In [2]:
```
# import the necessary packages
from sklearn.preprocessing import LabelBinarizer
from sklearn.metrics import classification_report
```

In [3]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.datasets import mnist
from tensorflow.keras import backend as K
```

In [4]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [5]:
```
import argparse
```

In [7]:
```
# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-o", "--output", required=True,
	help="path to the output loss/accuracy plot")
```

Out[7]:
```
_StoreAction(option_strings=['-o', '--output'], dest='output', nargs=None, const=Non
e, default=None, type=None, choices=None, help='path to the output loss/accuracy plo
t', metavar=None)
```

In [8]:
```
args = vars(ap.parse_args())
```

```
usage: ipykernel_launcher.py [-h] -o OUTPUT
ipykernel_launcher.py: error: the following arguments are required: -o/--output
An exception has occurred, use %tb to see the full traceback.

SystemExit: 2
```

In [10]:
```
# grab the MNIST dataset (if this is your first time using this
# dataset then the 11MB download may take a minute)
print("[INFO] accessing MNIST...")
((trainX, trainY), (testX, testY)) = mnist.load_data()
```

```
[INFO] accessing MNIST...
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mni
st.npz
11490434/11490434 [==============================] - 2s 0us/step
```

In [11]:
```
# each image in the MNIST dataset is represented as a 28x28x1
# image, but in order to apply a standard neural network we must
# first "flatten" the image to be simple list of 28x28=784 pixels
trainX = trainX.reshape((trainX.shape[0], 28 * 28 * 1))
testX = testX.reshape((testX.shape[0], 28 * 28 * 1))
```

In [12]:
```
# scale data to the range of [0, 1]
trainX = trainX.astype("float32") / 255.0
testX = testX.astype("float32") / 255.0
```

In [13]:
```
# convert the labels from integers to vectors
lb = LabelBinarizer()
trainY = lb.fit_transform(trainY)
testY = lb.transform(testY)
```

In [14]:
```
# define the 784-256-128-10 architecture using Keras
model = Sequential()
model.add(Dense(256, input_shape=(784,), activation="sigmoid"))
model.add(Dense(128, activation="sigmoid"))
model.add(Dense(10, activation="softmax"))
```

In [15]:
```
# train the model using SGD
print("[INFO] training network...")
sgd = SGD(0.01)
model.compile(loss="categorical_crossentropy", optimizer=sgd,
        metrics=["accuracy"])
H = model.fit(trainX, trainY, validation_data=(testX, testY),
        epochs=100, batch_size=128)
```

