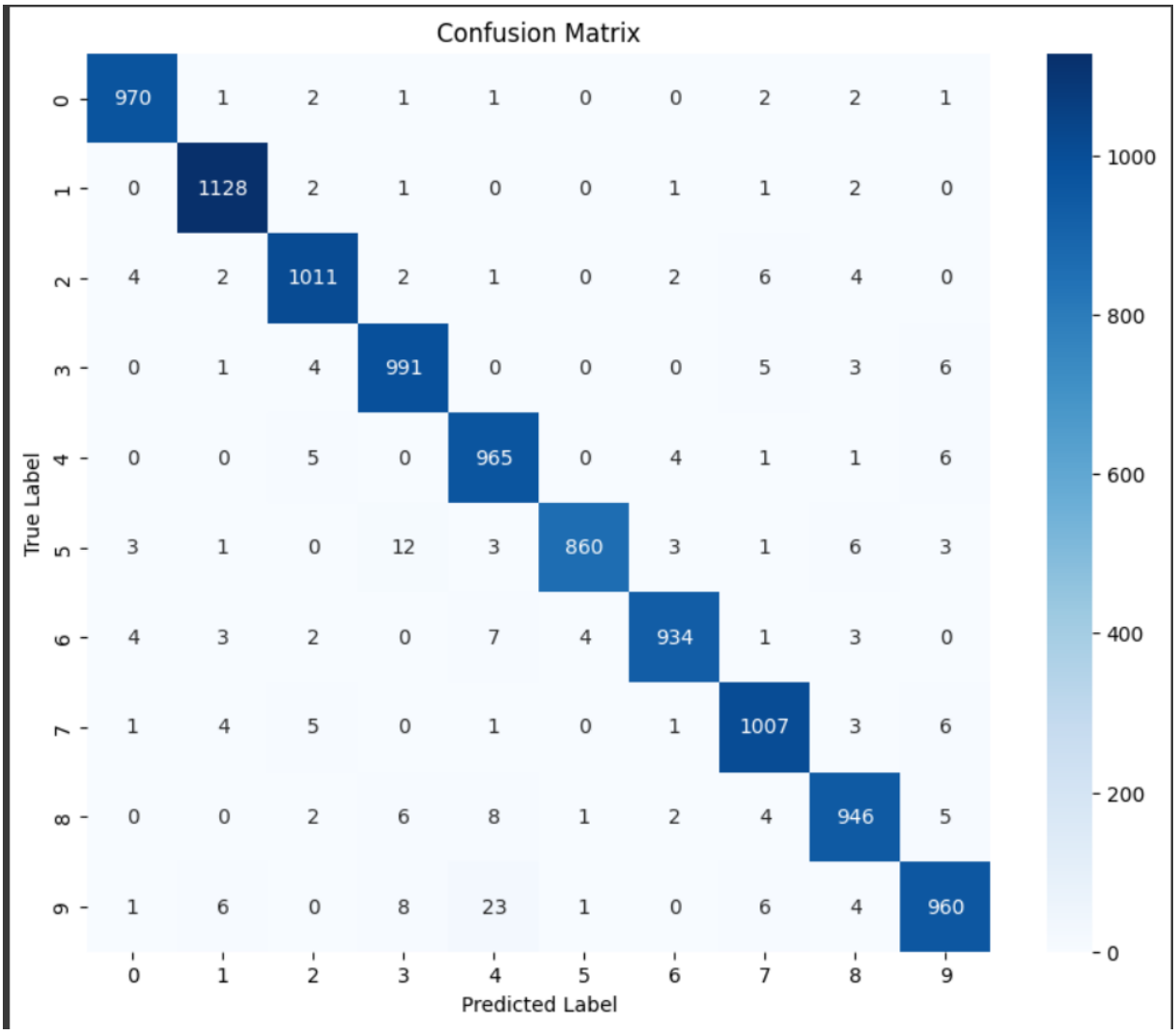
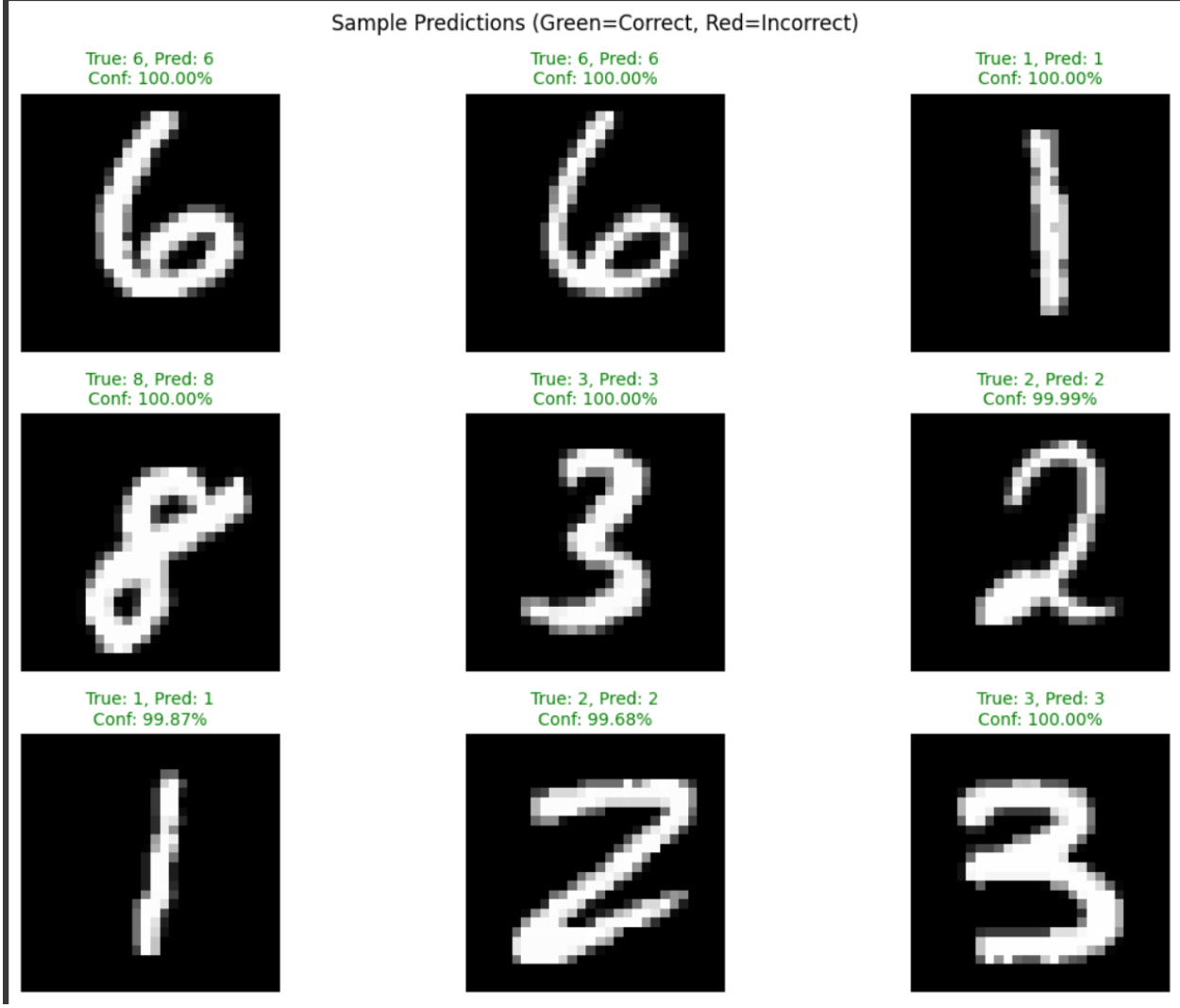
RESULTS:

