Automated Driving

Visual Evaluation of Drone Data

Abstract

A software to provide an intuitive GUI for analysis of recorded images from a drone footage of vehicles in a parking-lot. Our image processing techniques extract the location of the vehicles in the image and the reference lane markings which can be used for further analysis.

Mirza, Ahmad Hassan
ahmadhasanmirza@gmail.com
Kulkarni, Pranav
pranavkulkarni24@gmail.com

Automated Driving - Visual Evaluation of Drone Data

AD-VED^2

A GUI is developed which can take images as input and processes the input image to extract some useful information. In the current version of the software, vehicles are detected (from an aerial, top down view) and bounding boxes are drawn around the detections, lane markings are detected and highlighted in green. This document gives an overview of the features available in current version of the software. How the solution has been implemented and how the software can be further improved i.e. features that are planned for implementation in future versions.

The software is developed as part of the coding competition[1] held by it-talents and IAV.

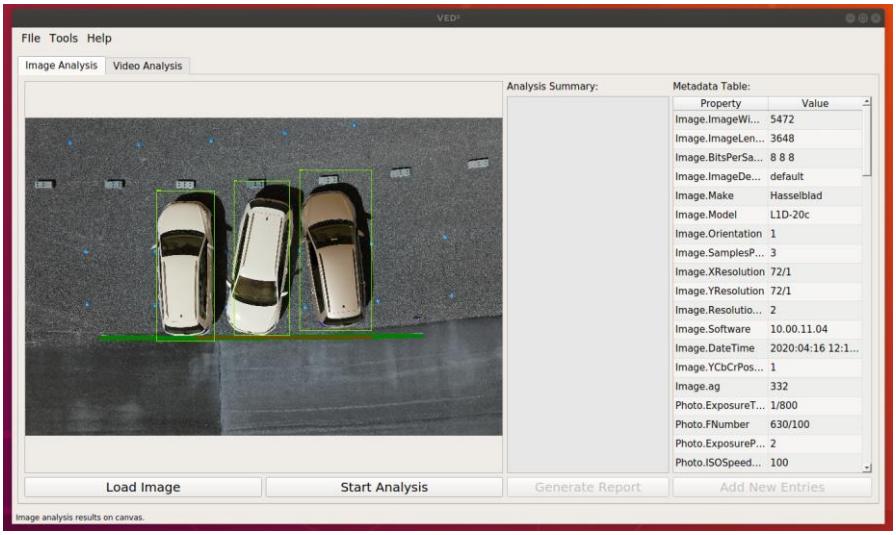

Figure 1 AD-VED^2 Graphical User Interface

1. Software Features – V. 1.0:

- Select and load image and video files for further processing in the software.
  - Video processing is not yet implemented
- Image Processing and visualization
  - Vehicle detection and visualization of detections on screen using bounding boxes.
  - Lane marking detection and visualization.
  - Extraction of meta data related to the image from EXIF data.
  - Display results of image analysis in the software.
  - Exporting of analysis results in json file format.

[1] https://www.it-talents.de/foerderung/code-competition/code-competition-05-2020

