# Verification and Validation Report: OCRacle

Phillip Tran

April 17, 2025

# 1 Revision History

| Date | Version | Notes |
|---|---|---|
| April 9, 2025 | 1.0 | Initial document creation |

# 2 Symbols, Abbreviations and Acronyms

| symbol | description |
|---|---|
| T | Test |
| GHA | GitHub Actions |

# Contents



# List of Tables



# List of Figures

# 3 Functional Requirements Evaluation

All functional requirements were evaluated via automated tests using pytest. All tests in the test suite passed successfully. To view the output of the automated test suite, see Appendix 11.1.

# 4 Nonfunctional Requirements Evaluation

## 4.1 Performance

T9 was supported by a test case in the automated test suite. The test case ensured that the overall accuracy of the program exceeded 67.4%, which is the accuracy of the previous OAR project (**?**).

The OCRacle program achieved an accuracy of 90.55% as indicated by the Model Evaluation Output in Appendix 11.2, which is a significant improvement over the previous OAR project.

## 4.2 Usability

T10 was not fully performed due to the time constraints of the project. However, the project author performed a manual usability test of the program and found that the documentation was accurate and complete.

## 4.3 Maintainability

To support T11, the code is run through the ruff linter, which checks for any problems with the code formatting and automatically fixes any problems that it can. In the final version of the code, there are no problems with the code.

To support T12, the Pyright static type checker was used to check for any problems with the type annotations in the code. In the final version of the code, there are no problems with the type annotations.

T13 was not fully performed due to the time constraints of the project. However, the project author did perform a manual code review of the code based upon the Source Code Checklist (**?**) as specified by T13 and found no major problems with the code in the final version.

### 4.4 Portability

To support T14, the automated Pytest test suite was run in a Windows, MacOS, and Linux environment on GHA. All tests passed successfully in all environments, validating the program's cross-platform compatibility.

## 5 Unit Testing

All unit tests were evaluated via automated tests using pytest. All tests in the test suite passed successfully. To view the output of the automated test suite, see Appendix 11.1.

## 6 Changes Due to Testing

The following major changes were made after a peer demonstration:

- The Accuracy Metrics Module was renamed the Model Evaluation Module in order to more clearly communicate the purpose of the module.
- The confusion matrix was renamed to the probability distribution in order to more clearly communicate the purpose of the model output.

## 7 Automated Testing

The automated test suite was setup to run on GHA. The test suite is run automatically on every commit to the main branch and for every pull request. The automated tests are stored in the .github/workflows directory.

## 8 Trace to Requirements

The following table outlines the traceability between the test cases and the requirements outlined in the SRS document. Note: The table has been generated using ChatGPT 4o. The output has been manually validated to ensure that it is correct. [1]

[1]The following query was used: "Fill out the following traceability matrix given the following table template and the information from this document: [table template and

| Test Case | R1 | R2 | R3 | R4 | R5 | NFR1 | NFR2 | NFR3 | NFR4 |
|---|---|---|---|---|---|---|---|---|---|
| T1 | X | | | | | | | | |
| T2 | X | | | | | | | | |
| T3 | X | | | | | | | | |
| T4 | | X | | | | | | | |
| T5 | | | X | | | | | | |
| T6 | | | | X | | | | | |
| T7 | | | | X | | | | | |
| T8 | | | | | X | | | | |
| T9 | | | | | | X | | | |
| T10 | | | | | | | X | | |
| T11 | | | | | | | | X | |
| T12 | | | | | | | | X | |
| T13 | | | | | | | | | X |

Table 1: Test Cases to Requirements Matrix

# 9 Trace to Modules

The following table outlines the traceability between test cases and modules outlined in the MG (**?**). Note: This table has been generated using ChatGPT 4o. The output has been manually validated to ensure that it is correct. [2]

# 10 Code Coverage Metrics

The following command was run to produce the code coverage metrics: [3]

information from this document]."

[2]The following query was used: "Here is an example of a table in LaTeX: [table template]. Please create a new table for tracking the tracability of test cases and modules in the same table format. Here is the information you'll need to complete this: [list of modules and the tests that validate each one]."

