Module Interface Specification for OCRacle

Phillip Tran

March 20, 2025

1 Revision History

Date	Version	Notes
March 13, 2025	1.0	Initial Document Creation

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at https://github.com/ptrandev/OCRacle/blob/main/docs/SRS/SRS.pdf for symbols, abbreviations and acronyms.

Contents

3 Introduction

The following document details the Module Interface Specifications for OCRacle, an optical character recognition (OCR) program for identifying Latin alphabet characters.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/ptrandev/OCRacle/.

4 Notation

[You should describe your notation. You can use what is below as a starting point. —SS]

The structure of the MIS for modules comes from **?**, with the addition that template modules have been adapted from **?**. The mathematical notation comes from Chapter 3 of **?**. For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | ... | c_n \Rightarrow r_n)$.

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

Data Type	**Notation**	**Description**
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)
bounded real	$\mathbb{R}_{\text{min} \leq x \leq \text{max}}$	a real number in the range [min, max]
bounded integer	$\mathbb{Z}_{\text{min} \leq x \leq \text{max}}$	an integer in the range [min, max]

The specification of OCRacle uses some derived data types: sequences, strings, and tuples. 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. In addition, OCRacle 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	Input Format Module Model Output Module Model Training Module Model Testing Module Prediction Model Module Application Module Image Preprocessing Module Accuracy Metrics Module
Software Decision Module	Graphical User Interface Module

Table 1: Module Hierarchy

6 MIS of Application Module

6.1 Module

main

6.2 Uses

- Graphical User Interface Module ??
- Input Format Module ??
- Model Output Module ??

6.3 Syntax

6.3.1 Exported Constants

N/A

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
main	-	-	-

6.4 Semantics

6.4.1 State Variables

N/A

6.4.2 Environment Variables

N/A

6.4.3 Assumptions

The Graphical User Interface module is responsible for handling all user inputs and outputs. The Application module is responsible for coordinating the interaction between the GUI, Input Format, and Model Output modules.

6.4.4 Access Routine Semantics

main():

- transition: The application is started and the user is able to interact with the GUI to input images and view the model's predictions.

6.4.5 Local Functions

N/A

7 MIS of Input Format Module

7.1 Module

input

7.2 Uses

- Image Preprocessing Module ??

7.3 Syntax

7.3.1 Exported Constants

N/A

7.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
input	filePath: string	imageMatrix: $\mathbb{Z}^{28\times28}_{0\leq x\leq 255}$	DimensionsError, FormatError, FileNotFound

7.4 Semantics

7.4.1 State Variables

N/A

7.4.2 Environment Variables

- filePath: string. The path to the input image file from the user.

7.4.3 Assumptions

N/A

7.4.4 Access Routine Semantics

input(filePath):

- output: imageMatrix := preprocessing(filePath). After ensuring that all exceptions are handled, call the preprocessing module **??** to perform the necessary transformations on the input image.
- exceptions:
  - A DimensionsError exception is thrown if the input image exceeds the bounds of the n_{min}, n_{max}, m_{min}, and m_{max} parameters as specified by Table 4 in the SRS (**?**, 4.2.6).
  - A FormatError exception is thrown if the input image is not one of the supported formats specified by R1 in the SRS (**?**, 5.1.1).
  - A FileNotFound exception is thrown if the input file does not exist.

7.4.5 Local Functions

N/A

8 MIS of Model Output Module

8.1 Module

output

8.2 Uses

- Prediction Model Module **??**

8.3 Syntax

8.3.1 Exported Constants

- LABELS: {'A',...,'Z'}

8.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
output	imageMatrix: $\mathbb{Z}^{28\times28}_{0\leq x\leq255}$	prediction: char, confidenceMatrix: $\mathbb{R}^{26}_{0\leq \mathrm{x}\leq1}$	-

8.4 Semantics

8.4.1 State Variables

N/A

8.4.2 Environment Variables

N/A

8.4.3 Assumptions

We assume that the imageMatrix input has been processed by the Input Format Module **??** before being passed to the Model Output Module.

8.4.4 Access Routine Semantics

output(imageMatrix):

- output:
  - confidenceMatrix := predict(imageMatrix). Call the prediction model module **??** to generate a confidence matrix for the input image.
  - prediction := LABELS[argmax(confidenceMatrix)]. Return the character representing the highest confidence value in the confidence matrix.

