

Configuration Manual

MSc Research Project
MSc in AI for Business

Luis Antonio Reyna Torres
Student ID: x23148802

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Luis Antonio Reyna Torres
Student ID:	x23148802
Programme:	MSc in AI for Business
Year:	2024
Module:	MSc Research Project
Supervisor:	Rejwanul Haque
Submission Due Date:	12/08/2024
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	3

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	11th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Luis Antonio Reyna Torres
x23148802

1 Systems Requirements

The model was developed using Python as the developing language in the Atom IDE. It was developed in a local environment in a Macbook with the following characteristics and versions.

Macbook Pro	2.3 GHz Dual-Core Intel Core i5
Python	3.6
Atom	1.60.0

Table 1: Developing Environment Versions

2 Environment Requirements

Due to the Convolutional Neural Network developed performs for two different environments, it was required to have a specific version of the frameworks imported in the classes in order to achieve a stable interaction between both environments. The versions of frameworks and libraries are detailed in the following table.

TensorFlow	2.12.0
coremltools	7.2

Table 2: Libraries imported in the CNN model.

The rest of the libraries imported in the classes such as deeplake, numpy, scikit-learn, opencv-python (cv2), and matplotlib were installed and imported on their last version.

3 Identify Files

The folder contains the following files:

- facialExpression.py
- predictionEmotion.py
- converterML.py
- 1 folder named models

- 7 test images of different facial expressions.

The class *facialExpression.py* contains the Convolutional Neural Network architecture and the code to obtain the dataset and train the CNN. The dataset implemented to train and test the model is obtained through *deeplake* request so it is not necessary to have a physical dataset. The class *predictionEmotion.py* loads the model trained and contains functions to load and preprocess one test image and deliver a resolution among the facial expression prediction. The class *converterML.py* contains the function to load the trained model and convert it to a file with extension *.mlpackage* which is a framework to be implemented in iOS environment. The folder named *models* contains the models already trained when this research was studied

4 Train the Model

1. Open a new session in the Terminal application
2. Go to the folder name *faceExpression* using the command

```
cd path/to/faceExpression
```

3. Install the libraries required in the project with the command

```
python3 pip install coremltools numpy deeplake scikit-learn opencv-python matplotlib
```

For tensorflow library, it is necessary to install the library manually from the URL: <https://github.com/tensorflow/tensorflow/releases/tag/v2.12.0>

4. Train the CNN with the command:

```
python3 facialExpression.py
```

This step produces a file named *emotion_detection_model.h5* which is the model saved in *keras* extension.

5 Test the Model

1. Specify the image to analyze in the class *predictionEmotion.py*. In line 10 inside the function *imread()* write the name of the image and its extension. Before this, ensure the image is located in the fold at the same level of the class *predictionEmotion.py*
2. Test the model running the command:

```
python3 predictionEmotion.py
```

The result of this step will produce a prediction from the image analyzed based on the 7 basic emotions.

6 Convert the model to CoreML model

1. Ensure the model is correctly written in line 7 within the class converterML.py
2. Run the following command to convert the model:

```
python3 converterML.py
```

This step produces a new file named emotionDetection.mlpackage which is the framework to be used in the iOS environment.