Assignment 13

In [11]:

```
import tensorflow as tf

# Load the MNIST dataset
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()

# Normalize pixel values to [0, 1]
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

# Add a channel dimension to the images
x_train = x_train[..., tf.newaxis]
x_test = x_test[..., tf.newaxis]

# Set up the layers of the network
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(10, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Train the model on the MNIST dataset
model.fit(x_train, y_train, epochs=3, validation_data=(x_test, y_test))

# Evaluate the model on the MNIST test data
test_loss, test_acc = model.evaluate(x_test, y_test)
print('Test accuracy:', test_acc)
```

```
Epoch 1/3
1875/1875 [==============================] - 81s 43ms/step - loss: 0.1569 - accuracy: 0.953
1 - val_loss: 0.0573 - val_accuracy: 0.9810
Epoch 2/3
1875/1875 [==============================] - 75s 40ms/step - loss: 0.0544 - accuracy: 0.983
0 - val_loss: 0.0438 - val_accuracy: 0.9852
Epoch 3/3
1875/1875 [==============================] - 72s 39ms/step - loss: 0.0391 - accuracy: 0.987
9 - val_loss: 0.0327 - val_accuracy: 0.9893
313/313 [==============================] - 5s 15ms/step - loss: 0.0327 - accuracy: 0.9893
Test accuracy: 0.989300012588501
```