1. AD-VED^2- User interface

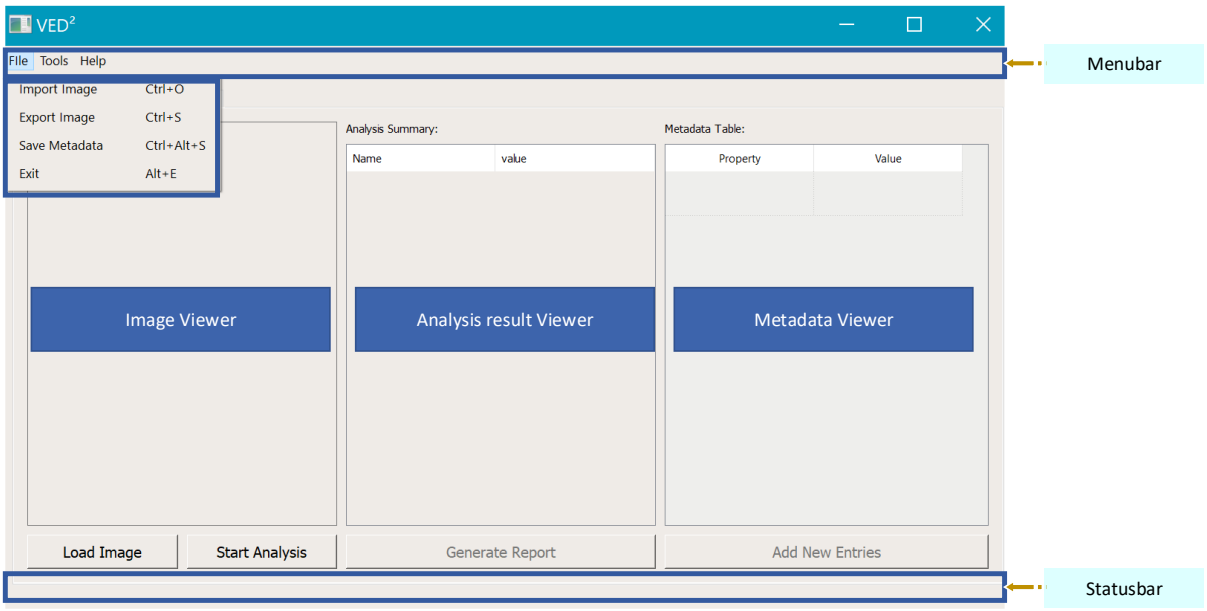

Figure 2 Overall structure of GUI

Keeping wide range of the user in the focus, we have designed the software with cross-platform functionalities as well as fluid and responsive design

2. Implementation Details:

The software is developed using Python 3.7. The GUI is designed in PyQt 5. Figure.1 shows the outlook of the GUI. for the image processing tasks several different libraries have been used. For vehicle detection an SSD Inception v2 (COCO) model has been trained using TensorFlow Object Detection API. The model is trained to detect vehicles from an aerial view (top view) and can be extended to detect other orientation of vehicles or even other types of objects. This allows for the software to be extended further to encompass other object detection tasks very easily.

The GUI features two tabs, Image Analysis and Video Analysis. This architecture gives the user option to choose what kind of input they want to work with. With the current version, the handling for video analysis is not implemented. So, most of the functions in this tab are just concept and are planned to be implemented in future versions. The image processing tab is functional, and the following paragraphs give an overview as to how various functionalities have been handled.

The input image is passed to Vehicle Detection Module. The image is run through the pre-trained model, and if any vehicles are detected in the input image, the co-ordinates of the bounding boxes around these vehicles as well as the image with the bounding boxes overlaid are returned. This image can then be drawn on the canvas or passed to the Lane Detection module to detect the lane markings.

