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0383 - val_loss:
0.0422
Epoch 69/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0382 - val_loss:
```

```
0.0419
Epoch 70/70
10/10 [==============================] - 0s 29ms/step - loss: 0.0382 - val_loss:
0.0401
Entrenando Denoising AE con Dim: 128, Noise Factor: 0.3
Epoch 1/70
10/10 [==============================] - 5s 161ms/step - loss: 0.1969 -
val_loss: 0.1822
Epoch 2/70
10/10 [==============================] - 0s 32ms/step - loss: 0.1749 - val_loss:
0.1425
Epoch 3/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1209 - val_loss:
0.1038
Epoch 4/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0958 - val_loss:
0.0952
Epoch 5/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0844 - val_loss:
0.0985
Epoch 6/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0781 - val_loss:
0.0880
Epoch 7/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0733 - val_loss:
0.0748
Epoch 8/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0704 - val_loss:
0.0764
Epoch 9/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0659 - val_loss:
0.0833
Epoch 10/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0630 - val_loss:
0.0844
Epoch 11/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0605 - val_loss:
0.0827
Epoch 12/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0590 - val_loss:
0.0676
Epoch 13/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0574 - val_loss:
0.0697
Epoch 14/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0563 - val_loss:
0.0710
Epoch 15/70
```

```
10/10 [==============================] - 0s 33ms/step - loss: 0.0558 - val_loss:
0.0646
Epoch 16/70
10/10 [==============================] - 0s 33ms/step - loss: 0.0555 - val_loss:
0.0648
Epoch 17/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0537 - val_loss:
0.0663
Epoch 18/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0539 - val_loss:
0.0605
Epoch 19/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0533 - val_loss:
0.0564
Epoch 20/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0519 - val_loss:
0.0577
Epoch 21/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0514 - val_loss:
0.0521
Epoch 22/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0512 - val_loss:
0.0553
Epoch 23/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0505 - val_loss:
0.0543
Epoch 24/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0500 - val_loss:
0.0500
Epoch 25/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0496 - val_loss:
0.0507
Epoch 26/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0491 - val_loss:
0.0486
Epoch 27/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0495 - val_loss:
0.0463
Epoch 28/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0488 - val_loss:
0.0486
Epoch 29/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0477 - val_loss:
0.0488
Epoch 30/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0475 - val_loss:
0.0464
Epoch 31/70
```

```
10/10 [==============================] - 0s 33ms/step - loss: 0.0469 - val_loss:
0.0454
Epoch 32/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0465 - val_loss:
0.0457
Epoch 33/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0463 - val_loss:
0.0449
Epoch 34/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0459 - val_loss:
0.0456
Epoch 35/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0456 - val_loss:
0.0439
Epoch 36/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0450 - val_loss:
0.0439
Epoch 37/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0449 - val_loss:
0.0424
Epoch 38/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0447 - val_loss:
0.0429
Epoch 39/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0443 - val_loss:
0.0428
Epoch 40/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0442 - val_loss:
0.0429
Epoch 41/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0440 - val_loss:
0.0422
Epoch 42/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0436 - val_loss:
0.0426
Epoch 43/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0434 - val_loss:
0.0419
Epoch 44/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0432 - val_loss:
0.0425
Epoch 45/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0431 - val_loss:
0.0421
Epoch 46/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0430 - val_loss:
0.0420
Epoch 47/70
```

```
10/10 [==============================] - 0s 32ms/step - loss: 0.0430 - val_loss:
0.0415
Epoch 48/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0427 - val_loss:
0.0422
Epoch 49/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0424 - val_loss:
0.0420
Epoch 50/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0424 - val_loss:
0.0436
Epoch 51/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0425 - val_loss:
0.0427
Epoch 52/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0421 - val_loss:
0.0415
Epoch 53/70
10/10 [==============================] - 0s 33ms/step - loss: 0.0418 - val_loss:
0.0417
Epoch 54/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0416 - val_loss:
0.0411
Epoch 55/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0415 - val_loss:
0.0415
Epoch 56/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0413 - val_loss:
0.0418
Epoch 57/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0413 - val_loss:
0.0407
Epoch 58/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0410 - val_loss:
0.0417
Epoch 59/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0410 - val_loss:
0.0415
Epoch 60/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0409 - val_loss:
0.0421
Epoch 61/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0409 - val_loss:
0.0424
Epoch 62/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0407 - val_loss:
0.0425
Epoch 63/70
```

```
10/10 [==============================] - 0s 31ms/step - loss: 0.0406 - val_loss:
0.0415
Epoch 64/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0405 - val_loss:
0.0438
Epoch 65/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0406 - val_loss:
0.0419
Epoch 66/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0403 - val_loss:
0.0419
Epoch 67/70
10/10 [==============================] - 0s 33ms/step - loss: 0.0401 - val_loss:
0.0425
```

[52]:
```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Conv2D, MaxPooling2D, UpSampling2D, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l1
from skimage.metrics import peak_signal_noise_ratio as psnr
from skimage.metrics import structural_similarity as ssim

# EarlyStopping para monitorear la pérdida de validación
early_stopping = EarlyStopping(monitor='val_loss', patience=10,␣
 ↪restore_best_weights=True)

# Configuraciones de Autoencoder Convolucional Sparsity
sparse_configs = [
    {'encoding_dim': 32, 'activation': 'relu', 'regularization': 1e-5},
    {'encoding_dim': 64, 'activation': 'relu', 'regularization': 1e-4},
    {'encoding_dim': 128, 'activation': 'relu', 'regularization': 1e-3},
]

# Entrenamiento del Autoencoder Convolucional Sparsity con diferentes␣
 ↪configuraciones
sparse_histories = []
# Lista para guardar los modelos Sparse Autoencoder
sparse_models = []


for config in sparse_configs:
    encoding_dim = config['encoding_dim']
    activation_fn = config['activation']
```

```
    regularization = config['regularization']

    # Definir el modelo de Autoencoder Convolucional
    input_img = Input(shape=(128, 128, 3))  # Respetar la forma redimensionada

    # Encoder
    x = Conv2D(64, (3, 3), activation=activation_fn, padding='same',␣
↪activity_regularizer=l1(regularization))(input_img)
    x = MaxPooling2D((2, 2), padding='same')(x)
    x = Dropout(0.1)(x)
    x = Conv2D(128, (3, 3), activation=activation_fn, padding='same',␣
↪activity_regularizer=l1(regularization))(x)
    x = MaxPooling2D((2, 2), padding='same')(x)
    x = Dropout(0.1)(x)
    encoded = Conv2D(encoding_dim, (3, 3), activation=activation_fn,␣
↪padding='same', activity_regularizer=l1(regularization))(x)

    # Decoder
    x = Conv2D(encoding_dim, (3, 3), activation=activation_fn,␣
↪padding='same')(encoded)
    x = UpSampling2D((2, 2))(x)
    x = Conv2D(128, (3, 3), activation=activation_fn, padding='same')(x)
    x = UpSampling2D((2, 2))(x)
    x = Conv2D(64, (3, 3), activation=activation_fn, padding='same')(x)
    decoded = Conv2D(3, (3, 3), activation='sigmoid', padding='same')(x)

    autoencoder = Model(input_img, decoded)
    autoencoder.compile(optimizer=Adam(learning_rate=1e-4),␣
↪loss='mean_absolute_error')

    # Entrenar el Autoencoder Convolucional Sparsity
    print(f"Entrenando Sparse AE con Dim: {encoding_dim}, Regularization:␣
↪{regularization}")
    history = autoencoder.fit(
        X_train, X_train,  # Usar entrada limpia y salida limpia
        epochs=70,
        batch_size=16,
        shuffle=True,
        validation_data=(X_val, X_val),  # Usar datos de validación limpios
        callbacks=[early_stopping],
        verbose=1
    )
    sparse_histories.append((config, history))
    sparse_models.append(autoencoder)  # Guardar el modelo específico para cada␣
↪configuración
```

```
Entrenando Sparse AE con Dim: 32, Regularization: 1e-05
```

```
Epoch 1/70
10/10 [==============================] - 2s 48ms/step - loss: 1.0503 - val_loss:
0.9383
Epoch 2/70
10/10 [==============================] - 0s 31ms/step - loss: 0.8976 - val_loss:
0.8157
Epoch 3/70
10/10 [==============================] - 0s 30ms/step - loss: 0.7882 - val_loss:
0.7238
Epoch 4/70
10/10 [==============================] - 0s 30ms/step - loss: 0.7051 - val_loss:
0.6533
Epoch 5/70
10/10 [==============================] - 0s 32ms/step - loss: 0.6377 - val_loss:
0.5975
Epoch 6/70
10/10 [==============================] - 0s 32ms/step - loss: 0.5773 - val_loss:
0.5503
Epoch 7/70
10/10 [==============================] - 0s 31ms/step - loss: 0.5137 - val_loss:
0.5108
Epoch 8/70
10/10 [==============================] - 0s 32ms/step - loss: 0.4385 - val_loss:
0.4868
Epoch 9/70
10/10 [==============================] - 0s 32ms/step - loss: 0.3672 - val_loss:
0.4678
Epoch 10/70
10/10 [==============================] - 0s 32ms/step - loss: 0.3119 - val_loss:
0.4009
Epoch 11/70
10/10 [==============================] - 0s 32ms/step - loss: 0.2686 - val_loss:
0.3484
Epoch 12/70
10/10 [==============================] - 0s 32ms/step - loss: 0.2348 - val_loss:
0.3112
Epoch 13/70
10/10 [==============================] - 0s 32ms/step - loss: 0.2072 - val_loss:
0.2824
Epoch 14/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1835 - val_loss:
0.2483
Epoch 15/70
10/10 [==============================] - 0s 32ms/step - loss: 0.1618 - val_loss:
0.2210
Epoch 16/70
10/10 [==============================] - 0s 32ms/step - loss: 0.1426 - val_loss:
0.1909
```

