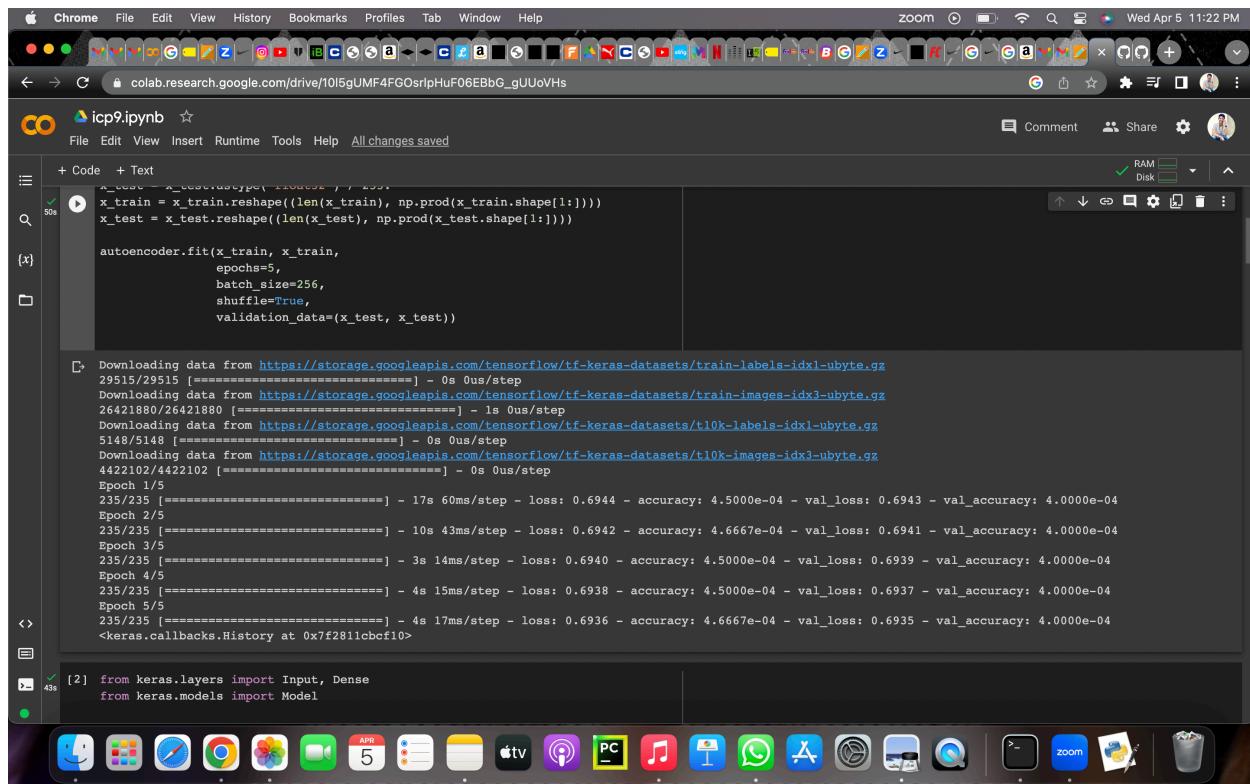


ICP 9 REPORT
KRISHNA VAMSI GIDUTHURI
700743211

- Dowloaded the content from cigar



Screenshot of a Google Colab notebook titled "icp9.ipynb". The code cell contains the following Python code:

