

**VCAM Focusing GUI**

**Setup and User Manual**

**Author: Oxin McKittrick**

**6/11/2020**

Contents

[1. Setup 1](#_Toc55557732)

[1.1 Creating the Dataset 1](#_Toc55557733)

[1.1.1 Parameter Consistency 1](#_Toc55557734)

[1.1.2 Creating the Dataset 2](#_Toc55557735)

[1.2 Accuracy Assessment 2](#_Toc55557736)

[1.3 Training the Model 3](#_Toc55557737)

[2. User Manual 4](#_Toc55557738)

[2.1 Checklist 4](#_Toc55557739)

[2.2 Step-by-Step Guide 4](#_Toc55557740)

# 1. Setup

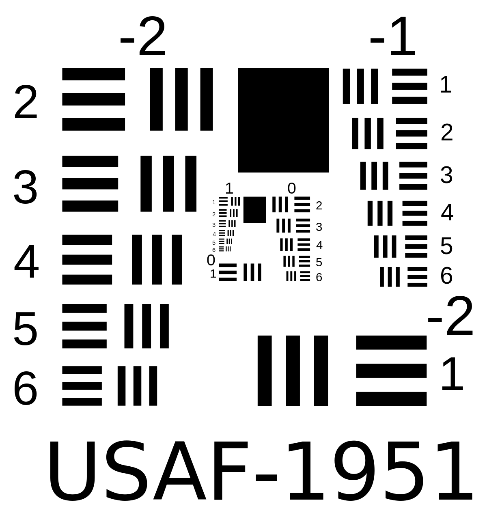
This section is for the purpose of setting up the necessary conditions so that the focusing can run without error, and should be carried out and tested prior to any critical VCAM focusing events.

## 1.1 Creating the Dataset

### 1.1.1 Parameter Consistency

A dataset needs to be developed to create a classification model, for the purpose of accuracy this dataset must be representative of the expected official VCAM focusing settings. To this, the dataset should ideally be developed in the exact setting that the VCAMs are expected to be focused in, that is that all parameters are kept consistent throughout.  
To achieve this the dataset should be created with a VCAM module (not a Pi Camera, phone camera, etc.), with the same lens (e.g. wide angle) to be focused. This dataset should also be set in the same setting expected for official focusing, for example if the VCAMs are to be focused in the clean room, then the dataset would ideally be created in this environment, under the same lighting conditions expected.

In terms of physical parameters, the target to be focused off must remain the same when creating the dataset and should be placed the same height as the VCAM recording it. Additionally, this target should be black and white and contain well defined horizontal and vertical lines, e.g. USAF-1951 resolution test chart seen below:



The VCAM must be perfectly perpendicular to the target and the distance from VCAM to target is to remain constant for all points within the dataset and eventual application of the VCAM focusing code. To achieve this, it is recommended that the VCAM and targets positions are fixed, e.g. via a mounting board or so.

### 1.1.2 Creating the Dataset

Once the conditions are setup correctly measurements may be taken in order to create the dataset, this can be done through use of the code 'CreateTrainingData.py', located in Réaltra’s server at: Z:\COMMON\Réaltra Software\Oxin's Scripts\VCAM Focusing Code.

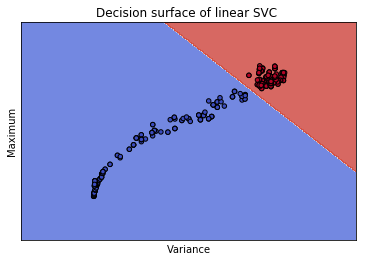
This will allow the user to run an altered version of the focusing code, where they similarly must first crop the FOV to perform focusing on (e.g. the target), it is important that this cropped area lines up with the borders of the target image so as to provide consistency with later focusing. The user themselves must then define whether an image is ‘blurry’ or ‘sharp’ (rather than being told). This definition is somewhat variable from person to person, however it will provide consistency across all VCAMs and to aid the user, the variance and maximum measurement of the VCAM’s image is provided within the GUI, these parameters are correlated with sharpness and should therefore provide an indication of how focused an image is beyond the accuracy of the human eye.

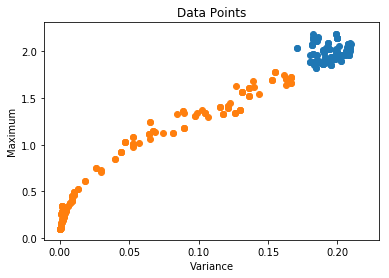
As the user saves sharp and blurry values, these are added to a text file labelled 'ML\_trainingData.txt'. The more data points gathered through this process, the more robust the classification model will ultimately be, so it is recommended that the dataset be made as large as possible.