```
Epoch 17/70
10/10 [==============================] - 0s 32ms/step - loss: 0.1264 - val_loss:
0.1725
Epoch 18/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1131 - val_loss:
0.1537
Epoch 19/70
10/10 [==============================] - 0s 32ms/step - loss: 0.1025 - val_loss:
0.1380
Epoch 20/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0944 - val_loss:
0.1284
Epoch 21/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0881 - val_loss:
0.1162
Epoch 22/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0828 - val_loss:
0.1092
Epoch 23/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0785 - val_loss:
0.1016
Epoch 24/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0751 - val_loss:
0.0933
Epoch 25/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0723 - val_loss:
0.0886
Epoch 26/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0699 - val_loss:
0.0856
Epoch 27/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0679 - val_loss:
0.0835
Epoch 28/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0664 - val_loss:
0.0811
Epoch 29/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0649 - val_loss:
0.0797
Epoch 30/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0635 - val_loss:
0.0791
Epoch 31/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0625 - val_loss:
0.0774
Epoch 32/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0614 - val_loss:
0.0743
```

```
Epoch 33/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0605 - val_loss:
0.0774
Epoch 34/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0598 - val_loss:
0.0735
Epoch 35/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0588 - val_loss:
0.0714
Epoch 36/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0583 - val_loss:
0.0752
Epoch 37/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0576 - val_loss:
0.0702
Epoch 38/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0571 - val_loss:
0.0734
Epoch 39/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0567 - val_loss:
0.0679
Epoch 40/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0561 - val_loss:
0.0670
Epoch 41/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0557 - val_loss:
0.0706
Epoch 42/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0554 - val_loss:
0.0709
Epoch 43/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0552 - val_loss:
0.0712
Epoch 44/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0549 - val_loss:
0.0693
Epoch 45/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0543 - val_loss:
0.0679
Epoch 46/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0537 - val_loss:
0.0680
Epoch 47/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0534 - val_loss:
0.0650
Epoch 48/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0531 - val_loss:
0.0651
```

```
Epoch 49/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0528 - val_loss:
0.0650
Epoch 50/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0526 - val_loss:
0.0649
Epoch 51/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0523 - val_loss:
0.0651
Epoch 52/70
10/10 [==============================] - 0s 33ms/step - loss: 0.0521 - val_loss:
0.0627
Epoch 53/70
10/10 [==============================] - 0s 28ms/step - loss: 0.0517 - val_loss:
0.0639
Epoch 54/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0514 - val_loss:
0.0638
Epoch 55/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0512 - val_loss:
0.0631
Epoch 56/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0511 - val_loss:
0.0632
Epoch 57/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0511 - val_loss:
0.0648
Epoch 58/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0507 - val_loss:
0.0638
Epoch 59/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0504 - val_loss:
0.0612
Epoch 60/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0503 - val_loss:
0.0587
Epoch 61/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0502 - val_loss:
0.0585
Epoch 62/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0500 - val_loss:
0.0582
Epoch 63/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0497 - val_loss:
0.0598
Epoch 64/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0492 - val_loss:
0.0608
```

```
Epoch 65/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0491 - val_loss:
0.0613
Epoch 66/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0490 - val_loss:
0.0630
Epoch 67/70
10/10 [==============================] - 0s 32ms/step - loss: 0.0489 - val_loss:
0.0636
Epoch 68/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0488 - val_loss:
0.0643
Epoch 69/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0487 - val_loss:
0.0637
Epoch 70/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0486 - val_loss:
0.0593
Entrenando Sparse AE con Dim: 64, Regularization: 0.0001
Epoch 1/70
10/10 [==============================] - 3s 49ms/step - loss: 6.2966 - val_loss:
5.3866
Epoch 2/70
10/10 [==============================] - 0s 31ms/step - loss: 5.0646 - val_loss:
4.4415
Epoch 3/70
10/10 [==============================] - 0s 31ms/step - loss: 4.2538 - val_loss:
3.8149
Epoch 4/70
10/10 [==============================] - 0s 31ms/step - loss: 3.6892 - val_loss:
3.3513
Epoch 5/70
10/10 [==============================] - 0s 31ms/step - loss: 3.2504 - val_loss:
2.9739
Epoch 6/70
10/10 [==============================] - 0s 31ms/step - loss: 2.8882 - val_loss:
2.6508
Epoch 7/70
10/10 [==============================] - 0s 31ms/step - loss: 2.5722 - val_loss:
2.3669
Epoch 8/70
10/10 [==============================] - 0s 31ms/step - loss: 2.2946 - val_loss:
2.1324
Epoch 9/70
10/10 [==============================] - 0s 31ms/step - loss: 2.0446 - val_loss:
1.9419
Epoch 10/70
10/10 [==============================] - 0s 31ms/step - loss: 1.8181 - val_loss:
```

```
1.7831
Epoch 11/70
10/10 [==============================] - 0s 31ms/step - loss: 1.6267 - val_loss:
1.6370
Epoch 12/70
10/10 [==============================] - 0s 31ms/step - loss: 1.4712 - val_loss:
1.5239
Epoch 13/70
10/10 [==============================] - 0s 31ms/step - loss: 1.3361 - val_loss:
1.4053
Epoch 14/70
10/10 [==============================] - 0s 31ms/step - loss: 1.2129 - val_loss:
1.2845
Epoch 15/70
10/10 [==============================] - 0s 31ms/step - loss: 1.0940 - val_loss:
1.1510
Epoch 16/70
10/10 [==============================] - 0s 31ms/step - loss: 0.9786 - val_loss:
1.0069
Epoch 17/70
10/10 [==============================] - 0s 32ms/step - loss: 0.8656 - val_loss:
0.8900
Epoch 18/70
10/10 [==============================] - 0s 31ms/step - loss: 0.7555 - val_loss:
0.7550
Epoch 19/70
10/10 [==============================] - 0s 32ms/step - loss: 0.6478 - val_loss:
0.6399
Epoch 20/70
10/10 [==============================] - 0s 32ms/step - loss: 0.5438 - val_loss:
0.5188
Epoch 21/70
10/10 [==============================] - 0s 32ms/step - loss: 0.4494 - val_loss:
0.4164
Epoch 22/70
10/10 [==============================] - 0s 32ms/step - loss: 0.3653 - val_loss:
0.3289
Epoch 23/70
10/10 [==============================] - 0s 32ms/step - loss: 0.2937 - val_loss:
0.2569
Epoch 24/70
10/10 [==============================] - 0s 32ms/step - loss: 0.2370 - val_loss:
0.2034
Epoch 25/70
10/10 [==============================] - 0s 33ms/step - loss: 0.1982 - val_loss:
0.1765
Epoch 26/70
10/10 [==============================] - 0s 33ms/step - loss: 0.1775 - val_loss:
```

```
0.1688
Epoch 27/70
10/10 [==============================] - 0s 32ms/step - loss: 0.1674 - val_loss:
0.1561
Epoch 28/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1606 - val_loss:
0.1548
Epoch 29/70
10/10 [==============================] - 0s 33ms/step - loss: 0.1558 - val_loss:
0.1471
Epoch 30/70
10/10 [==============================] - 0s 33ms/step - loss: 0.1513 - val_loss:
0.1449
Epoch 31/70
10/10 [==============================] - 0s 32ms/step - loss: 0.1477 - val_loss:
0.1434
Epoch 32/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1448 - val_loss:
0.1407
Epoch 33/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1425 - val_loss:
0.1380
Epoch 34/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1406 - val_loss:
0.1357
Epoch 35/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1373 - val_loss:
0.1321
Epoch 36/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1355 - val_loss:
0.1302
Epoch 37/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1336 - val_loss:
0.1292
Epoch 38/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1319 - val_loss:
0.1281
Epoch 39/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1293 - val_loss:
0.1255
Epoch 40/70
10/10 [==============================] - 0s 33ms/step - loss: 0.1277 - val_loss:
0.1241
Epoch 41/70
10/10 [==============================] - 0s 32ms/step - loss: 0.1259 - val_loss:
0.1228
Epoch 42/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1247 - val_loss:
```

```
0.1216
Epoch 43/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1238 - val_loss:
0.1203
Epoch 44/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1224 - val_loss:
0.1211
Epoch 45/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1220 - val_loss:
0.1183
Epoch 46/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1189 - val_loss:
0.1157
Epoch 47/70
10/10 [==============================] - 0s 32ms/step - loss: 0.1169 - val_loss:
0.1145
Epoch 48/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1159 - val_loss:
0.1140
Epoch 49/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1147 - val_loss:
0.1123
Epoch 50/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1137 - val_loss:
0.1124
Epoch 51/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1129 - val_loss:
0.1119
Epoch 52/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1127 - val_loss:
0.1099
Epoch 53/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1112 - val_loss:
0.1093
Epoch 54/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1098 - val_loss:
0.1071
Epoch 55/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1079 - val_loss:
0.1054
Epoch 56/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1069 - val_loss:
0.1047
Epoch 57/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1063 - val_loss:
0.1031
Epoch 58/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1052 - val_loss:
```

```
0.1033
Epoch 59/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1046 - val_loss:
0.1029
Epoch 60/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1038 - val_loss:
0.1009
Epoch 61/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1027 - val_loss:
0.1000
Epoch 62/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1020 - val_loss:
0.1030
Epoch 63/70
10/10 [==============================] - 0s 30ms/step - loss: 0.1033 - val_loss:
0.1012
Epoch 64/70
10/10 [==============================] - 0s 31ms/step - loss: 0.1012 - val_loss:
0.0982
Epoch 65/70
10/10 [==============================] - 0s 30ms/step - loss: 0.1007 - val_loss:
0.1002
Epoch 66/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0994 - val_loss:
0.0964
Epoch 67/70
10/10 [==============================] - 0s 30ms/step - loss: 0.0984 - val_loss:
0.0971
Epoch 68/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0979 - val_loss:
0.0960
Epoch 69/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0968 - val_loss:
0.0953
Epoch 70/70
10/10 [==============================] - 0s 31ms/step - loss: 0.0977 - val_loss:
0.0968
Entrenando Sparse AE con Dim: 128, Regularization: 0.001
Epoch 1/70
10/10 [==============================] - 2s 49ms/step - loss: 70.4664 -
val_loss: 60.2412
Epoch 2/70
10/10 [==============================] - 0s 31ms/step - loss: 56.5650 -
val_loss: 50.0025
Epoch 3/70
10/10 [==============================] - 0s 31ms/step - loss: 47.6837 -
val_loss: 43.0622
Epoch 4/70
```

```
10/10 [==============================] - 0s 31ms/step - loss: 41.5122 -
val_loss: 37.9892
Epoch 5/70
10/10 [==============================] - 0s 31ms/step - loss: 36.8284 -
val_loss: 33.9760
Epoch 6/70
10/10 [==============================] - 0s 31ms/step - loss: 32.9795 -
val_loss: 30.5501
Epoch 7/70
10/10 [==============================] - 0s 31ms/step - loss: 29.6930 -
val_loss: 27.6380
Epoch 8/70
10/10 [==============================] - 0s 31ms/step - loss: 26.8763 -
val_loss: 25.1162
Epoch 9/70
10/10 [==============================] - 0s 33ms/step - loss: 24.3509 -
val_loss: 22.7619
Epoch 10/70
10/10 [==============================] - 0s 33ms/step - loss: 21.9667 -
val_loss: 20.4599
Epoch 11/70
10/10 [==============================] - 0s 33ms/step - loss: 19.6425 -
val_loss: 18.1919
Epoch 12/70
10/10 [==============================] - 0s 34ms/step - loss: 17.3654 -
val_loss: 15.9478
Epoch 13/70
10/10 [==============================] - 0s 32ms/step - loss: 15.1280 -
val_loss: 13.7663
Epoch 14/70
10/10 [==============================] - 0s 34ms/step - loss: 12.9796 -
val_loss: 11.7013
Epoch 15/70
10/10 [==============================] - 0s 33ms/step - loss: 11.0086 -
val_loss: 9.9065
Epoch 16/70
10/10 [==============================] - 0s 33ms/step - loss: 9.3233 - val_loss:
8.4246
Epoch 17/70
10/10 [==============================] - 0s 33ms/step - loss: 7.9440 - val_loss:
7.1888
Epoch 18/70
10/10 [==============================] - 0s 32ms/step - loss: 6.7733 - val_loss:
6.1617
Epoch 19/70
10/10 [==============================] - 0s 31ms/step - loss: 5.8367 - val_loss:
5.3298
Epoch 20/70
```

```
10/10 [==============================] - 0s 31ms/step - loss: 5.0241 - val_loss:
4.5697
Epoch 21/70
10/10 [==============================] - 0s 31ms/step - loss: 4.2770 - val_loss:
3.8605
Epoch 22/70
10/10 [==============================] - 0s 31ms/step - loss: 3.6028 - val_loss:
3.2370
Epoch 23/70
10/10 [==============================] - 0s 31ms/step - loss: 3.0056 - val_loss:
2.6624
Epoch 24/70
10/10 [==============================] - 0s 32ms/step - loss: 2.4697 - val_loss:
2.1588
Epoch 25/70
10/10 [==============================] - 0s 31ms/step - loss: 1.9985 - val_loss:
1.7411
Epoch 26/70
10/10 [==============================] - 0s 31ms/step - loss: 1.6378 - val_loss:
1.4493
Epoch 27/70
10/10 [==============================] - 0s 32ms/step - loss: 1.3925 - val_loss:
1.2422
Epoch 28/70
10/10 [==============================] - 0s 33ms/step - loss: 1.1986 - val_loss:
1.0627
Epoch 29/70
10/10 [==============================] - 0s 32ms/step - loss: 1.0281 - val_loss:
0.8990
Epoch 30/70
10/10 [==============================] - 0s 32ms/step - loss: 0.8783 - val_loss:
0.7723
Epoch 31/70
10/10 [==============================] - 0s 31ms/step - loss: 0.7895 - val_loss:
0.7256
Epoch 32/70
10/10 [==============================] - 0s 31ms/step - loss: 0.7502 - val_loss:
0.6985
Epoch 33/70
10/10 [==============================] - 0s 31ms/step - loss: 0.7206 - val_loss:
0.6701
Epoch 34/70
10/10 [==============================] - 0s 31ms/step - loss: 0.6946 - val_loss:
0.6553
Epoch 35/70
10/10 [==============================] - 0s 31ms/step - loss: 0.6720 - val_loss:
0.6345
Epoch 36/70
```

```
10/10 [==============================] - 0s 31ms/step - loss: 0.6512 - val_loss:
0.6179
Epoch 37/70
10/10 [==============================] - 0s 31ms/step - loss: 0.6327 - val_loss:
0.5949
Epoch 38/70
10/10 [==============================] - 0s 31ms/step - loss: 0.6155 - val_loss:
0.5795
Epoch 39/70
10/10 [==============================] - 0s 31ms/step - loss: 0.5996 - val_loss:
0.5650
Epoch 40/70
10/10 [==============================] - 0s 32ms/step - loss: 0.5846 - val_loss:
0.5532
Epoch 41/70
10/10 [==============================] - 0s 31ms/step - loss: 0.5729 - val_loss:
0.5386
Epoch 42/70
10/10 [==============================] - 0s 32ms/step - loss: 0.5586 - val_loss:
0.5262
Epoch 43/70
10/10 [==============================] - 0s 31ms/step - loss: 0.5431 - val_loss:
0.5125
Epoch 44/70
10/10 [==============================] - 0s 30ms/step - loss: 0.5289 - val_loss:
0.5017
Epoch 45/70
10/10 [==============================] - 0s 31ms/step - loss: 0.5170 - val_loss:
0.4896
Epoch 46/70
10/10 [==============================] - 0s 32ms/step - loss: 0.5055 - val_loss:
0.4842
Epoch 47/70
10/10 [==============================] - 0s 31ms/step - loss: 0.4945 - val_loss:
0.4721
Epoch 48/70
10/10 [==============================] - 0s 31ms/step - loss: 0.4829 - val_loss:
0.4584
Epoch 49/70
10/10 [==============================] - 0s 31ms/step - loss: 0.4728 - val_loss:
0.4483
Epoch 50/70
10/10 [==============================] - 0s 32ms/step - loss: 0.4638 - val_loss:
0.4397
Epoch 51/70
10/10 [==============================] - 0s 31ms/step - loss: 0.4559 - val_loss:
0.4295
Epoch 52/70
```

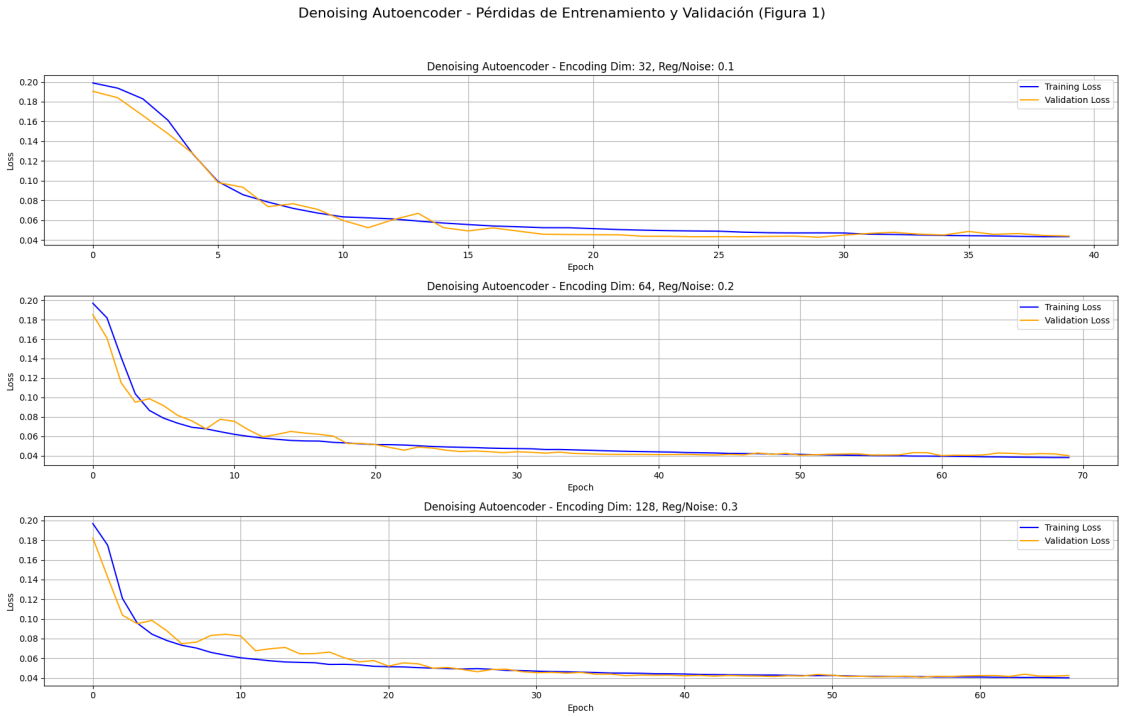
```
10/10 [==============================] - 0s 31ms/step - loss: 0.4437 - val_loss:
0.4220
Epoch 53/70
10/10 [==============================] - 0s 31ms/step - loss: 0.4339 - val_loss:
0.4144
Epoch 54/70
10/10 [==============================] - 0s 31ms/step - loss: 0.4254 - val_loss:
0.4063
Epoch 55/70
10/10 [==============================] - 0s 31ms/step - loss: 0.4176 - val_loss:
0.3969
Epoch 56/70
10/10 [==============================] - 0s 31ms/step - loss: 0.4101 - val_loss:
0.3895
Epoch 57/70
10/10 [==============================] - 0s 31ms/step - loss: 0.4021 - val_loss:
0.3846
Epoch 58/70
10/10 [==============================] - 0s 30ms/step - loss: 0.3947 - val_loss:
0.3766
Epoch 59/70
10/10 [==============================] - 0s 31ms/step - loss: 0.3873 - val_loss:
0.3789
Epoch 60/70
10/10 [==============================] - 0s 31ms/step - loss: 0.3828 - val_loss:
0.3675
Epoch 61/70
10/10 [==============================] - 0s 31ms/step - loss: 0.3742 - val_loss:
0.3553
Epoch 62/70
10/10 [==============================] - 0s 31ms/step - loss: 0.3675 - val_loss:
0.3513
Epoch 63/70
10/10 [==============================] - 0s 31ms/step - loss: 0.3611 - val_loss:
0.3439
Epoch 64/70
10/10 [==============================] - 0s 31ms/step - loss: 0.3558 - val_loss:
0.3428
Epoch 65/70
10/10 [==============================] - 0s 31ms/step - loss: 0.3496 - val_loss:
0.3326
Epoch 66/70
10/10 [==============================] - 0s 31ms/step - loss: 0.3476 - val_loss:
0.3284
Epoch 67/70
10/10 [==============================] - 0s 30ms/step - loss: 0.3420 - val_loss:
0.3374
Epoch 68/70
```

```
10/10 [==============================] - 0s 31ms/step - loss: 0.3364 - val_loss:
0.3261
Epoch 69/70
10/10 [==============================] - 0s 31ms/step - loss: 0.3319 - val_loss:
0.3139
Epoch 70/70
10/10 [==============================] - 0s 32ms/step - loss: 0.3257 - val_loss:
0.3083
```

[53]:
```python
import matplotlib.pyplot as plt
import math

# Parámetros de configuración para los gráficos
plots_per_figure = 3  # Número de gráficos por figura para una mejor␣
 ↪visualización

# Función para graficar pérdidas
def graficar_perdidas(histories, titulo):
    num_plots = len(histories)
    num_figures = math.ceil(num_plots / plots_per_figure)

    for fig in range(num_figures):
        plt.figure(figsize=(18, 12))
        for i in range(plots_per_figure):
            plot_idx = fig * plots_per_figure + i
            if plot_idx >= num_plots:
                break

            config, history = histories[plot_idx]
            encoding_dim = config.get('encoding_dim', 'N/A')
            reg_or_noise = config.get('regularization', config.
 ↪get('noise_factor', 'N/A'))

            # Graficar las pérdidas de entrenamiento y validación
            plt.subplot(plots_per_figure, 1, i + 1)
            plt.plot(history.history['loss'], label='Training Loss',␣
 ↪color='blue', linestyle='-')
            plt.plot(history.history['val_loss'], label='Validation Loss',␣
 ↪color='orange', linestyle='-')
            plt.title(f"{titulo} - Encoding Dim: {encoding_dim}, Reg/Noise:␣
 ↪{reg_or_noise}")
            plt.xlabel('Epoch')
            plt.ylabel('Loss')
            plt.legend()
            plt.grid(True)
```

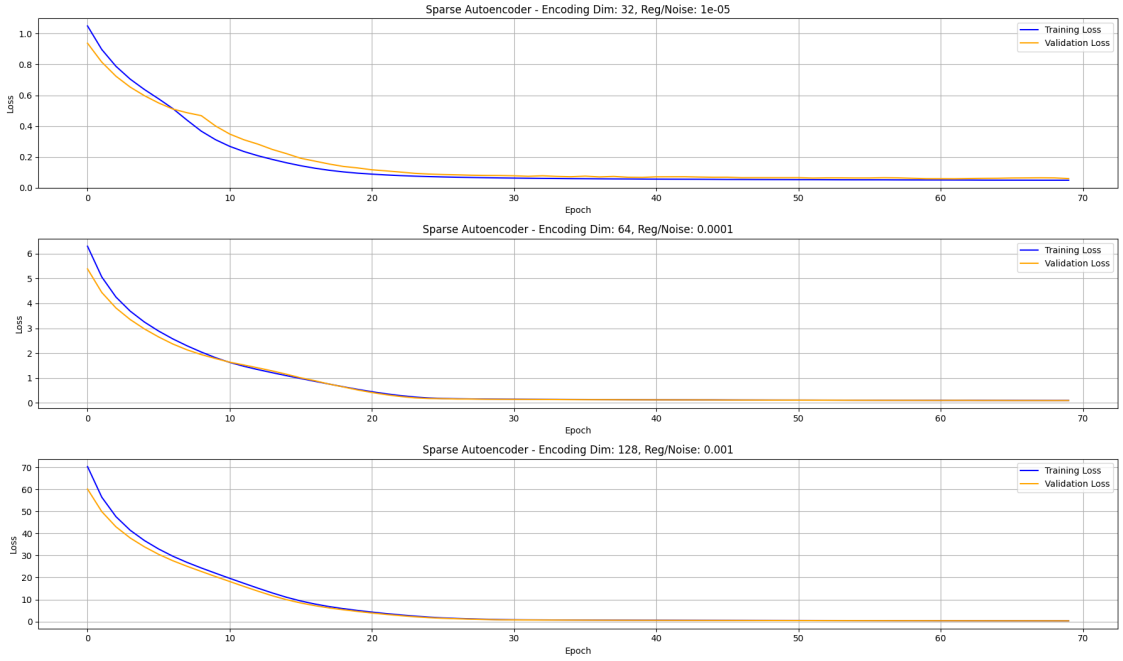
```
        plt.suptitle(f"{titulo} - Pérdidas de Entrenamiento y Validación␣
 ↪(Figura {fig + 1})", fontsize=16)
        plt.tight_layout(rect=[0, 0.03, 1, 0.95])
        plt.show()

# Visualizar pérdidas para el Autoencoder Convolucional Denoising
print("\nVisualización de Pérdidas - Autoencoder Convolucional Denoising")
graficar_perdidas(denoising_histories, "Denoising Autoencoder")

# Visualizar pérdidas para el Autoencoder Convolucional Sparsity
print("\nVisualización de Pérdidas - Autoencoder Convolucional Sparsity")
graficar_perdidas(sparse_histories, "Sparse Autoencoder")
```

```
Visualización de Pérdidas - Autoencoder Convolucional Denoising
```




```
Visualización de Pérdidas - Autoencoder Convolucional Sparsity
```

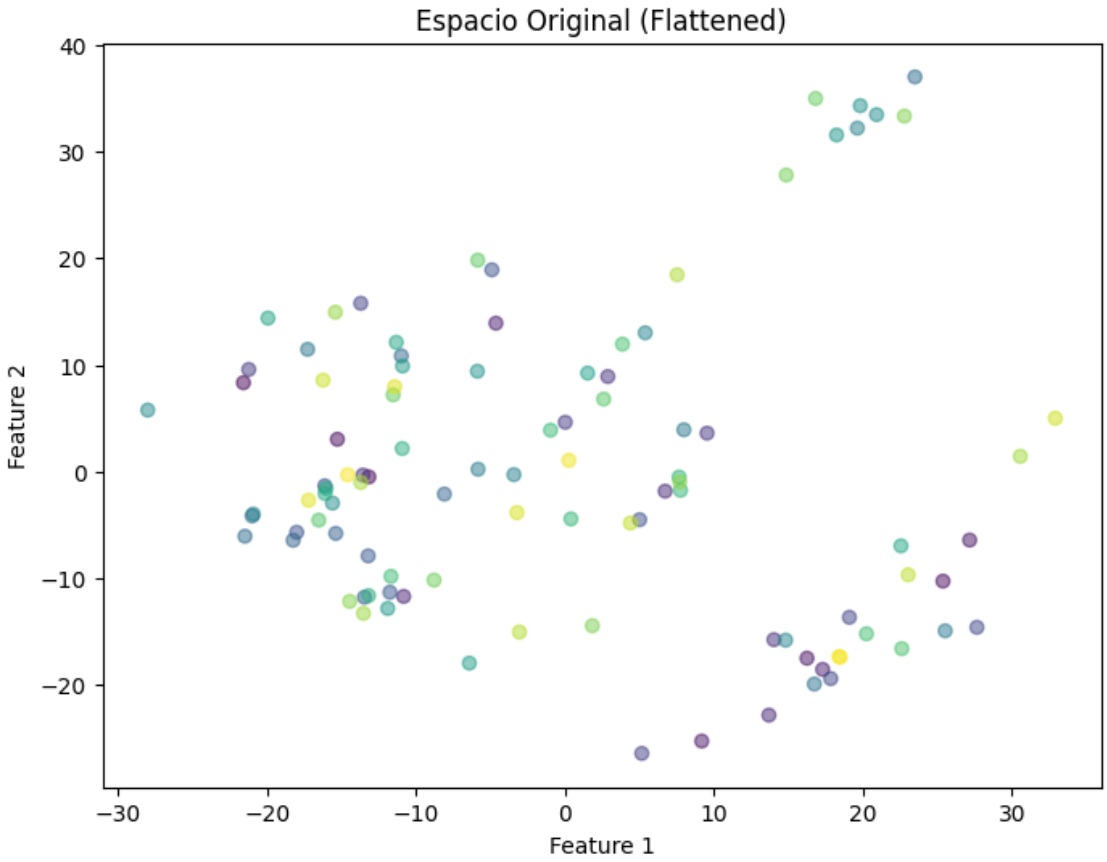
Sparse Autoencoder - Pérdidas de Entrenamiento y Validación (Figura 1)



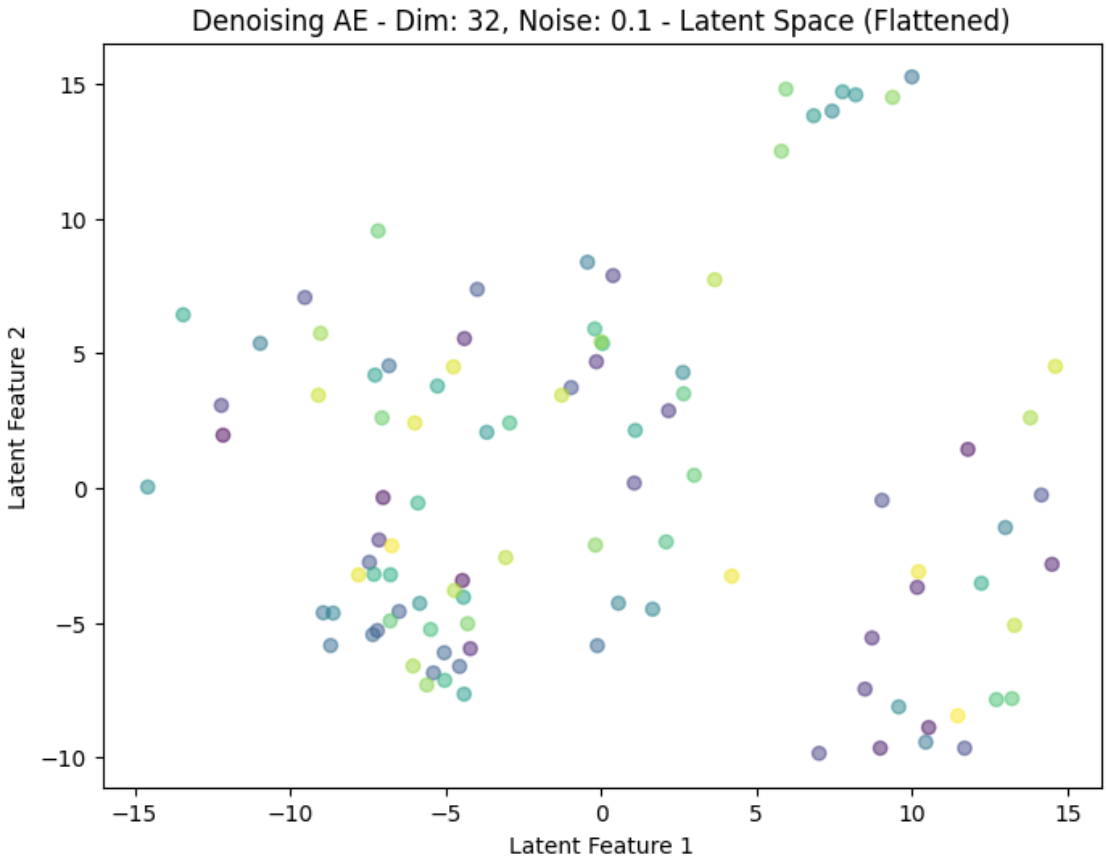
[54]:
```
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Función para graficar el espacio original (aplanado y reducido con PCA)
def graficar_espacio_original(X, title="Espacio Original", num_samples=100):
    num_samples = min(len(X), num_samples)
    X_sample = X[:num_samples].reshape(num_samples, -1)
    pca = PCA(n_components=2)
    original_pca = pca.fit_transform(X_sample)