```
[INFO] training network...
Epoch 1/100
469/469 [==============================] - 2s 4ms/step - loss: 2.2793 - accuracy: 0.2
024 - val_loss: 2.2444 - val_accuracy: 0.4344
Epoch 2/100
469/469 [==============================] - 2s 5ms/step - loss: 2.2111 - accuracy: 0.3
995 - val_loss: 2.1703 - val_accuracy: 0.3639
Epoch 3/100
469/469 [==============================] - 3s 5ms/step - loss: 2.1254 - accuracy: 0.5
034 - val_loss: 2.0659 - val_accuracy: 0.5420
Epoch 4/100
469/469 [==============================] - 2s 5ms/step - loss: 2.0006 - accuracy: 0.5
695 - val_loss: 1.9129 - val_accuracy: 0.5918
Epoch 5/100
469/469 [==============================] - 2s 5ms/step - loss: 1.8241 - accuracy: 0.6
123 - val_loss: 1.7100 - val_accuracy: 0.6264
Epoch 6/100
469/469 [==============================] - 2s 5ms/step - loss: 1.6111 - accuracy: 0.6
539 - val_loss: 1.4886 - val_accuracy: 0.6770
Epoch 7/100
469/469 [==============================] - 2s 5ms/step - loss: 1.4007 - accuracy: 0.6
913 - val_loss: 1.2910 - val_accuracy: 0.7133
Epoch 8/100
469/469 [==============================] - 2s 5ms/step - loss: 1.2227 - accuracy: 0.7
214 - val_loss: 1.1318 - val_accuracy: 0.7435
Epoch 9/100
469/469 [==============================] - 2s 5ms/step - loss: 1.0823 - accuracy: 0.7
459 - val_loss: 1.0088 - val_accuracy: 0.7574
Epoch 10/100
469/469 [==============================] - 2s 5ms/step - loss: 0.9728 - accuracy: 0.7
640 - val_loss: 0.9119 - val_accuracy: 0.7791
Epoch 11/100
469/469 [==============================] - 2s 5ms/step - loss: 0.8863 - accuracy: 0.7
797 - val_loss: 0.8346 - val_accuracy: 0.7932
Epoch 12/100
469/469 [==============================] - 2s 5ms/step - loss: 0.8165 - accuracy: 0.7
934 - val_loss: 0.7709 - val_accuracy: 0.8058
Epoch 13/100
469/469 [==============================] - 2s 5ms/step - loss: 0.7592 - accuracy: 0.8
057 - val_loss: 0.7185 - val_accuracy: 0.8164
Epoch 14/100
469/469 [==============================] - 2s 5ms/step - loss: 0.7112 - accuracy: 0.8
162 - val_loss: 0.6741 - val_accuracy: 0.8258
Epoch 15/100
469/469 [==============================] - 2s 5ms/step - loss: 0.6705 - accuracy: 0.8
253 - val_loss: 0.6372 - val_accuracy: 0.8317
Epoch 16/100
469/469 [==============================] - 2s 5ms/step - loss: 0.6355 - accuracy: 0.8
335 - val_loss: 0.6043 - val_accuracy: 0.8390
Epoch 17/100
469/469 [==============================] - 2s 5ms/step - loss: 0.6052 - accuracy: 0.8
413 - val_loss: 0.5754 - val_accuracy: 0.8479
Epoch 18/100
469/469 [==============================] - 2s 5ms/step - loss: 0.5789 - accuracy: 0.8
467 - val_loss: 0.5502 - val_accuracy: 0.8537
Epoch 19/100
469/469 [==============================] - 2s 5ms/step - loss: 0.5556 - accuracy: 0.8
531 - val_loss: 0.5281 - val_accuracy: 0.8600
Epoch 20/100
469/469 [==============================] - 2s 5ms/step - loss: 0.5350 - accuracy: 0.8
```

