

In [4]: `pip install opencv-python`

Requirement already satisfied: opencv-python in c:\users\akankasha\anaconda3\lib\site-packages (4.9.0.80)
Requirement already satisfied: numpy>=1.21.2 in c:\users\akankasha\anaconda3\lib\site-packages (from opencv-python) (1.24.3)
Note: you may need to restart the kernel to use updated packages.

In [6]: `import tensorflow as tf
from tensorflow.keras.models import Sequential`

```
# Example usage  
model = Sequential()  
model.compile(optimizer='adam',  
              loss=tf.compat.v1.losses.sparse_softmax_cross_entropy,  
              metrics=['accuracy'])
```

WARNING:tensorflow:From C:\Users\Akankasha\anaconda3\Lib\site-packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\Akankasha\anaconda3\Lib\site-packages\keras\src\optimizers_init_.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

In [5]: `import matplotlib.pyplot as plt
import cv2
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Flatten, Conv2D, MaxPool2D, Dropout
from keras.optimizers import SGD, Adam
from keras.callbacks import ReduceLROnPlateau, EarlyStopping
from keras.utils import to_categorical
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle`

In [8]: `data = pd.read_csv(r'C:\Users\Akankasha\OneDrive\Desktop\ds\A_Z Handwritten Data.csv')
print(data.head(10))`

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	...	0.639	0.640	0.641	\
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	

	0.642	0.643	0.644	0.645	0.646	0.647	0.648
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[10 rows x 785 columns]

```
In [9]: X = data.drop('0',axis = 1)
        y = data['0']
```

```
In [10]: train_x, test_x, train_y, test_y = train_test_split(X, y, test_size = 0.2)

train_x = np.reshape(train_x.values, (train_x.shape[0], 28,28))
test_x = np.reshape(test_x.values, (test_x.shape[0], 28,28))

print("Train data shape: ", train_x.shape)
print("Test data shape: ", test_x.shape)
```

Train data shape: (297960, 28, 28)
Test data shape: (74490, 28, 28)

```
In [11]: word_dict = {0:'A',1:'B',2:'C',3:'D',4:'E',5:'F',6:'G',7:'H',8:'I',9:'J',10:'K',11:'L',12:'M',13:'N',14:'O',15:'P',16:'Q',17:'R',18:'S',19:'T',20:'U',21:'V',22:'W',23:'X',24:'Y',25:'Z'}
```

```
In [12]: y_int = np.int0(y)
count = np.zeros(26, dtype='int')
for i in y_int:
    count[i] +=1

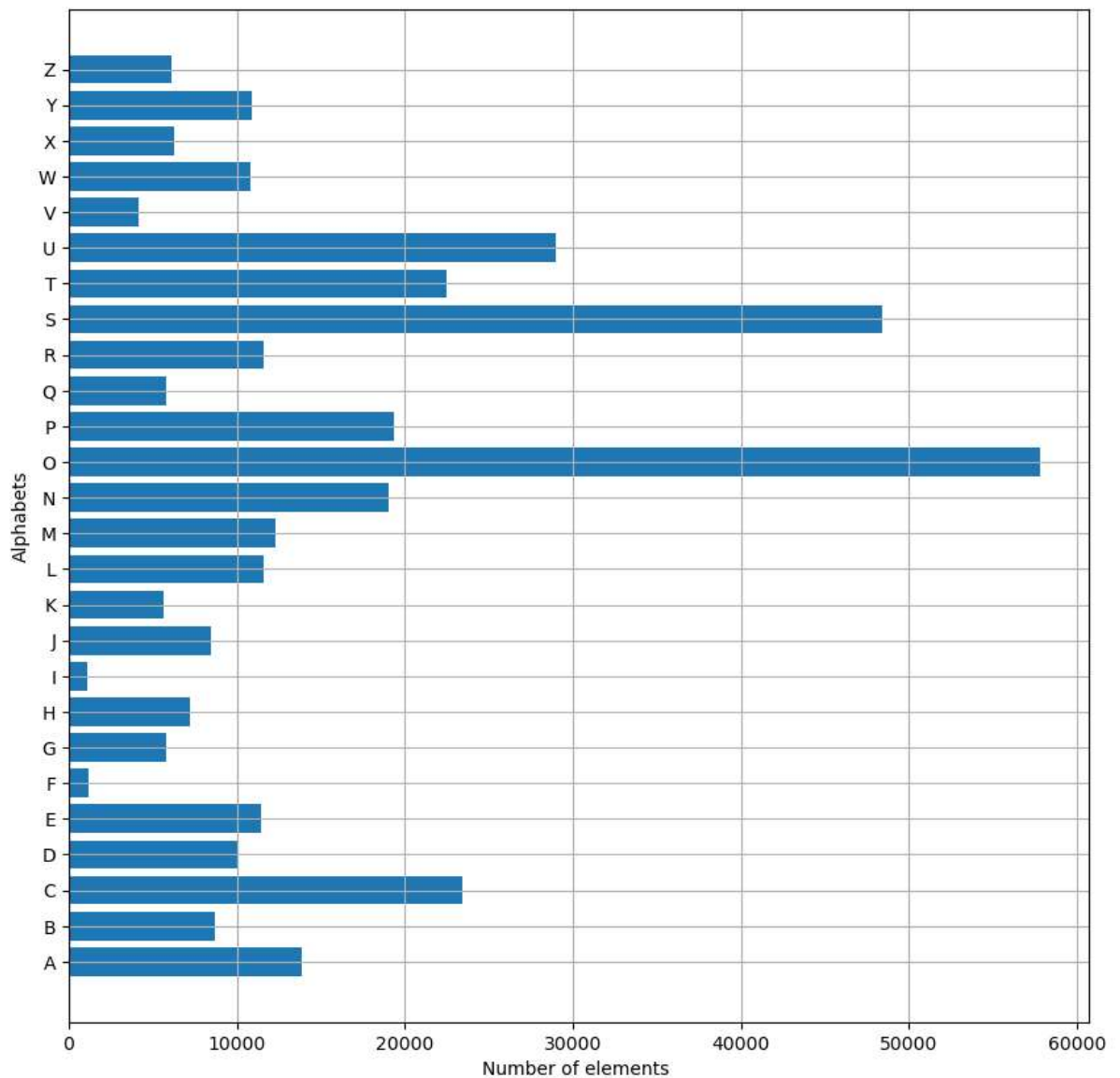
alphabets = []
for i in word_dict.values():
    alphabets.append(i)

fig, ax = plt.subplots(1,1, figsize=(10,10))
ax.barh(alphabets, count)

plt.xlabel("Number of elements ")
plt.ylabel("Alphabets")
plt.grid()
plt.show()
```