[3]Third-party libraries from the `src/libraries` directory were omitted from the coverage report as this was considered out of scope for this project. Additionally, `if __name__ == '__main__'` blocks were also omitted since they are not executed when the module is imported.

| Test Case | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M9 | M10 |
|---|---|---|---|---|---|---|---|---|---|
| T1 | | | X | | | | | | |
| T2 | | | X | | | | | | |
| T3 | | | X | | | | | | |
| T4 | | | | | X | | | | |
| T5 | | | X | | | | X | | |
| T6 | | | | | | X | | | |
| T7 | | | | | | X | | | |
| T8 | | | | | | | X | | |
| T9 | | | | | | | | X | |
| T10 | X | X | X | X | X | X | X | X | X |
| T11 | X | X | X | X | X | X | X | X | X |
| T12 | X | X | X | X | X | X | X | X | X |
| T13 | X | X | X | X | X | X | X | X | X |
| T14 | X | X | X | X | X | X | X | X | X |
| T15 | | | | X | | | | | |
| T16 | | | | | | | | | X |
| T17 | | | | | | | | | X |
| T18 | | | X | | | | | | |
| T19 | | | X | | | | | | |

Table 2: Traceability Matrix of Test Cases to Modules

```
coverage run -m pytest test/test.py && coverage report -m
```

This produced the following output:

```
Name                    Stmts   Miss  Cover   Missing
-----------------------------------------------------
src/data.py                14      0   100%
src/evaluation.py          17      0   100%
src/input.py               25      0   100%
src/model.py                5      0   100%
src/output.py               9      0   100%
```

```
src/preprocessing.py      11      0   100%
src/train.py              18      0   100%
test/test.py             112      0   100%
-----------------------------------------------------
TOTAL                    211      0   100%
```

# 11 Appendix

## 11.1 Pytest Test Suite Output

```
================== test session starts ==================
platform darwin -- Python 3.9.6, pytest-8.3.5, pluggy-1.5.0
-- /Users/phillip/Git/OCRacle/env/bin/python
cachedir: .pytest_cache
rootdir: /Users/phillip/Git/OCRacle
configfile: pyproject.toml
plugins: emoji-0.2.0, md-0.2.0, cov-6.1.1
collected 14 items

test/test.py::testJpegAcceptance PASSED [  7%]
test/test.py::testPngAcceptance PASSED
test/test.py::testNonSupportedFormatRejection PASSED
test/test.py::testImagePreProcessing PASSED
test/test.py::testCharacterPrediction PASSED
test/test.py::testProbabilityVectorSum PASSED
test/test.py::testProbabilityVectorLength PASSED
test/test.py::testHumanReadableOutput PASSED
test/test.py::testAccuracyMeasurement PASSED
test/test.py::testModelTraining PASSED
test/test.py::testLoadTrainSubset PASSED
test/test.py::testLoadTestSubset PASSED
test/test.py::testFileNotFound PASSED
test/test.py::testInvalidDimensions PASSED

==================
warnings summary
==================
```

```
env/lib/python3.9/site-packages/urllib3/__init__.py:35
  /Users/phillip/Git/OCRacle/env/lib/python3.9/site-packages/urllib3/__init__.py:3
    warnings.warn(

test/test.py: 16 warnings
  /Users/phillip/Git/OCRacle/test/../src/libraries/emnist.py:226: DeprecationWarni
    dim_size = int(numpy.frombuffer(data[offset : offset + 4], dtype=">u4"))

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
================== 14 passed, 17 warnings in 59.12s ==================
```

