

Module Interface Specification for OAR

Hunter Ceranic

March 8, 2024

1 Revision History

Date	Version	Notes
March 8, 2024	1.0	Initial Revision

2 Symbols, Abbreviations and Acronyms

See SRS Documentation (Ceranic, 2024) at <https://github.com/cer-hunter/OAR-CAS741/blob/main/docs/SRS/SRS.pdf>

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Introduction	1
4	Notation	1
5	Module Decomposition	2
6	MIS of Application Control Module	3
6.1	Module	3
6.2	Uses	3
6.3	Syntax	3
6.3.1	Exported Constants	3
6.3.2	Exported Access Programs	3
6.4	Semantics	3
6.4.1	State Variables	3
6.4.2	Environment Variables	3
6.4.3	Assumptions	3
6.4.4	Access Routine Semantics	3
6.4.5	Local Functions	3
7	MIS of Output Module	4
7.1	Module	4
7.2	Uses	4
7.3	Syntax	4
7.3.1	Exported Constants	4
7.3.2	Exported Access Programs	4
7.4	Semantics	4
7.4.1	State Variables	4
7.4.2	Environment Variables	4
7.4.3	Assumptions	4
7.4.4	Access Routine Semantics	4
7.4.5	Local Functions	5
8	MIS of Input Data Read Module	5
8.1	Module	5
8.2	Uses	5
8.3	Syntax	5
8.3.1	Exported Constants	5
8.3.2	Exported Access Programs	5

8.4	Semantics	5
8.4.1	State Variables	5
8.4.2	Environment Variables	5
8.4.3	Assumptions	6
8.4.4	Access Routine Semantics	6
8.4.5	Local Functions	6
9	MIS of Input Classifier Module	6
9.1	Module	6
9.2	Uses	6
9.3	Syntax	6
9.3.1	Exported Constants	6
9.3.2	Exported Access Programs	6
9.4	Semantics	7
9.4.1	State Variables	7
9.4.2	Environment Variables	7
9.4.3	Assumptions	7
9.4.4	Access Routine Semantics	7
9.4.5	Local Functions	7
10	MIS of OAR Model Data Module	7
10.1	Module	7
10.2	Uses	7
10.3	Syntax	7
10.3.1	Exported Constants	7
10.3.2	Exported Access Programs	7
10.4	Semantics	8
10.4.1	State Variables	8
10.4.2	Environment Variables	8
10.4.3	Assumptions	8
10.4.4	Access Routine Semantics	8
10.4.5	Local Functions	8
11	MIS of OAR Model Equations Module	8
11.1	Module	8
11.2	Uses	8
11.3	Syntax	8
11.3.1	Exported Constants	8
11.3.2	Exported Access Programs	9
11.4	Semantics	9
11.4.1	State Variables	9
11.4.2	Environment Variables	9
11.4.3	Assumptions	9

11.4.4	Access Routine Semantics	9
11.4.5	Local Functions	10
12	MIS of OAR Model Training Module	10
12.1	Module	10
12.2	Uses	10
12.3	Syntax	10
12.3.1	Exported Constants	10
12.3.2	Exported Access Programs	11
12.4	Semantics	11
12.4.1	State Variables	11
12.4.2	Environment Variables	11
12.4.3	Assumptions	11
12.4.4	Access Routine Semantics	11
12.4.5	Local Functions	11
13	MIS of OAR Model Testing Module	11
13.1	Module	11
13.2	Uses	12
13.3	Syntax	12
13.3.1	Exported Constants	12
13.3.2	Exported Access Programs	12
13.4	Semantics	12
13.4.1	State Variables	12
13.4.2	Environment Variables	12
13.4.3	Assumptions	13
13.4.4	Access Routine Semantics	13
13.4.5	Local Functions	13
14	MIS of Confusion Matrix Module	13
14.1	Module	13
14.2	Uses	13
14.3	Syntax	13
14.3.1	Exported Constants	13
14.3.2	Exported Access Programs	14
14.4	Semantics	14
14.4.1	State Variables	14
14.4.2	Environment Variables	14
14.4.3	Assumptions	14
14.4.4	Access Routine Semantics	14
14.4.5	Local Functions	14

15 MIS of Input Processing Module	14
15.1 Module	14
15.2 Uses	14
15.3 Syntax	15
15.3.1 Exported Constants	15
15.3.2 Exported Access Programs	15
15.4 Semantics	15
15.4.1 State Variables	15
15.4.2 Environment Variables	15
15.4.3 Assumptions	15
15.4.4 Access Routine Semantics	15
15.4.5 Local Functions	15
16 MIS of Graphical User Interface	15
16.1 Module	15
16.2 Uses	16
16.3 Syntax	16
16.3.1 Exported Constants	16
16.3.2 Exported Access Programs	16
16.4 Semantics	16
16.4.1 State Variables	16
16.4.2 Environment Variables	16
16.4.3 Assumptions	17
16.4.4 Access Routine Semantics	17
16.4.5 Local Functions	17

3 Introduction

The following document details the Module Interface Specifications for the OAR (Optical Alphabet Recognition) program. This document specifies how each module interfaces with other parts of the program.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at <https://github.com/cer-hunter/OAR-CAS741>.

