

# User Manual for OCR-Based Manufacturing Equipment Display Reader

**Author:** Diego Lafeuille

**Date:** 15.08.2023

## Inhaltsverzeichnis

|   |    |
|---|----|
| Introduction.....                             | 3  |
| System requirements .....                     | 3  |
| Setup.....                                    | 4  |
| Software installation .....                   | 4  |
| Cloning the Git repository .....              | 4  |
| Installing Python .....                       | 4  |
| Using a virtual environment .....             | 5  |
| Installing the required python libraries..... | 5  |
| Installing the camera SDK.....                | 6  |
| Installing the Tesseract model.....           | 6  |
| Hardware setup .....                          | 6  |
| Setting Up the Charuco Board .....            | 7  |
| Camera Positioning.....                       | 7  |
| Lens Selection .....                          | 9  |
| Other considerations.....                     | 10 |
| Getting Started .....                         | 11 |
| Launching the Application .....               | 11 |
| Overview of the GUI .....                     | 11 |
| Main Window .....                             | 11 |
| Indicate target surface window.....           | 13 |
| Workflow .....                                | 13 |
| Retrieving Stored Data from Database .....    | 16 |
| Data retrieval.....                           | 16 |
| Data post-processing.....                     | 16 |

## Introduction

The OCR-Based Manufacturing Equipment Display Reader is a tool designed to extract and record values from legacy manufacturing equipment displays (extruders for example). The primary objective of this software is to capture data points such as temperature, pressure, torque, etc., which aren't digitally accessible to us, and store them in a database. This is achieved by utilizing a camera to visually record the display and then using Optical Character Recognition (OCR) technology to interpret and catalog the data. An essential feature of this system is its ability to recalculate the position in the captured pictures of the looked for data. As production environment can be hectic, this ensures that the measurement can continue without a need to be recalibrated if someone shoves the camera inadvertently. This manual will guide users on how to effectively use and maximize the benefits of this software.

## System requirements

[Section should be expanded through further testing]

Works with Windows 10, should be tested on Linux.

A CUDA compatible GPU speeds up dramatically the time needed for the OCR function, especially the “Best” detection method, but is not necessary for the correct operation of the code. Here is a [tutorial for the correct installation of CUDA for Windows 10](#). If you wish to leverage a GPU, before the installation step [Installing the required python libraries](#), you will need to open the requirements.txt document, and uncomment the commented lines (remove the “#” at the beginning of the line and save the file).

```
# tensorboard==2.11.2
# tensorboard-data-server==0.6.1
# tensorboard-plugin-wit==1.8.1
# tensorflow==2.11.0
# tensorflow-estimator==2.11.0
# tensorflow-intel==2.11.0
# tensorflow-io-gcs-filesystem==0.31.0
# torch==2.0.1+cu118
# torchaudio==2.0.2+cu118
# torchvision==0.15.2+cu118
```

Figure 1: Commented out libraries to leverage GPU

These are the libraries pertaining to the use of a GPU. They have been commented out to avoid automatically installing them, as they are quite heavy packages, and only relevant if you wish to make use of a GPU.

# Setup

## Software installation

To ensure the proper functioning of the software, it's essential to correctly install all required software components. This chapter will guide you step-by-step through the installation process.

While it is not necessary to run this software, the use of an IDE such as Visual Studio Code is recommended, as it puts many useful tools at disposition to automate some of the following steps.

## Cloning the Git repository

Before the installation, you'll need to clone the software repository from Git:

1. Open your terminal or command prompt (Windows key + R, then type in "cmd" and enter).
2. Navigate to the directory where you want to save the software using the command:

```
cd [directory-name]
```

3. Enter the following command to clone the repository:

```
git clone [repository-url]
```

4. After completion, navigate into the cloned directory:

```
cd [repository-name]
```

## Installing Python

The software runs on Python, specifically version 3.10. It should work with any version of 3.10, but has been extensively tested with 3.10.6. To install Python:

1. Check if you have a (correct) Python package on your system by opening your terminal or command prompt and typing:

```
python --version
```

If it indicates that you do not have Python installed on your machine, or your version of Python is not a version of 3.10, follow the next steps. Otherwise, you can skip to [Using a virtual environment](#).

2. Download the [Python 3.10 installer](#) for your operating system, or specifically [Python 3.10.6](#).
3. Launch the installer. Ensure that you check the box "Add Python 3.10 to PATH" during installation. This step is crucial for the system to recognize python commands from the command line.
4. Follow the on-screen instructions to complete the installation.
5. Verify that Python has been correctly installed by trying the step 1 again. It should display `Python 3.10.X`, with the X being your installed version of Python 3.10.

## Using a virtual environment

For isolated and conflict-free software running, it's recommended to create a virtual environment. Here are instructions on how to do it with "venv":

1. In your terminal or command prompt, navigate to the software directory (where you cloned the git repository).

```
cd [path/to/directory]
```

2. Enter the following command to create a virtual environment named 'env' (or any name you prefer):

```
python -m venv venv
```

To activate your virtual environment:

- On Windows: `.\venv\Scripts\activate`

- On macOS and Linux: `source venv/bin/activate`

The environment needs to be activated before using the software, so every time you open a new terminal. When creating the environment in VS Code with the repository open, a prompt should appear asking you if you wish to set this environment as default for this directory. Clicking yes will ensure that the environment gets activated every time a new terminal gets opened from within VS Code.

## Installing the required python libraries

With the virtual environment activated, you now have to install the external libraries used by the software. Still in the software directory, type:

```
pip install -r requirements.txt
```

Some libraries installed through dependencies can cause conflicts, so after this step, enter the following commands:

```
pip uninstall opencv-contrib-python
```

```
pip uninstall opencv-python
```

```
pip uninstall opencv-python-headless
```

```
pip install opencv-contrib-python
```

This step will require some time, but only needs to be done once.

## Installing the camera SDK

For the software to be able to communicate with the camera, you will also need to download and install the camera's Software Development Kit (SDK). Download the installer by clicking on the link "*Windows SDK USB2+USB3+GigE (including Directshow + Python) Galaxy V1.23.2305.9161*" on [this page](#), and simply follow the instructions of the installer. You will have to restart your computer after this step for it to take effect.

## Installing the Tesseract model

You will need to install the Google's Tesseract model to make a full use of the software. You can find the installer for Windows [here](#). Simply download the installer and follow the instructions. The default file path of the Tesseract executable should be '*C:\Program Files\Tesseract-OCR\tesseract*'. If not, make sure to change the given default path to this one.

## Hardware setup

Ensuring a proper hardware setup is crucial for the optimal performance of the software, as it will have an extensive impact on the sharpness of the image, and thus on the accuracy of the OCR. Below are guidelines and recommendations to set up the system correctly.

Already launching the software in order to have a preview of the camera feed is recommended, as it helps see the differences in image quality when changing one of the following parameters.

## Setting Up the Charuco Board

A Charuco board is a combined checkerboard and ArUco marker grid, used primarily for computer vision tasks like camera calibration and 3D reconstruction. In our application, it assists in identifying any deviations in the camera's position, enabling the software to recalibrate the regions of interest effectively.

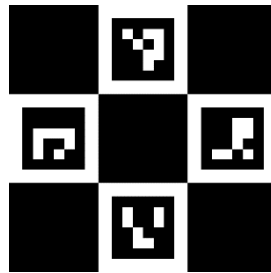


Figure 2: 3x3 Charuco Board

1. **Preparation:** Different designs can be found in the [path/to/charuco] directory of the repository, with the smallest size being 3x3. The bigger the board, the more precise the calculations of the camera position are, but the 3x3 board should be precise enough for our use. If you need a new marker, print it and mount it on a flat, rigid surface. If the board is warped or not flat, it might impair the software's capacity to robustly determine the position of the ROIs.
2. **Placement:** The Charuco board should be positioned around the screen. It does not have to be on the same plane as the screen. It can for example be placed on the side of the screen if there is not enough room on its borders. It should however remain visible to the camera. You can attach it to the side of the screen or anywhere adjacent, ensuring the camera has a clear view.
3. **Security:** Secure the Charuco board firmly. Movement can disrupt the calibration, making a new setup in the GUI necessary since the software calculates the position of the display in relation to the board.

## Camera Positioning

1. **Distance:** It is generally hard to choose freely the distance at which to position the camera, as the deployment environments generally will not allow too many options. Given the opportunity try to stay as close as possible to the optimal distances of the different lenses (50cm, 100cm and 200cm) and if those distances do not allow placement, try going under them rather than above, as image sharpness decreases faster when going beyond the optimal distance than when going under.
2. **Angle:** Aim to minimize the camera's angle relative to the screen. Try avoiding angles above 45°, both vertically and horizontally, to ensure the best image quality and accuracy in data extraction.

3. **Considering the spacing of values:** If the camera is placed closely to the display with a considerable angle, and there are values you wish to observe that spaced apart by a wide gap, it is possible that the closer value will appear brighter than the further one. This can make it hard to find camera settings that account for the appearance of both of them. In this case, try repositioning the camera to mitigate this effect by reducing the angle of the camera and/or increasing the distance of the camera to the screen.
4. **Orientation:** The software can also function with the camera upside down or on its side. However, considering the typical widescreen (width > height) resolution of the displays as well as the camera images, it is advised to position the camera either upright or upside down. This allows the camera to be closer to the screen, capturing a detailed image without unnecessary empty space.
5. **Stability:** Attach the camera to a stable platform or fixture. Avoid surfaces that vibrate, as it can blur the image and decrease the data extraction accuracy.
6. **Lighting:** Ensure that there are no strong light reflections on any part of the display where values are to be read. Reflections can distort data and compromise the accuracy of the OCR process.
7. **Moiré Effect:** The moiré effect, an interference pattern created when two grids overlay imperfectly, can sometimes appear on screens. This effect can introduce significant noise into the captured image, severely decreasing the accuracy of the software. If you notice the Moiré effect:
  - First, try repositioning the camera slightly – even a change of a few centimeters or degrees in angle might resolve the issue.
  - If adjusting the camera's position is not feasible, experiment with different exposure times (can be set via the software GUI) and aperture combinations.

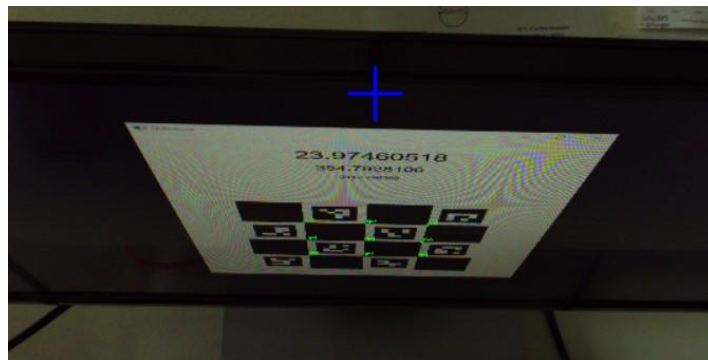


Figure 3: Moiré effect example



## Lens Selection

At the time this manual is being drafted, we have three lenses available, each with different focal lengths:

- 6mm with an optimal distance of 50 cm from the screen.
- 12mm with an optimal distance of 100 cm from the screen.
- 25mm with an optimal distance of 200 cm from the screen.

To choose the right lens:

1. **Measure Distance:** Measure the distance available between the planned camera position and the display.
2. **Opt for Longer Focal Length:** The image sharpness degrades faster when exceeding the optimal distance compared to being under it. If the installation setup does not allow setting the camera close to one of the optimal distances, it is recommended to take the lens with the optimal distance above the actual distance. Use the following graph to help you make your decision. Of course, testing with the two lenses whose optimal distances are above and under the actual distance is best to determine which provides the clearest and most accurate image.

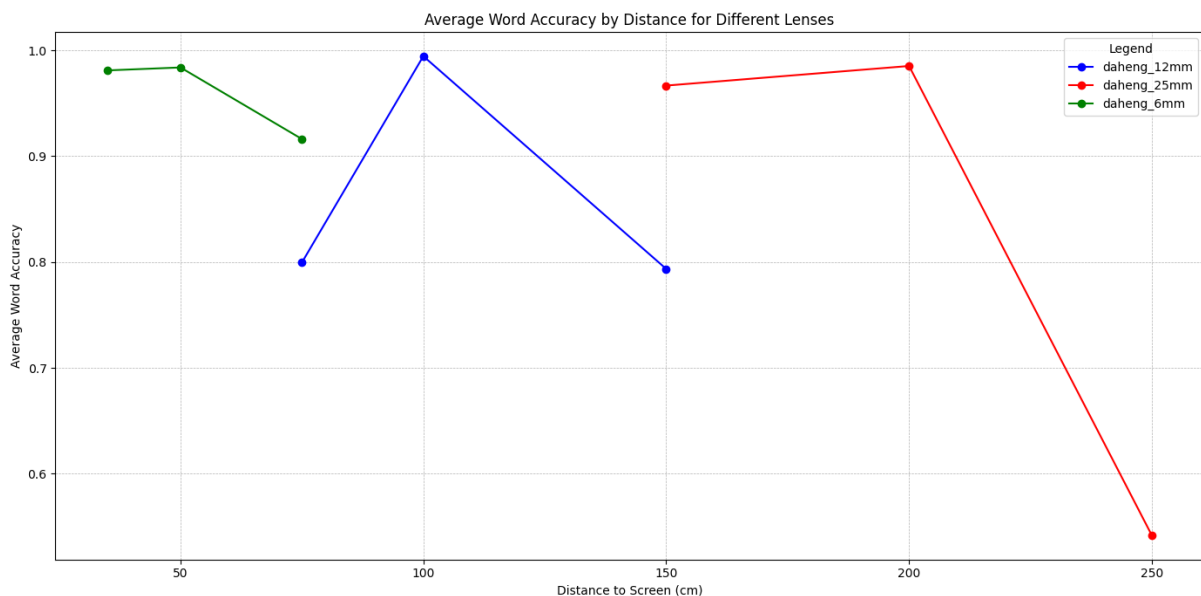


Figure 4: Calculated software accuracy during control experiment based on distance and lens

## Other considerations

Accurate calculations of the display's position, and the camera's relation to the Charuco board, are contingent upon a meticulous camera calibration for each specific lens and its set focal length. This calibration ensures that the system can reliably and robustly recalibrate the regions of interest even when the camera's position changes, which leads to the following considerations:

1. **Avoid Altering the Set Focal Length:** Although our lenses offer a bit of flexibility in terms of focal length adjustment, it is advisable not to tweak it. Altering the preset focal length can compromise the calibration's precision. This, in turn, can diminish the accuracy of the regions of interest recalibration, especially when the camera's position changes.
2. **Exceptions to the Rule:** If you are confident that the camera will remain stationary throughout the measurement phase, you might consider adjusting the lens's focal length to obtain the sharpest image. This is an exception and should be exercised with caution.
3. **Reverting to Calibration Focal Length:** If the lens's focal length has been modified, and you need to return it to the calibrated setting, follow this procedure:
  - Position the camera at the lens's optimal distance from an object.
  - Gradually adjust the lens until the image appears at its sharpest. A tip here is to utilize a shorter exposure time. This makes the depth of field shallower, which in turn makes the point of optimal sharpness more discernible.

# Getting Started

## Launching the Application

To launch the script, simply double-click on the *CVT\_OCR.bat* file.

## Overview of the GUI

### Main Window

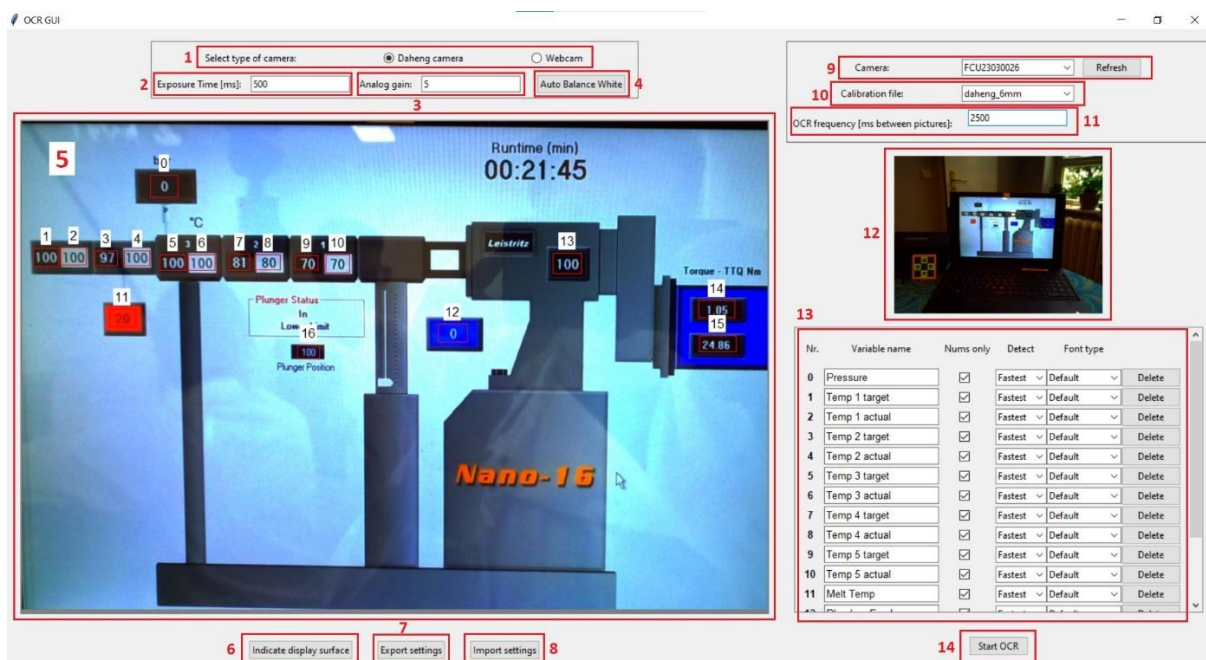


Figure 5: GUI main window screenshot

1. **Camera type:** Choose the type of camera you wish to use between a webcam and a Daheng industrial camera. (At the time of drafting this manual, choosing webcam when a Daheng camera is plugged in sets a hidden parameter of the Daheng camera to a value making it impossible to use by simply clicking the Daheng option again. If this happens, close the program, unplug and replug the camera, and start over).
2. **Exposure time:** Set an exposure time between 1 and 999 ms. Value must be an integer. Only accessible for Daheng cameras. Press the “Enter” key to confirm. Only accessible for Daheng cameras.
3. **Analog gain:** Set an analog gain value between 0 and 24. Value must be a float. Press the “Enter” key to confirm. Only accessible for Daheng cameras.
4. **AutoBalanceWhite:** Rebalances the colors depending on the lighting conditions. Does not have a big impact on OCR accuracy. Only accessible for Daheng cameras.

5. **Target surface video feed:** Display of the camera feed zoomed in on the target surface (screen displaying values). If target surface has not been indicated yet, original feed is displayed. You can click on this image to indicate ROIs on the display.
6. **Indicate target surface:** Opens the window to indicate the position of the target surface and calculate its coordinates.
7. **Export parameters:** After selecting your settings, identifying the target surface, creating your ROIs, and determining their parameters, you can export this information. This allows you to run the script without the GUI or import the settings back into the GUI at a later time, eliminating the need to manually repeat the setup for the same configuration. Clicking the button will prompt you to choose the settings file name.
8. **Import parameters:** Opens a window to browse your file and choose a settings file as created by the export button.
9. **Camera input:** Select the camera you wish to use from the list of detected cameras. Make sure the camera is not being used by another software or it will not be accessible.
10. **Camera calibration file:** Choose from the list the file containing the calibration parameters for the used combination of camera and lens.
11. **OCR frequency:** Choose the frequency at which the OCR should be performed (in ms -> 500 ms is equal to 2 operations per second). Value should be an integer. This is the highest frequency possible. Actual frequency may be lower based on processing time needed.
12. **Original feed:** Display of full unwarped camera feed.
13. **ROI list:** List of demarcated ROIs on the target surface video feed.
  - a) **Variable name:** Name of variable displayed on target surface (ex: Temperature), as to be saved in the database
  - b) **Numbers only:** Check box to indicate if the text in the indicated ROI can only be a number (default = true). Recognizable characters in this case are {0123456789.+}. If normal text should also be able to be recognized, uncheck the box.
  - c) **Detect:** Choose the text detection method. "Fastest" is custom-made, much faster, but more susceptible to noise. "Best" is considerably slower but handles image noise better.
  - d) **Font type:** Choose from the list appropriate kind of font depicted. Changes the image processing pipeline. Particularly important for seven segment displays.
  - e) **Delete button:** Click to delete this ROI if unnecessary or if you are unhappy with its demarcation on the image.
14. **Start/Stop OCR button:** Starts and stops the measurement. When "Start" is clicked, a prompt window will open to ask the name of the measurement, and if wished a commentary to store it with. The name of the measurement must be unique and giving a measurement name already stored in the database will lead to an error message.
15. **Preview button** *[Functionality in development]*: Allows seeing a preview of the ROIs and how they look after their respective image processing pipeline applied, as well as their resulting OCR-extracted text. This should be used to ensure there is no consistent problem before launching the measurement and storing the data in the database.

## Indicate target surface window

[Insert screenshot]

This window is used to indicate where the display is in the picture, and calculate its coordinates in relation to the Charuco board. This step is crucial to ensure that the software can zoom in on the target surface again if the position of the camera changes. How to correctly use it is described in detail in the chapter [insert chapter #].

1. **Live camera feed:** Displays the camera feed, with the pose estimation (position and angle) of the Charuco board if detected. Corners of the target surface can be indicated by clicking on them on this image. If you are unhappy with the position of the corners you indicated, clicking a fifth time anywhere on the image after having indicated the four corners will start
2. **Confirm button:** If satisfied with the demarcation you have made click on this button to save this information (has to be done twice with different angles).
3. **Cancel:** Closes the window without storing the data from the surfaces indicated by the user.

## Workflow

Here is a description of a typical workflow you can follow to ensure a good use of the software:

1. **Install Charuco board:** Fix the Charuco board strategically as described in the chapter [Setting Up the Charuco Board](#).
2. **Launching the software:** You should now start the script, and choose the appropriate settings (camera type, camera input, calibration file and Charuco board size).
3. **Target surface first perspective:** Place the camera in a first position looking at the display with the Charuco board visible. Use the GUI to indicate the four corners of the target surface. The target surface does not necessarily need to be the whole screen, if a smaller area contains all the desired values. Leaving unnecessary space out will make demarcating the ROIs in step 9 easier, as the displayed image will be more zoomed in. The target surface should however have four distinct corners, as you will have to do this again, and indicating disparities in the indicated points will have a negative impact of the robustness of the software to camera movement.
4. **Target surface second perspective:** Move the camera to get a second perspective of the display, and repeat the previous step. It is recommended to have an angle close to 90° between the two perspectives to have a better precision, but good values can still be calculated with other angles (as long as it is not close to 0° or 180°). After having confirmed the second perspective, you can display the recalculated surface position if you wish to. When moving the camera around, the surface (appearing with a green border) should reposition itself to appear on the desired surface. Unexpected behavior here could indicate that the calibration file chosen is not the appropriate one for this camera/lens combination. If you are happy with the displayed surface, you can now close the window.
5. **Camera setup:** Install the camera in its final position as instructed in the [Camera Positioning](#) chapter.
6. **Aperture selection:** Choose an aperture (one of the two adjustable rings on the lens) to maximize the sharpness of the image. Starting with f/8 is a good value. Start there and adjust as necessary. A smaller aperture increases the Depth of Field (DoF), meaning the space in which the observed objects appear in focus is increased. As you should avoid changing the focus (the other adjustable

ring on the lens) of the camera if possible, reducing the aperture when the position of the camera is not at its optimal value for the used lens can help have a sharper image. However an aperture too small can cause diffraction effects and should be avoided. You will notice that a smaller aperture lets less light come into the camera sensor, but this can be compensated with a higher exposure time and a higher analog gain.

7. **Exposure time selection:** Once you are satisfied with the sharpness of the image you can adjust the exposure time to give the image an appropriate brightness. A higher exposure time will however cause the image to be more sensitive to movement and vibration. As our setup is static, this should not be an issue. It is still recommended not to go above 500ms, as a going higher increases the chance of values regularly changing during the exposure, making the number unrecognizable.



Figure 6: Exposure time too long example 1



Figure 7: Exposure time too long example 2

8. **Analog gain selection:** If an appropriate brightness is not attainable without the exposure time going over 500ms, you can increase the analog gain (sensor sensitivity to light). This should be limited as it introduces more noise in the image.
9. **Determine the ROIs:** Now that the target surface is found and that the camera has been appropriately adjusted, you can demarcate the different areas of the target surface containing the desired values. Pay attention to leave enough margin so that if the displayed number gets more digits, that the whole new value stays in the demarcation, but try not to include outside elements, such as part of a frame around the number, within the demarcation (a vertical line next to the number can be interpreted as a one).
10. **Choose the ROI settings:** Give each of your ROIs a name (ex: Temp. 1 Soll, Temp. 1 Ist), and set the following parameters appropriately:
  - a) Numbers only: Indicates if the values expected in the ROI are numbers only, or if text is also expected. Choosing not to check this box even though no text is expected to appear here is highly discouraged, as the accuracy will unnecessarily deteriorate (easy to misinterpret “0” as “O” for example).
  - b) Detect: Chooses the text detection method. “Fast” is considerably faster than “Best”, but less impervious to noise. Furthermore, it is highly sensitive to outside components intruding in the ROI. If a part of a frame encasing the value is within the ROI, choose “Best”.
  - c) Font: Choose what type of font is within the ROI. Sets the appropriate image processing pipeline used for this part of the image.
11. **Preview the ROI results:** Now that the settings are done, click on the “Preview” button to see the original image of the ROIs, their processed versions, and the interpreted value. If the value recognized is consistently wrong, try the following troubleshooting steps:
  - a) Moiré effect: If you observe a Moiré effect on the ROI image, see point (7) of the [Camera Positioning](#) chapter. If you still do not manage to correct this without compromising the accuracy of the text recognition for other ROIs, try choosing the “Best” detection method.
  - b) Original image quality is bad: Try re-adjusting the aperture, exposure time and analog gain to improve the image quality

- c) Value in processed image is unclear: Make sure the font you have chosen for this ROI is the right one
- d) Value recognized is consistently wrong although the value in the processed image seems clear:  
Try changing the selected text detection method

If the interpreted values are still consistently wrong after these troubleshooting steps, you might have to reset the camera to a more favorable position and angle. Reducing the angle with the screen can homogenize how the different ROIs are perceived from the camera, and thus make finding an aperture / exposure time / gain combination more appropriate for all ROIs easier. Changing the distance of the camera to the screen can also have a considerable effect on the sharpness of the image. As a last resort, you can also adjust the focal length on the lens. This means making the software less impervious to displacement, as the recalculation of the target surface's position in the camera feed will be less precise, and the values displayed might come out of their respective demarcated ROIs. If you still decide to use this, you should frequently make sure that it is not the case (a quick look on the GUI is sufficient). You should also make sure that the focal length has been reset to the original value (see [Other Considerations](#) chapter) before the next use.

If you still don't manage to get acceptable results for all ROIs, you might have to forego extracting the values of certain ROIs. In this case, make a picture of the setup (camera, screen, Charuco board), a picture of the display, export the used parameters so they can be found again, and take a screenshot of the previewed ROIs. Provide the technical team in charge of this software with this information, so they may find a solution to avoid this issue in the future if possible. If you are satisfied with your settings, you can now close the preview.

12. **Export your settings:** Export your settings so you can import them at a later date when doing measurements with the same setup. It is recommended to always export your settings when you are done, even a single measurement is planned with this setup, to avoid having to do the set all the parameters all over again (naming all the ROIs can be especially time consuming) if a problem arises and the software has to be launched again.
13. **Check for disparity on the spectrometer PC and the PC running the software:** The values from the ROIs are stored alongside the timestamp at which the picture they stem from has been taken. The clock of the computer running the software sets this timestamp. The data gathered by the spectrometer is timestamped relatively to the spectrometer computer. To be able to correlate both datasets, you should determine if both clocks are synchronized. If not, you should make a note of the time difference.
14. **Start your measurement:** Click on the "Start OCR" button. You will be prompted to give the measurement a name. You also have the option here to save a comment with this measurement. You can for example save here a potential time difference between the spectrometer clock and the software's clock.
15. **Checking sporadically:** Make sure at times that everything is in order. For example, the software has been developed to be resistant to small camera displacements to avoid having to reset everything if someone bumps against the camera, or at the start of a new measurement if the camera has been put away in stored away between two days. Big movements however, such as moving the camera from one side of the screen to the opposite side, can cause small displacements of the ROI demarcations, leading to the values spilling out. Checking sporadically the GUI to make sure that something like this has not happened is recommended.
16. **Stop the measurement:** Click on the "Stop OCR" button when you are finished with the measurement.

## Retrieving Stored Data from Database

### Data retrieval

The data is stored in an SQLite database. This is a local file stored in “ocr\_script/ocr\_databsae.db”. The data can be viewed directly within VS Studio with the “SQLite Viewer” extension, or it can be downloaded in csv format using the database GUI. To open the Database GUI, simply double-click the *CVT\_OCR\_DB.bat* file.

In this GUI you can search the database by measurement name and ID to retrieve all the data from the searched measurement, or you can look for a specific stored variable, also by name or ID. Clicking on the “Search” button with “Measurement Name” or “Variable Name” without any text given will output the full list of all stored measurements or variables. You can merge measurements and variables within on file by selecting multiple ones before clicking on the “Download” button. This merging operation will only succeed if the all the selected measurements have the same variables (if searched by measurement), or if all the variables have the same name (if searched by variable).

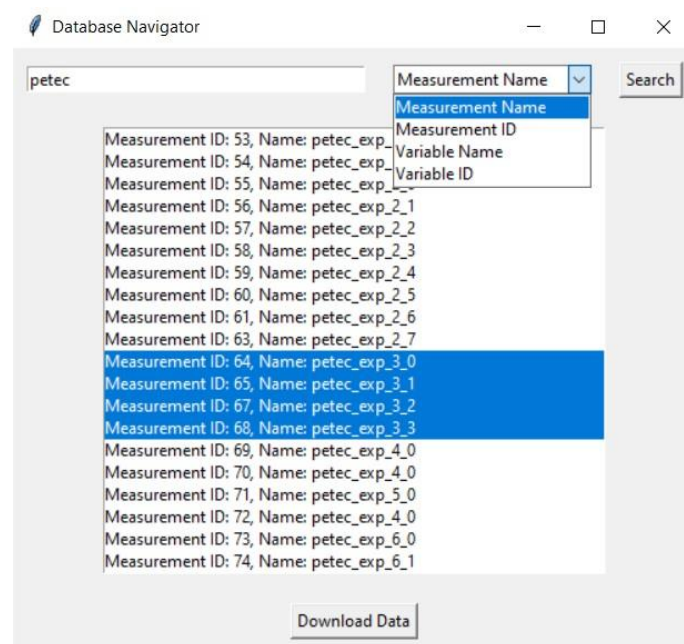


Figure 8: Database GUI

### Data post-processing

[Functionality in development. Will be integrated within the Database GUI to be able to opt-in for the data clean-up before downloading the data, with a live preview of the changes.]