```
from sklearn.datasets import fetch_lfw_people
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split
```

```
lfw = fetch_lfw_people (min_faces_per_person=100)
```

```
n_samples, h, w = lfw.images.shape
print("Number of sample faces and its height and width: ", n_samples, h, w)
```

```
Number of sample faces and its height and width:  1140 62 47
```

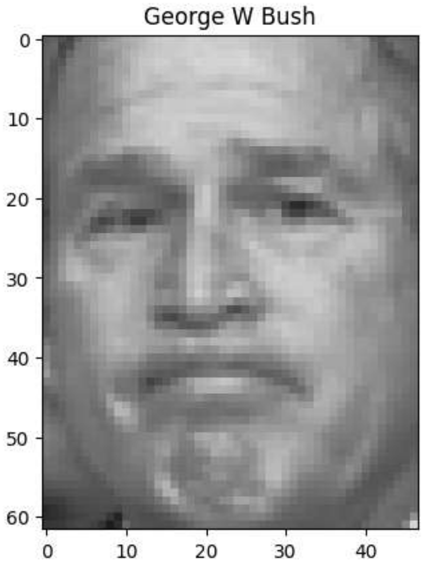
```
X = lfw.data
y = lfw.target
target_names = lfw.target_names
print("input data shape: ", X.shape)
print("Target length: ", len(y))
print("target_names: ", target_names)
```

```
input data shape:  (1140, 2914)
Target length:  1140
target_names:  ['Colin Powell' 'Donald Rumsfeld' 'George W Bush' 'Gerhard Schroeder'
 'Tony Blair']
```

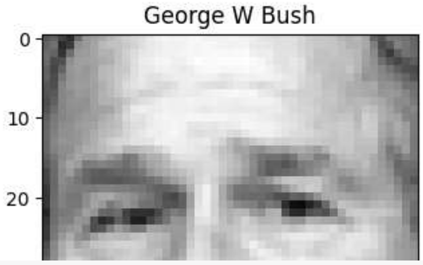
```
X[0]
```

```
array([0.32026145, 0.34771243, 0.26013073, ..., 0.4       , 0.5542484 ,
       0.82483655], dtype=float32)
```

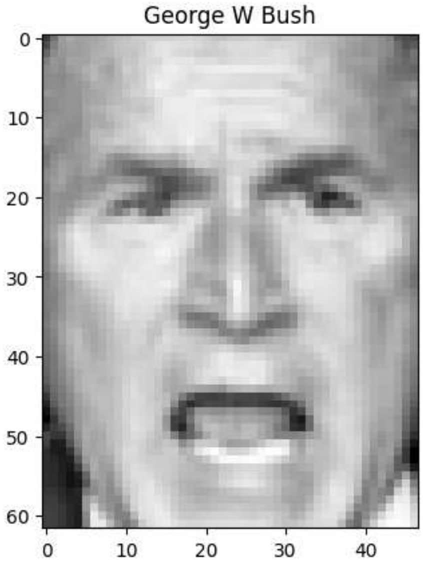
```
plt.imshow(lfw.images[0])
plt.title(target_names [y[0]])
plt.show()
```



```
plt.imshow(lfw.images[0],cmap='gray')
plt.title(target_names [y[0]])
plt.show()
```

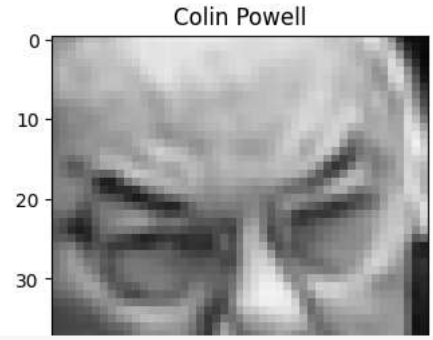


```
plt.imshow(lfw.images[100],cmap='gray')
plt.title(target_names [y[100]])
plt.show()
```



```
plt.imshow(lfw.images[101],cmap='gray')
plt.title(target_names [y[101]])
plt.show()
```

```
plt.imshow(lfw.images[105],cmap='gray')
plt.title(target_names [y[105]])
plt.show()
```



```
target_names.shape [0]
```

```
5
```

```
X.shape[0]
```

```
1140
```

```
X.shape[1]
```

```
2914
```

