STM32 F401RE CNN Evaluation

```
import numpy as np
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix, classification_report
from sklearn.metrics import roc_auc_score
```

DATA

Dataset: 256 entry **Test_Dataset**: 25 entry

```
y_train =
[0.00,1.00,0.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,0.00,0.00,1.00
,0.00,0.00,1.00,0.00,0.00,1.00,0.00,1.00,1.00,1.00,1.00,1.00,1.00,0.00
,1.00,0.00,1.00,0.00,1.00,1.00,1.00,0.00,1.00,1.00,0.00,0.00,0.00,1.00
,0.00,1.00,1.00,0.00,1.00,0.00,0.00,1.00,1.00,0.00,1.00,0.00,0.00,1.00
,0.00,0.00,1.00,0.00,0.00,0.00,1.00,0.00,0.00,1.00,1.00,0.00,0.00,1.00
,0.00,1.00,1.00,1.00,1.00,1.00,1.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
,1.00,1.00,1.00,0.00,1.00,1.00,1.00,1.00,1.00,0.00,0.00,1.00,1.00,1.00
,0.00,1.00,1.00,1.00,1.00,0.00,1.00,1.00,0.00,0.00,0.00,1.00,1.00,1.00
,1.00,1.00,1.00,0.00,0.00,0.00,1.00,1.00,0.00,0.00,0.00,1.00,0.00,1.00
,0.00,0.00,1.00,1.00,1.00,0.00,1.00,1.00,1.00,1.00,0.00,1.00,0.00,1.00
,0.00,1.00,0.00,0.00,0.00,0.00,1.00,0.00,1.00,1.00,1.00,1.00,0.00,0.00
,1.00,0.00,1.00,0.00,0.00,1.00,1.00,1.00,0.00,0.00,0.00,1.00,0.00,1.00
,1.00,1.00,0.00,1.00,0.00,1.00,0.00,0.00,1.00,1.00,1.00,0.00,1.00,1.00
,0.00,0.00,1.00,1.00,1.00,0.00,1.00,0.00,1.00,1.00,1.00,1.00,0.00,1.00
,0.00,0.00,1.00,1.00,0.00,1.00,1.00,0.00,1.00,0.00,1.00,0.00,0.00,1.00
,1.00,1.00,1.00,1.00,1.00,0.00,0.00,1.00,0.00,0.00,0.00,0.00,1.00,0.00
,0.00,1.00,0.00,0.00,0.00,1.00,0.00,1.00,0.00,0.00,1.00,0.00,0.00,0.00
,1.00,1.00,0.00,1.00,1.00,0.00,1.00,0.00,0.00,0.00,0.00,1.00,1.00,1.00
,0.00,1.00,0.00,0.00]
y_train_pred =
np.array([0.18,0.98,0.00,0.98,0.98,0.98,0.98,0.98,0.98,0.05,0.98,0.01,
0.00,0.98,0.00,0.00,0.98,0.01,0.01,0.98,0.01,0.98,0.98,0.80,0.98,0.98,
0.98,0.01,0.98,0.04,0.98,0.00,0.98,0.98,0.98,0.01,0.98,0.98,0.04,0.05,
0.00,0.98,0.00,0.00,0.98,0.00,0.98,0.00,0.18,0.98,0.98,0.00,0.98,0.00,
0.01,0.98,0.01,0.01,0.98,0.05,0.09,0.00,0.98,0.00,0.01,0.98,0.98,0.00,
0.01,0.98,0.00,0.98,0.98,0.98,0.98,0.98,0.98,0.05,0.00,0.01,0.00,0.00,
0.00,0.00,0.98,0.98,0.98,0.00,0.98,0.98,0.98,0.98,0.98,0.00,0.00,0.98,
0.98,0.98,0.51,0.98,0.00,0.98,0.98,0.00,0.98,0.98,0.00,0.00,0.02,0.98,
0.80,0.98,0.98,0.98,0.88,0.00,0.00,0.00,0.98,0.98,0.00,0.06,0.04,0.98,
0.01,0.98,0.01,0.04,0.98,0.98,0.98,0.05,0.98,0.98,0.98,0.00,0.18,0.98,
0.00,0.98,0.29,0.98,0.00,0.00,0.01,0.00,0.98,0.01,0.80,0.80,0.98,0.98,
0.01,0.00,0.98,0.00,0.98,0.98,0.01,0.98,0.98,0.98,0.00,0.00,0.00,0.98,
0.00,0.98,0.98,0.98,0.00,0.98,0.01,0.98,0.00,0.00,0.98,0.98,0.98,0.00,
0.98,0.98,0.01,0.00,0.98,0.98,0.98,0.00,0.98,0.00,0.98,0.80,0.80,0.98,
0.00,0.98,0.00,0.00,0.98,0.98,0.05,0.98,0.98,0.00,0.98,0.00,0.98,0.01,
0.18,0.98,0.98,0.98,0.98,0.98,0.98,0.00,0.00,0.18,0.01,0.02,0.00,0.00,
0.98,0.00,0.01,0.98,0.04,0.01,0.00,0.98,0.01,0.98,0.01,0.01,0.98,0.00,
```

```
0.01,0.00,0.88,0.98,0.00,0.98,0.98,0.01,0.98,0.04,0.01,0.05,0.01,0.98,
0.98,0.18,0.05,0.98,0.00,0.01])

y_test =
[1.00,1.00,0.00,1.00,1.00,1.00,0.00,1.00,1.00,1.00,1.00,1.00,0.00,0.00
,0.00,0.00,1.00,0.00,1.00,0.00,1.00,0.00,0.00,0.00,1.00]
y_test_pred =
np.array([0.18,0.98,0.00,0.98,0.98,0.98,0.98,0.98,0.98,0.05,0.98,0.01,
0.00,0.98,0.00,0.00,0.98,0.01,0.01,0.98,0.01,0.98,0.98,0.80,0.98])
```