C:\Users\Akankasha\AppData\Local\Temp\ipykernel_17300\2258927801.py:1: Deprecation Warning: `np.int0` is a deprecated alias for `np.intp`. (Deprecated NumPy 1.24)

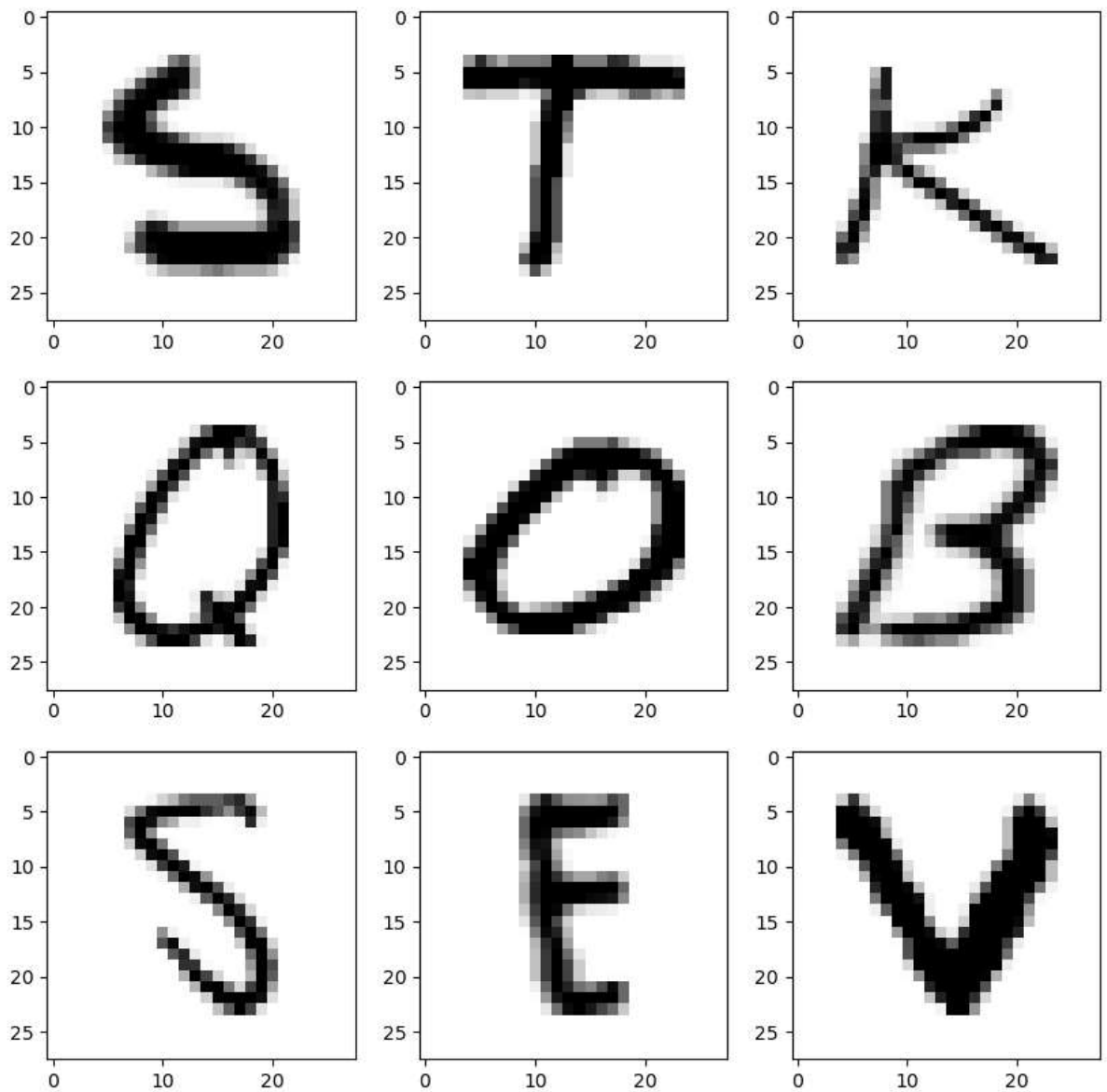
y_int = np.int0(y)



```
In [13]: shuff = shuffle(train_x[:100])

fig, ax = plt.subplots(3,3, figsize = (10,10))
axes = ax.flatten()

for i in range(9):
    _, shu = cv2.threshold(shuff[i], 30, 200, cv2.THRESH_BINARY)
    axes[i].imshow(np.reshape(shuff[i], (28,28)), cmap="Greys")
plt.show()
```