```
584 - val_loss: 0.5083 - val_accuracy: 0.8644
Epoch 21/100
469/469 [==============================] - 2s 5ms/step - loss: 0.5166 - accuracy: 0.8
628 - val_loss: 0.4910 - val_accuracy: 0.8686
Epoch 22/100
469/469 [==============================] - 2s 5ms/step - loss: 0.5001 - accuracy: 0.8
668 - val_loss: 0.4755 - val_accuracy: 0.8726
Epoch 23/100
469/469 [==============================] - 2s 5ms/step - loss: 0.4852 - accuracy: 0.8
704 - val_loss: 0.4618 - val_accuracy: 0.8748
Epoch 24/100
469/469 [==============================] - 3s 5ms/step - loss: 0.4719 - accuracy: 0.8
737 - val_loss: 0.4486 - val_accuracy: 0.8806
Epoch 25/100
469/469 [==============================] - 2s 5ms/step - loss: 0.4597 - accuracy: 0.8
767 - val_loss: 0.4368 - val_accuracy: 0.8828
Epoch 26/100
469/469 [==============================] - 2s 5ms/step - loss: 0.4486 - accuracy: 0.8
792 - val_loss: 0.4266 - val_accuracy: 0.8847
Epoch 27/100
469/469 [==============================] - 2s 5ms/step - loss: 0.4386 - accuracy: 0.8
818 - val_loss: 0.4173 - val_accuracy: 0.8865
Epoch 28/100
469/469 [==============================] - 2s 5ms/step - loss: 0.4295 - accuracy: 0.8
840 - val_loss: 0.4089 - val_accuracy: 0.8879
Epoch 29/100
469/469 [==============================] - 3s 6ms/step - loss: 0.4210 - accuracy: 0.8
857 - val_loss: 0.4004 - val_accuracy: 0.8895
Epoch 30/100
469/469 [==============================] - 3s 6ms/step - loss: 0.4134 - accuracy: 0.8
881 - val_loss: 0.3938 - val_accuracy: 0.8912
Epoch 31/100
469/469 [==============================] - 3s 5ms/step - loss: 0.4063 - accuracy: 0.8
897 - val_loss: 0.3863 - val_accuracy: 0.8941
Epoch 32/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3998 - accuracy: 0.8
906 - val_loss: 0.3807 - val_accuracy: 0.8944
Epoch 33/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3938 - accuracy: 0.8
921 - val_loss: 0.3747 - val_accuracy: 0.8962
Epoch 34/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3881 - accuracy: 0.8
937 - val_loss: 0.3698 - val_accuracy: 0.8973
Epoch 35/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3829 - accuracy: 0.8
946 - val_loss: 0.3648 - val_accuracy: 0.8979
Epoch 36/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3782 - accuracy: 0.8
959 - val_loss: 0.3602 - val_accuracy: 0.8991
Epoch 37/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3736 - accuracy: 0.8
967 - val_loss: 0.3563 - val_accuracy: 0.9005
Epoch 38/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3694 - accuracy: 0.8
976 - val_loss: 0.3521 - val_accuracy: 0.9012
Epoch 39/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3654 - accuracy: 0.8
983 - val_loss: 0.3489 - val_accuracy: 0.9013
Epoch 40/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3617 - accuracy: 0.8
```

```
994 - val_loss: 0.3452 - val_accuracy: 0.9032
Epoch 41/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3581 - accuracy: 0.9
002 - val_loss: 0.3416 - val_accuracy: 0.9036
Epoch 42/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3547 - accuracy: 0.9
008 - val_loss: 0.3385 - val_accuracy: 0.9046
Epoch 43/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3516 - accuracy: 0.9
016 - val_loss: 0.3359 - val_accuracy: 0.9055
Epoch 44/100
469/469 [==============================] - 3s 5ms/step - loss: 0.3486 - accuracy: 0.9
018 - val_loss: 0.3326 - val_accuracy: 0.9064
Epoch 45/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3457 - accuracy: 0.9
031 - val_loss: 0.3305 - val_accuracy: 0.9063
Epoch 46/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3430 - accuracy: 0.9
034 - val_loss: 0.3281 - val_accuracy: 0.9070
Epoch 47/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3403 - accuracy: 0.9
039 - val_loss: 0.3251 - val_accuracy: 0.9081
Epoch 48/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3378 - accuracy: 0.9
047 - val_loss: 0.3233 - val_accuracy: 0.9071
Epoch 49/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3354 - accuracy: 0.9
052 - val_loss: 0.3214 - val_accuracy: 0.9084
Epoch 50/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3331 - accuracy: 0.9
054 - val_loss: 0.3192 - val_accuracy: 0.9092
Epoch 51/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3309 - accuracy: 0.9
063 - val_loss: 0.3169 - val_accuracy: 0.9093
Epoch 52/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3286 - accuracy: 0.9
065 - val_loss: 0.3144 - val_accuracy: 0.9099
Epoch 53/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3265 - accuracy: 0.9
075 - val_loss: 0.3131 - val_accuracy: 0.9098
Epoch 54/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3246 - accuracy: 0.9
075 - val_loss: 0.3110 - val_accuracy: 0.9110
Epoch 55/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3227 - accuracy: 0.9
079 - val_loss: 0.3092 - val_accuracy: 0.9114
Epoch 56/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3208 - accuracy: 0.9
085 - val_loss: 0.3075 - val_accuracy: 0.9121
Epoch 57/100
469/469 [==============================] - 3s 5ms/step - loss: 0.3190 - accuracy: 0.9
085 - val_loss: 0.3057 - val_accuracy: 0.9118
Epoch 58/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3172 - accuracy: 0.9
095 - val_loss: 0.3039 - val_accuracy: 0.9125
Epoch 59/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3154 - accuracy: 0.9
094 - val_loss: 0.3028 - val_accuracy: 0.9124
Epoch 60/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3138 - accuracy: 0.9
```