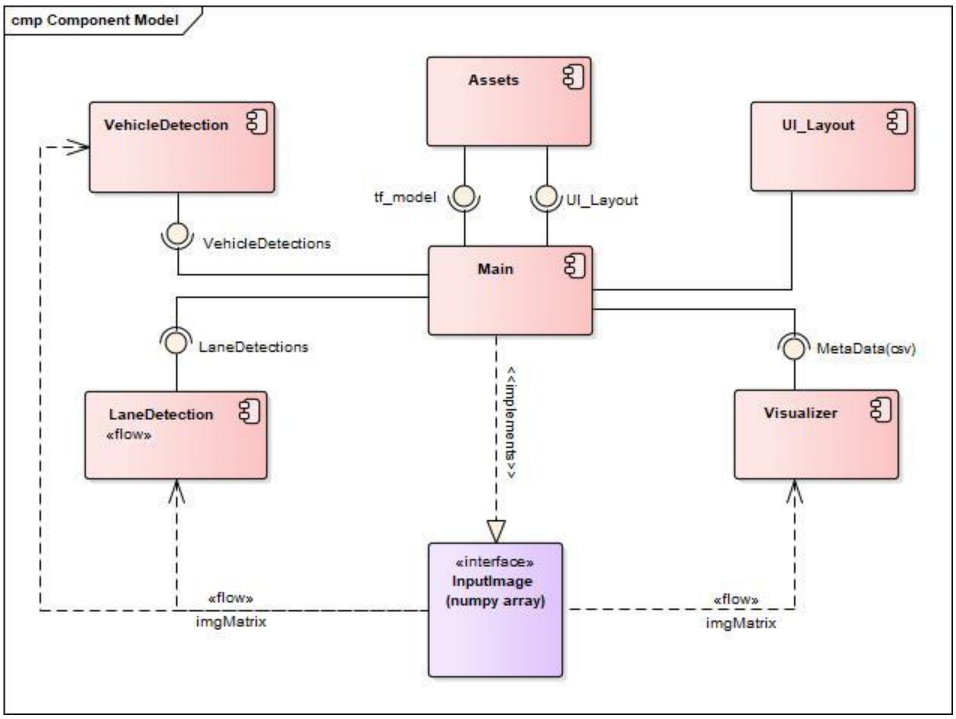

Figure 3 Component Diagram

For lane detection OpenCV is used and several image processing algorithms are used to extract the location of the lanes as accurately as possible. The center-axis of the detected lane is also extracted and given in the results, the idea is that this central axis can be used to calculate relevant information such as the orientation of the vehicles with respect to the lane markings which can include the angle and distance information.

3. Extracting Metadata

Exchangeable Image File Format (Exif) is widely used by drone manufactures for encoding the wide range of information such as GPS locations. Images encoded with Exif tags are capable of providing vast information to users as a standalone source. Our software can extract metadata from image directly, without any need of importing external data such as csv file.

As soon as user import drone image (with exif image), software will automatically extract all the metadata from the image header and load into GUI in a human readable table format.

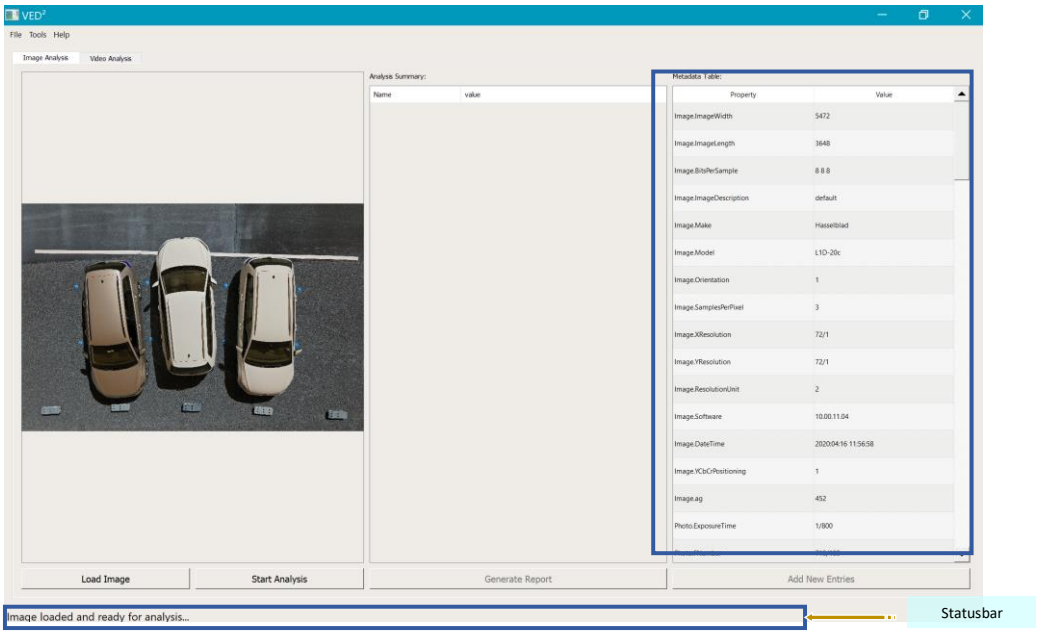

Figure 4 Metadata Loader

4. Analysis

With the current implementation the software gives user the flexibility between choosing which analysis they want to perform. For example, From the “Tools” tab the user can either choose to process the image to extract only the Lane Markings information, only the vehicle detections, or both. The button on the main GUI captioned “Start Analysis” performs both processes and return the combined results.

Figure.2 shows the Component model of the implemented solution and the interfaces implemented by each component. The architecture was designed to allow for easy expansion and adaptability of the software to new use cases.

5. Future Work

In further versions of the software, the functionalities of “Video Analysis” tab are to be implemented. Apart from that, more analysis functionalities are to be added, where the distance between vehicles and between vehicles and lane markings could be extracted. Depending on the use-cases. Any number of analysis functionalities can be added and visualized by simple function calls already implemented in the software.

The report generation feature will also be enhanced so that it can provide a pdf report. The planned implementation for this feature is to have a pre-built latex template which will be filled programmatically from the python scripts resulting in an auto generated pdf report which shall include the calculated data from the image and video analysis processes.