In [14]: *# Now we reshape the train & test image dataset so that they can be put in the model*

```
train_X = train_x.reshape(train_x.shape[0], train_x.shape[1], train_x.shape[2], 1)
print("New shape of train data: ", train_X.shape)
```

```
test_X = test_x.reshape(test_x.shape[0], test_x.shape[1], test_x.shape[2], 1)
print("New shape of test data: ", test_X.shape)
```

New shape of train data: (297960, 28, 28, 1)

New shape of test data: (74490, 28, 28, 1)

In [15]: `train_yOHE = to_categorical(train_y, num_classes = 26, dtype='int')`
`print("New shape of train labels: ", train_yOHE.shape)`

```
test_yOHE = to_categorical(test_y, num_classes = 26, dtype='int')
print("New shape of test labels: ", test_yOHE.shape)
```

New shape of train labels: (297960, 26)

New shape of test labels: (74490, 26)

In [16]: `model = Sequential()`

```
model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(28, 28, 1)))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
```

```
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = 'same'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
```

```

model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding = 'val
model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Flatten())

model.add(Dense(64,activation = "relu"))
model.add(Dense(128,activation = "relu"))

model.add(Dense(26,activation = "softmax"))

```

WARNING:tensorflow:From C:\Users\Akankasha\anaconda3\Lib\site-packages\keras\src\layers\pooling\max_pooling2d.py:161: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

In [18]: `model.compile(optimizer = Adam(learning_rate=0.001), loss='categorical_crossentropy', history = model.fit(train_X, train_yOHE, epochs=1, validation_data = (test_X, test_`

WARNING:tensorflow:From C:\Users\Akankasha\anaconda3\Lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\Akankasha\anaconda3\Lib\site-packages\keras\src\engine\base_layer_utils.py:384: The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

9312/9312 [=====] - 267s 28ms/step - loss: 0.1674 - accuracy: 0.9554 - val_loss: 0.0891 - val_accuracy: 0.9751