Set an "acceptance" limit

```
y_train_pred_bin = (y_train_pred >= 0.05).astype(int)
y_test_pred_bin = (y_test_pred >= 0.05).astype(int)
```

Helper function in order to evaluate our model

```
def evaluate_model(y_true, y_pred, dataset_name=""):
    print(f"=== Evaluation for {dataset_name} ===")
    print(f"Accuracy: {accuracy_score(y_true, y_pred):.4f}")
    print(f"Precision: {precision_score(y_true, y_pred):.4f}")
    print(f"Recall: {recall_score(y_true, y_pred):.4f}")
    print(f"F1 Score: {f1_score(y_true, y_pred):.4f}")
    print("Matrice di confusione:")
    print(confusion_matrix(y_true, y_pred))
    print("Report dettagliato:")
    print(classification_report(y_true, y_pred))
    print("\n")
```

Evaluation on training data

```
evaluate_model(y_train, y_train_pred_bin, "Training Set")
```

```
=== Evaluation for Training Set ===
Accuracy: 0.9258
Precision: 0.8940
Recall: 0.9783
F1 Score: 0.9343
Matrice di confusione:
[[102  16]
 [  3 135]]
Report dettagliato:
              precision    recall  f1-score   support

         0.0       0.97      0.86      0.91       118
         1.0       0.89      0.98      0.93       138

    accuracy                           0.93       256
   macro avg       0.93      0.92      0.92       256
weighted avg       0.93      0.93      0.93       256
```

```
auc = roc_auc_score(y_train, y_train_pred)
print("ROC AUC Score on training data:", auc)
```

```
ROC AUC Score on training data: 0.9782915745517071
```

Evaluation on test data

```
evaluate_model(y_test, y_test_pred_bin, "Test Set")
```

```
=== Evaluation for Test Set ===
Accuracy: 0.6400
Precision: 0.6471
Recall: 0.7857
F1 Score: 0.7097
Matrice di confusione:
[[ 5  6]
 [ 3 11]]
Report dettagliato:
              precision    recall  f1-score   support

         0.0       0.62      0.45      0.53        11
         1.0       0.65      0.79      0.71        14

    accuracy                           0.64        25
   macro avg       0.64      0.62      0.62        25
weighted avg       0.64      0.64      0.63        25
```

```
auc = roc_auc_score(y_test, y_test_pred)
print("ROC AUC Score on test data:", auc)
```

```
ROC AUC Score on test data: 0.6493506493506493
```