BHARAT DATA SCIENCE INTERNSHIP

TASK-3: Number Recognition

Handwritten digit recognition system not only detects scanned images of handwritten digits. Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of handwritten digits.

In [12]:
```
import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [4]:
```
(train_images, train_labels), (test_images, test_labels) = keras.datasets.mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz (https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz)
11490434/11490434 [==============================] - 5s 0us/step
```

In [5]:
```
train_images = train_images / 255.0
test_images = test_images / 255.0
```

In [6]:
```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(10, activation='softmax')
])
```

In [7]:
```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

In [8]:
```
model.fit(train_images, train_labels, epochs=10)
```

```
Epoch 1/10
1875/1875 [==============================] - 5s 2ms/step - loss: 0.2857 - accuracy: 0.9171
Epoch 2/10
1875/1875 [==============================] - 5s 3ms/step - loss: 0.1359 - accuracy: 0.9589
Epoch 3/10
1875/1875 [==============================] - 5s 3ms/step - loss: 0.1051 - accuracy: 0.9683
Epoch 4/10
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0838 - accuracy: 0.9742
Epoch 5/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.0712 - accuracy: 0.9775
Epoch 6/10
1875/1875 [==============================] - 7s 3ms/step - loss: 0.0610 - accuracy: 0.9809
Epoch 7/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.0567 - accuracy: 0.9821
Epoch 8/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.0516 - accuracy: 0.9828
Epoch 9/10
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0464 - accuracy: 0.9843
Epoch 10/10
1875/1875 [==============================] - 5s 2ms/step - loss: 0.0429 - accuracy: 0.9855
```

Out[8]:
```
<keras.src.callbacks.History at 0x2187a7d3290>
```

In [9]:
```
test_loss, test_acc = model.evaluate(test_images, test_labels)
print(f"Test accuracy: {test_acc}")
```

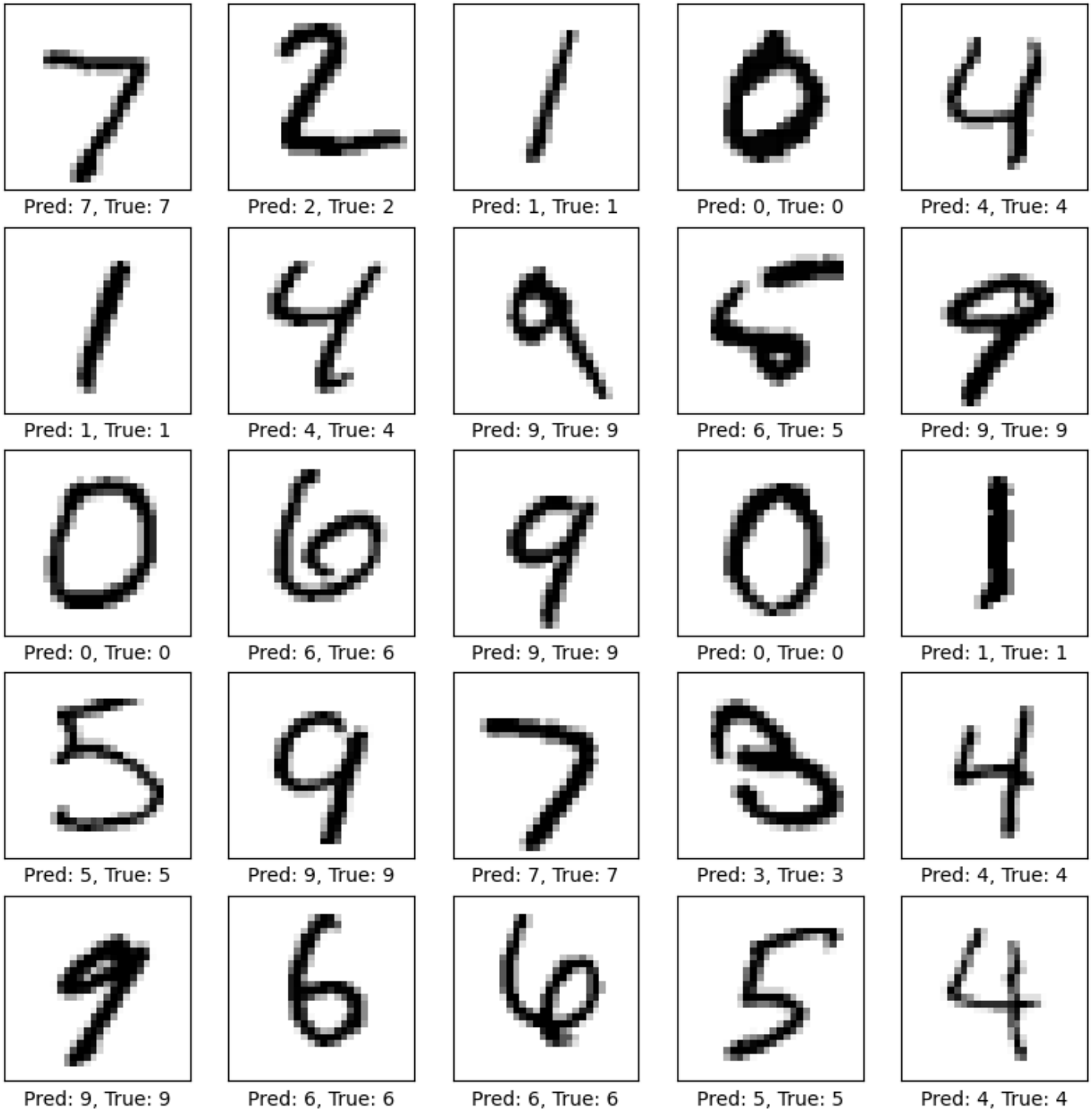
```
313/313 [==============================] - 1s 1ms/step - loss: 0.0765 - accuracy: 0.9795
Test accuracy: 0.9794999957084656
```

In [10]:
```
predictions = model.predict(test_images)
```

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

In [11]:

```
plt.figure(figsize=(10, 10))
for i in range(25):
    plt.subplot(5, 5, i + 1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(test_images[i], cmap=plt.cm.binary)
    predicted_label = np.argmax(predictions[i])
    true_label = test_labels[i]
    plt.xlabel(f"Pred: {predicted_label}, True: {true_label}")
plt.show()
```


Pred: 7, True: 7
Pred: 2, True: 2
Pred: 1, True: 1
Pred: 0, True: 0
Pred: 4, True: 4
Pred: 1, True: 1
Pred: 4, True: 4
Pred: 9, True: 9
Pred: 6, True: 5
Pred: 9, True: 9
Pred: 0, True: 0
Pred: 6, True: 6
Pred: 9, True: 9
Pred: 0, True: 0
Pred: 1, True: 1
Pred: 5, True: 5
Pred: 9, True: 9
Pred: 7, True: 7
Pred: 3, True: 3
Pred: 4, True: 4
Pred: 9, True: 9
Pred: 6, True: 6
Pred: 6, True: 6
Pred: 5, True: 5
Pred: 4, True: 4