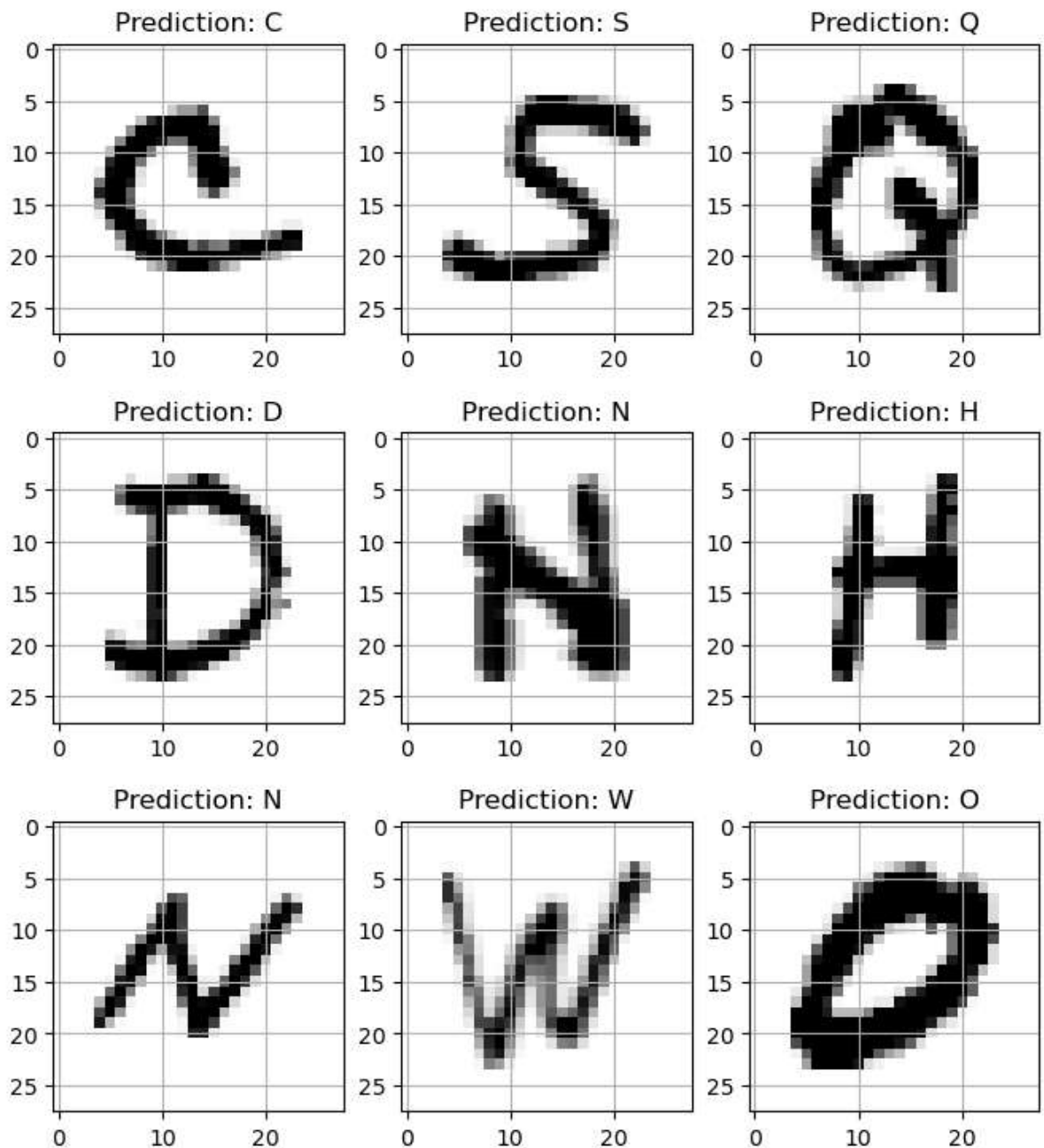
In [19]: `print("The validation accuracy is :", history.history['val_accuracy'])
print("The training accuracy is :", history.history['accuracy'])
print("The validation loss is :", history.history['val_loss'])
print("The training loss is :", history.history['loss'])`

The validation accuracy is : [0.9750704765319824]
The training accuracy is : [0.9554403424263]
The validation loss is : [0.08914036303758621]
The training loss is : [0.1674167662858963]

In [20]: `fig, axes = plt.subplots(3,3, figsize=(8,9))
axes = axes.flatten()

for i,ax in enumerate(axes):
 img = np.reshape(test_X[i], (28,28))
 ax.imshow(img, cmap="Greys")

 pred = word_dict[np.argmax(test_yOHE[i])]
 ax.set_title("Prediction: "+pred)
 ax.grid()`



```
In [58]: img = cv2.imread(r'C:\Users\Akankasha\OneDrive\Desktop\ds\TEST1\1.jpg')
img_copy = img.copy()
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (400,440))
```

```
In [59]: img_copy = cv2.GaussianBlur(img_copy, (7,7), 0)
img_gray = cv2.cvtColor(img_copy, cv2.COLOR_BGR2GRAY)
_, img_thresh = cv2.threshold(img_gray, 100, 255, cv2.THRESH_BINARY_INV)

img_final = cv2.resize(img_thresh, (28,28))
img_final = np.reshape(img_final, (1,28,28,1))
```

```
In [60]: img_pred = word_dict[np.argmax(model.predict(img_final))]

cv2.putText(img, "Dataflair _ _ _", (20,25), cv2.FONT_HERSHEY_TRIPLEX, 0.7, color
cv2.putText(img, "Prediction: " + img_pred, (20,410), cv2.FONT_HERSHEY_DUPLEX, 1.3,
cv2.imshow('Dataflair handwritten character recognition _ _ _', img)
```

1/1 [=====] - 0s 53ms/step

```
In [ ]: while (1):
k = cv2.waitKey(1) & 0xFF
```

```
    if k == 27:  
        break  
cv2.destroyAllWindows()
```

In []: