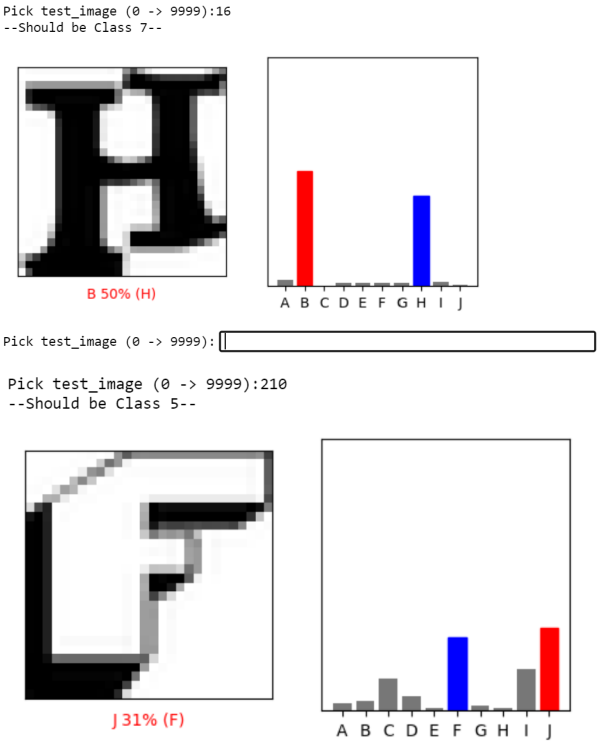
Partial Model:

The model is Sequential. I used 3 layers. The first one is the input layer. The second is the hidden layer. This layer consists of 32 neurons. The last layer is the output, and the number of neurons is 10. For the input and hidden layer, we use the TANH as an activation function, and the last layer uses the SOFTMAX. The used optimizer is ADAM, and for loss function, my choice was sparse_categorical_crossentropy. The metric is accuracy. The number of epochs is 2. The batch_size value is 40.

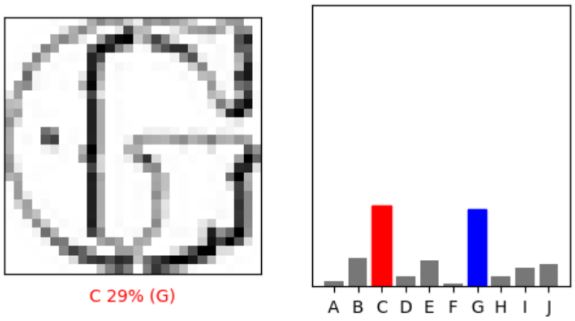
```
print("--Make model--")
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(32, activation='tanh'),
  tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

print("--Fit model--")
model.fit(x_train, y_train, epochs=2 , batch_size = 40, verbose=2)
```

Three Images (False Prediction):



```
Pick test_image (0 -> 9999):252
--Should be Class 6--
```



Train Accuracy is 84.9% and the Test accuracy is 90.8%.

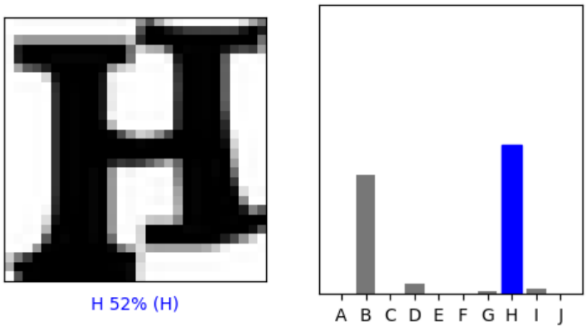
Complete Model:

The model is Sequential. I used 4 layers this time instead of 3 layers. The first one is the input layer. The second and third are the hidden layers. The number of neurons was increased, and the hidden layers consist of 128 neurons each. The last layer is the output, and the number of neurons is 10. For the hidden layers, I used the RELU as an activation function instead of TANH, and the last layer stays the same using the SOFTMAX and the same case for the optimizer is ADAM and for loss function. The metric is accuracy. The number of epochs this time is 12 before it was 2. The batch_size value is 40. I have added a dropout layer to enhance the model. I added a dropout layer before the output with 0.30 probability. The dropout layer will ignore some neurons in a random number of time for each epoch.

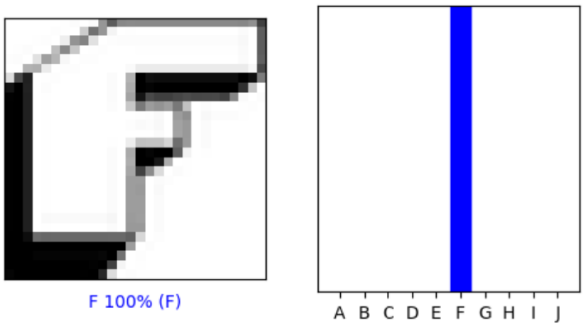
```
print("--Make model--")
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Flatten(input_shape=(28, 28)))
model.add(tf.keras.layers.Dense(128, activation='relu'))
model.add(tf.keras.layers.Dense(128, activation='relu'))
model.add(tf.keras.layers.Dropout(0.30))
model.add(tf.keras.layers.Dense(10, activation='softmax'))
```

Three Images (True Prediction):

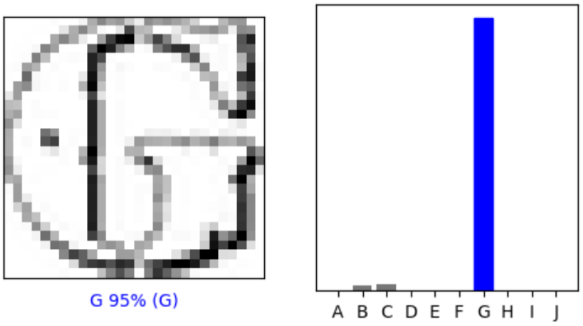
```
Pick test_image (0 -> 9999):16
--Should be Class 7--
```



```
Pick test_image (0 -> 9999):210
--Should be Class 5--
```




```
Pick test_image (0 -> 9999):252
--Should be Class 6--
```



For the new model, the Train Accuracy is 92.8% (it was 84.9%) and the test Accuracy 94.1% (it was 90.8%).