```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow as tf

from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.model_selection import train_test_split
from tensorflow.keras import layers, losses
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Model
```

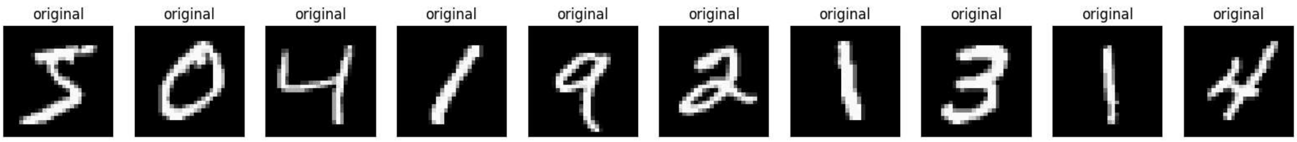
```
(x_train, _), (x_test, _) = mnist.load_data()
x_train, x_val = x_train[:-10000], x_train[-10000:]

x_train = x_train.astype('float32') / 255.
x_test = x_test.astype('float32') / 255.
x_val = x_val.astype('float32') / 255.

print(x_train.shape)
print(x_test.shape)
print(x_val.shape)
```

```
(50000, 28, 28)
(10000, 28, 28)
(10000, 28, 28)
```

```
#visualize
n=10
plt.figure(figsize=(20, 4))
for i in range(n):
    ax=plt.subplot(2, n, i+1)
    plt.imshow(x_train[i])
    plt.title("original")
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
```



```
latent_dim = 64

class Autoencoder(Model):
  def __init__(self, latent_dim):
    super(Autoencoder, self).__init__()
    self.latent_dim = latent_dim
    self.encoder = tf.keras.Sequential([
      layers.Flatten(),
      layers.Dense(latent_dim, activation='relu'),
    ])
    self.decoder = tf.keras.Sequential([
      layers.Dense(784, activation='sigmoid'),
      layers.Reshape((28, 28))
    ])

  def call(self, x):
    encoded = self.encoder(x)
    decoded = self.decoder(encoded)
    return decoded

autoencoder = Autoencoder(latent_dim)
```

```
#Define Autoencoder class

autoencoder.compile(optimizer='adam', loss=losses.MeanSquaredError())

#Train Autoencoder

autoencoder.fit(x_train, x_train,
                epochs=10,
                shuffle=True,
                validation_data=(x_val, x_val))
```

```
Epoch 1/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 2ms/step - loss: 0.0477 - val_loss: 0.0112
Epoch 2/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - loss: 0.0093 - val_loss: 0.0062
Epoch 3/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - loss: 0.0058 - val_loss: 0.0050
Epoch 4/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 6s 4ms/step - loss: 0.0049 - val_loss: 0.0047
Epoch 5/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 4s 3ms/step - loss: 0.0045 - val_loss: 0.0044
Epoch 6/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - loss: 0.0044 - val_loss: 0.0043
Epoch 7/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 8s 2ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 8/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 2ms/step - loss: 0.0042 - val_loss: 0.0042
Epoch 9/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 2ms/step - loss: 0.0041 - val_loss: 0.0042
Epoch 10/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 5s 2ms/step - loss: 0.0041 - val_loss: 0.0041
<keras.src.callbacks.history.History at 0x7c8e8239ed70>
```

```
print(autoencoder.encoder.summary())
```

```
Model: "sequential_2"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━
┃ Layer (type)                        ┃ Output Shape                ┃       Param
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━
│ flatten_1 (Flatten)                 │ (None, 784)                 │
├─────────────────────────────────────┼─────────────────────────────┼────────────
│ dense_2 (Dense)                     │ (None, 64)                  │       50,24
└─────────────────────────────────────┴─────────────────────────────┴────────────
 Total params: 50,240 (196.25 KB)
 Trainable params: 50,240 (196.25 KB)
 Non-trainable params: 0 (0.00 B)
None
```

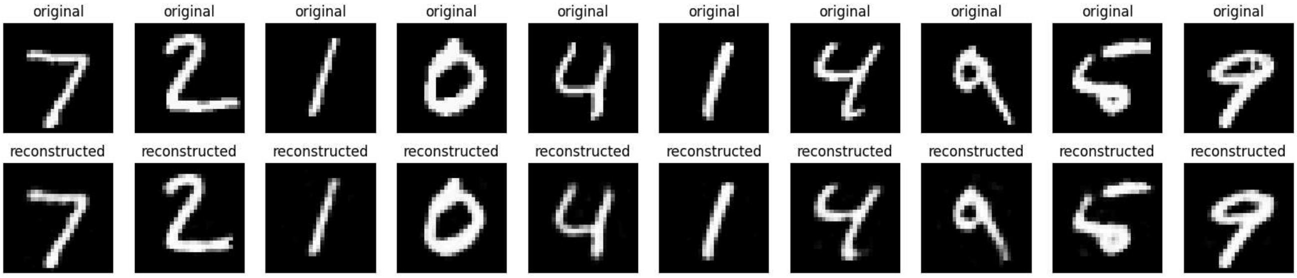
```
print(autoencoder.decoder.summary())
```

```
Model: "sequential_3"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━
┃ Layer (type)                        ┃ Output Shape                ┃       Param
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━
│ dense_3 (Dense)                     │ (None, 784)                 │       50,96
├─────────────────────────────────────┼─────────────────────────────┼────────────
│ reshape_1 (Reshape)                 │ (None, 28, 28)              │
└─────────────────────────────────────┴─────────────────────────────┴────────────
 Total params: 50,960 (199.06 KB)
 Trainable params: 50,960 (199.06 KB)
 Non-trainable params: 0 (0.00 B)
None
```

```
encoded_imgs = autoencoder.encoder(x_test).numpy()
decoded_imgs = autoencoder.decoder(encoded_imgs).numpy()
```

```
n = 10
plt.figure(figsize=(20, 4))
for i in range(n):
    # display original
    ax = plt.subplot(2, n, i +1)
    plt.imshow(x_test[i])
    plt.title("original")
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

    #display reconstruction
    ax = plt.subplot(2, n, i + 1 + n)
    plt.imshow(decoded_imgs[i])
    plt.title("reconstructed")
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```



Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.