Lab Report – 7

Autoencoders, Object Detection

Goutham Krishnan

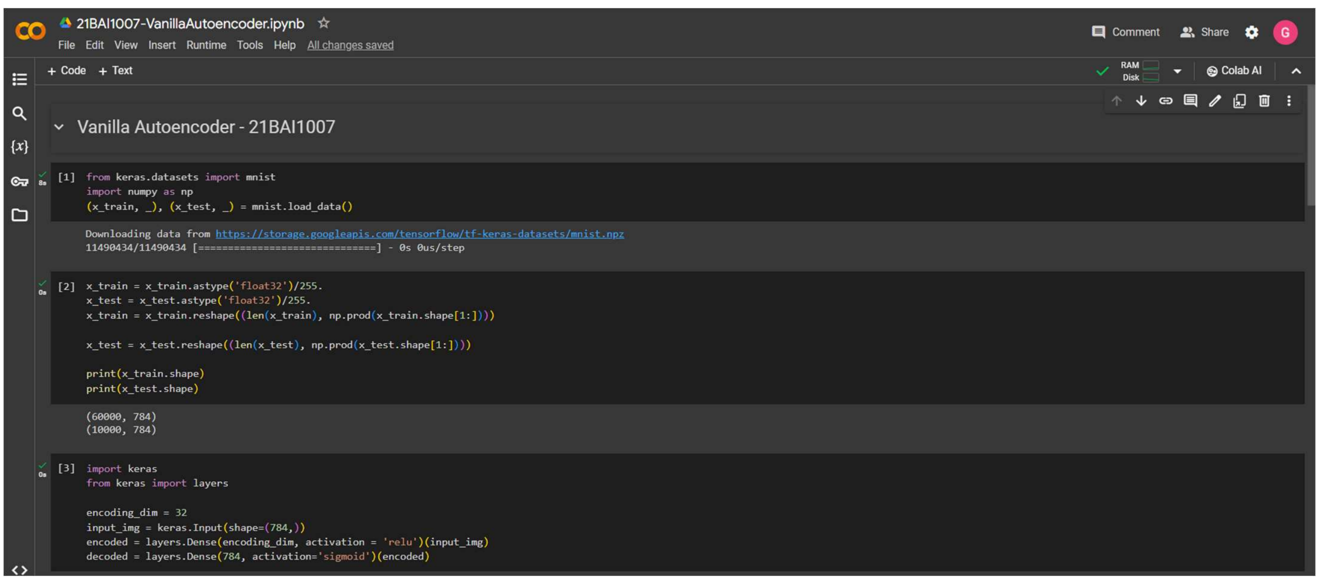
21BAI1007

Aim

1. Execute the two sample programs, one for vanilla autoencoder and one for denoising autoencoder on the MNIST dataset.
2. Create an object detection model using YOLO that detects the eyes and face.
3. Using autoencoders, implement the dimensionality reduction of MNIST handwritten image dataset.

Observations and Output

<u>For the sample code</u>



21BAI1007-VanillaAutoencoder.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```python
autoencoder = keras.Model(input_img, decoded)
encoder=keras.Model(input_img, encoded)

encoded_input = keras.Input(shape=(encoding_dim,))

decoder_layer = autoencoder.layers[-1]

decoder = keras.Model(encoded_input, decoder_layer(encoded_input))
autoencoder.compile(optimizer='adam', loss='binary_crossentropy')
autoencoder.fit(x_train, x_train, epochs=50, batch_size=256, shuffle=True, validation_data=(x_test, x_test))
```

