In [1]:

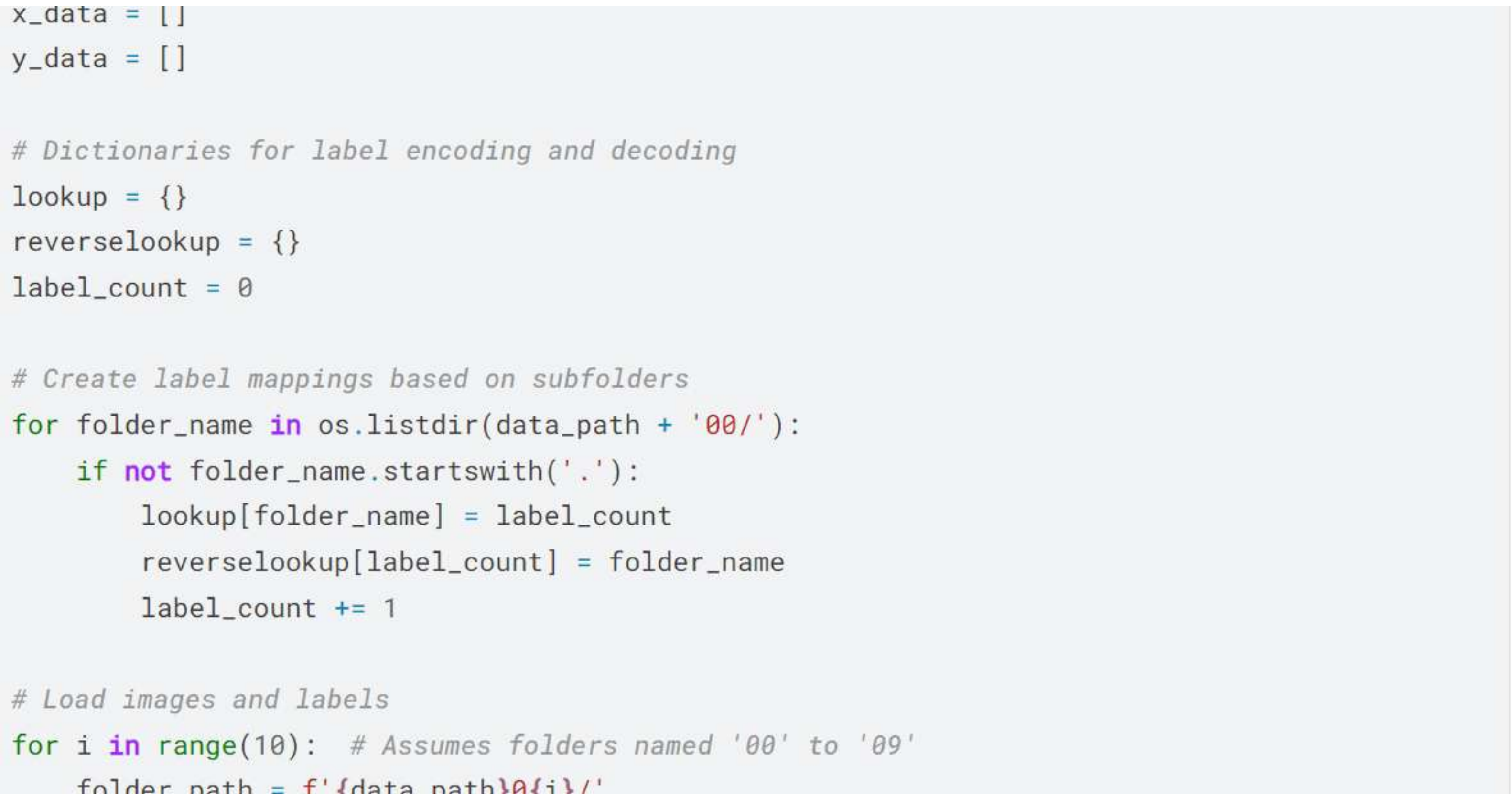
```
# Import necessary libraries
import numpy as np
import pandas as pd
import os
import cv2

import matplotlib.pyplot as plt
import seaborn as sns
import random

from scipy.stats import norm

import keras
from keras import backend as K
K.clear_session()
```

In [2]:

