



University of Illinois at Urbana-Champaign
Computer Science Program

CS 445

Computation Photography

Final Project Report

*Image Processing Based Vehicle Number Plate Detection
and Speeding Radar*

Submission Date: 8th May 2020

Mohammad Khorasani

Motivation

The scope of this project has been to implement an image processing-based traffic radar that detects vehicle number plates and subsequently measures the instantaneous vehicle speed. This application of computational photography/image processing was selected in order to develop an open source and cost-effective alternative to current speeding radar systems that can carry a price tag upwards of \$6,500 per unit¹. As an open source technique, this will enable local authorities, municipalities, and facilities to implement their own low-cost and convenient traffic monitoring systems with off the shelf devices and equipment.

Approach

To implement this application, a set of easily accessible and relatively inexpensive items and programs were selected. Namely, an iPhone camera with 60 fps, 1080p resolution, exposure time of 1/120 seconds and an activated flashlight was used for nighttime monitoring, while a Nikon D7000 camera with 24 fps, 1080p resolution and an exposure time of 1/60 seconds was used for daytime monitoring. Both cameras were setup with an adjustable tripod that was situated immediately to the side of the road with an inclination of 0°, a horizontal angle of approximately 20° towards the road and an elevation of approximately 2 feet off the ground. A total of 10 test cases were experimented using 3 different vehicles of varying sizes and speeds ranging from 20kph to 70kph at various ambient lighting conditions ranging from morning to evening. The test matrix can be seen in figure 1. Subsequently the recorded continuous footages were processed using several Python scripts to detect and report the number plates and vehicle speeds; a detailed description of this step can be found in the proceeding 'Implementation' section.

Implementation

Once the footage was shot, it would then be processed by several scripts written in Python 3.7. These scripts make use of several open source Python toolkits, namely cv2, numpy and matplotlib. These toolkits were used to process the footage, to create template images, to compare pixels and to process output images, other than that all other algorithms were written originally. The implementation can be summarized as follows:

1. **Extract_frames.py**: fragment footage into individual frames
2. **Resize_numbers.py**: create templates of numbers 0-9 with varying sizes and lighting conditions (executed only once)
3. **Main.py**:
 - a. Determine ambient lighting conditions (day/night)
 - b. Compare each template (numbers of varying sizes and light) to each patch within each extracted frame using the normalized correlation coefficient algorithm
 - i. If similarity score exceeds a certain threshold, record the number and location
 - ii. If distance between locations of numbers exceed a certain threshold, disregard any new numbers (false number rejection)
 - iii. Append detected digits in order of appearance along x-axis
 - iv. If length of detected number equals six digits, register digits as number plate
 - c. Count the number of frames where at least 1 digit is detected; use the speed measurement model to determine the vehicle speed based on the total number of frames with detected plate numbers
 - d. Display and save the image of the vehicle with the detected number plate and speed.
4. **Record_video.py**: optionally you may choose to recreate a video of the plate detection and speed measurement process using the output images from main.py.

As mentioned in step 3b, the technique used to compare template images to the source image for matching is the normalized correlation coefficient equation shown below, where NCC is the similarity score, T is the template image, S is the source image, x and y are coordinates in the source image, x' and y' are coordinated in the template image.

$$NCC(x, y) = \frac{\sum_{x', y'} T'(x', y') \cdot S'(x + x', y + y')}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} S'(x + x', y + y')^2}}$$

The benefit to using the normalized correlation coefficient is that, where the pixels in the template and source images both have lighter (+ve · +ve) or darker (-ve · -ve) intensities, the result will be a large positive value indicating a match due to the dot product. Whereas, if one image has a light intensity and the other a dark intensity (+ve · -ve), the result will be a large negative value indicating a mismatch. This process of comparing the correlations or convolutions of two images, renders this technique as being relatively insensitive to absolute differences and ambient lighting conditions.

As detailed in step 3c, the following speed measurement model (power regression) was used in the second iteration to determine the speed, where V is the vehicle speed and f is the number of frames with a detected plate number. The speed measurement technique was entirely novel in its inception and implementation without using any open source input.

$$V = 341 * f^{-0.891}$$

Results

The following test matrix in figure 1, displays the test cases and results for each. Please note, that the number of frames corresponds to the frames whereby the script was able to detect any and all numbers in the footage. The test cases were split into train-test datasets and were used to iteratively train a more accurate vehicle speed measurement model. The results for the first and second iterations are presented, respectively.