4 Notation

The structure of the MIS for modules comes from [Hoffman and Strooper \(1995\)](#), with the addition that template modules have been adapted from [Ghezzi et al. \(2003\)](#). The mathematical notation comes from Chapter 3 of [Hoffman and Strooper \(1995\)](#). For instance, the symbol $:=$ is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | \dots | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by OAR.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	\mathbf{Z}	a number without a fractional component in $(-\infty, \infty)$
positive integer	\mathbf{Z}_+	a positive integer (\mathbf{Z}) in $(0, \infty)$
unsigned 8-bit integer	\mathbf{U}	a number without a fractional component in $(0, 255)$
natural number	\mathbf{N}	a number without a fractional component in $[1, \infty)$
real	\mathbf{R}	any number in $(-\infty, \infty)$
positive real	\mathbf{R}_+	any real number (\mathbf{R}) in $(0, \infty)$
image data	$\mathbf{I}_{x,y}$	data: a one-dimensional array of unsigned 8-bit integers in order from the top-left pixel of the image to the bottom-right pixel. Has a width: \mathbf{Z}_+ width of x and height: \mathbf{Z}_+ height of y .
matrix	$\mathbf{M}_{x,y}$	data: a one-dimensional array of real numbers, with a width: \mathbf{Z}_+ width of x and height: \mathbf{Z}_+ height of y .

The specification of OAR uses some derived data types: sequences, strings, tuples, and booleans. Sequences are lists filled with elements of the same data type. Strings are sequences

of characters. Tuples contain a list of values, potentially of different types. Booleans can be represented in different ways but only have two possible values: true or false. In addition, OAR uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	Level 2
Hardware-Hiding Module	
Behaviour-Hiding Module	Application Control Output Module Input Data Read Module Input Classifier Module OAR Model Data Module OAR Model Equations Module OAR Model Training Module OAR Model Testing Module
Software Decision Module	Confusion Matrix Module Input Processing Module Graphical User Interface

Table 1: Module Hierarchy

6 MIS of Application Control Module

6.1 Module

main

6.2 Uses

- Graphical User Interface (GUI) Module Specification ([16](#))

6.3 Syntax

6.3.1 Exported Constants

None.

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
main	-	-	-

6.4 Semantics

6.4.1 State Variables

None.

6.4.2 Environment Variables

None.

6.4.3 Assumptions

The GUI is running and displayed without issue.

6.4.4 Access Routine Semantics

main():

- transition: Initializes the program and the GUI module [16](#)

6.4.5 Local Functions

None.

7 MIS of Output Module

7.1 Module

output

7.2 Uses

- Input Data Read Module Specification (8)
- Input Classifier Module Specification (9)

7.3 Syntax

7.3.1 Exported Constants

- DEC_FIXED: Used for fixed decimal number length rounding (ex. "5.8923" at fixed length "2" results in "5.89")

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
output	-	inputImage ($\mathbf{I}_{x,y}$), resultLabel (String), resultConf (String)	-

7.4 Semantics

7.4.1 State Variables

None.

7.4.2 Environment Variables

None.

7.4.3 Assumptions

The input image is valid.

7.4.4 Access Routine Semantics

output():

- output: The inputImage as ($\mathbf{I}_{x,y}$) and the predicted label resultLabel and confidence in the classification resultConf as strings, to be used by the GUI module 16.

7.4.5 Local Functions

None.

8 MIS of Input Data Read Module

8.1 Module

input

8.2 Uses

- Input Processing Module Specification ([15](#))

8.3 Syntax

8.3.1 Exported Constants

- **MAX_SIZE**: A value (\mathbf{Z}_+) describing both width and height (in pixels) for maximum acceptable size of the input image (currently a square).
- **MIN_SIZE**: A value (\mathbf{Z}_+) describing both width and height (in pixels) for minimum acceptable size of the input image (currently a square).
- **MODEL_IMG_SIZE**: The required size of the input image matrix to be used by the classification model $\mathbf{I}_{x,y}$.