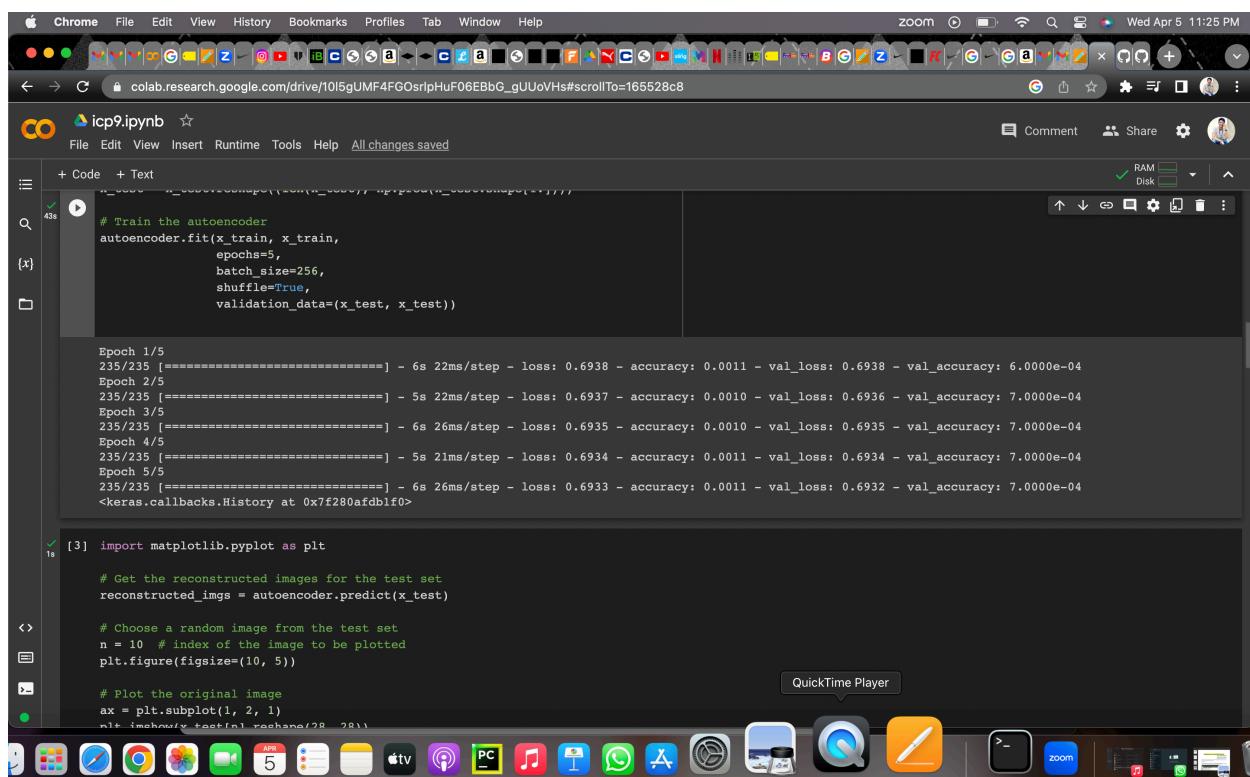
```
x_test = x_test.astype('float32') / 255.
x_train = x_train.reshape(len(x_train), np.prod(x_train.shape[1:]))
x_test = x_test.reshape((len(x_test), np.prod(x_test.shape[1:])))

autoencoder.fit(x_train, x_train,
                 epochs=5,
                 batch_size=256,
                 shuffle=True,
                 validation_data=(x_test, x_test))

[ Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
  29515/29515 [=====] - 0s 0us/step
  Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
  26421880/26421880 [=====] - 1s 0us/step
  Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
  5148/5148 [=====] - 0s 0us/step
  Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
  4422102/4422102 [=====] - 0s 0us/step
  Epoch 1/5
  235/235 [=====] - 17s 60ms/step - loss: 0.6944 - accuracy: 4.5000e-04 - val_loss: 0.6943 - val_accuracy: 4.0000e-04
  Epoch 2/5
  235/235 [=====] - 10s 43ms/step - loss: 0.6942 - accuracy: 4.6667e-04 - val_loss: 0.6941 - val_accuracy: 4.0000e-04
  Epoch 3/5
  235/235 [=====] - 3s 14ms/step - loss: 0.6940 - accuracy: 4.5000e-04 - val_loss: 0.6939 - val_accuracy: 4.0000e-04
  Epoch 4/5
  235/235 [=====] - 4s 15ms/step - loss: 0.6938 - accuracy: 4.5000e-04 - val_loss: 0.6937 - val_accuracy: 4.0000e-04
  Epoch 5/5
  235/235 [=====] - 4s 17ms/step - loss: 0.6936 - accuracy: 4.6667e-04 - val_loss: 0.6935 - val_accuracy: 4.0000e-04
<keras.callbacks.History at 0x7f2811cbcf10>

[2] from keras.layers import Input, Dense
    from keras.models import Model
```