## 11.2 Model Evaluation Output

```
Loss: 0.3176746666431427
Accuracy: 0.9055288434028625
Confusion Matrix:
  tf.Tensor(
[[735   3   3   6  10   1   5   6   0   1   0   0   1   2   7   6   9   0
    0   0   2   0   0   1   0   2]
  [  5 754   0   4   3   0   4  13   1   1   0   6   0   1   3   0   1   1
    1   0   1   0   0   0   0   1]
  [  0   1 744   3  35   1   1   0   1   0   0   4   0   0   2   0   0   2
    2   1   2   0   1   0   0   0]
  [ 11   7   0 720   0   1   1   1   0   5   0   1   0   2  39   4   4   2
    0   0   0   0   0   0   0   2]
  [  1   1  13   0 769   0   2   0   2   0   0   3   0   0   2   2   1   2
    0   1   1   0   0   0   0   0]
  [  0   0   0   1   5 731   2   0   2   1   0   1   0   0   0  24   1   5
    4  23   0   0   0   0   0   0]
  [ 25  19   7   2  15   3 588   1   0   6   0   1   0   1   2   2 120   1
    4   1   0   0   0   0   2   0]
  [  8   1   0   1   0   1   0 739   1   1   6  13   4  10   0   0   0   1
    0   2   5   0   2   3   2   0]
  [  0   0   2   0   2   3   1   1 551  16   1 210   0   0   0   0   1   1
    1   0   0   1   1   2   0   6]
  [  1   1   0   8   0   1   4   0  29 734   0   3   0   0   0   0   1   0
    3  10   0   1   0   3   1   0]
  [  2   4   2   1   3   3   0  15   0   0 723   5   2   1   0   0   0   9
```

```
  0   6   4   2   0  16   1   1]
 [  0   1  11   0   0   1   0   4 158   1   0 621   0   0   1   0   1   0
   0   1   0   0   0   0   0   0]
 [  1   1   0   0   0   0   0   5   0   0   1   0 780   3   0   1   0   0
   0   1   3   0   2   2   0   0]
 [ 26   0   0   8   0   0   0  17   0   2   2   1  23 682   0   6   0  12
   0   2   5   2   8   4   0   0]
 [  3   0   1  12   2   0   0   0   0   0   0   0   0   1 775   2   2   1
   0   0   1   0   0   0   0   0]
 [  0   0   0   3   1   8   1   0   0   0   0   0   0   0   1 783   0   1
   0   1   0   0   0   1   0   0]
 [ 34   4   3   2   6   1  69   0   2   1   0   0   0   1   7   4 654   5
   0   1   2   0   1   0   3   0]
 [  9   4   2   0   5   4   0   0   2   0   4   2   1   1   0   3   1 727
   1   7   0  14   0   7   3   3]
 [  5   4   0   2   2   4  13   1   1  11   0   0   0   1   0   0   0   0
 756   0   0   0   0   0   0   0]
 [  1   1   1   0   3   2   0   0   3   1   1   2   0   0   1   1   1   2
   1 770   0   0   0   4   3   2]
 [  5   0   1   3   0   0   0   4   0   3   1   1   2   3   8   0   1   1
   1   0 737  25   2   1   1   0]
 [  0   0   0   2   0   0   0   0   0   1   0   1   0   1   0   0   0   8
   0   1  32 737   0   0  17   0]
 [  1   0   0   3   0   0   0   3   0   0   2   0   3   7   0   0   0   0
   0   1  12   1 765   2   0   0]
 [  4   0   0   2   0   1   1   1   0   1  10   1   0   0   0   1   1   2
   1   2   0   1   1 758  10   2]
 [  0   2   0   1   0   0   6   3   1   8   0   2   0   0   0   2   1   6
   1   4   2  14   0   8 739   0]
 [  4   1   1   2  10   1   5   1   3   0   0   0   0   0   0   1   1   1
   0   0   0   0   0   6   0 763]], shape=(26, 26), dtype=int32)
Class Accuracy:
  {'P': 0.97875, 'M': 0.975, 'O': 0.96875, 'T': 0.9625, 'E': 0.96125,
  'W': 0.95625, 'Z': 0.95375, 'X': 0.9475, 'S': 0.945, 'B': 0.9425, 'C': 0.93,
  'H': 0.92375, 'Y': 0.92375, 'U': 0.92125, 'V': 0.92125, 'A': 0.91875,
  'J': 0.9175, 'F': 0.91375, 'R': 0.90875, 'K': 0.90375, 'D': 0.9, 'N': 0.8525,
  'Q': 0.8175, 'L': 0.77625, 'G': 0.735, 'I': 0.68875}
```