9 MIS of Model Training Module

9.1 Module

train

9.2 Uses

- Input Format Module **??**
- Model Output Module **??**
- Prediction Model Module **??**
- Accuracy Metrics Module **??**

9.3 Syntax

9.3.1 Exported Constants

N/A

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
train	-	model: $\mathbb{R}^{m \times n}$	-

9.4 Semantics

9.4.1 State Variables

- dataset: $(\mathbb{Z}^{28\times 28}_{0\leq x\leq 255}$, char)[]
- epochs: $\mathbb{N}$
- batch_size: $\mathbb{N}$
- learning_rate: $\mathbb{R}$
- epsilon: $\mathbb{R}$
- time_step: $\mathbb{R}$
- first_moment: $\mathbb{R}$
- second_moment: $\mathbb{R}$
- beta1: $\mathbb{R}$
- beta2: $\mathbb{R}$
- gradient: $\mathbb{R}^{m\times n}$
- theta: $\mathbb{R}^{m\times n}$
- y_hat: $\mathbb{R}^{m\times n}$

9.4.2 Environment Variables

N/A

9.4.3 Assumptions

N/A

9.4.4 Access Routine Semantics

train():

- transition: The model is trained on the dataset using the hyperparameters epochs, batch_size, learning_rate, epsilon, time_step, first_moment, second_moment, beta1, and beta2. These are manipulated via the ADAM optimizer as specified by the SRS.
- output: model := $f_{train}()$

9.4.5 Local Functions

relu(x):

- output: result := max(0, x)

softmax(x):

- output: result := $e^x / \sum_{i=1}^{n} e^{x_i}$

cnn_forward(x, theta):

- transition: y_hat := $f_{cnn\_forward}(x, theta)$. This function calculates the output of the model given the input x and the weights theta.

cnn_backward(x, y, y_hat, theta):

- transition: gradient := $f_{cnn\_backward}(x, y, y\_hat, theta)$. This function calculates the gradient of the model given the input x, the target y, the output y_hat, and the weights theta.

updateTheta(gradient, theta, time_step, first_moment, second_moment, beta1, beta2, epsilon):

- transition: theta := $f_{updateTheta}(gradient, theta, time\_step, first\_moment, second\_moment, beta1, beta$ This function updates the weights of the model using the ADAM optimizer.

10 MIS of Model Testing Module

10.1 Module

test

10.2 Uses

- Accuracy Metrics Module **??**
- Model Output Module **??**
- Input Format Module **??**

10.3 Syntax

10.3.1 Exported Constants

N/A

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
test	-	loss: $\mathbb{R}_{0 \leq x \leq 1}$, accuracy: $\mathbb{R}_{0 \leq x \leq 1}$, confusionMatrix: ${\mathbb{R}_{0 \leq x \leq 1}}^{26 \times 26}$	-

10.4 Semantics

10.4.1 State Variables

- dataset: $(\mathbb{Z}_{0 \leq x \leq 255}^{28 \times 28}$, char)[]. The dataset of images and their corresponding labels used for testing the model.
- loss: $\mathbb{R}_{0 \leq x \leq 1}$. The cross-entropy loss of the model, which is a measure of the model's performance.
- confusionMatrix: ${\mathbb{R}_{0 \leq x \leq 1}}^{26 \times 26}$. A matrix that shows the model's predictions versus the actual labels.

10.4.2 Environment Variables

N/A

10.4.3 Assumptions

Assume that the model has been trained using the Model Training Module **??**.

10.4.4 Access Routine Semantics

test():

- transition: The model is tested on a dataset of unseen images. The loss, and confusion matrix are calculated using the model's predictions.
- output:
  - loss := crossEntropyLoss(prediction, target). The prediction from the model is compared to the true target values from the dataset. The Accuracy Metrics Module **??** is used to calculate the cross-entropy loss.
  - confusionMatrix := confusionMatrix(prediction, target). The prediction from the model is compared to the true target values from the dataset. The Accuracy Metrics Module **??** is used to calculate the confusion matrix.

10.4.5 Local Functions

N/A

11 MIS of Prediction Model Module

11.1 Module

model

11.2 Uses

11.3 Syntax

11.3.1 Exported Constants

N/A

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
predict	imageMatrix: $\mathbb{Z}^{28\times 28}_{0\leq x\leq 255}$	confidenceMatrix: $\mathbb{R}^{26}_{0\leq \mathrm{x}\leq 1}$	-

11.4 Semantics

11.4.1 State Variables

- model: $\mathbb{R}^{m\times n}$. The representation of the model's weights for the prediction model.

11.4.2 Environment Variables

N/A

11.4.3 Assumptions

We assume that the input image has been preprocessed as specified in the SRS before being passed to the prediction model.

11.4.4 Access Routine Semantics

predict(imageMatrix):

- output: confidenceMatrix. Where each element in the confidence matrix represents the model's confidence in the corresponding character. The model is used to generate a confidence matrix for the input image using a CNN as specified by GD3 in the SRS (**?**, 4.2.3). GD3 provides a high-level description of the model's architecture, but the implemented model may differ.

11.4.5 Local Functions

N/A

12 MIS of Image Preprocessing Module

12.1 Module

preprocessing

12.2 Uses

12.3 Syntax

12.3.1 Exported Constants

N/A

12.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
preprocessing	file : string	imageMatrix: $\mathbb{Z}^{28\times 28}_{0\leq x\leq 255}$	-

12.4 Semantics

12.4.1 State Variables

N/A

12.4.2 Environment Variables

N/A

12.4.3 Assumptions

All input data has been validated by the Input Format Module before being processed by the Image Preprocessing Module.

12.4.4 Access Routine Semantics

preprocessing(file):

- output: imageMatrix := normalization(bicubicInterpolation(file)). The input file has been transformed such that it conforms to the preprocessing requirements as specified by A3 in the SRS (**?**, 4.2.1).

12.4.5 Local Functions

bicubicInterpolation(imageMatrix):

- output: Perform bicubic interpolation such that the imageMatrix is transformed to a 28x28 matrix. as specified by TM:BI in the SRS (**?**, 4.2.2).

normalization(imageMatrix):

- output: Perform normalization such that the image is turned into a greyscale image with pixel values in the range [0, 255]. as specified by TM:N in the SRS (**?**, 4.2.2).

13 MIS of Accuracy Metrics Module

13.1 Module

performance

13.2 Uses

13.3 Syntax

13.3.1 Exported Constants

N/A

13.3.2 Exported Access Programs

Name	In	Out	Exceptions
crossEntropyLoss	prediction: $\mathbb{R}^{26}_{0\leq \mathrm{x}\leq 1}$, target: $\mathbb{R}^{26}_{0\leq \mathrm{x}\leq 1}$	loss: $\mathbb{R}_{0\leq \mathrm{x}\leq 1}$	DimensionsError
confusionMatrix	prediction: $\mathbb{R}^{26}_{0\leq \mathrm{x}\leq 1}$, target: $\mathbb{R}^{26}_{0\leq \mathrm{x}\leq 1}$	confusionMatrix: ${\mathbb{R}_{0\leq \mathrm{x}\leq 1}}^{26\times 26}$	DimensionsError

13.4 Semantics

13.4.1 State Variables

N/A

13.4.2 Environment Variables

N/A

13.4.3 Assumptions

N/A

13.4.4 Access Routine Semantics

crossEntropyLoss(prediction, target):

- output: loss := $-\sum_{i=1}^{n} target_i \log(prediction_i)$. The cross entropy loss is calculated for the model's predictions and the target values as specified by TM:CELF in the SRS (**?**, 4.2.2).
- exceptions:
  - A DimensionsError exception is thrown if the dimensions of the prediction and target matrices do not match.

confusionMatrix(prediction, target):

- output: confusionMatrix := $\text{confusionMatrix}_{i,j} = \sum_{k=1}^{n} \text{target}_{k,i} \times \text{prediction}_{k,j}$. A confusion matrix is generated for the model's predictions and the target values to fulfill NFR1 in the SRS (**?**, 5.2).
- exceptions:
  - A DimensionsError exception is thrown if the dimensions of the prediction and target matrices do not match.

13.4.5 Local Functions

N/A

14 MIS of Graphical User Interface Module

14.1 Module

gui

14.2 Uses

- Hardware-Hiding Module

14.3 Syntax

14.3.1 Exported Constants

N/A

14.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
-	-	-	-

14.4 Semantics

14.4.1 State Variables

N/A

14.4.2 Environment Variables

- displayWindow (dimensions: $\mathbb{Z}_+ \times \mathbb{Z}_+$)
- keyboard (keypress: char)
- mouse (location : $\mathbb{Z}_+ \times \mathbb{Z}_+$, click: bool)

14.4.3 Assumptions

This behavior will be sufficiently abstracted by the use of Python Notebook, which will handle all UI elements and interactions with the user. As a result, the GUI modules does not need to have any pre-defined state variables or access routines.

14.4.4 Access Routine Semantics

N/A

14.4.5 Local Functions

N/A

References

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.

Phillip Tran. System requirements specification. https://github.com/ptrandev/OCRacle/blob/main/docs/SRS/SRS.pdf, 2025.

15 Appendix

[Extra information if required —SS]