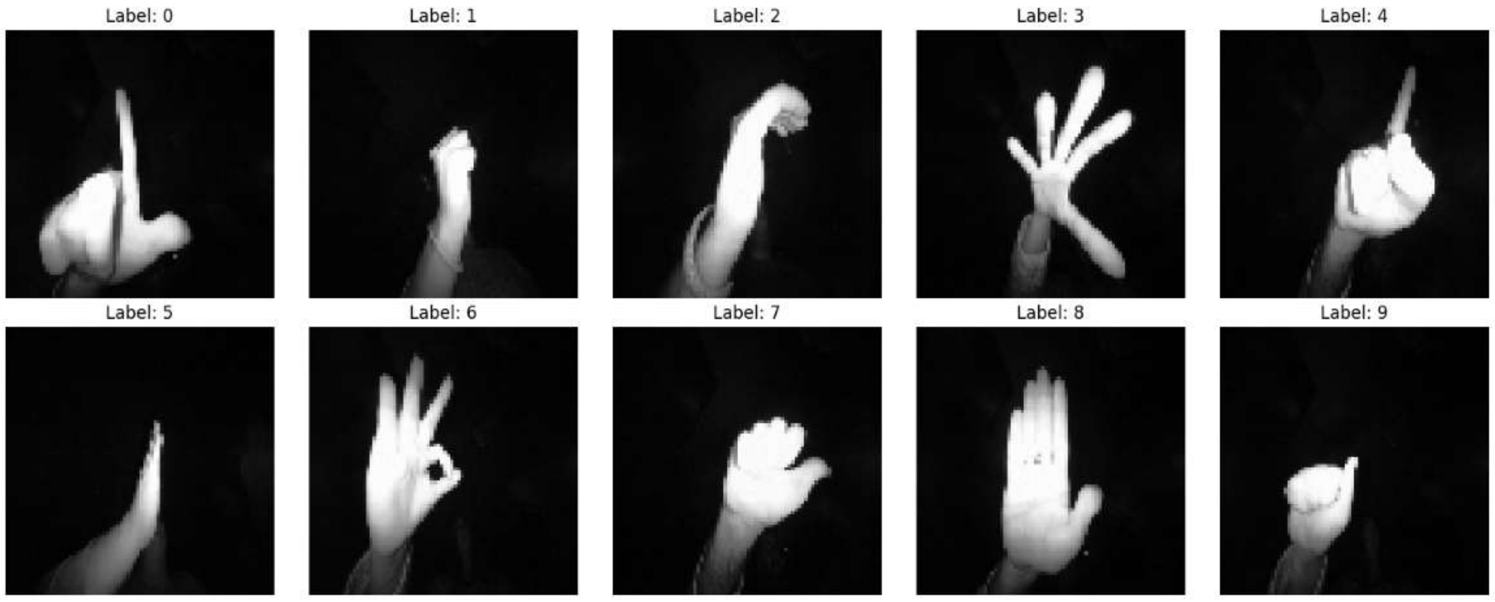
```
image_shape = (80, 80, 1)  # Adjusted to match your hand gesture images' size and grayscale
batch_size = 64
latent_dim = 16
epochs = 5
```

In [3]:

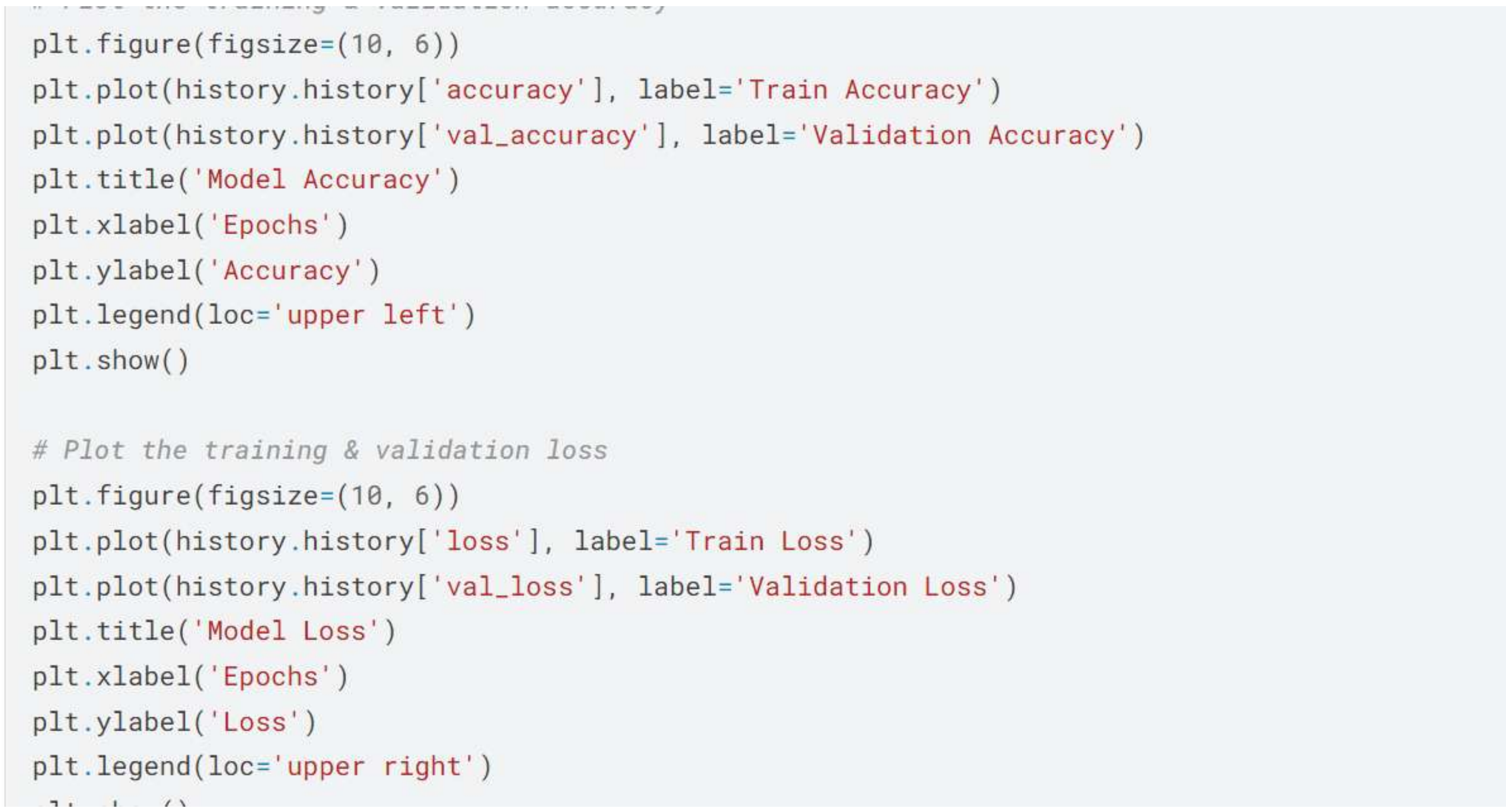
```
import os
import cv2
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle  # Import the shuffle function
from tensorflow.keras.utils import to_categorical

# Parameters
IMG_SIZE = 80  # Image size to resize to
data_path = '../input/leapgestrecog/leapGestRecog/'  # Path to dataset folder
```