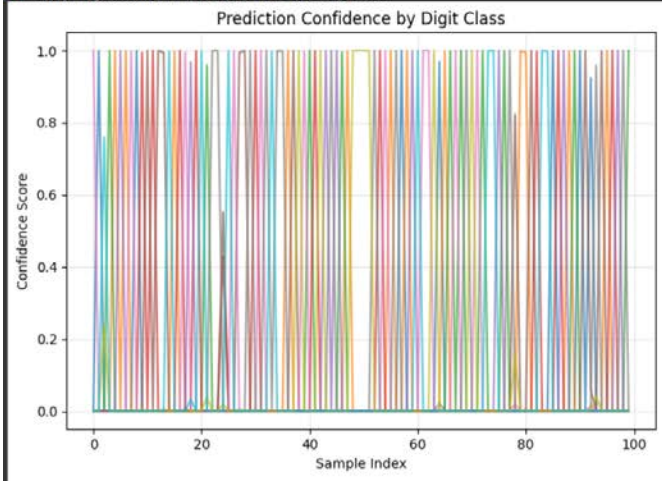
```
================================================
Features:
✅ CNN model with >95% accuracy on MNIST
✅ Comprehensive model evaluation
✅ Training history visualization
✅ Confusion matrix analysis
✅ Sample prediction testing
✅ Custom digit creation and testing
✅ NEW: Advanced confidence analysis with line graphs
================================================
✅ Model loaded successfully from file
✅ Model recompiled successfully
✅ Compiled metrics built successfully

📊 Test Results:
✅ Accuracy: 0.9772 (97.72%)

📋 Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       980
           1       0.98      0.99      0.99      1135
           2       0.98      0.98      0.98      1032
           3       0.97      0.98      0.98      1010
           4       0.96      0.98      0.97       982
           5       0.99      0.96      0.98       892
           6       0.99      0.97      0.98       958
           7       0.97      0.98      0.98      1028
           8       0.97      0.97      0.97       974
           9       0.97      0.95      0.96      1009

    accuracy                           0.98     10000
   macro avg       0.98      0.98      0.98     10000
weighted avg       0.98      0.98      0.98     10000
```