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
input	inputPath (String)	inputImage ($\mathbf{I}_{x,y}$)	InvalidSize, InvalidFormat

8.4 Semantics

8.4.1 State Variables

None.

8.4.2 Environment Variables

- **inputPath**: the File System path or location as a string pointing to where the base input image is located.

8.4.3 Assumptions

The input path location is valid, readable and accessible.

8.4.4 Access Routine Semantics

input(inputPath):

- output: Pre-processed `inputImage` as $\mathbf{I}_{x,y}$ ready for classification.
- exception: `InvalidSize` if the size of the base image is outside of the range of `MIN_SIZE` - `MAX_SIZE`, `InvalidFormat` if the file type at the `inputPath` is not supported by the OAR Program (according to [R1](#))

8.4.5 Local Functions

None.

9 MIS of Input Classifier Module

9.1 Module

`classify`

9.2 Uses

- OAR Model Data Module Specification ([10](#))
- OAR Model Equations Module Specification ([11](#))

9.3 Syntax

9.3.1 Exported Constants

None.

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
<code>classify</code>	<code>inputImage</code> ($\mathbf{I}_{x,y}$), <code>oarModel</code> ($\mathbf{M}_{x,y}$)	<code>resultLabel</code> (String), <code>confPercent</code> \mathbf{R}_+	-

9.4 Semantics

9.4.1 State Variables

None.

9.4.2 Environment Variables

None.

9.4.3 Assumptions

The input image is valid.

9.4.4 Access Routine Semantics

`classify(inputImage):`

- output: The predicted label of the input image `resultLabel` as a String and the associated confidence level in the prediction as a \mathbf{R}_+ .

9.4.5 Local Functions

None.

10 MIS of OAR Model Data Module

10.1 Module

`model`

10.2 Uses

- OAR Model Testing Module Specification ([13](#))

10.3 Syntax

10.3.1 Exported Constants

None.

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
<code>model</code>	-	-	-

10.4 Semantics

10.4.1 State Variables

- `oarModel`: Data structure designed to store the matrix of weights and biases associated with the trained OAR classification model as a tuple of $\mathbf{M}_{x,y}$ and \mathbf{R} .

10.4.2 Environment Variables

None.

10.4.3 Assumptions

None.

10.4.4 Access Routine Semantics

`model()`:

- `transition`: This module is a simple tuple ($\mathbf{M}_{x,y}$ and \mathbf{R}) data structure for storing the OAR classification model weights and biases.

10.4.5 Local Functions

None.

11 MIS of OAR Model Equations Module

11.1 Module

`oarUtils`

11.2 Uses

None.

11.3 Syntax

11.3.1 Exported Constants

None.

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
sigmoid	sigIn (\mathbf{R})	sigOut (\mathbf{R})	-
logLossFunc	trueVal (\mathbf{R}), predVal (\mathbf{R})	logLoss (\mathbf{R})	-
predict	inputImage ($\mathbf{I}_{x,y}$), weight ($\mathbf{M}_{x,y}$), bias (\mathbf{R})	predVal (\mathbf{R})	-
gradientW	inputImage ($\mathbf{I}_{x,y}$), trueVal (\mathbf{R}), weight ($\mathbf{M}_{x,y}$), bias (\mathbf{R}), regParam (\mathbf{R}), trainSize (\mathbf{Z}_+)	gradW (\mathbf{R})	-
gradientB	inputImage ($\mathbf{I}_{x,y}$), trueVal (\mathbf{R}), weight ($\mathbf{M}_{x,y}$), bias (\mathbf{R})	gradB (\mathbf{R})	-
gradientDescent	inputImage ($\mathbf{I}_{x,y}$), trueVal (\mathbf{R}), weight ($\mathbf{M}_{x,y}$), bias (\mathbf{R}), regParam (\mathbf{R}_+), learnRate (\mathbf{R}_+), trainSize (\mathbf{Z}_+)	weight ($\mathbf{M}_{x,y}$), bias (\mathbf{R})	-

11.4 Semantics

11.4.1 State Variables

None.

11.4.2 Environment Variables

None.

11.4.3 Assumptions

The input image is valid.

11.4.4 Access Routine Semantics

sigmoid(sigIn):

- output: Computes the sigmoid function of sigIn according to TM1 and returns the value sigOut as \mathbf{R} .

logLossFunc(trueVal, predVal):

- output: Computes the value of the log loss function using `trueVal` and `predVal` according to TM2 and returns the value `logLoss` as \mathbf{R} .