- Encoded the representation of the input and also load the MNIST dataset



Screenshot of a Google Colab notebook titled "icp9.ipynb". The code cell contains the following Python code:

```
# Train the autoencoder
autoencoder.fit(x_train, x_train,
                 epochs=5,
                 batch_size=256,
                 shuffle=True,
                 validation_data=(x_test, x_test))

[ Epoch 1/5
  235/235 [=====] - 6s 22ms/step - loss: 0.6938 - accuracy: 0.0011 - val_loss: 0.6938 - val_accuracy: 6.0000e-04
  Epoch 2/5
  235/235 [=====] - 5s 22ms/step - loss: 0.6937 - accuracy: 0.0010 - val_loss: 0.6936 - val_accuracy: 7.0000e-04
  Epoch 3/5
  235/235 [=====] - 6s 26ms/step - loss: 0.6935 - accuracy: 0.0010 - val_loss: 0.6935 - val_accuracy: 7.0000e-04
  Epoch 4/5
  235/235 [=====] - 5s 21ms/step - loss: 0.6934 - accuracy: 0.0011 - val_loss: 0.6934 - val_accuracy: 7.0000e-04
  Epoch 5/5
  235/235 [=====] - 6s 26ms/step - loss: 0.6933 - accuracy: 0.0011 - val_loss: 0.6932 - val_accuracy: 7.0000e-04
<keras.callbacks.History at 0x7f280afdb1f0>

[3] import matplotlib.pyplot as plt

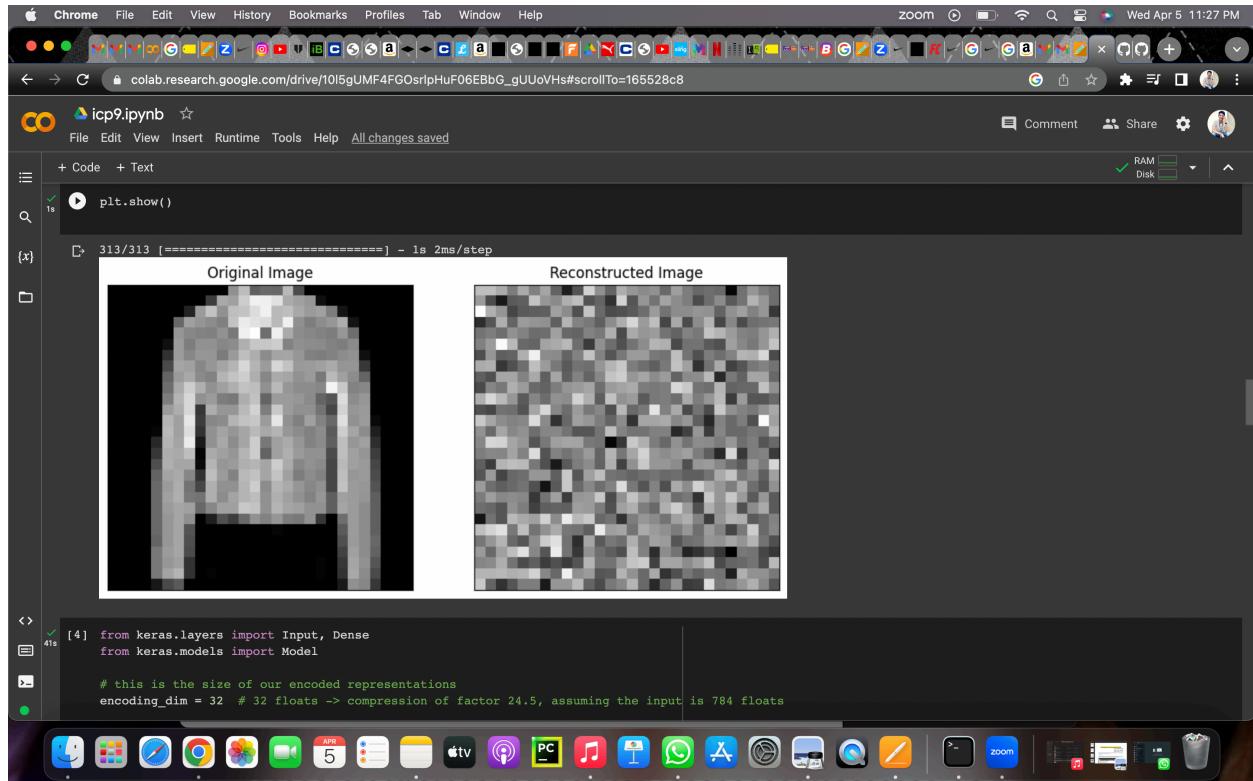
# Get the reconstructed images for the test set
reconstructed_imgs = autoencoder.predict(x_test)

# Choose a random image from the test set
n = 10 # index of the image to be plotted
plt.figure(figsize=(10, 5))

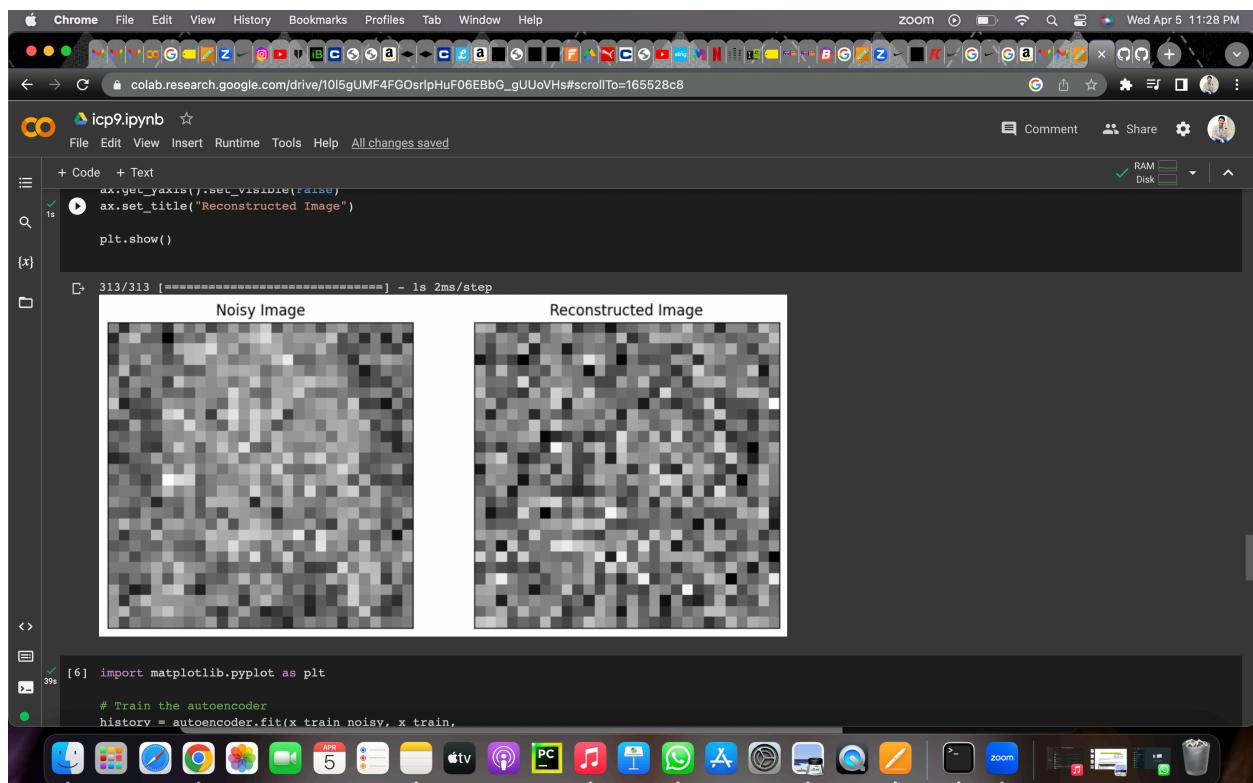
# Plot the original image
ax = plt.subplot(1, 2, 1)
plt.imshow(x_test[0], cmap='gray')

# Plot the reconstructed image
ax = plt.subplot(1, 2, 2)
plt.imshow(reconstructed_imgs[0], cmap='gray')
```

Reconstructed images for set test data and with original and reconstructed images



NOISY AND RECONSTRUCTED IMAGE



Plotting the loss and accuracy

