# Introduction

This Framework is intended to be use as base for your project.

It contains 2 types of Main files:

* **Main-UploadDataset** – To be used as an application to upload photos to the dataset, which then will be used to train the ML package.
* **Main-Identify** – To be used to identify the user after the ML package has been trained and made available.

# Prerequisites

In order to run the framework, please **install Python 3.6** on the machine where the robot is running.

You can use the “install.bat” inside the Prerequisites folder to install the additional packages to run the scripts.

**Make sure to update the Python Path in the Config.xlsx file.**

# ML Instructions

Please follow the steps below in order. *Make sure you have python and required packages installed.*

## Create Project

1. Go to AI Centre and create a new project.

Graphical user interface, text, application

Description automatically generated

## Create Dataset

1. Open the project create on step 1.
2. Go to Datasets
3. Create new dataset

Graphical user interface, application

Description automatically generated

1. Make dataset public

Graphical user interface, text, application, email

Description automatically generated

1. Click ‘Create’
2. Copy the Endpoint and the API Key

Graphical user interface, application

Description automatically generated

1. Create one Credential asset in Orchestrator for the API Key and one text asset for the Endpoint. Make sure to update the Settings and Assets tabs in the Config file with the names of the assets.

## Upload dataset

You can either upload a dataset manually or use the Framework to create one.

**At least 2 people/folder need to be uploaded to the dataset for the training.**

### Upload dataset manually

Make sure you follow the correct folder hierarchy.

Folder hierarchy:

* Dataset *(Folder)*
  + Training *(Folder)*
    - Name1 *(Folder) - Place photos from Name1 here*
    - Name2 *(Folder) - Place photos from Name1 here*
    - Name3 *(Folder) - Place photos from Name1 here*
    - …
  + Evaluation *(Folder)*
    - Name1 *(Folder) - Place photos from Name1 here*
    - Name2 *(Folder) - Place photos from Name1 here*
    - Name3 *(Folder) - Place photos from Name1 here*
    - …

### Upload using the Framework

The first step of the framework (Main-UploadDataset.xaml) robot is to create a dataset. This will be done by taking pictures using the computer camera and then uploading them to the dataset created on Step 2.

**Make sure the Config files are updated with the dataset endpoint and api key.**

## Create ML Package

1. Open the AI Centre project.
2. Go to ML Packages.
3. Go to Upload zip file
4. Upload the file FaceRecognition.zip from the Requirements folder in the framework.
   1. Make sure to select input type as File
   2. Make sure to select Python 38 OPENCV as Language
   3. Make sure to Enable Training

Graphical user interface, application

Description automatically generated

1. Click on ‘Create’.
2. Wait until the validation is completed.
3. The status of the ML Package should be ‘Undeployed’.

## Train

1. Open the AI Centre project.
2. Got to Pipelines.
3. Create a new Full pipeline run, or a Train run if you don’t need the evaluation.
4. Choose the ML Package uploaded on [Step 4](#_Create_ML_Package) and select the latest version.
5. For the Input dataset choose the Training dataset. This will be the **Training folder** inside the dataset created on [Step 2](#_Create_Dataset).
6. For the evaluation dataset choose the **Evaluation folder** in the dataset created on [Step 2](#_Create_Dataset).

Graphical user interface, application

Description automatically generated

1. Wait until the training is completed. This may take some time.
2. The status of the pipeline should be ‘**Successful’**.

## Create ML Skill

* 1. Open the AIC project.
  2. Go to ML Skills and click on ‘Create new’.
  3. Select the ML package trained on [Step 5](#_Train).
     1. Select the latest major and minor version
  4. Click on ‘Create’
  5. Wait until the skill is completed. The final status should be ‘Available’.
  6. Open the new skill.
  7. Click on ‘Modify current deployment’.
  8. Make ML Skill Public.

Graphical user interface, text, application, chat or text message

Description automatically generated

* 1. Click ‘Confirm’.
  2. Wait until the changes are completed. The final status should be ‘Available’.
  3. Copy the Url and the API Key form the ML Skill

Graphical user interface, text, application, email

Description automatically generated

* 1. Create a text asset in Orchestrator for the Url (Endpoint) and a credential asset for the api key. **Make sure to update the Config file with the asset names.**

## Identify

This can be done using the framework (Main-Identify.xaml), and it uses the ML Skill created on Step 6 to recognize the person from the photo/camera.