Verification and Validation Report: Artificial Neural Network (ANN)

Tanya Djavaherpour

April 15, 2024

1 Revision History

Date	Version	Notes
Apr. 15, 2024	1.0	Initial Draft

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

For more details, please refer to the SRS document (Djavaherpour, 2024a) HERE.

Contents

List of Tables

List of Figures

This document summarizes the Verification and Validation (VnV) processes applied to the ANN. It includes results from executing all test cases outlined in the VnV Plan (Djavaherpour, 2024b). Detailed outcomes for the functional and non-functional requirements are presented in Sections 3 and 4, respectively. Additional sections provide comprehensive analyses of the test results.

3 Functional Requirements Evaluation

The functional requirements are evaluated in this section. A detailed description of all of these tests can be found in the VnV Plan (Djavaherpour, 2024b).

3.1 T1

This test checks the format of the input image to ensure it meets the size and format requirements and constraints. In this test, we verify that exceptions work properly, so that the end user will not be able to use files that are unacceptable. The implemented test is available here. As it is indicated in the log file, all of the tests are passed.

3.2 T2

This is a classifier test to make sure the implemented model can predict properly. For this purpose, a test is run using Pytest (Pytest Testing Framework, 2023), which is available here. The result of this test is accessible here. As it is indicated in the result file, all the tests are passed.

4 Nonfunctional Requirements Evaluation

The nonfunctional requirements are evaluated in this section.

4.1 Accuracy

The accuracy results are shown in Table 1 and the cost values are indicater in Figure 1. Also, accuracy results are accessible here.

Dataset	Accuracy
train data	22.234 %
test data	21.959 %

Table 1: Accuracy Results

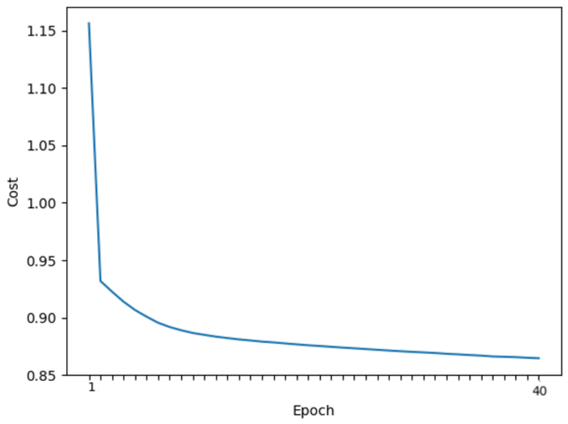

Figure 1: Cost Value over 40 Epochs

4.2 Usability

As the author has not yet received any feedback from usability surveys, reporting about this area is postponed to future drafts.

4.3 Maintainability

As it is mentioned in the VnV Plan (Djavaherpour, 2024b), Pylint (Python Code Quality Authority, 2023) library checked the quality of all of the modules. Furthermore, we conducted Walkthrough Meetings to identify areas for improvement in system maintainability and documentation.

4.4 Portability

We tested this system on Windows and MacOS devices and it could pass all the tests on bothh of them.

4.5 User Interaction Efficiency

This area will be discussed in future drafts since the author has not yet received feedback from end users.

5 Comparison to Existing Implementation

This project is a re-implementation of a previous one, originally developed in a Jupyter Notebook without distinct modules. Also, in previous implementation the model was not saved, whcih means there was not any feature for classifying uploaded images.

The previous model was trained on only 4 classes, while we trained this model on ten different classes, including airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The previous implementation has the accuracy of 45%. In this project, after increasing the number of classes the reached accuracy is as shown in Table 1.

In ANN, we expanded our dataset by 2.5 times and initially anticipated a proportional decrease in model accuracy, specifically expecting it to drop to almost 18% (45 divided by 2.5) of its original value. However, the observed decline in accuracy was significantly less than predicted, indicating a robustness in our model that can handle increased data variability effectively. This unexpected outcome not only validates our approach but also highlights the enhanced generalization capability of our model, underscoring clear progress in our project objectives.

6 Unit Testing

The following section includes different unit tests as described in the VnV Plan (Djavaherpour, 2024b). All the unit tests and their results are available here. As you can see, all of the tests are passed.

6.1 ANN Control Module Unit Test

This unit test ensures that the main function of the software wokrs properly. This function includes two parts: **Training Model** and **Classifying the Input Image**. This test is available here and the results are shown here with the OK received result.

7 Changes Due to Testing

During testing, there were no significant changes. The majority of the changes were related to exceptions. This commit contains these changes.

8 Automated Testing

All the module are tested with Pylint (Python Code Quality Authority, 2023). Additionally, the unit tests and Pytest (Pytest Testing Framework, 2023) are applied on modules that are available on GitHub repository.

9 Trace to Requirements

The traceability between tests cases and requirements is displayed in Table 2.

	R1	R2	R3	NFR1	NFR2	NFR3	NFR4
T1	X	X					
T2			X				
T3				X			
T4					X		
T5						X	
T6							X

Table 2: Tracebility between test cases and requirements

10 Trace to Modules

Table 3 indicates the traceability between tests and modules.

	M2	M3	M4	M5	M6	M7	M8	M9	M10
T1					X				
T2		X	X	X					X
test-id1 - test-id2	X								
test-id3 - test-id10		X				X			X
test-id11 - test-id18			X	X		X			
test-id19 - test-id26							X	X	

Table 3: Tracebility between test cases and modules

11 Code Coverage Metrics

The result of code coverage is available here. They have been tested based on unit tets and pytests. The resuls are shown in Table 4. There are some important points about these results:

- Having a main function to run the software is the reason control.py did not receive 100%.
- Because input_prep.py uses input_image methods, this module is not tested separately.
- Different methods of the train_and_test.py module are used in different classes, and since the model can be run and tested, there is no special test.

Name	Stmts	Miss	Cover	Teset
input_image.py	16	0	100%	test_image_properties.py
classifier.py	16	0	100%	test_classifier.py, output_unit.py
training_model.py	15	0	100%	test_classifier.py, output_unit.py, model_unit.py
data.py	57	0	100%	data_prep_unit.py
control.py	18	1	94%	control_unit.py
output.py	38	0	100%	output_unit.py
model.py	34	0	100%	model_unit.py

Table 4: Tracebility Between Test Cases and Requirements

References

Tanya Djavaherpour. System requirements specification. https://github.com/tanya-jp/ANN-CAS741/blob/main/docs/SRS/SRS.pdf, 2024a.

Tanya Djavaherpour. System verification and validation plan. https://github.com/tanya-jp/ANN-CAS741/blob/main/docs/VnVPlan/VnVPlan.pdf, 2024b.

Pytest Testing Framework. Pytest. https://docs.pytest.org/en/8.0.x/, March 2023. Python testing framework.

Python Code Quality Authority. Pylint. https://pypi.org/project/pylint/, December 2023. Python static code analysis tool.