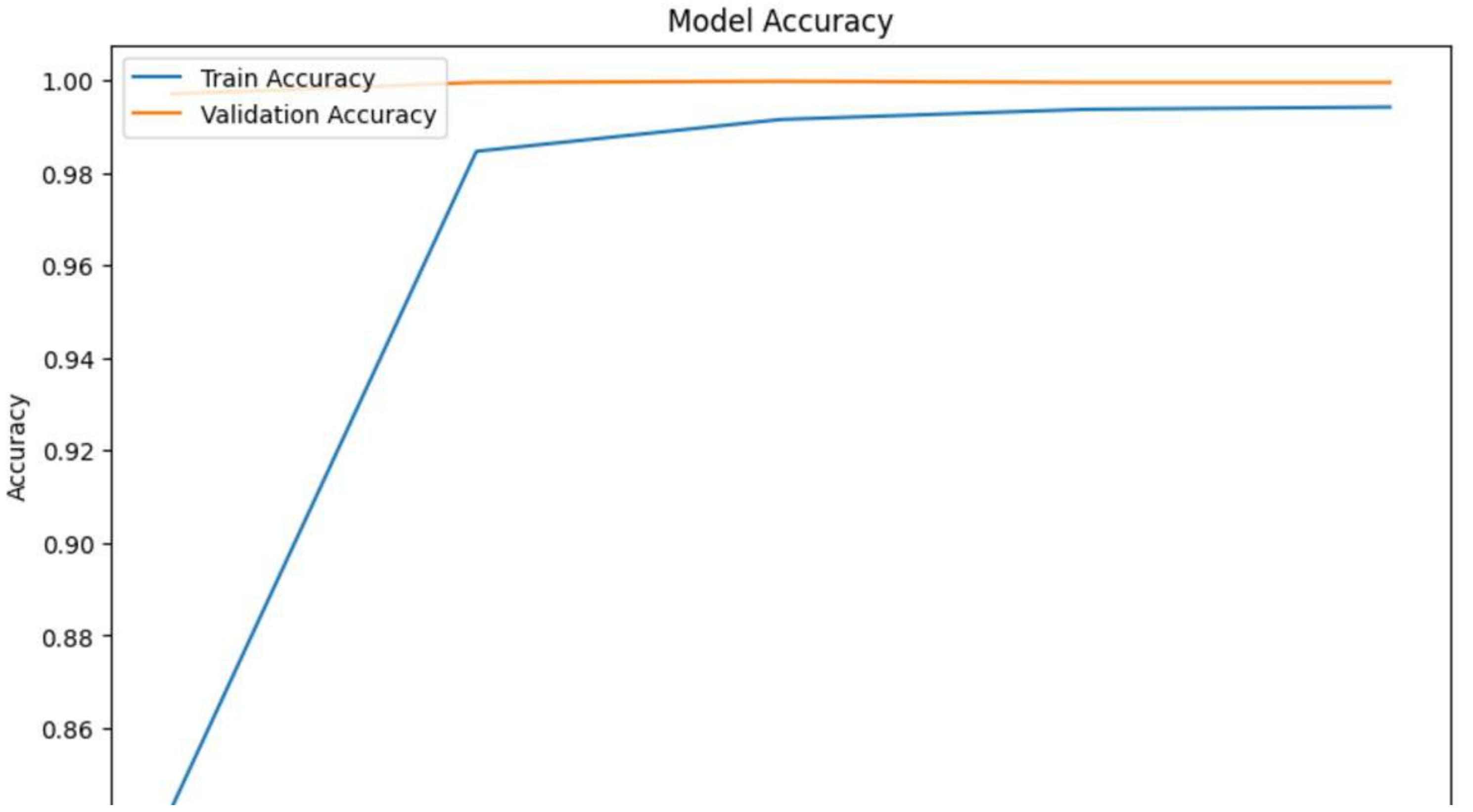
```
x_data = []
y_data = []

# Dictionaries for label encoding and decoding
lookup = {}
reverselookup = {}
label_count = 0

# Create label mappings based on subfolders
for folder_name in os.listdir(data_path + '00/'):
    if not folder_name.startswith('.'):
        lookup[folder_name] = label_count
        reverselookup[label_count] = folder_name
        label_count += 1

# Load images and labels
for i in range(10):  # Assumes folders named '00' to '09'
    folder_path = f'{data_path}0{i}/'
```


Label: 0
Label: 1
Label: 2
Label: 3
Label: 4
Label: 5
Label: 6
Label: 7
Label: 8
Label: 9

```python
x_train, x_val, y_train, y_val = train_test_split(x_data, y_data, test_size=0.2, random_sta
te=42)

# Display dataset info
print("Dataset loaded.")
print(f"Total images: {len(x_data)}")
print(f"Image shape: {x_data[0].shape}")
print(f"Number of classes: {len(lookup)}")

# Optional: Check label mappings
print("Label mappings:", lookup)

# Optional: Check the shapes of the train/validation split
print(f"Training set shape: {x_train.shape}, {y_train.shape}")
print(f"Validation set shape: {x_val.shape}, {y_val.shape}")
```



```
plt.figure(figsize=(10, 6))
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(loc='upper left')
plt.show()

# Plot the training & validation loss
plt.figure(figsize=(10, 6))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend(loc='upper right')
```


Model Accuracy
Accuracy
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
Train Accuracy
Validation Accuracy