`predict(inputImage, weight, bias):`

- output: Calls `sigmoid` to calculate the predicted label of the `inputImage` using `weight` and `bias` from the model according to GD1.

`gradientW(inputImage, trueVal, weight, bias, regParam, trainSize):`

- output: Calculates the gradient of the log loss function using all the input variables with respect to the weights according to according to GD2.

`gradientB(inputImage, trueVal, weight, bias):`

- output: Calculates the gradient of the log loss function using all the input variables with respect to the bias according to according to GD3.

`gradientDescent(inputImage, trueVal, weight, bias, regParam, learnRate, trainSize):`

- output: Calls `gradientW` and `gradientB` to execute the algorithm to update the weights and biases using all the input variables for one epoch according to IM1.

11.4.5 Local Functions

None.

12 MIS of OAR Model Training Module

12.1 Module

`train`

12.2 Uses

- OAR Model Equations Module Specification (11)

12.3 Syntax

12.3.1 Exported Constants

- LAMBDA: The regularization parameter used during model training as \mathbf{R}_+ .
- ALPHA: The learning rate parameter used during model training as \mathbf{R}_+ .

12.3.2 Exported Access Programs

Name	In	Out	Exceptions
train	trainSet ($\mathbf{M}_{x,y}$), trainVals ($\mathbf{M}_{x,y}$), weightBiasMatrix (tuple of $\mathbf{M}_{x,y}$ and \mathbf{R}), trainSize (\mathbf{Z}_+)	weightBiasMatrix (tuple of $\mathbf{M}_{x,y}$ and \mathbf{R})	-

12.4 Semantics

12.4.1 State Variables

- **weight**: the weight portion of the **weightBiasMatrix** input as $\mathbf{M}_{x,y}$ for each label.
- **bias**: the bias portion of the **weightBiasMatrix** input as \mathbf{R} for each label.

12.4.2 Environment Variables

None.

12.4.3 Assumptions

None.

12.4.4 Access Routine Semantics

train(trainSet, trainVals, weightBiasMatrix, trainSize):

- **output**: Executes the algorithm for training the weights and biases of the model given the training dataset information and functions from the OAR Model Equations Module (11), and returns the updated version of the **weightBiasMatrix** as a tuple of $\mathbf{M}_{x,y}$ and \mathbf{R} .

12.4.5 Local Functions

None.

13 MIS of OAR Model Testing Module

13.1 Module

test

13.2 Uses

- OAR Model Data Module Specification (11)
- OAR Model Equations Module Specification (11)
- OAR Model Training Module Specification (12)
- Confusion Matrix Module Specification (14)

13.3 Syntax

13.3.1 Exported Constants

- EPOCHS: The the number of times the model training regression algorithm is ran as \mathbf{Z}_+ .
- TRAIN_SIZE: The size of the training data used during model training as \mathbf{Z}_+ .
- TEST_SIZE: The size of the testing data used during model training as \mathbf{Z}_+ .
- DATA_IMG_SIZE: The size of the input image matrix used during model training as $\mathbf{I}_{x,y}$.
- LABELS: The set of possible labels for the classification model as a tuple of Strings.

13.3.2 Exported Access Programs

Name	In	Out	Exceptions
test	-	oarModel (tuple of $\mathbf{M}_{x,y}$ and \mathbf{R})	-

13.4 Semantics

13.4.1 State Variables

- **dataSet**: The set of pre-processed images and their associated labels that will be used for training the classification model as a tuple of $\mathbf{I}_{x,y}$ and \mathbf{Z}_+ .
- **weight**: the weight portion of the **weightBiasMatrix** input as $\mathbf{M}_{x,y}$ for each label.
- **bias**: the bias portion of the **weightBiasMatrix** input as \mathbf{R} for each label.
- **predictionData**: matrix which tracks the number predictions made for each test image as $\mathbf{M}_{x,y}$.

13.4.2 Environment Variables

- **dataSetPath**: the File System path or location as a string pointing to where the data set is located.

13.4.3 Assumptions

The `dataSetPath` is valid, readable and accessible.

13.4.4 Access Routine Semantics

`train()`:

- output: The matrix of weights and biases representing the `oarModel` as $\mathbf{M}_{x,y}$.

