Vehicle Dash Cameras with Artificial Intelligence

Presenter: Matthew M. Stranz, Major: CIS · Mentor: Dr. Sarbani Banerjee, CIS Department

Abstract

The purpose of this research project is to create a vehicle dash camera that assists the user, police, and insurance company in easy license plate recognition and number retrieval in an event of an accident and/or hit and run. Right now, personal vehicle dash cameras are very limited and only record data, and since accidents are very chaotic, it can be difficult extracting the required information. To implement these specific aspects, the vehicle license plate would be detected by YOLOv4, a real-time object recognition system and machine learning model. Following detection, the number within the license plate would be filtered through OpenCV real-time computer vision and printed with Tesseract OCR optical character recognition engine. Currently, this is a software capability that is only found in government, police, drone, and stationary security cameras. Supporting software for this research would be obtained from Google Colaboratory, Jupyter Notebook, Anaconda (Miniconda for Raspberry Pi), Nvidia CUDA Toolkit 10.1, and Git. Additionally, the hardware that would be needed is a Nvidia GPU, Raspberry Pi 4 Model B, a storage device microSD card, Arducam day and night vision camera, Raspberry Pi 7" touch screen display, USB to USB Type-C with 15W car charger, and a housing unit that would incorporate the hardware along with a mounting for the windshield or the mirror. The final goal is to have all license plate numbers be enlarged and printed above the license plate, and all detections be saved onto the storage device.

References

The AI Guy Video Tutorial:
https://www.youtube.com/watch?v=AAPZLK41rek&t=1s

theAIGuysCode/yolov4-custom-functions:
https://github.com/theAIGuysCode/yolov4-custom-functions

Nicholas Renotte's Video Tutorial:
https://www.youtube.com/watch?v=NApYP_5wlKY

nicknochnack/ANPRwithPython:
https://github.com/nicknochnack/ANPRwithPython

Otsu Method:
https://en.wikipedia.org/wiki/Otsu%27s_method

Findings/Results

Step 1: YOLOv4 subimage

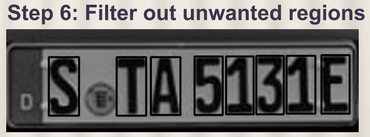
Step 2: Grayscale and Gaussian blur

Step 3: Thresholded to black & white

Step 4: Dilated

Step 5: Sort contours

Step 6: Filter out unwanted regions



Step 7: Segment each character, apply bitwise_not mask for black and white, and apply small medium blur. Finally, pass it through Tesseract.

Interpretation of Findings

YOLOV4 is used to create a bounding box around the license plate. It allows for a specific location or subimage to be targeted and converted. Without it, the rest of the environment can create too much background noise which can skew the result. The next steps are all implemented with OpenCV. The subimage is resized for a bigger view, converted to grayscale, and smoothed out by Gaussian blur. Next, the Otsu's method is applied which thresholded the image from grayscale to white text with a black background and then dilated to make contours more detectable. The contours identified around the letters and numbers are bounded with a box and then sorted from left to right. If a region's height, width, and area do not meet the size criteria, they are filtered out and removed. Finally, each character is segmented into its own subimage, converted back to black and white via bitwise not mask, and apply a small median blur. Each character subimage is converted from image to string via Tesseract OCR and then printed boldly above YOLOV4's license plate detection box.

Conclusions

This project is still ongoing and currently being tested with different methods and programming. The first image example was converted through Pytesseract and the second through EasyOCR. The reason being that not all license plate numbers are correctly identified and printed. However, running the camera in real-time allows the license plate to be filtered with multiple iterations allowing for better accuracy. Additionally, a fast CPU and/or GPU would also increase the accuracy. Finally, instead of piecing hardware together, it would be better to have a 3D printer in order to make a touch screen housing that incorporates a camera housing.

Acknowledgements

This project is supported by the Office of Undergraduate Research at SUNY Buffalo State and Dr. Sarbani Banerjee.