```
104 - val_loss: 0.3016 - val_accuracy: 0.9140
Epoch 61/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3122 - accuracy: 0.9
104 - val_loss: 0.2998 - val_accuracy: 0.9143
Epoch 62/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3105 - accuracy: 0.9
108 - val_loss: 0.2982 - val_accuracy: 0.9145
Epoch 63/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3089 - accuracy: 0.9
114 - val_loss: 0.2966 - val_accuracy: 0.9152
Epoch 64/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3076 - accuracy: 0.9
116 - val_loss: 0.2957 - val_accuracy: 0.9157
Epoch 65/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3060 - accuracy: 0.9
122 - val_loss: 0.2944 - val_accuracy: 0.9168
Epoch 66/100
469/469 [==============================] - 2s 5ms/step - loss: 0.3046 - accuracy: 0.9
123 - val_loss: 0.2925 - val_accuracy: 0.9166
Epoch 67/100
469/469 [==============================] - 3s 5ms/step - loss: 0.3031 - accuracy: 0.9
128 - val_loss: 0.2923 - val_accuracy: 0.9153
Epoch 68/100
469/469 [==============================] - 3s 6ms/step - loss: 0.3019 - accuracy: 0.9
130 - val_loss: 0.2909 - val_accuracy: 0.9156
Epoch 69/100
469/469 [==============================] - 3s 5ms/step - loss: 0.3004 - accuracy: 0.9
136 - val_loss: 0.2890 - val_accuracy: 0.9166
Epoch 70/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2991 - accuracy: 0.9
136 - val_loss: 0.2891 - val_accuracy: 0.9175
Epoch 71/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2978 - accuracy: 0.9
142 - val_loss: 0.2873 - val_accuracy: 0.9180
Epoch 72/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2965 - accuracy: 0.9
145 - val_loss: 0.2858 - val_accuracy: 0.9182
Epoch 73/100
469/469 [==============================] - 3s 5ms/step - loss: 0.2953 - accuracy: 0.9
147 - val_loss: 0.2845 - val_accuracy: 0.9185
Epoch 74/100
469/469 [==============================] - 3s 5ms/step - loss: 0.2941 - accuracy: 0.9
150 - val_loss: 0.2837 - val_accuracy: 0.9188
Epoch 75/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2928 - accuracy: 0.9
153 - val_loss: 0.2835 - val_accuracy: 0.9190
Epoch 76/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2917 - accuracy: 0.9
156 - val_loss: 0.2820 - val_accuracy: 0.9189
Epoch 77/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2906 - accuracy: 0.9
156 - val_loss: 0.2805 - val_accuracy: 0.9194
Epoch 78/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2893 - accuracy: 0.9
161 - val_loss: 0.2798 - val_accuracy: 0.9197
Epoch 79/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2882 - accuracy: 0.9
166 - val_loss: 0.2788 - val_accuracy: 0.9197
Epoch 80/100
469/469 [==============================] - 3s 5ms/step - loss: 0.2872 - accuracy: 0.9
```

