

User documentation

Note: To see the application in action without installation and running, the docs/appliacion_use_videos/ folder contains videos of usage of individual functions of the application.

(detectionVisualisation_video1.mkv, houghParametersTuning.mkv, sizeEstimationTesting.mkv). They are also temporarily available online:

(Detection visualisation)

https://youtu.be/c_y6aQq1-xo

<https://youtu.be/RoeSsnaeEFY>

https://youtu.be/0_AxzF86HXI

(Hough parameters tuning)

https://youtu.be/II_Rf5y7rHs

(Size estimation testing)

<https://youtu.be/rc5vIKXzW7Q>

Note: Due to the large space consumption of the whole dataset (tenths of gbs compressed), the attachment contains only a small portion of video data. For the way to request more data, see technical documentation.

In order to run the interactive application, the environment needs to be installed. Please refer to the technical documentation for environment installation.

Some commands are Windows specific. Please refer to python and anaconda documentation for running the application on different OS.

Run application

In order to run the application, please open two anaconda consoles. In both consoles change the directory to the root folder of the project (master-thesis-implementation/).

```
cd ./master-thesis-implementation/
```

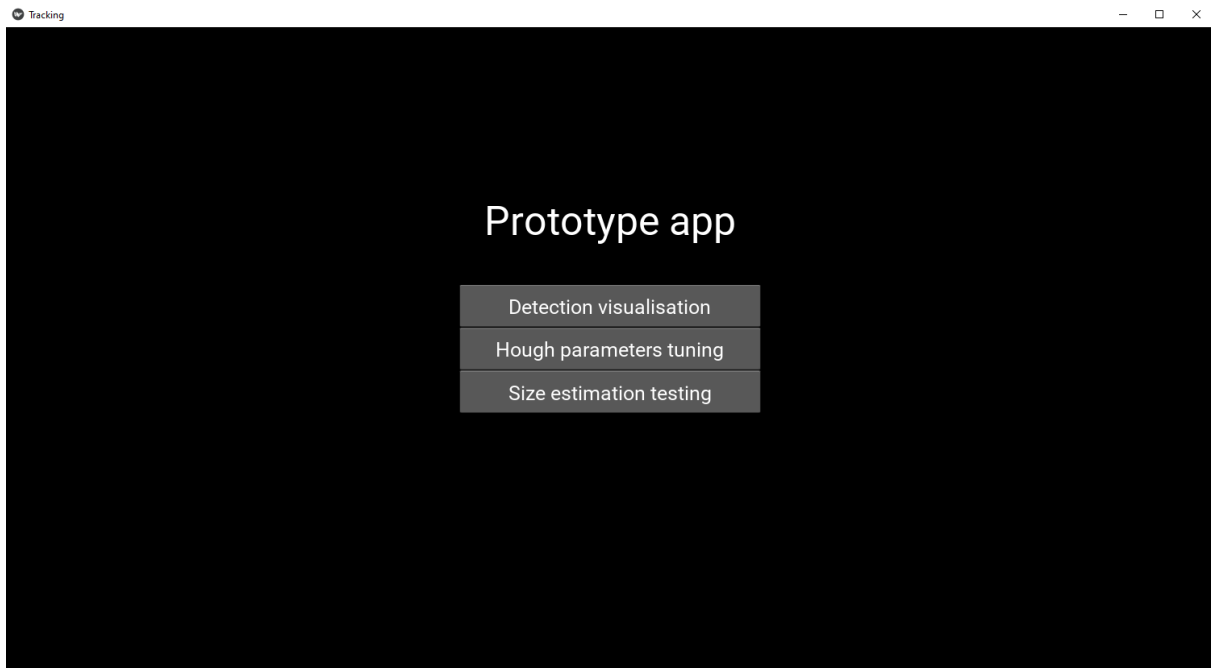
In the first anaconda console window, activate try1 conda environment and run classification_server script in the bin folder:

```
conda activate try1  
python .\bin\classification_server.py
```

In the second anaconda console window, activate tfg7 conda environment and run the kivy application:

```
conda activate tfg7  
python .\bin\trackingapp\TrackingApp.py
```

The window with the application should open:

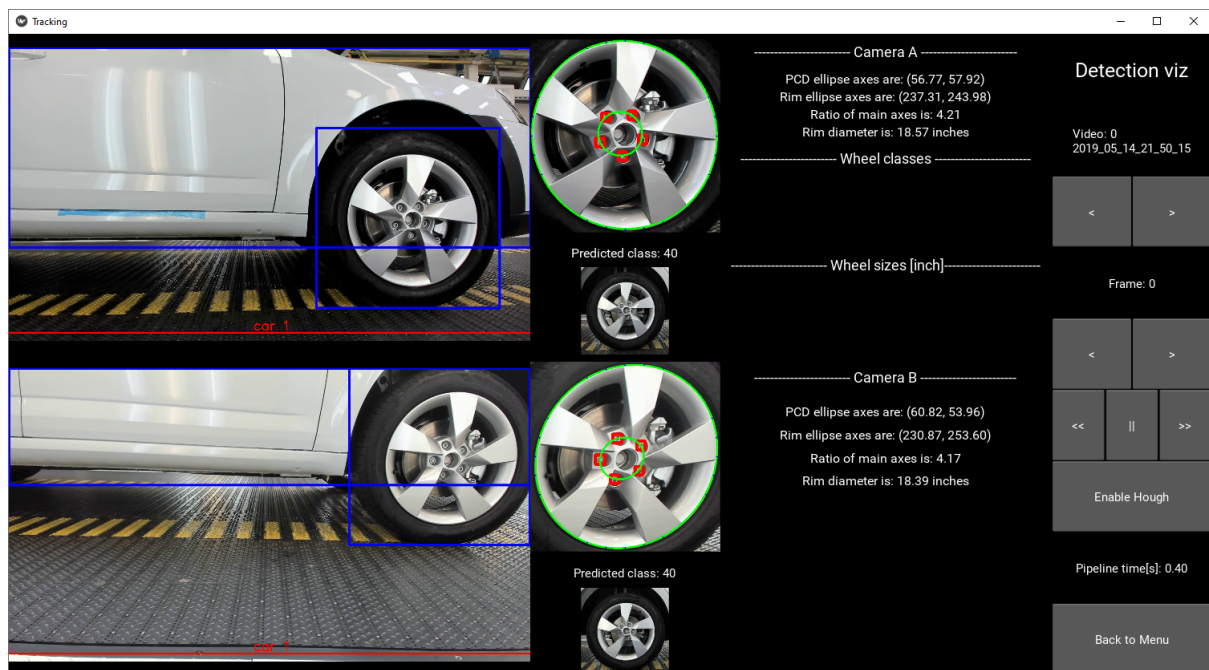


Detection Visualisation

In order to have data for this part, at least one video needs to be parsed to frames first.
Please run script `./bin/parse_videos_to_frames.py` and follow instructions

```
python .\bin\parse_videos_to_frames.py
```

By clicking on “Detection visualisation” in the menu, the UI changes to:
(a minor freeze of a few seconds can be experienced, it is caused by loading yolo models)



In the right column is the menu. Described from the top to the bottom, we can see

- Number and name of the current video that is analyzed
- Buttons for switching between videos
- Number of currently analyzed frame
- Buttons for switching between frames (also mapped to right and left arrow, or keys A, D can be used)
- Buttons for continuous animation of the video with a framerate of about 2.5 fps (also mapped to keys Q, W, E)
- Button to enable visualisation of Hough transform (need to switch frame after to see the result)
- Time of the inference of the whole pipeline
- Button for return to the menu

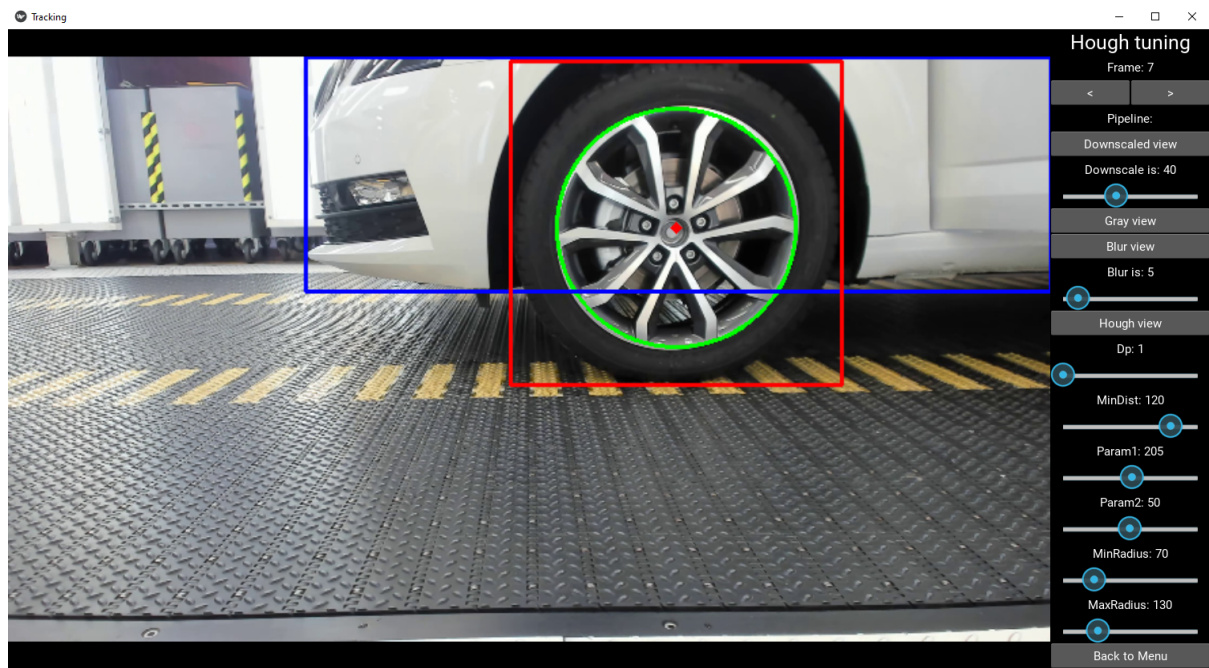
The left part is split into almost identical parts. The upper part describes camera A and lower part that describes camera B. Description of the upper part:

- On the left is a frame from the camera with impainted yolo bounding boxes(blue rectangles) and the position of individual objects as they are tracked by heuristic (red line and 'car1' text)

- Next to it is the upper image that contains a visualisation of wheel bolts detection (red boxes), and pitch circle and rim contour ellipse fits (green ellipses)
- Under it is a current prediction of the wheel class + representative image of that class
- On the right, under the heading 'Camera A', there are current values of ellipses axes, their ratio and estimate of rim size in inches
- The rest of the text is the summary of aggregated values for wheel classes and wheel sizes for the current car. (not present in camera B part)

Hough Parameters tuning

By clicking on “Hough Parameters tuning” in the menu, the UI changes to:



This UI also follows division into the right menu column and data visualisations in the left one. In the larger part of the window, there is a frame with gold bounding boxes (red for wheel, blue for car) and Hough transform predictions (green circle(s)).

The right menu is described from top to bottom:

- Number of current frame and button for switching between frames
- The following part are controls for the processing pipeline, by clicking on the button, we display the frame in that part of the pipeline, by changing values via sliders, we modify the pipeline parameters, which should manifest in the change of the properties of the current frame
- Button for return to the menu

Size Estimation Testing

By clicking on “Detection visualisation” in the menu, the UI changes to:



On the left, there is a current image with impainted detections of the wheel bolts (red rectangles), and ellipses for pitch circle and rim contour (green ellipses). Also, notice red points on the contour of the rim, those are points sampled by the rays in the algorithm described in the thesis.

The right menu contains:

- Number of current frame and buttons to switch it
- Current values of axes of individual ellipses and value of rim diameter estimate
- Button for return to the menu