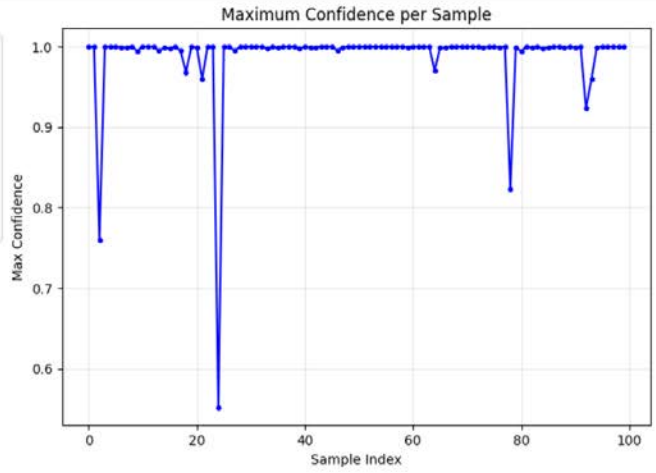

Confusion Matrix
True Label
Predicted Label
0 970 1 2 1 1 0 0 2 2 1
1 0 1128 2 1 0 0 1 1 2 0
2 4 2 1011 2 1 0 2 6 4 0
3 0 1 4 991 0 0 0 5 3 6
4 0 0 5 0 965 0 4 1 1 6
5 3 1 0 12 3 860 3 1 6 3
6 4 3 2 0 7 4 934 1 3 0
7 1 4 5 0 1 0 1 1007 3 6
8 0 0 2 6 8 1 2 4 946 5
9 1 6 0 8 23 1 0 6 4 960
0 1 2 3 4 5 6 7 8 9
1000
800
600
400
200
0

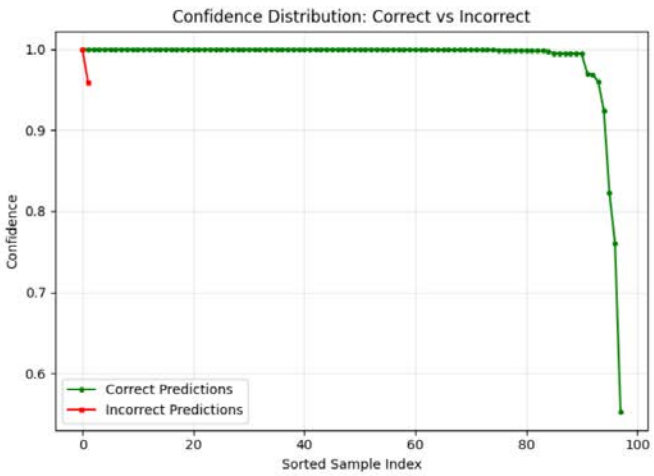
Testing Sample Predictions:

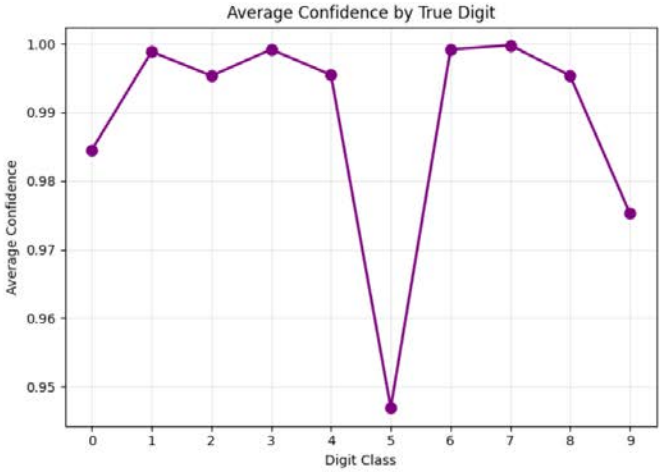


```
Generating Confidence Analysis Line Graphs:
Analyzing confidence patterns for 100 samples...
```

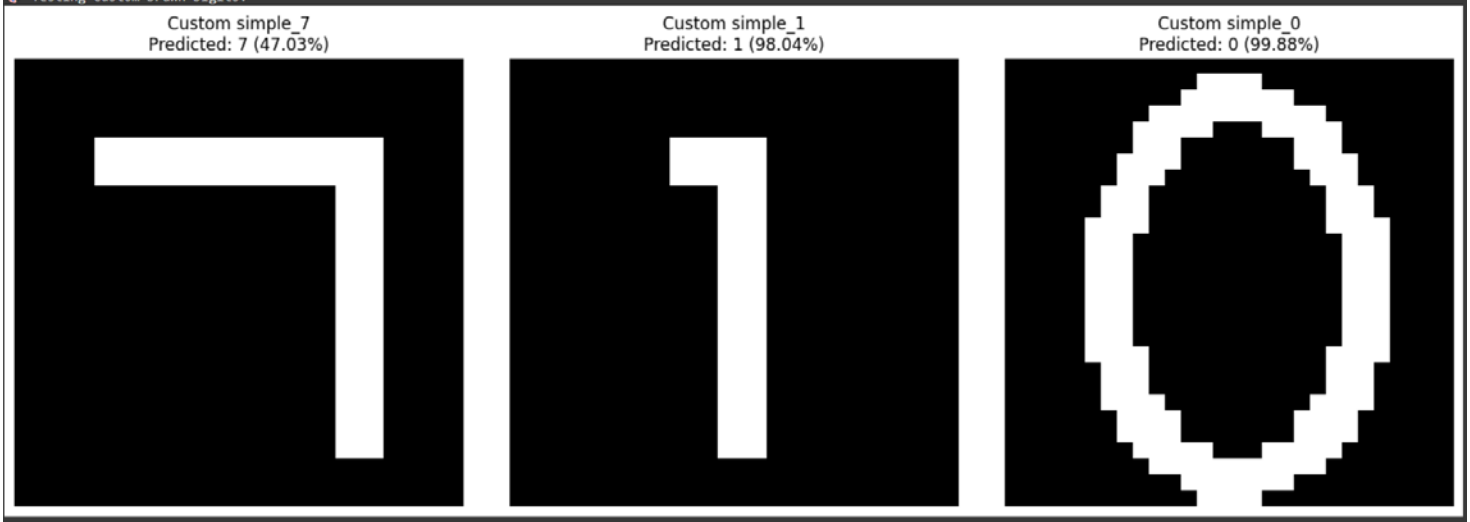








```
Confidence Analysis Results:
Correct predictions: 98
Incorrect predictions: 2
Average confidence (correct): 0.989
Average confidence (incorrect): 0.979

Testing Custom Drawn Digits:
```




```
System demonstration complete!

To use the system programmatically:
1. system = DigitRecognitionSystem()
2. prediction = system.predict_digit(your_28x28_image)
3. predicted_digit, confidence, all_probs = prediction
4. system.plot_prediction_confidence_lines(num_samples=100)
```

THE END