    plt.figure(figsize=(8, 6))
    plt.scatter(original_pca[:, 0], original_pca[:, 1], alpha=0.5, c=np.
 ↪arange(len(original_pca)), cmap='viridis')
    plt.title(f"{title} (Flattened)")
    plt.xlabel("Feature 1")
    plt.ylabel("Feature 2")
    plt.show()

# Función para graficar el espacio latente del autoencoder
def graficar_espacio_latente(autoencoder, X, title="Espacio Latente",␣
 ↪num_samples=100):
    num_samples = min(len(X), num_samples)
    encoder = Model(inputs=autoencoder.input, outputs=autoencoder.layers[-7].
 ↪output)
```

```
    X_sample = X[:num_samples]
    latent_space = encoder.predict(X_sample)
    latent_flattened = latent_space.reshape(len(latent_space), -1)

    pca = PCA(n_components=2)
    latent_pca = pca.fit_transform(latent_flattened)

    plt.figure(figsize=(8, 6))
    plt.scatter(latent_pca[:, 0], latent_pca[:, 1], alpha=0.5, c=np.
 ↪arange(len(latent_pca)), cmap='viridis')
    plt.title(f"{title} - Latent Space (Flattened)")
    plt.xlabel("Latent Feature 1")
    plt.ylabel("Latent Feature 2")
    plt.show()

# Visualizar el espacio original
print("Espacio Original")
graficar_espacio_original(X_train)

# Visualizar el espacio latente para cada configuración de Denoising Autoencoder
print("\nEspacio Latente - Denoising Autoencoder")
for i, (config, history) in enumerate(denoising_histories):
    encoding_dim = config['encoding_dim']
    noise_factor = config['noise_factor']
    autoencoder = denoising_models[i]
    graficar_espacio_latente(autoencoder, X_train, title=f"Denoising AE - Dim:␣
 ↪{encoding_dim}, Noise: {noise_factor}")

# Visualizar el espacio latente para cada configuración de Sparse Autoencoder
print("\nEspacio Latente - Sparse Autoencoder")
for i, (config, history) in enumerate(sparse_histories):
    encoding_dim = config['encoding_dim']
    regularization = config['regularization']
    autoencoder = sparse_models[i]
    graficar_espacio_latente(autoencoder, X_train, title=f"Sparse AE - Dim:␣
 ↪{encoding_dim}, Reg: {regularization}")
```

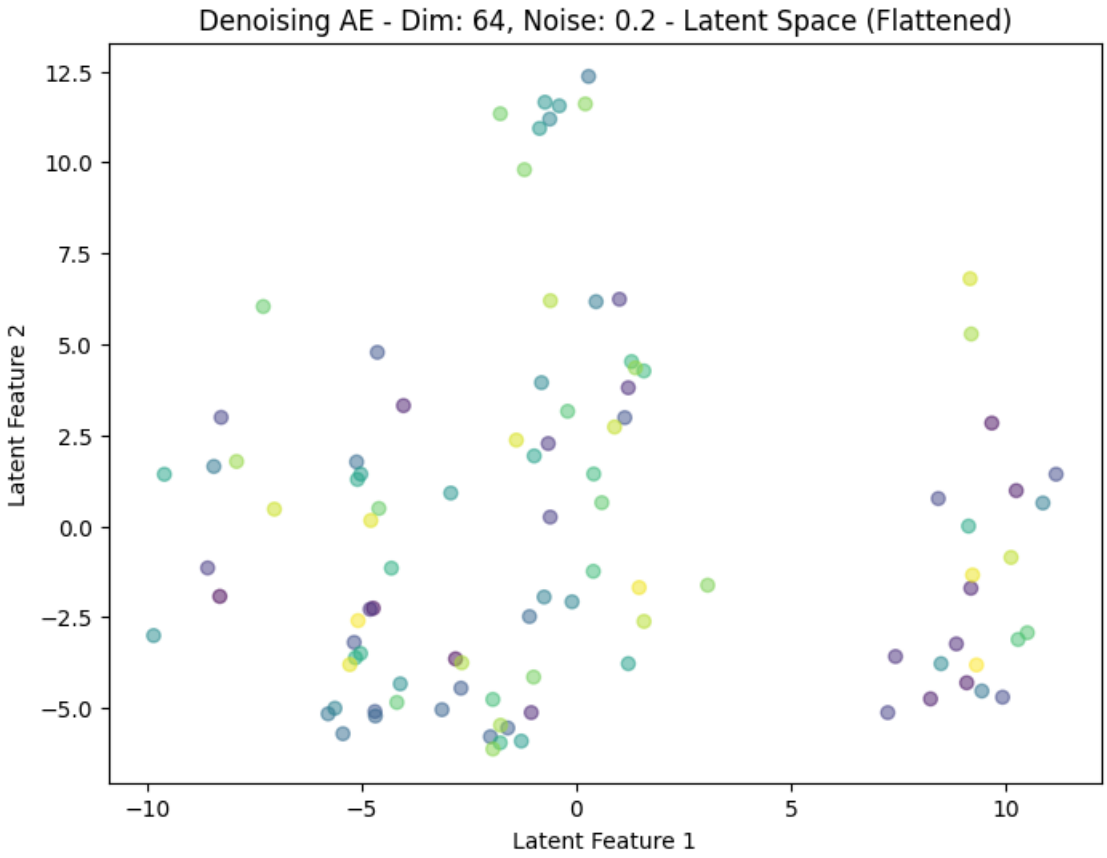
```
Espacio Original
```



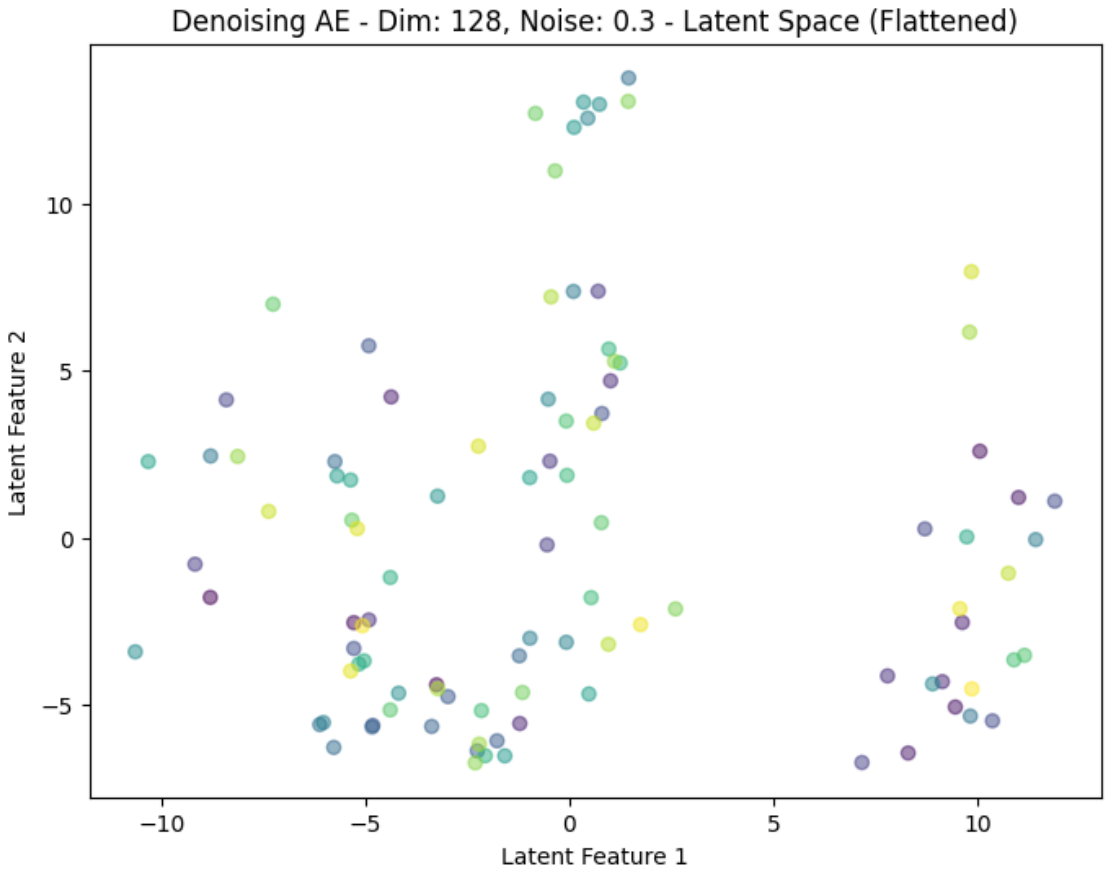

```
Espacio Latente - Denoising Autoencoder
4/4 [==============================] - 3s 429ms/step
```



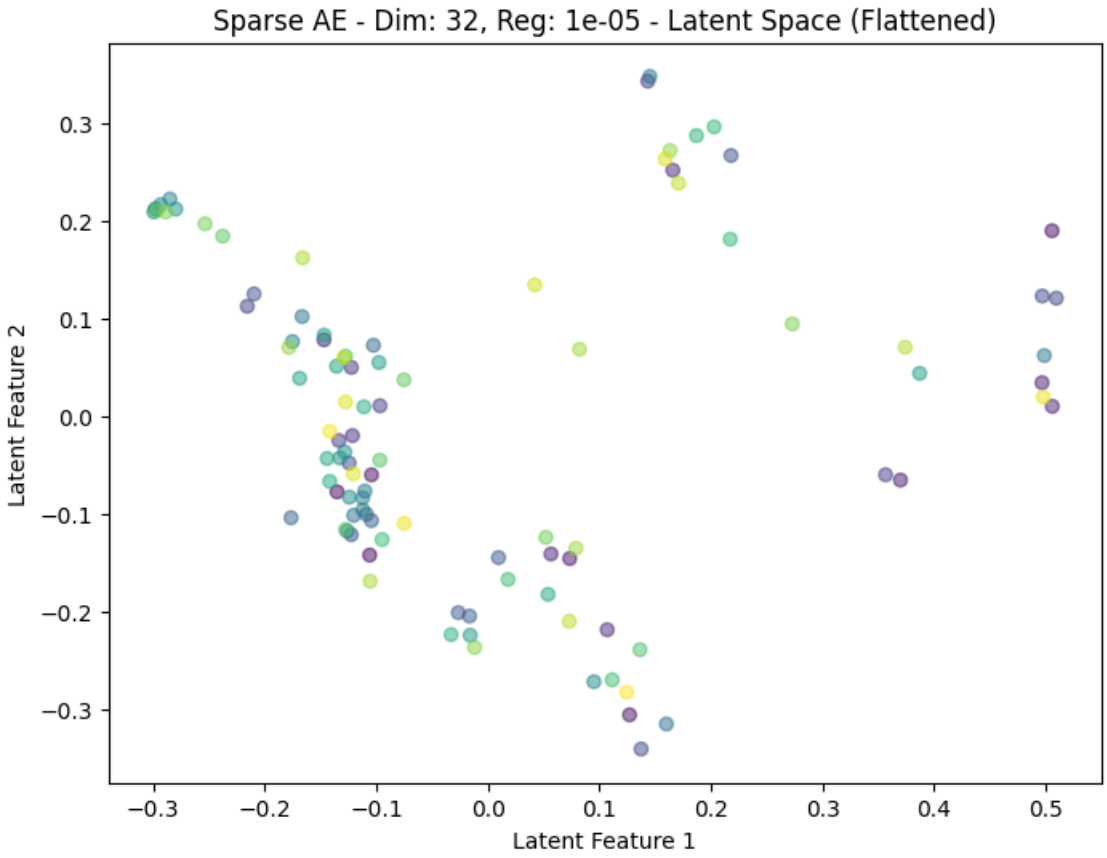

```
4/4 [==============================] - 1s 141ms/step
```



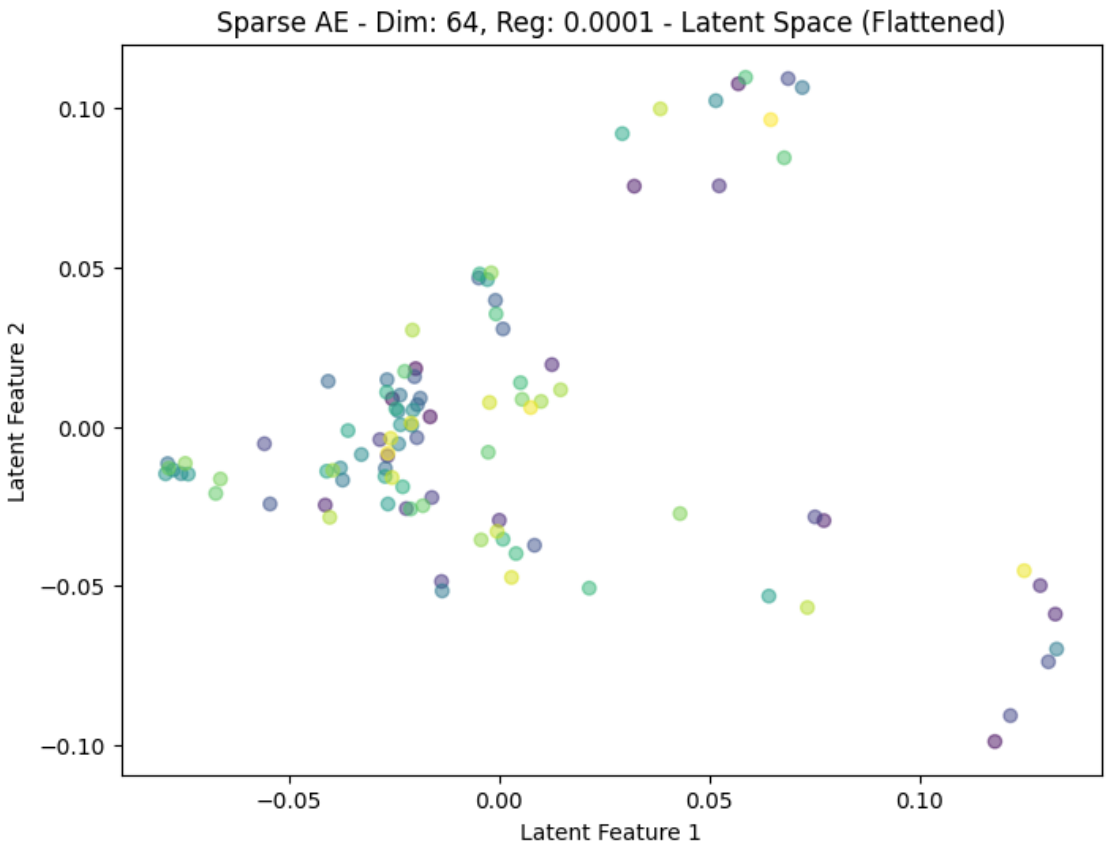

```
4/4 [==============================] - 1s 184ms/step
```



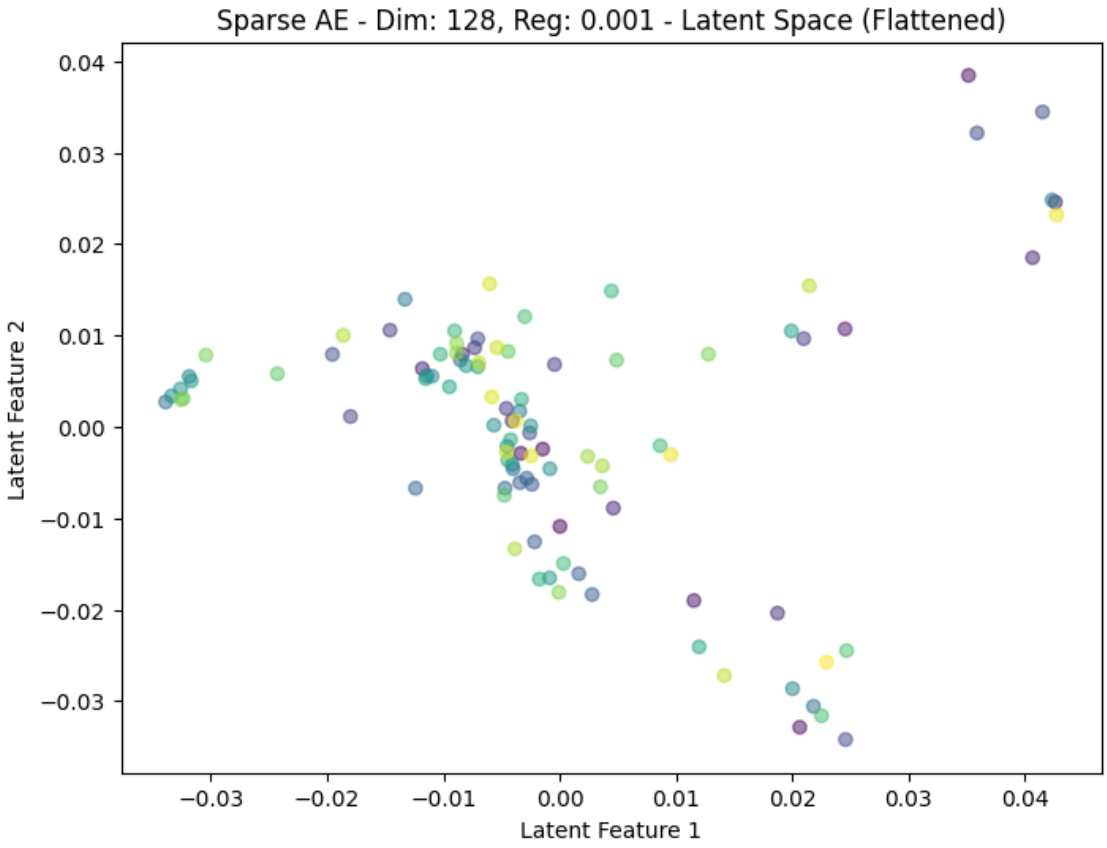

```
Espacio Latente - Sparse Autoencoder
4/4 [==============================] - 0s 3ms/step
```




```
4/4 [==============================] - 0s 3ms/step
```




```
4/4 [==============================] - 0s 3ms/step
```



Funcion para calcular metricas de reconstruccion

[55]:
```
from skimage.metrics import peak_signal_noise_ratio as psnr
from skimage.metrics import structural_similarity as ssim
from sklearn.metrics import mean_squared_error

def calcular_metricas_reconstruccion(original, reconstruido):
    # Calcular MSE
    mse_val = mean_squared_error(original.flatten(), reconstruido.flatten())

    # Calcular PSNR
    psnr_val = psnr(original, reconstruido, data_range=original.max() -␣
 ↪original.min())

    # Calcular SSIM
    ssim_val = ssim(original, reconstruido, multichannel=True,␣
 ↪data_range=original.max() - original.min())

    return mse_val, psnr_val, ssim_val
```

Funcion para mostrar reconstruccion visual

[56]:
```
import matplotlib.pyplot as plt

def mostrar_reconstruccion_visual(autoencoder, X_original, num_samples=5,␣
 ↪title="Reconstrucción"):
    # Seleccionar muestras aleatorias
    indices = np.random.choice(len(X_original), num_samples, replace=False)
    X_sample = X_original[indices]

    # Obtener las imágenes reconstruidas
    X_reconstruido = autoencoder.predict(X_sample)

    # Mostrar imágenes originales, reconstruidas y métricas
    plt.figure(figsize=(12, 4 * num_samples))

    for i in range(num_samples):
        original = X_sample[i]
        reconstruido = X_reconstruido[i]

        # Calcular métricas
        mse_val, psnr_val, ssim_val =␣
 ↪calcular_metricas_reconstruccion(original, reconstruido)

        # Mostrar imagen original
        plt.subplot(num_samples, 2, 2 * i + 1)
        plt.imshow(original)
        plt.title(f"Original")
        plt.axis("off")

        # Mostrar imagen reconstruida con métricas
        plt.subplot(num_samples, 2, 2 * i + 2)
        plt.imshow(reconstruido)
        plt.title(f"Reconstrucción\nMSE: {mse_val:.4f}, PSNR: {psnr_val:.2f},␣
 ↪SSIM: {ssim_val:.4f}")
        plt.axis("off")

    plt.suptitle(title, fontsize=16)
    plt.tight_layout()
    plt.show()
```

Aplicacion a cada escenario de AE

[57]:
```
# Mostrar la reconstrucción visual y las métricas para cada configuración de␣
 ↪Denoising Autoencoder
print("Reconstrucción Visual y Métricas - Denoising Autoencoder")
for i, (config, _) in enumerate(denoising_histories):
    encoding_dim = config['encoding_dim']
    noise_factor = config['noise_factor']
```

```
    autoencoder = denoising_models[i]
    mostrar_reconstruccion_visual(autoencoder, X_train, num_samples=5,
                                  title=f"Denoising AE - Dim: {encoding_dim},␣
 ↪Noise: {noise_factor}")

# Mostrar la reconstrucción visual y las métricas para cada configuración de␣
 ↪Sparse Autoencoder
print("\nReconstrucción Visual y Métricas - Sparse Autoencoder")
for i, (config, _) in enumerate(sparse_histories):
    encoding_dim = config['encoding_dim']
    regularization = config['regularization']
    autoencoder = sparse_models[i]
    mostrar_reconstruccion_visual(autoencoder, X_train, num_samples=5,
                                  title=f"Sparse AE - Dim: {encoding_dim}, Reg:␣
 ↪{regularization}")
```

```
Reconstrucción Visual y Métricas - Denoising Autoencoder
1/1 [==============================] - 3s 3s/step
```

```
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
```

Denoising AE - Dim: 32, Noise: 0.1

Original

Reconstrucción
MSE: 0.0314, PSNR: 14.40, SSIM: 0.5927

Original

Reconstrucción
MSE: 0.0340, PSNR: 13.60, SSIM: 0.5619

Original

Reconstrucción
MSE: 0.0232, PSNR: 16.30, SSIM: 0.5776

Original

Reconstrucción
MSE: 0.0346, PSNR: 14.01, SSIM: 0.6088

Original

Reconstrucción
MSE: 0.0310, PSNR: 14.23, SSIM: 0.6010

```
1/1 [==============================] - 1s 1s/step

<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
```

Denoising AE - Dim: 64, Noise: 0.2

Original

Reconstrucción
MSE: 0.0410, PSNR: 13.27, SSIM: 0.6440

Original

Reconstrucción
MSE: 0.0443, PSNR: 12.75, SSIM: 0.6255

Original

Reconstrucción
MSE: 0.0422, PSNR: 13.00, SSIM: 0.6250

Original

Reconstrucción
MSE: 0.0519, PSNR: 11.45, SSIM: 0.5922

Original

Reconstrucción
MSE: 0.0498, PSNR: 12.25, SSIM: 0.6343

```
1/1 [==============================] - 1s 565ms/step
```

```
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
```

Denoising AE - Dim: 128, Noise: 0.3

Original

Reconstrucción
MSE: 0.0328, PSNR: 14.60, SSIM: 0.6173

Original

Reconstrucción
MSE: 0.0451, PSNR: 13.46, SSIM: 0.5899

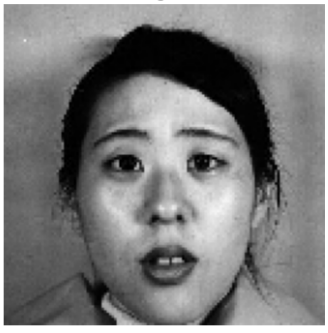
Original

Reconstrucción
MSE: 0.0733, PSNR: 10.26, SSIM: 0.5676

Original

Reconstrucción
MSE: 0.0667, PSNR: 11.12, SSIM: 0.5582

Original



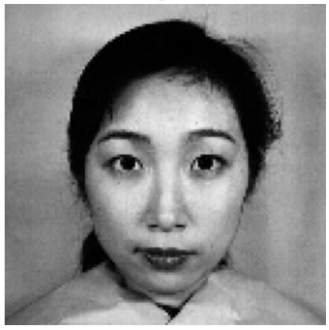
Reconstrucción
MSE: 0.0580, PSNR: 12.09, SSIM: 0.5845

```
Reconstrucción Visual y Métricas - Sparse Autoencoder
1/1 [==============================] - 0s 108ms/step
```

```
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
```

Sparse AE - Dim: 32, Reg: 1e-05

Original

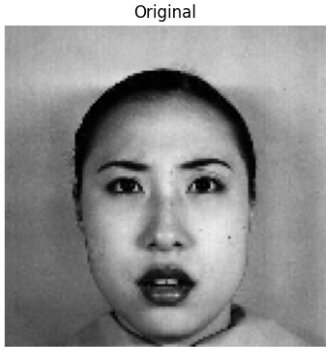


Reconstrucción
MSE: 0.0058, PSNR: 22.13, SSIM: 0.7629

Original

Reconstrucción
MSE: 0.0043, PSNR: 23.03, SSIM: 0.7433

Original



Reconstrucción
MSE: 0.0054, PSNR: 22.16, SSIM: 0.7478

Original

Reconstrucción
MSE: 0.0041, PSNR: 22.97, SSIM: 0.7395

Original

Reconstrucción
MSE: 0.0038, PSNR: 23.39, SSIM: 0.7965

```
1/1 [==============================] - 0s 105ms/step
```

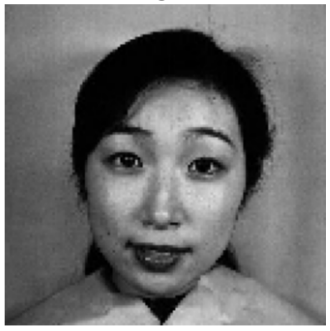
```
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
```

Sparse AE - Dim: 64, Reg: 0.0001

Original

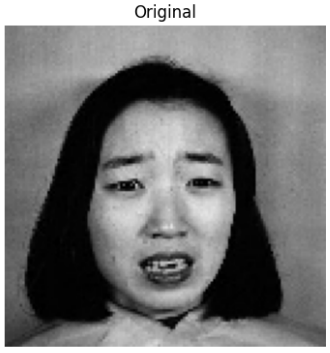
Reconstrucción
MSE: 0.0064, PSNR: 21.07, SSIM: 0.6658

Original



Reconstrucción
MSE: 0.0054, PSNR: 21.53, SSIM: 0.6697

Original



Reconstrucción
MSE: 0.0051, PSNR: 22.00, SSIM: 0.7072

Original

Reconstrucción
MSE: 0.0083, PSNR: 19.59, SSIM: 0.6228

Original

Reconstrucción
MSE: 0.0065, PSNR: 21.17, SSIM: 0.6544

```
1/1 [==============================] - 0s 107ms/step

<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
<ipython-input-55-258969855426>:13: FutureWarning: `multichannel` is a
deprecated argument name for `structural_similarity`. It will be removed in
version 1.0. Please use `channel_axis` instead.
  ssim_val = ssim(original, reconstruido, multichannel=True,
data_range=original.max() - original.min())
```

Sparse AE - Dim: 128, Reg: 0.001

Original

Reconstrucción
MSE: 0.0175, PSNR: 17.58, SSIM: 0.5079

Original

Reconstrucción
MSE: 0.0286, PSNR: 14.24, SSIM: 0.4009

Original

Reconstrucción
MSE: 0.0117, PSNR: 18.75, SSIM: 0.5281

Original

Reconstrucción
MSE: 0.0106, PSNR: 18.85, SSIM: 0.5377

Original

Reconstrucción
MSE: 0.0179, PSNR: 16.12, SSIM: 0.4718

[60]:
```
from sklearn.cluster import AgglomerativeClustering, KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score, calinski_harabasz_score

def evaluar_espacio_latente(autoencoder, X, n_clusters=3, use_pca=False,␣
 ↪clustering_method='kmeans'):
    # Generar el espacio latente
    encoder = Model(inputs=autoencoder.input, outputs=autoencoder.layers[-7].
 ↪output)
    latent_space = encoder.predict(X)
    latent_flattened = latent_space.reshape(len(latent_space), -1)

    # Normalizar el espacio latente
    scaler = StandardScaler()
    latent_scaled = scaler.fit_transform(latent_flattened)

    # Opcional: Reducción de dimensionalidad usando PCA
    if use_pca:
        pca = PCA(n_components=2)
        latent_reduced = pca.fit_transform(latent_scaled)
    else:
        latent_reduced = latent_scaled

    # Selección del método de clustering
    if clustering_method == 'kmeans':
        clusterer = KMeans(n_clusters=n_clusters, random_state=42)
    elif clustering_method == 'agglomerative':
        clusterer = AgglomerativeClustering(n_clusters=n_clusters)
    else:
        raise ValueError("Método de clustering no soportado: usa 'kmeans' o␣
 ↪'agglomerative'")

    # Aplicar clustering
    cluster_labels = clusterer.fit_predict(latent_reduced)

    # Calcular las métricas de calidad del clustering
    silhouette_avg = silhouette_score(latent_reduced, cluster_labels)
    calinski_harabasz = calinski_harabasz_score(latent_reduced, cluster_labels)

    # Imprimir los resultados
    print(f"Clustering Method: {clustering_method.upper()}")
    print(f"Latent Dimensionality: {latent_space.shape[1]}")
    print(f"Use PCA: {use_pca}")
```

```
    print(f"Number of Clusters: {n_clusters}")
    print(f"Silhouette Score: {silhouette_avg:.4f}")
    print(f"Calinski-Harabasz Index: {calinski_harabasz:.4f}")
    print("\n" + "-"*40 + "\n")

    return silhouette_avg, calinski_harabasz

# Ejemplo de uso con diferentes configuraciones
print("Evaluación de Calidad del Espacio Latente")

# Evaluación para Denoising Autoencoder
for i, (config, _) in enumerate(denoising_histories):
    encoding_dim = config['encoding_dim']
    noise_factor = config['noise_factor']
    autoencoder = denoising_models[i]

    print(f"Evaluación para Denoising AE - Dim: {encoding_dim}, Noise Factor:␣
 ↪{noise_factor}")

    # Prueba de clustering con diferentes configuraciones
    evaluar_espacio_latente(autoencoder, X_train, n_clusters=3, use_pca=False,␣
 ↪clustering_method='kmeans')
    evaluar_espacio_latente(autoencoder, X_train, n_clusters=3, use_pca=True,␣
 ↪clustering_method='kmeans')
    evaluar_espacio_latente(autoencoder, X_train, n_clusters=3, use_pca=False,␣
 ↪clustering_method='agglomerative')
    evaluar_espacio_latente(autoencoder, X_train, n_clusters=3, use_pca=True,␣
 ↪clustering_method='agglomerative')

# Evaluación para Sparse Autoencoder
for i, (config, _) in enumerate(sparse_histories):
    encoding_dim = config['encoding_dim']
    regularization = config['regularization']
    autoencoder = sparse_models[i]

    print(f"Evaluación para Sparse AE - Dim: {encoding_dim}, Regularization:␣
 ↪{regularization}")

    # Prueba de clustering con diferentes configuraciones
    evaluar_espacio_latente(autoencoder, X_train, n_clusters=3, use_pca=False,␣
 ↪clustering_method='kmeans')
    evaluar_espacio_latente(autoencoder, X_train, n_clusters=3, use_pca=True,␣
 ↪clustering_method='kmeans')
    evaluar_espacio_latente(autoencoder, X_train, n_clusters=3, use_pca=False,␣
 ↪clustering_method='agglomerative')
```

```
    evaluar_espacio_latente(autoencoder, X_train, n_clusters=3, use_pca=True,␣
↪clustering_method='agglomerative')
```

```
Evaluación de Calidad del Espacio Latente
Evaluación para Denoising AE - Dim: 32, Noise Factor: 0.1
5/5 [==============================] - 0s 4ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.1953
Calinski-Harabasz Index: 32.4739

----------------------------------------

5/5 [==============================] - 0s 4ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5513
Calinski-Harabasz Index: 226.6611

----------------------------------------

5/5 [==============================] - 0s 3ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.1953
Calinski-Harabasz Index: 32.4739

----------------------------------------

5/5 [==============================] - 0s 4ms/step
Clustering Method: AGGLOMERATIVE
```

```
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5513
Calinski-Harabasz Index: 226.6607

----------------------------------------

Evaluación para Denoising AE - Dim: 64, Noise Factor: 0.2
5/5 [==============================] - 0s 3ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.1714
Calinski-Harabasz Index: 28.5000

----------------------------------------

5/5 [==============================] - 0s 3ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5706
Calinski-Harabasz Index: 245.9239

----------------------------------------

5/5 [==============================] - 0s 3ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.1438
Calinski-Harabasz Index: 25.5822

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

```
5/5 [==============================] - 0s 4ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5693
Calinski-Harabasz Index: 242.9856

----------------------------------------

Evaluación para Denoising AE - Dim: 128, Noise Factor: 0.3
5/5 [==============================] - 0s 3ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.1659
Calinski-Harabasz Index: 26.4289

----------------------------------------

5/5 [==============================] - 0s 4ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5903
Calinski-Harabasz Index: 259.3354

----------------------------------------

5/5 [==============================] - 0s 4ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.1428
```

```
Calinski-Harabasz Index: 23.9274

----------------------------------------

5/5 [==============================] - 0s 3ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5894
Calinski-Harabasz Index: 256.4067

----------------------------------------

Evaluación para Sparse AE - Dim: 32, Regularization: 1e-05
5/5 [==============================] - 0s 3ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.0976
Calinski-Harabasz Index: 16.2710

----------------------------------------

5/5 [==============================] - 0s 4ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5681
Calinski-Harabasz Index: 207.6326

----------------------------------------

5/5 [==============================] - 0s 4ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
```

```
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.1707
Calinski-Harabasz Index: 15.5690

----------------------------------------

5/5 [==============================] - 0s 3ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5681
Calinski-Harabasz Index: 207.6330

----------------------------------------

Evaluación para Sparse AE - Dim: 64, Regularization: 0.0001
5/5 [==============================] - 0s 4ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.0425
Calinski-Harabasz Index: 9.5547

----------------------------------------

5/5 [==============================] - 0s 3ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5382
Calinski-Harabasz Index: 226.2074

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

```
5/5 [==============================] - 0s 3ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.0687
Calinski-Harabasz Index: 9.6424

----------------------------------------

5/5 [==============================] - 0s 4ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5479
Calinski-Harabasz Index: 220.8955

----------------------------------------

Evaluación para Sparse AE - Dim: 128, Regularization: 0.001
5/5 [==============================] - 0s 3ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.0561
Calinski-Harabasz Index: 10.9030

----------------------------------------

5/5 [==============================] - 0s 3ms/step

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

Clustering Method: KMEANS
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5297
Calinski-Harabasz Index: 240.5233
```

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

5/5 [==============================] - 0s 3ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: False
Number of Clusters: 3
Silhouette Score: 0.1251
Calinski-Harabasz Index: 11.0390

----------------------------------------

5/5 [==============================] - 0s 3ms/step
Clustering Method: AGGLOMERATIVE
Latent Dimensionality: 32
Use PCA: True
Number of Clusters: 3
Silhouette Score: 0.5307
Calinski-Harabasz Index: 233.1754

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

2 Mejor opcion: Denoising AE con Dim: 128 y Noise Factor: 0.3

[62]:
```python
from sklearn.metrics import davies_bouldin_score, silhouette_score
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

# Definir la mejor configuración encontrada
best_autoencoder = denoising_models[2]  # Este índice debe coincidir con el␣
 ↪Denoising AE con Dim: 128 y Noise Factor: 0.3
n_clusters = 3
use_pca = True

# Generar el espacio latente para el conjunto de prueba
encoder = Model(inputs=best_autoencoder.input, outputs=best_autoencoder.
 ↪layers[-7].output)
latent_space_test = encoder.predict(X_test)
latent_flattened_test = latent_space_test.reshape(len(latent_space_test), -1)

# Normalizar el espacio latente
scaler = StandardScaler()
latent_scaled_test = scaler.fit_transform(latent_flattened_test)
```

```
# Aplicar PCA si es necesario
if use_pca:
    pca = PCA(n_components=2)
    latent_reduced_test = pca.fit_transform(latent_scaled_test)
else:
    latent_reduced_test = latent_scaled_test

# Aplicar el método de clustering (KMeans)
clusterer = KMeans(n_clusters=n_clusters, random_state=42)
cluster_labels_test = clusterer.fit_predict(latent_reduced_test)

# Calcular las métricas de clasificación en el conjunto de prueba
silhouette_avg_test = silhouette_score(latent_reduced_test, cluster_labels_test)
davies_bouldin_test = davies_bouldin_score(latent_reduced_test,␣
 ↪cluster_labels_test)

# Imprimir los resultados
print("Resultados en el conjunto de prueba externo:")
print(f"Silhouette Score: {silhouette_avg_test:.4f}")
print(f"Davies-Bouldin Index: {davies_bouldin_test:.4f}")
```

```
1/1 [==============================] - 0s 68ms/step
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

```
Resultados en el conjunto de prueba externo:
Silhouette Score: 0.5759
Davies-Bouldin Index: 0.4069
```

Interpretación del Silhouette Score: * Cercano a 1: Indica clusters bien definidos y separados; los puntos están cerca de su propio cluster y lejos de los clusters vecinos. * Cercano a 0: Sugiere que los clusters se superponen y que los puntos están cerca de los límites de otros clusters, por lo que la separación no es ideal. * Negativo: Indica que muchos puntos pueden estar en el cluster incorrecto o que el clustering no tiene una estructura clara.

Interpretación del DBI: * DBI cercano a 0: Indica una excelente separación y compactación, lo cual es ideal en clustering. * Valores de DBI bajos (por debajo de 1.0 generalmente se considera bueno): Sugiere que los clusters están razonablemente bien formados y separados. * Valores altos de DBI (mayores que 1.0): Indican que los clusters están menos bien definidos y que los puntos están menos agrupados, posiblemente con una superposición significativa entre clusters.

- 0.57 indica que los clusters tienen una separación aceptable y cohesión interna moderada, lo cual puede ser adecuado para muchos problemas de clustering.
- Este valor, junto con un Davies-Bouldin Index de 0.4, sugiere que el esquema de clustering es en general bueno, con clusters razonablemente definidos y bien separados.