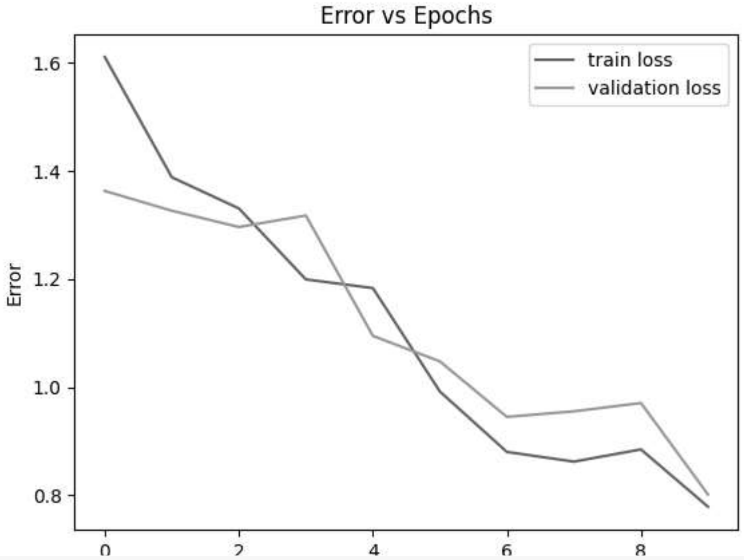
```
model = Sequential()
model.add(Dense(256, input_dim = X.shape[1], activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(target_names.shape[0], activation = 'softmax'))
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 dense_1 (Dense)             (None, 256)               746240    
                                                                 
 dense_2 (Dense)             (None, 128)               32896     
                                                                 
 dense_3 (Dense)             (None, 5)                 645       
                                                                 
=================================================================
Total params: 779781 (2.97 MB)
Trainable params: 779781 (2.97 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics=['accuracy'])
history = model.fit (X, y, batch_size = 32, epochs = 10, validation_split = 0.2)
```

```
Epoch 1/10
29/29 [==============================] - 2s 23ms/step - loss: 1.6102 - accuracy: 0.4013 - val_loss: 1.3626 - val_accuracy: 0.5000
Epoch 2/10
29/29 [==============================] - 1s 20ms/step - loss: 1.3879 - accuracy: 0.4572 - val_loss: 1.3261 - val_accuracy: 0.5307
Epoch 3/10
29/29 [==============================] - 1s 19ms/step - loss: 1.3304 - accuracy: 0.5186 - val_loss: 1.2960 - val_accuracy: 0.5132
Epoch 4/10
29/29 [==============================] - 1s 20ms/step - loss: 1.1991 - accuracy: 0.5691 - val_loss: 1.3172 - val_accuracy: 0.5263
Epoch 5/10
29/29 [==============================] - 1s 19ms/step - loss: 1.1831 - accuracy: 0.5570 - val_loss: 1.0948 - val_accuracy: 0.5395
Epoch 6/10
29/29 [==============================] - 1s 20ms/step - loss: 0.9919 - accuracy: 0.6327 - val_loss: 1.0476 - val_accuracy: 0.6711
Epoch 7/10
29/29 [==============================] - 0s 15ms/step - loss: 0.8803 - accuracy: 0.6820 - val_loss: 0.9450 - val_accuracy: 0.6886
Epoch 8/10
29/29 [==============================] - 0s 14ms/step - loss: 0.8623 - accuracy: 0.7018 - val_loss: 0.9553 - val_accuracy: 0.6447
Epoch 9/10
29/29 [==============================] - 0s 13ms/step - loss: 0.8849 - accuracy: 0.6842 - val_loss: 0.9707 - val_accuracy: 0.6316
Epoch 10/10
29/29 [==============================] - 0s 13ms/step - loss: 0.7790 - accuracy: 0.7094 - val_loss: 0.8016 - val_accuracy: 0.6886
```

```
plt.plot(history.history['loss'], label = 'train loss')
plt.plot(history.history['val_loss'], label = 'validation loss')
plt.title("Error vs Epochs")
plt.xlabel("Epochs")
plt.ylabel("Error")
plt.legend()
plt.show()
```



```
model.compile(tf.keras.optimizers.Adadelta(learning_rate=0.0001, rho=0.9), loss ='sparse_categorical_crossentropy', metrics=['accuracy'])
history = model.fit(X, y, batch_size = 64, epochs = 25, validation_split = 0.2)
```