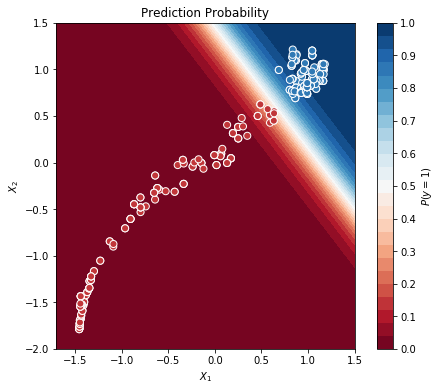
## 1.2 Accuracy Assessment

Before creating the final classification model, its accuracy may be examined through graphs. 'Model\_GraphRepresentation.py' will create the classification model from the dataset made previous (without saving it) and investigate the classifiers cut off points. This will output three graphs to give the user a visual representation of their model’s accuracy. This process can also be used to acknowledge any potential misclassifications carried out by the user when creating the dataset. For example, if the 2D-graph shows a clean cut-off between blurry and sharp images, but one data point appears to be in the completely wrong region, chances are that this was a misclassification by the user (perhaps pressing the wrong classification button by accident) while creating the dataset. If this occurs, for the purpose of accuracy, the user may go into the dataset text file and omit any of these potential mistakes.

An example of the three-sample graph produced can be seen below:







## 1.3 Training the Model

Once all data points are collected, 'CreateClassificationModel.py' (also located in Réaltra’s server) is to be run, this will create the classification model off of the 'ML\_trainingData.txt' file created previous. This code will ‘pickle’ the model and save it under 'focus\_model.pkl'. This will then be accessed by the final code in order to employ it in its predictions in its eventual predictive focusing.

\*\*\* It is important that this whole process be done once for each VCAM lens. This means two classification models need to be made, from two separate datasets, each corresponding to one of the potential lenses. Both of these ‘pickled’ classification models should then be saved within two separate folders, e.g. ‘WideAngleLens’ and ‘NarrowAngleLens’, with a copy of the final focusing code in each folder, for a lab technician to easily navigate when VCAM focusing occurs. \*\*\*

# 2. User Manual

## 2.1 Checklist

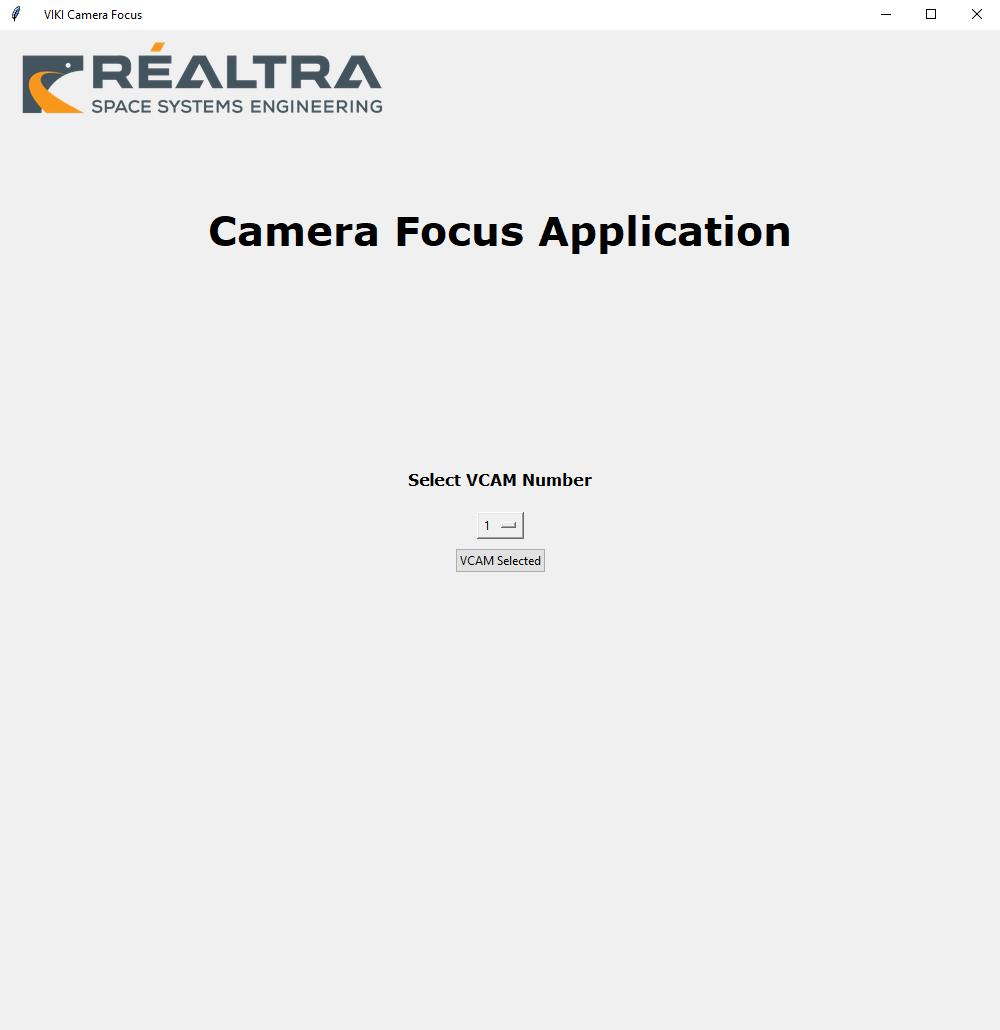
For the purpose of accuracy, it is important that VCAM focusing is consistent with the parameters used when the training set was defined. For this reason, it is important to ensure that all parameters in the checklist are met.  
They are as follows:

* The target used for focusing is \_\_ (TBC).
* The VCAM FOV is positioned perpendicular to the face of the focusing target.
* The distance between the VCAM and target is \_\_ m (TBC).
* The VCAM height is identical to that of the target’s centre.
* The VCAM and target positions are fixed from start to end of the focusing process.
* The VCAM is connected to the computer in use via an Ethernet cable.
* The focusing environment is bright.

## 2.2 Step-by-Step Guide

The following is a step-by-step guide on how to navigate the GUI and ultimately arrive at a successfully focused VCAM module.

* Locate the folder related to VCAM focusing at \_\_ (TBC).
* Within this folder select the folder corresponding to the lens of the VCAM in use, (e.g. wide or narrow).
* From here, the focusing software labelled \_\_ (TBC) is run.
* The landing page will be as follows:



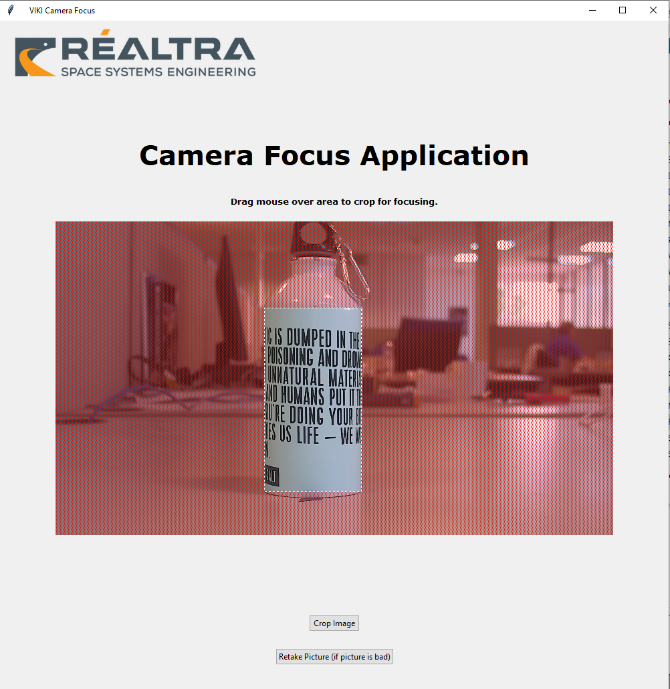
* From the drop-down menu, the user must select which VCAM they are focusing, in order for the program to connect to its corresponding UDP video stream. Once chosen, click on the button ‘VCAM Selected’.
* The following page will then be displayed:



* Reconfirm that the system is setup correctly and if so, select the ‘Take Picture’ button.
* The following page will then be displayed:



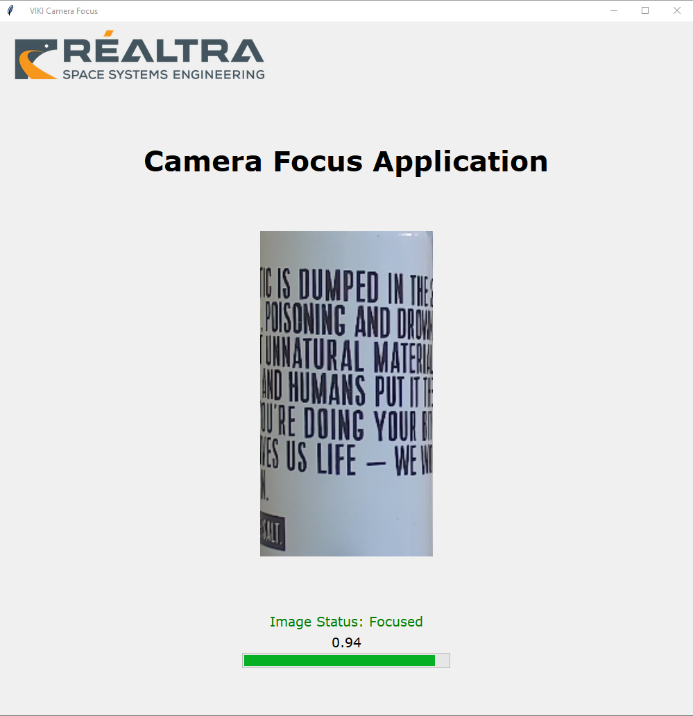
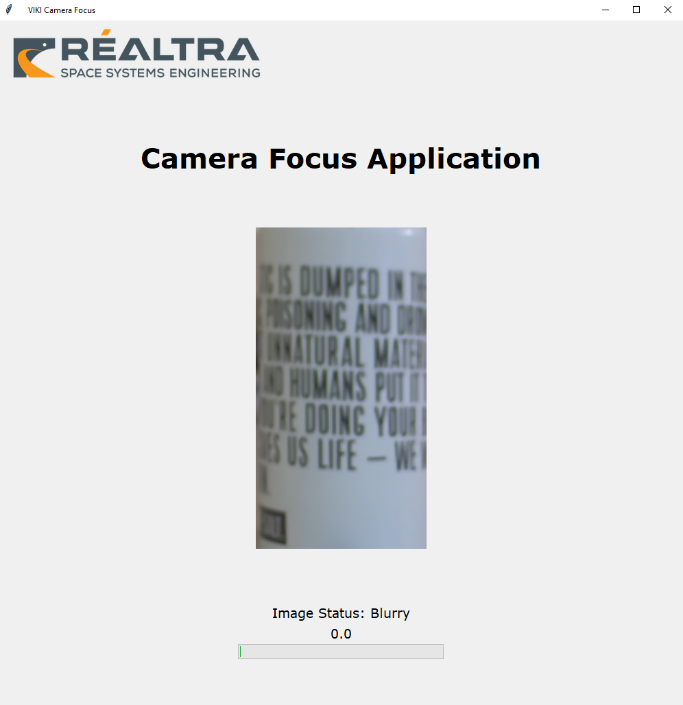
* If the resultant image is unsuitable for whatever reason, the user may select the ‘Retake Picture’ button. If the image is suitable, proceed to the next step.
* Using the mouse, click and drag over the target area in the image taken. The area selected for cropping should match up with the target image’s borders, to provide consistency with how the program was ‘taught’. This will look like so:

  
\*Note: The image used above is simply to provide an example and does not reflect expectation.

* Once the area defined for cropping appears as required, the ‘Crop Image’ button is selected.
* The following page will then be displayed:



* The cropped image is then inspected and if deemed unsuitable, the ‘Re-Crop’ button is selected, which will take the user back to previous page.
* If the cropped image is suitable, the ‘Begin Focusing’ button is selected.
* The following page will then be displayed:

   
\*Two images of the same page have been presented to show the assessment of a sharp and blurry image.

* This page shows a live video feed of the VCAM’s field of view (of low frame rate), while concurrently running a live assessment of whether the VCAM is deemed focused or not. The value presented and its associated ‘progress bar’, is a measure of ‘how certain’ the code is that the image is focused and therefore a good indicator of ‘how focused’ the image is.
* The VCAM focus is then slowly tuned and the corresponding focus measurement on the GUI is consulted. It is important to allow for a slight time delay between changing the focus on the VCAM and the subsequent measurement on the GUI.
* As the GUI’s focus reading shows the VCAM coming close to fully focused, the user must become more incremental and meticulous with changing the VCAM’s focus.
* Once the GUI displays a focus reading of \_\_ (TBC), the VCAM can be considered focused and the program exited.