```
169 - val_loss: 0.2778 - val_accuracy: 0.9196
Epoch 81/100
469/469 [==============================] - 3s 6ms/step - loss: 0.2861 - accuracy: 0.9
168 - val_loss: 0.2765 - val_accuracy: 0.9206
Epoch 82/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2850 - accuracy: 0.9
176 - val_loss: 0.2758 - val_accuracy: 0.9206
Epoch 83/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2840 - accuracy: 0.9
178 - val_loss: 0.2748 - val_accuracy: 0.9208
Epoch 84/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2829 - accuracy: 0.9
180 - val_loss: 0.2741 - val_accuracy: 0.9208
Epoch 85/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2818 - accuracy: 0.9
183 - val_loss: 0.2731 - val_accuracy: 0.9213
Epoch 86/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2808 - accuracy: 0.9
186 - val_loss: 0.2726 - val_accuracy: 0.9214
Epoch 87/100
469/469 [==============================] - 3s 6ms/step - loss: 0.2799 - accuracy: 0.9
188 - val_loss: 0.2712 - val_accuracy: 0.9219
Epoch 88/100
469/469 [==============================] - 3s 5ms/step - loss: 0.2789 - accuracy: 0.9
187 - val_loss: 0.2705 - val_accuracy: 0.9220
Epoch 89/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2779 - accuracy: 0.9
192 - val_loss: 0.2696 - val_accuracy: 0.9221
Epoch 90/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2770 - accuracy: 0.9
192 - val_loss: 0.2686 - val_accuracy: 0.9228
Epoch 91/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2759 - accuracy: 0.9
198 - val_loss: 0.2680 - val_accuracy: 0.9223
Epoch 92/100
469/469 [==============================] - 2s 5ms/step - loss: 0.2750 - accuracy: 0.9
205 - val_loss: 0.2674 - val_accuracy: 0.9233
Epoch 93/100
469/469 [==============================] - 3s 6ms/step - loss: 0.2741 - accuracy: 0.9
202 - val_loss: 0.2668 - val_accuracy: 0.9230
Epoch 94/100
469/469 [==============================] - 3s 6ms/step - loss: 0.2732 - accuracy: 0.9
203 - val_loss: 0.2660 - val_accuracy: 0.9238
Epoch 95/100
469/469 [==============================] - 3s 6ms/step - loss: 0.2723 - accuracy: 0.9
209 - val_loss: 0.2649 - val_accuracy: 0.9236
Epoch 96/100
469/469 [==============================] - 3s 6ms/step - loss: 0.2714 - accuracy: 0.9
210 - val_loss: 0.2643 - val_accuracy: 0.9241
Epoch 97/100
469/469 [==============================] - 3s 6ms/step - loss: 0.2705 - accuracy: 0.9
213 - val_loss: 0.2632 - val_accuracy: 0.9241
Epoch 98/100
469/469 [==============================] - 3s 6ms/step - loss: 0.2696 - accuracy: 0.9
214 - val_loss: 0.2625 - val_accuracy: 0.9243
Epoch 99/100
469/469 [==============================] - 3s 6ms/step - loss: 0.2688 - accuracy: 0.9
218 - val_loss: 0.2614 - val_accuracy: 0.9249
Epoch 100/100
```

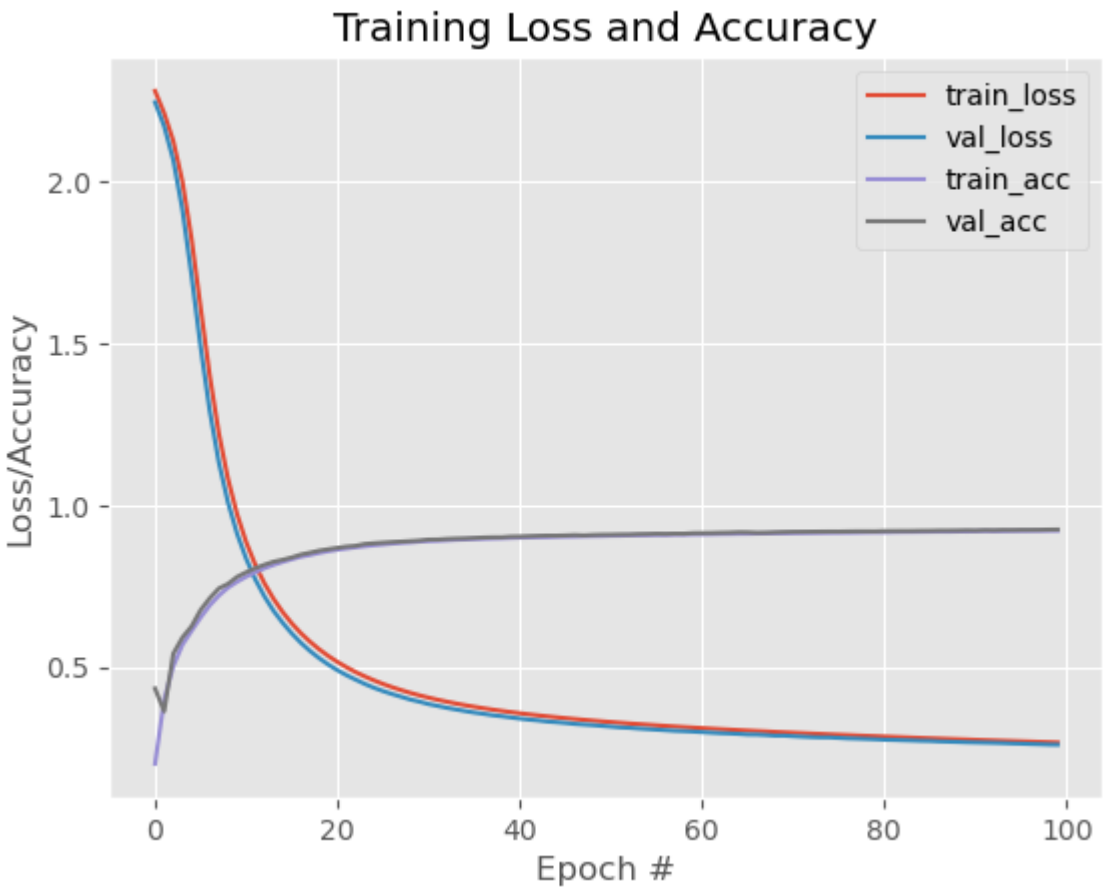
```
469/469 [==============================] - 3s 6ms/step - loss: 0.2678 - accuracy: 0.9
219 - val_loss: 0.2610 - val_accuracy: 0.9252
```

In [16]:
```
# evaluate the network
print("[INFO] evaluating network...")
predictions = model.predict(testX, batch_size=128)
print(classification_report(testY.argmax(axis=1),
        predictions.argmax(axis=1),
        target_names=[str(x) for x in lb.classes_]))
```