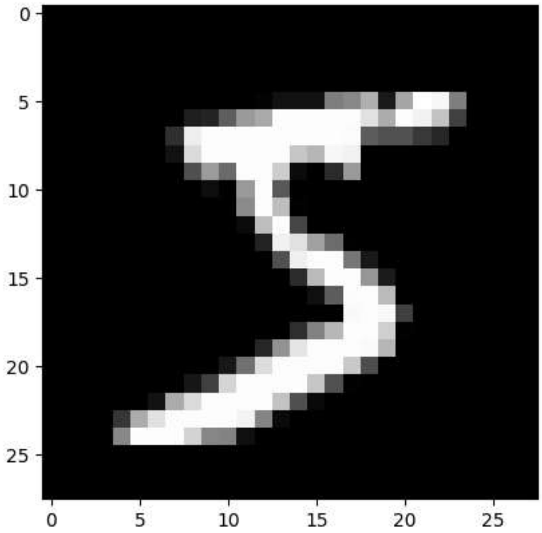
```
Epoch 1/25
15/15 [==============================] - 1s 34ms/step - loss: 0.6530 - accuracy: 0.7336 - val_loss: 0.8005 - val_accuracy: 0.6930
Epoch 2/25
15/15 [==============================] - 0s 19ms/step - loss: 0.6521 - accuracy: 0.7325 - val_loss: 0.7995 - val_accuracy: 0.6930
Epoch 3/25
15/15 [==============================] - 0s 19ms/step - loss: 0.6513 - accuracy: 0.7325 - val_loss: 0.7985 - val_accuracy: 0.6930
Epoch 4/25
15/15 [==============================] - 0s 20ms/step - loss: 0.6504 - accuracy: 0.7325 - val_loss: 0.7974 - val_accuracy: 0.6930
Epoch 5/25
15/15 [==============================] - 0s 19ms/step - loss: 0.6495 - accuracy: 0.7336 - val_loss: 0.7963 - val_accuracy: 0.6930
Epoch 6/25
15/15 [==============================] - 0s 18ms/step - loss: 0.6485 - accuracy: 0.7336 - val_loss: 0.7952 - val_accuracy: 0.6930
Epoch 7/25
15/15 [==============================] - 0s 19ms/step - loss: 0.6476 - accuracy: 0.7346 - val_loss: 0.7941 - val_accuracy: 0.6930
Epoch 8/25
15/15 [==============================] - 0s 20ms/step - loss: 0.6466 - accuracy: 0.7357 - val_loss: 0.7931 - val_accuracy: 0.6930
Epoch 9/25
15/15 [==============================] - 0s 19ms/step - loss: 0.6458 - accuracy: 0.7357 - val_loss: 0.7920 - val_accuracy: 0.6930
Epoch 10/25
15/15 [==============================] - 0s 19ms/step - loss: 0.6448 - accuracy: 0.7357 - val_loss: 0.7910 - val_accuracy: 0.6930
Epoch 11/25
15/15 [==============================] - 0s 19ms/step - loss: 0.6439 - accuracy: 0.7379 - val_loss: 0.7899 - val_accuracy: 0.6930
Epoch 12/25
15/15 [==============================] - 0s 18ms/step - loss: 0.6430 - accuracy: 0.7401 - val_loss: 0.7888 - val_accuracy: 0.6930
Epoch 13/25
15/15 [==============================] - 0s 18ms/step - loss: 0.6421 - accuracy: 0.7434 - val_loss: 0.7879 - val_accuracy: 0.6930
Epoch 14/25
15/15 [==============================] - 0s 18ms/step - loss: 0.6413 - accuracy: 0.7434 - val_loss: 0.7868 - val_accuracy: 0.6930
Epoch 15/25
15/15 [==============================] - 0s 28ms/step - loss: 0.6404 - accuracy: 0.7467 - val_loss: 0.7858 - val_accuracy: 0.6930
Epoch 16/25
15/15 [==============================] - 0s 30ms/step - loss: 0.6395 - accuracy: 0.7489 - val_loss: 0.7847 - val_accuracy: 0.6930
Epoch 17/25
15/15 [==============================] - 0s 30ms/step - loss: 0.6386 - accuracy: 0.7511 - val_loss: 0.7836 - val_accuracy: 0.6930
Epoch 18/25
15/15 [==============================] - 0s 29ms/step - loss: 0.6377 - accuracy: 0.7511 - val_loss: 0.7826 - val_accuracy: 0.6930
Epoch 19/25
15/15 [==============================] - 0s 28ms/step - loss: 0.6368 - accuracy: 0.7555 - val_loss: 0.7816 - val_accuracy: 0.6930
Epoch 20/25
15/15 [==============================] - 0s 30ms/step - loss: 0.6360 - accuracy: 0.7555 - val_loss: 0.7806 - val_accuracy: 0.6930
Epoch 21/25
15/15 [==============================] - 0s 30ms/step - loss: 0.6351 - accuracy: 0.7566 - val_loss: 0.7796 - val_accuracy: 0.6930
Epoch 22/25
15/15 [==============================] - 0s 20ms/step - loss: 0.6343 - accuracy: 0.7566 - val_loss: 0.7786 - val_accuracy: 0.6930
Epoch 23/25
15/15 [==============================] - 0s 17ms/step - loss: 0.6334 - accuracy: 0.7566 - val_loss: 0.7776 - val_accuracy: 0.6930
Epoch 24/25
15/15 [==============================] - 0s 18ms/step - loss: 0.6326 - accuracy: 0.7566 - val_loss: 0.7766 - val_accuracy: 0.6974
Epoch 25/25
15/15 [==============================] - 0s 20ms/step - loss: 0.6318 - accuracy: 0.7566 - val_loss: 0.7757 - val_accuracy: 0.6974
```

```
import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
```

```
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 1s 0us/step
```

```
plt.imshow(x_train[0],cmap='gray')
plt.show()
```


0
5
10
15
20
25
0
5
10
15
20
25