Test Case	Actual vehicle speed (kph)	Number of frames	1st iteration predicted speed (kph)	2nd iteration predicted speed (kph)	1st iteration error (%)	2nd iteration error (%)
1*	20	33	15	17	25	15
2	21	20	24	25	14	19
3	22	23	21	22	5	0
4	24	20	24	25	0	4
5	35	11	40	38	14	9
6	41	9	48	44	17	7
7	42	10	44	41	5	2
8	61	5	81	69	33	13
9	62	7	60	54	3	13
10	70	6	69	60	1	14
Average error (%)					11.7	9.6

Figure 1: Test matrix with results for 1st and 2nd vehicle speed measurement iterations.

*Test case 1 is the only experiment conducted at night during extremely poor ambient lighting conditions. For this case, the iPhone camera was used with 60 fps and 1/120 seconds exposure time; likewise, a different set of template images were used for the template matching and a different speed measurement model was used to determine the vehicle speed. As evident in the 2nd iteration speed measurement, the average error dropped to below 10% (9.6%), and consequently this could be regarded as a successful outcome, given that the industry standard for speeding violations offers a 10% margin of errorⁱⁱ. However, it should be stressed that these results were largely conducted at 24 fps, whereas a higher frame rate may have offered greater accuracy. With regards to number plate detection, all 10 test cases were read without error (a sample of results may be viewed in figure 3) which equates to an overall error rate of 0%. This therefore proves the potential of this image processing-based technique as being an effective and inexpensive (\$1,700) alternative to legacy speeding radars.

Challenges and Innovations

The main challenge posed by this project was the issue of ambient lighting conditions. Early on, it was apparent that varying lighting levels due to day, night and cloud cover would have a drastic effect on the detection of numbers in the source image based on comparison to the template images. This was solved by using the normalized correlation coefficient algorithm with varying thresholds and templates for matching for each number. As a result, the final implementation is far less susceptible to absolute lighting conditions. Furthermore, it was found that due to longer filming exposure times required at night, frames had a lower temporal resolution which adversely affected detection; this was solved by using a camera with a flash light, higher frame rate and lower exposure time as detailed in the 'approach' section. The innovations of this project were the development of algorithms for false number plate rejection (detailed in step 3bii) and vehicle speed measurement (detailed in step 3d). Both original techniques enable image processing to be able to reliably detect and measure number plates and vehicle speeds similar to a speeding radar, albeit with a fraction of the cost. The expected grade for this project is 100%.

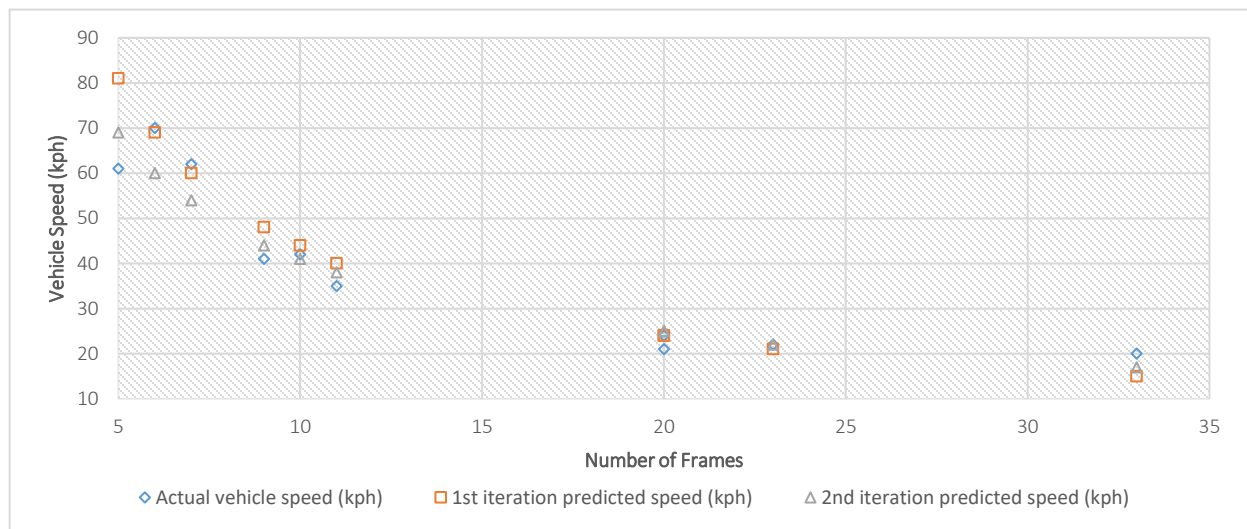


Figure 2: Graph of vehicle speed (actual and predicted) vs. number of frames with a detect number plate (see step 3c).

Links to number plate and speed detection videos:

- Video at 24fps: <https://www.youtube.com/watch?v=otAKOXekY5w>
- Video at 12fps: <https://www.youtube.com/watch?v=eGuRIqdugiU>

Number plate detection and speed measurement results:

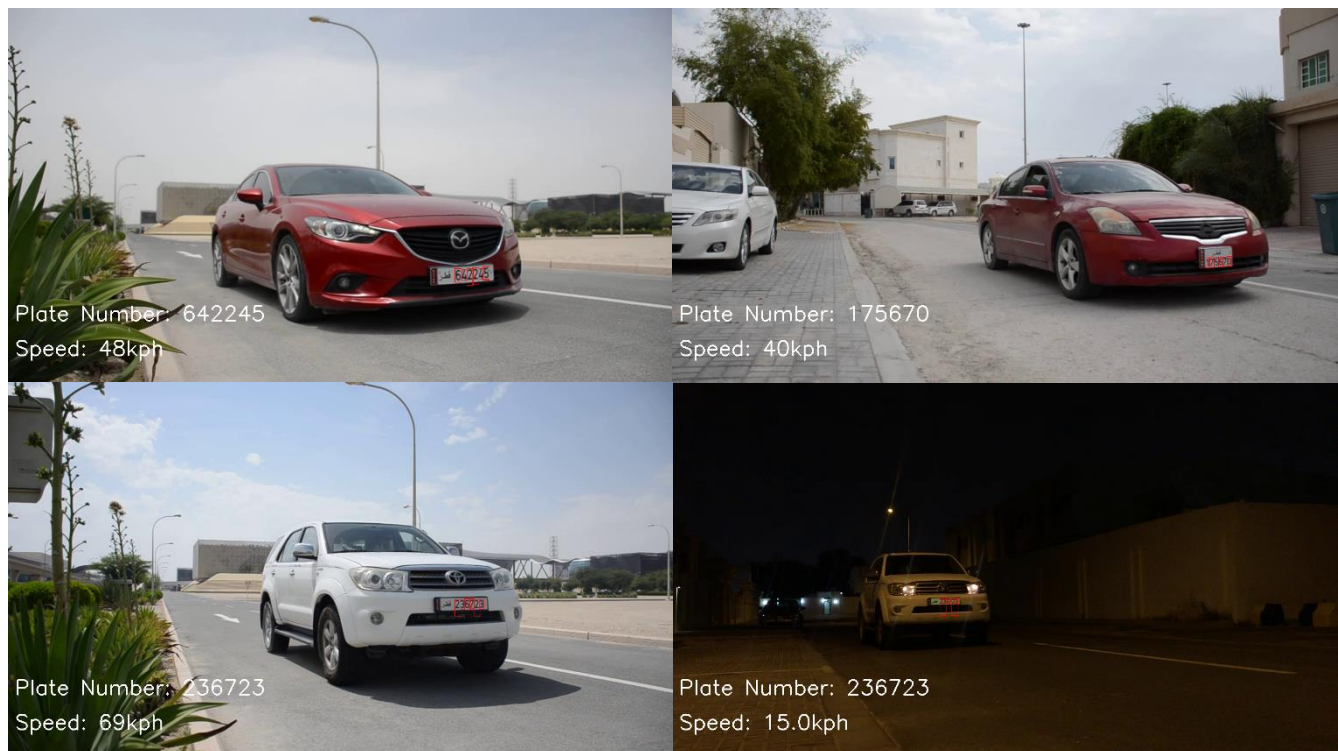


Figure 3: Output images of main.py for number plate detection and speed measurement.

ⁱ https://www.alibaba.com/product-detail/Speed-Radar-Detector-Best-Hidden-Cameras_60706967606.html?spm=a2700.7724857.normalList.70.1417bd44wfHNCI

ⁱⁱ <https://www.clm.co.uk/blog/new-tougher-speeding-fines/>