```
[INFO] evaluating network...
79/79 [==============================] - 0s 3ms/step
              precision    recall  f1-score   support

           0       0.94      0.98      0.96       980
           1       0.97      0.97      0.97      1135
           2       0.92      0.91      0.91      1032
           3       0.90      0.91      0.91      1010
           4       0.92      0.93      0.93       982
           5       0.91      0.86      0.88       892
           6       0.93      0.94      0.94       958
           7       0.94      0.92      0.93      1028
           8       0.89      0.90      0.89       974
           9       0.91      0.91      0.91      1009

    accuracy                           0.93     10000
   macro avg       0.92      0.92      0.92     10000
weighted avg       0.93      0.93      0.93     10000
```

In [17]:
```
# plot the training loss and accuracy
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, 100), H.history["loss"], label="train_loss")
plt.plot(np.arange(0, 100), H.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, 100), H.history["accuracy"], label="train_acc")
plt.plot(np.arange(0, 100), H.history["val_accuracy"], label="val_acc")
plt.title("Training Loss and Accuracy")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend()
plt.savefig(args["output"])
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [17], in <cell line: 12>()
     10 plt.ylabel("Loss/Accuracy")
     11 plt.legend()
---> 12 plt.savefig(args["output"])

NameError: name 'args' is not defined
```


Training Loss and Accuracy
train_loss
val_loss
train_acc
val_acc
Loss/Accuracy
2.0
1.5
1.0
0.5
0
20
40
60
80
100
Epoch #

In [ ]: