```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import tensorflow as tf

# 2.1 Load the data
df = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Q2/Dry_Bean_Dataset.xlsx')

# 2.2 Convert the data to a suitable format
class_map = {'BARBUNYA': 0, 'BOMBAY': 1, 'CALI': 2, 'DERMASON': 3, 'HOROZ': 4, 'SEKER': 5, 'SIRA': 6}
df['Class'] = df['Class'].map(class_map)

X = df.drop(['Class'], axis=1).values
y = df['Class'].values

scaler = StandardScaler()
X = scaler.fit_transform(X)

# 2.3 Separate training and testing data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 2.4 Using TensorFlow, define and train an ANN
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(128, input_shape=(16,), activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(7, activation='softmax')
])

model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train, validation_split=0.2, epochs=50, batch_size=32)

# 2.5 Train the ANN
test_loss, test_acc = model.evaluate(X_test, y_test)

# 2.6 Measure the performance of ANN against test data
print(f'Test accuracy: {test_acc:.2f}')
```

```
Epoch 1/50
273/273 [==============================] - 2s 3ms/step - loss: 0.4444 - accuracy: 0.8699 - val_loss: 0.2359 - val_accuracy: 0.9247
Epoch 2/50
273/273 [==============================] - 1s 4ms/step - loss: 0.2127 - accuracy: 0.9239 - val_loss: 0.2358 - val_accuracy: 0.9137
Epoch 3/50
273/273 [==============================] - 1s 5ms/step - loss: 0.2028 - accuracy: 0.9227 - val_loss: 0.2103 - val_accuracy: 0.9284
Epoch 4/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1977 - accuracy: 0.9248 - val_loss: 0.2106 - val_accuracy: 0.9288
Epoch 5/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1939 - accuracy: 0.9266 - val_loss: 0.2199 - val_accuracy: 0.9197
Epoch 6/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1896 - accuracy: 0.9285 - val_loss: 0.2118 - val_accuracy: 0.9261
Epoch 7/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1877 - accuracy: 0.9295 - val_loss: 0.2206 - val_accuracy: 0.9252
Epoch 8/50
273/273 [==============================] - 1s 3ms/step - loss: 0.1883 - accuracy: 0.9289 - val_loss: 0.2079 - val_accuracy: 0.9270
Epoch 9/50
273/273 [==============================] - 1s 4ms/step - loss: 0.1844 - accuracy: 0.9295 - val_loss: 0.2075 - val_accuracy: 0.9307
Epoch 10/50
273/273 [==============================] - 1s 3ms/step - loss: 0.1806 - accuracy: 0.9320 - val_loss: 0.2103 - val_accuracy: 0.9284
Epoch 11/50
273/273 [==============================] - 1s 3ms/step - loss: 0.1847 - accuracy: 0.9276 - val_loss: 0.2206 - val_accuracy: 0.9256
Epoch 12/50
273/273 [==============================] - 1s 3ms/step - loss: 0.1829 - accuracy: 0.9296 - val_loss: 0.2090 - val_accuracy: 0.9265
Epoch 13/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1798 - accuracy: 0.9309 - val_loss: 0.2193 - val_accuracy: 0.9238
Epoch 14/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1793 - accuracy: 0.9320 - val_loss: 0.2075 - val_accuracy: 0.9302
Epoch 15/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1790 - accuracy: 0.9308 - val_loss: 0.2113 - val_accuracy: 0.9298
Epoch 16/50
273/273 [==============================] - 1s 3ms/step - loss: 0.1767 - accuracy: 0.9324 - val_loss: 0.2073 - val_accuracy: 0.9298
Epoch 17/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1750 - accuracy: 0.9318 - val_loss: 0.2065 - val_accuracy: 0.9343
Epoch 18/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1763 - accuracy: 0.9334 - val_loss: 0.2026 - val_accuracy: 0.9343
Epoch 19/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1754 - accuracy: 0.9330 - val_loss: 0.2011 - val_accuracy: 0.9311
Epoch 20/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1726 - accuracy: 0.9333 - val_loss: 0.2082 - val_accuracy: 0.9311
```

```
Epoch 21/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1721 - accuracy: 0.9351 - val_loss: 0.2141 - val_accuracy: 0.9279
Epoch 22/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1706 - accuracy: 0.9358 - val_loss: 0.2103 - val_accuracy: 0.9316
Epoch 23/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1716 - accuracy: 0.9342 - val_loss: 0.2066 - val_accuracy: 0.9302
Epoch 24/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1711 - accuracy: 0.9340 - val_loss: 0.1998 - val_accuracy: 0.9316
Epoch 25/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1691 - accuracy: 0.9347 - val_loss: 0.2046 - val_accuracy: 0.9316
Epoch 26/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1677 - accuracy: 0.9361 - val_loss: 0.2082 - val_accuracy: 0.9307
Epoch 27/50
273/273 [==============================] - 1s 2ms/step - loss: 0.1687 - accuracy: 0.9351 - val_loss: 0.2056 - val_accuracy: 0.9343
Epoch 28/50
273/273 [==============================] - 1s 5ms/step - loss: 0.1684 - accuracy: 0.9367 - val_loss: 0.2071 - val_accuracy: 0.9247
Epoch 29/50
273/273 [==============================] - 1s 5ms/step - loss: 0.1688 - accuracy: 0.9343 - val_loss: 0.2114 - val_accuracy: 0.9288
```

Colab paid products - Cancel contracts here

✓ 48s completed at 2:50 PM