```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten,Dropout

INPUT_FILE = "./train.csv"
RNG_SEED = 31415926
LEARNING_RATE = .001
BATCH_SIZE = 128
EPOCHS = 50


def load_data(input: str):
    # gets data without the nan labels
    csv_data = np.genfromtxt(input, delimiter=",")[1::]
    labels = csv_data.T[0]  # seperates the labels
    unstructured_data = csv_data.T[1::].T
    data = unstructured_data.reshape(len(unstructured_data), 28, 28)
    return labels, data


def align_data(data: np.ndarray):
    return data.reshape(data.shape[0],28,28,1)

def split(data: np.ndarray, label: np.array, rng):
    indexes = rng.permutation(len(label))
    r_data = data[indexes]
    r_label = label[indexes]
    split = int(.8 * len(r_label))
    train_data = r_data[0:split]
    train_label = r_label[0:split]
    validiation_split = int(.8 * len(train_label))

    return align_data(train_data[0:validiation_split]), tf.keras.utils.to_catego
rical(train_label[0:validiation_split]), \
        align_data(train_data[validiation_split::]), tf.keras.utils.to_categoric
al(train_label[validiation_split::]), \
        align_data(r_data[split::]), tf.keras.utils.to_categorical(r_label[split
::])


def plotImg(img: np.ndarray, label: str):
    plt.imshow(img, cmap='gray')
    plt.title(label)
    plt.show()


def main():
    seed_sequence = np.random.SeedSequence(RNG_SEED)
    [np_seed] = seed_sequence.spawn(1)
    np_rng = np.random.default_rng(np_seed)

    labels, data = load_data(INPUT_FILE)
    x_train, y_train, x_validate, y_validate, x_test, y_test =  \
        split(data, labels, np_rng)

    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
        MaxPooling2D((2, 2)),
        Dropout(.2),
        Conv2D(64, (3, 3),  activation='relu'),
```

```
        MaxPooling2D((2, 2)),
        Conv2D(64, (3, 3), activation='relu'),
        Flatten(),
        Dense(64, activation='relu'),
        Dense(10, activation='softmax', kernel_regularizer='l2')
    ])

    model.summary()
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accura
)
    history = model.fit(x_train, y_train, epochs=EPOCHS, batch_size=BATCH_SI
validation_data=(x_validate, y_validate))

    _, acc = model.evaluate(x_test, y_test)
    print("acc: " + str(acc))


if __name__ == "__main__":
    main()
```

```
Metal device set to: Apple M1
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0         
 )                                                               
                                                                 
 dropout (Dropout)           (None, 13, 13, 32)        0         
                                                                 
 conv2d_1 (Conv2D)           (None, 11, 11, 64)        18496     
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 5, 5, 64)         0         
 2D)                                                             
                                                                 
 conv2d_2 (Conv2D)           (None, 3, 3, 64)          36928     
                                                                 
 flatten (Flatten)           (None, 576)               0         
                                                                 
 dense (Dense)               (None, 64)                36928     
                                                                 
 dense_1 (Dense)             (None, 10)                650       
                                                                 
=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0


Training Information removed for viewing purposes

acc: 0.9864285588264465

systemMemory: 16.00 GB
maxCacheSize: 5.33 GB
```