```
235/235 [==============================] - 3s 12ms/step - loss: 0.0933 - val_loss: 0.0920
Epoch 23/50
235/235 [==============================] - 3s 12ms/step - loss: 0.0932 - val_loss: 0.0920
Epoch 24/50
235/235 [==============================] - 3s 11ms/step - loss: 0.0931 - val_loss: 0.0919
Epoch 25/50
235/235 [==============================] - 2s 10ms/step - loss: 0.0931 - val_loss: 0.0919
Epoch 26/50
235/235 [==============================] - 3s 11ms/step - loss: 0.0930 - val_loss: 0.0919
Epoch 27/50
235/235 [==============================] - 3s 13ms/step - loss: 0.0930 - val_loss: 0.0919
Epoch 28/50
235/235 [==============================] - 3s 11ms/step - loss: 0.0930 - val_loss: 0.0918
Epoch 29/50
235/235 [==============================] - 2s 11ms/step - loss: 0.0929 - val_loss: 0.0917
Epoch 30/50
235/235 [==============================] - 2s 11ms/step - loss: 0.0929 - val_loss: 0.0917
Epoch 31/50
235/235 [==============================] - 2s 11ms/step - loss: 0.0929 - val_loss: 0.0916
Epoch 32/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0928 - val_loss: 0.0917
Epoch 33/50
235/235 [==============================] - 2s 10ms/step - loss: 0.0928 - val_loss: 0.0917
Epoch 34/50
235/235 [==============================] - 2s 10ms/step - loss: 0.0928 - val_loss: 0.0917
Epoch 35/50
235/235 [==============================] - 2s 10ms/step - loss: 0.0928 - val_loss: 0.0916
```

21BAI1007-VanillaAutoencoder.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[5]

```python
encoded_imgs = encoder.predict(x_test)
decoded_imgs = decoder.predict(encoded_imgs)
```

```
313/313 [==============================] - 1s 2ms/step
313/313 [==============================] - 0s 1ms/step
```

[6]

```python
import matplotlib.pyplot as plt
n=10
plt.figure(figsize=(20, 4))

for i in range(n):
  ax = plt.subplot(2, n, i+1)
  plt.imshow(x_test[i].reshape(28, 28))
  plt.gray()
  ax.get_xaxis().set_visible(False)
  ax.get_yaxis().set_visible(False)
  plt.imshow(decoded_imgs[i].reshape(28, 28))
  plt.gray()
  ax.get_xaxis().set_visible(False)
  ax.get_yaxis().set_visible(False)
  plt.show()
```

21BAI1007-VanillaAutoencoder.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[6]

21BAI1007-Denoising_Autoencoder.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Denoising Autoencoders - 21BAI1007

```
from keras.datasets import mnist
import numpy as np
(x_train, _), (x_test, _) = mnist.load_data()
x_train = x_train.astype('float32')/255.
x_test = x_test.astype('float32')/255.
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

[2]
```
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:])))

noise_factor = 0.5

x_train_noisy = x_train + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_train.shape)
x_test_noisy = x_test + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_test.shape)
x_train_noisy = np.clip(x_train_noisy, 0., 1.)
x_test_noisy = np.clip(x_test_noisy, 0., 1.)
```

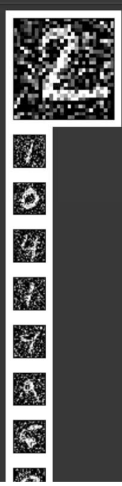
[3]
```
import matplotlib.pyplot as plt
n = 10
plt.figure(figsize = (20, 2))

for i in range(1, n+1):
  ax = plt.subplot(1, n, i)
  plt.imshow(x_test_noisy[i].reshape(28, 28))
  plt.gray()
  ax.get_xaxis().set_visible(False)
  ax.get_yaxis().set_visible(False)
  plt.show()
```

21BAI1007-Denoising_Autoencoder.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[3]



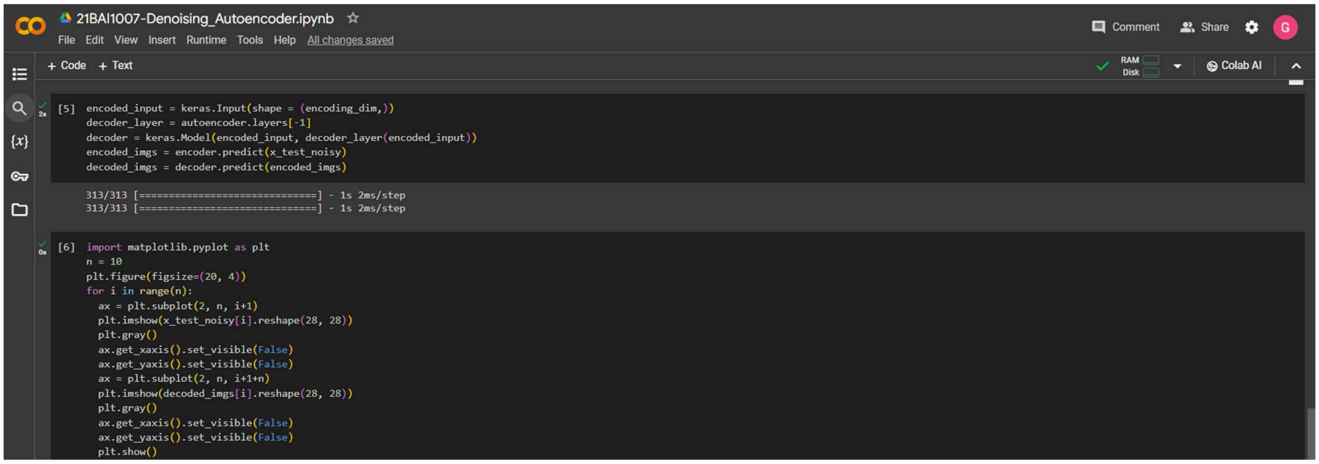
21BAI1007-Denoising_Autoencoder.ipynb

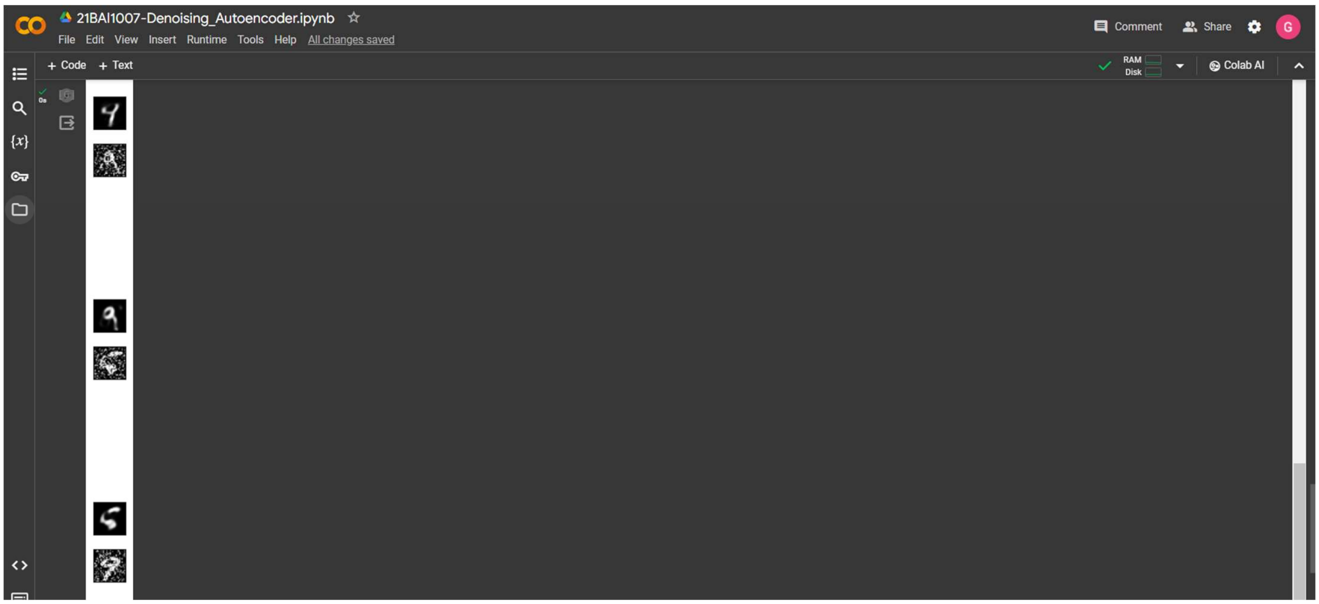
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[4]
```
import keras
from keras import layers
import tensorflow as tf
from keras.callbacks import TensorBoard
encoding_dim = 32
input_img = keras.Input(shape=(784,))

encoded = layers.Dense(encoding_dim, activation = 'relu')(input_img)
decoded = layers.Dense(784, activation = 'sigmoid')(encoded)

autoencoder = keras.Model(input_img, decoded)
autoencoder.compile(optimizer = 'adam', loss='binary_crossentropy')
autoencoder.fit(x_train_noisy, x_train, epochs = 100, batch_size = 128, shuffle = True, validation_data=(x_test_noisy, x_test),)
encoder = keras.Model(input_img, encoded)
```

```
Epoch 72/100
469/469 [==============================] - 3s 6ms/step - loss: 0.1255 - val_loss: 0.1247
Epoch 73/100
469/469 [==============================] - 3s 6ms/step - loss: 0.1255 - val_loss: 0.1247
Epoch 74/100
469/469 [==============================] - 4s 9ms/step - loss: 0.1255 - val_loss: 0.1246
Epoch 75/100
469/469 [==============================] - 3s 7ms/step - loss: 0.1255 - val_loss: 0.1246
Epoch 76/100
469/469 [==============================] - 3s 6ms/step - loss: 0.1255 - val_loss: 0.1245
Epoch 77/100
469/469 [==============================] - 3s 6ms/step - loss: 0.1255 - val_loss: 0.1245
Epoch 78/100
469/469 [==============================] - 4s 9ms/step - loss: 0.1255 - val_loss: 0.1245
Epoch 79/100
469/469 [==============================] - 3s 6ms/step - loss: 0.1255 - val_loss: 0.1249
Epoch 80/100
469/469 [==============================] - 3s 6ms/step - loss: 0.1255 - val_loss: 0.1246
Epoch 81/100
469/469 [==============================] - 3s 7ms/step - loss: 0.1255 - val_loss: 0.1247
Epoch 82/100
469/469 [==============================] - 4s 9ms/step - loss: 0.1254 - val_loss: 0.1245
Epoch 83/100
```





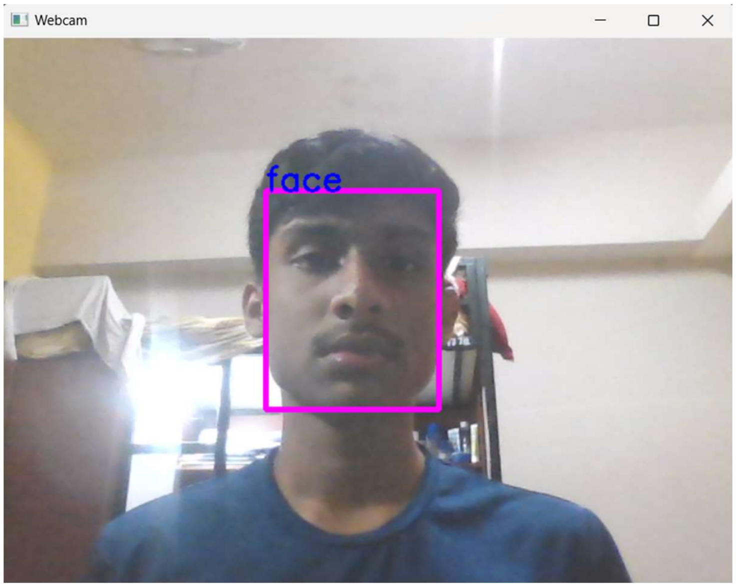
For Face Detection Model using YOLO

Code:

```
from ultralytics import YOLO
import cv2
import math
# start webcam
cap = cv2.VideoCapture(0)
cap.set(3, 640)
cap.set(4, 480)

# model
model = YOLO("yolov8n-face.pt")

# object classes
classNames = ["face"]


while True:
    success, img = cap.read()
    results = model(img, stream=True)
```

```
    # coordinates
    for r in results:
        boxes = r.boxes

        for box in boxes:
            # bounding box
            x1, y1, x2, y2 = box.xyxy[0]
            x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2) # convert
to int values

            # put box in cam
            cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 255), 3)

            # confidence
            confidence = math.ceil((box.conf[0]*100))/100
            print("Confidence --->",confidence)

            # class name
            cls = int(box.cls[0])
            print("Class name -->", classNames[cls])

            # object details
            org = [x1, y1]
            font = cv2.FONT_HERSHEY_SIMPLEX
            fontScale = 1
            color = (255, 0, 0)
            thickness = 2

            cv2.putText(img, classNames[cls], org, font, fontScale, color,
thickness)

    cv2.imshow('Webcam', img)
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Output:



For dimensionality reduction using Autoencoders

21BAI1007-DimReduction-MNIST.ipynb

Dimensionality Reduction of MNIST Dataset using AutoEncoders

Goutham Krishnan 21BAI1007

```
[1] import numpy as np
    import matplotlib.pyplot as plt
    from keras.datasets import mnist
    from keras.models import Model, Sequential
    from keras.layers import Reshape, Flatten, Dense, Lambda
    from keras import losses
```

Loading the data

```
[2] (train_images, _), (test_images, _) = mnist.load_data()

    Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
    11490434/11490434 [==============================] - 0s 0us/step
```

Scaling the data

```
[3] # Scaling
    x_train = train_images.astype('float32')/255.
    x_test = test_images.astype('float32')/255.

    print(x_train.shape)
    print(x_test.shape)

    (60000, 28, 28)
    (10000, 28, 28)
```

21BAI1007-DimReduction-MNIST.ipynb

Creating the Autoencoder model

```
latent_dim = 64

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

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

autoencoder = Autoencoder(latent_dim)
```

```
[5] autoencoder.compile(optimizer='adam', loss=losses.MeanSquaredError())
```

21BAI1007-DimReduction-MNIST.ipynb

Train the model

```
autoencoder.fit(x_train, x_train, epochs=20, shuffle=True)

Epoch 1/20
1875/1875 [==============================] - 8s 3ms/step - loss: 0.0239
Epoch 2/20
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0068
Epoch 3/20
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0050
Epoch 4/20
1875/1875 [==============================] - 5s 2ms/step - loss: 0.0045
Epoch 5/20
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0043
Epoch 6/20
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0042
Epoch 7/20
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0042
Epoch 8/20
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0041
Epoch 9/20
1875/1875 [==============================] - 5s 2ms/step - loss: 0.0040
Epoch 10/20
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0040
Epoch 11/20
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0040
Epoch 12/20
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0039
Epoch 13/20
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0039
Epoch 14/20
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0039
Epoch 15/20
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0039
Epoch 16/20
1875/1875 [==============================] - 5s 2ms/step - loss: 0.0039
```

21BAI1007-DimReduction-MNIST.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
[7] print(autoencoder.encoder.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (32, 784)                 0

 dense (Dense)               (32, 64)                  50240

=================================================================
Total params: 50240 (196.25 KB)
Trainable params: 50240 (196.25 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

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

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_1 (Dense)             (32, 784)                 50960

 reshape (Reshape)           (32, 28, 28)              0

=================================================================
Total params: 50960 (199.06 KB)
Trainable params: 50960 (199.06 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

21BAI1007-DimReduction-MNIST.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Reconstruct the test images

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

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

    ax = plt.subplot(2, n, i+1+n)
    plt.imshow(decoded_imgs[i])
    plt.title("Reconstructed images")
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

plt.show()
```

21BAI1007-DimReduction-MNIST.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
    ax.get_yaxis().set_visible(False)

    ax = plt.subplot(2, n, i+1+n)
    plt.imshow(decoded_imgs[i])
    plt.title("Reconstructed images")
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

plt.show()
```