13.4.5 Local Functions

- `splitDataSet(dataSet)`:
 - output: Takes the `dataSet` as an input and splits it into distinct parts for training and testing the classification model. The following values are output:
 - * `trainData`: The part of the `dataSet` used to train the model as $\mathbf{M}_{x,y}$.
 - * `trainVals`: The part of the `dataSet` corresponding to the true labels of the `trainData` as $\mathbf{M}_{x,y}$.
 - * `testData`: The part of the `dataSet` used to test the model as $\mathbf{M}_{x,y}$.
 - * `testVals`: The part of the `dataSet` corresponding to the true labels of the `testData` as $\mathbf{M}_{x,y}$.
- `evalModelData(oarModel, confusionMatrix)`:
 - output: Evaluates the performance of the `oarModel` based on the `confusionMatrix` and returns the values that should be written to the OAR Model Data Module [10](#).

14 MIS of Confusion Matrix Module

14.1 Module

`confMatrix`

14.2 Uses

None.

14.3 Syntax

14.3.1 Exported Constants

None.

14.3.2 Exported Access Programs

Name	In	Out	Exceptions
confMatrix	predictionData ($\mathbf{M}_{x,y}$)	confusionMatrix ($\mathbf{M}_{x,y}$)	
printConfMatrix	confusionMatrix ($\mathbf{M}_{x,y}$)	confusionMatrix ($\mathbf{I}_{x,y}$)	

14.4 Semantics

14.4.1 State Variables

None.

14.4.2 Environment Variables

None.

14.4.3 Assumptions

None.

14.4.4 Access Routine Semantics

confMatrix(predictionData):

- output: Outputs the confusion matrix representing the performance of the model based on the predictionData as $\mathbf{M}_{x,y}$.

printConfMatrix(confusionMatrix):

- output: Outputs the confusion matrix representing the performance of the model based on the predictionData as a graphical image ($\mathbf{I}_{x,y}$).

14.4.5 Local Functions

None.

15 MIS of Input Processing Module

15.1 Module

preprocess

15.2 Uses

None.

15.3 Syntax

15.3.1 Exported Constants

None.

15.3.2 Exported Access Programs

Name	In	Out	Exceptions
preprocess	baseImage ($\mathbf{I}_{x,y}$)	inputImage ($\mathbf{I}_{x,y}$)	-

15.4 Semantics

15.4.1 State Variables

None.

15.4.2 Environment Variables

None.

15.4.3 Assumptions

The format and parameters of the base image was already verified to be within the requirements.

15.4.4 Access Routine Semantics

preprocess(baseImage):

- output: Performs transformations on the **baseImage** such that the resulting **inputImage** as $\mathbf{I}_{x,y}$, is normalized to be able to be used by the classification model.

15.4.5 Local Functions

None.

16 MIS of Graphical User Interface

16.1 Module

gui

16.2 Uses

- Hardware-Hiding Module
- Output Module (7)

16.3 Syntax

16.3.1 Exported Constants

- GUI_BOXSIZE: A value (\mathbf{Z}_+) describing both width and height (in pixels) used for the image display "box" (currently always a square)

16.3.2 Exported Access Programs

Name	In	Out	Exceptions
gui	inputImage ($\mathbf{I}_{x,y}$), resultLabel (String), resultConf (String)	displayWindow, event handlers	-

16.4 Semantics

16.4.1 State Variables

- inputImage: The processed input image and given by the Output Module 7 as $\mathbf{I}_{x,y}$
- resultLabel: The label output as given by the Output Module 7 as a string.
- resultConf: The confidence probability output as given by the Output Module 7 as a string.

16.4.2 Environment Variables

- Keyboard (\mathbf{Z}_+ for keycodes describing the key pressed)
- Mouse (Boolean for click state and \mathbf{Z}_+ for cursor position)
- Screen (\mathbf{Z}_+ for width and height in pixels)
- Button (String for a file location) to provide an input image from the file system

16.4.3 Assumptions

- The file system is able to read and provide the image file as specified by the user through an OS file-open dialog. Otherwise if the file is not found, denied access or cancelled, no changes should occur.
- The OS is able to provide basic text or number input user controls with some basic built-in validation, and is able to handle events from Human Interface Devices (HIDs such as mouse, keyboard or touchscreen).

16.4.4 Access Routine Semantics

`gui()`:

- `transition`: Sets up user control event handlers (i.e., mouse clicks or drag, button presses, text input change, ...) as needed for the user input. Calls the Output Module (7) to classify the input image. The input image and output results are then pushed to the `displayWindow`.

16.4.5 Local Functions

None.

References

- Hunter Ceranic. System requirements specification. <https://github.com/cer-hunter/OAR-CAS741/blob/main/docs/SRS/SRS.pdf>, 2024.
- Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.
- Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL <http://citeseer.ist.psu.edu/428727.html>.