Learning to Recognize Distance to Stop Signs Using the Virtual World of Grand Theft Auto 5

Artur Filipowicz
Princeton University
229 Sherrerd Hall, Princeton, NJ 08544
T: +01 732-593-9067
Email: arturf@princeton.edu
corresponding author

Jeremiah Liu
Princeton University
229 Sherrerd Hall, Princeton, NJ 08544
T: +01 202-615-5905
Email: jerryliu@princeton.edu

Alain Kornhauser
Princeton University
229 Sherrerd Hall, Princeton, NJ 08544
T: +01 609-258-4657 F: +01 609-258-1563
Email: alaink@princeton.edu

Word Count: 7,209 words = 5,209 + 2,000 (3 Figures + 5 Tables)

Paper submitted for consideration for presentation for the 96th TRB Annual Meeting in January 2017 and for publication in the Transportation Research Record
ABSTRACT

This paper examines the use of a convolutional neural network and a virtual environment to detect stop signs and estimate distances to them based on individual images. To train the network, we develop a method to automatically collect labeled data from Grand Theft Auto 5, a video game. Using this method, we collect a dataset of 1.4 million images with and without stop signs across different environments, weather conditions, and times of day. Convolutional neural network trained and tested on this data can detect 95.5% of the stops signs within 20 meters of the vehicle with a false positive rate of 5.6% and an average error in distance of 1.2m to 2.4m. We also discovered that the performance our approach is limited in distance to about 20m. The applicability of these results to real world driving are promising and must be studied further.
2 INTRODUCTION

With increases in automation of the driving task, vehicles are expected to safely navigate the same roadway environments as human divers. To meet this expectation, future driver assistance systems and self-driving vehicles require information to determine two fundamental questions of driving: the location where the vehicle should go next and when it should stop. To find an answer, a vehicle needs to know its relative position and orientation, not only with respect to a lane or other vehicles, but also with respect to the environment and existing roadway infrastructure. For example, an autonomous vehicle should be able to recognize the same unaltered stop signs that are readily recognized by human drivers and stop, just like human drivers.

In the last decade of autonomous vehicle research, a range of approaches for enabling vehicles to perceive the environment have emerged depending on the extent to which the existing roadway infrastructure is augmented from that which exists to serve today’s human drivers and the amount of historical intelligence that is captured in digital databases and is available in real-time to the autonomous vehicle system. While humans have recently begun to use turn-by-turn “GPS” systems that contain historically coded digital map data and freshly captured and processed traffic and other crowd-sourced intelligence to navigate to destinations, the continuous task of human driving is accomplished through a purely real-time autonomous approach in which we do not need any other information from the environment but a picture. In 2005, Princeton University’s entry in the DARPA Challenge, Prospect 11, followed this idea, using radar and cameras to identify and locate obstacles. Based on these measurements, the on-board computer would create a world model and find a safe path within the limits of a desired set of 2D GPS digital map data points (1). Other projects, such as (2), followed the same approach.

Since then the Google Car (3) vastly extended the use of pre-made digital data by creating, maintaining and distributing pre-made highly detailed digital 3D maps of existing roadway environments that are then used, in combination with real-time on-board sensors to locate the vehicle within the existing roadway infrastructure and, consequently, relative to important stationary elements of that infrastructure such as lane markings and stop signs. All of this being accomplished without the need for any augmentation of the existing roadway infrastructure. Other approaches, motivated by US DoT connected vehicle research, have proposed the creation of an intelligent infrastructure with electronic elements that could be readily identified and located by intelligent sensors, thus helping autonomous vehicles but being of little, if any, direct assistance to existing human drivers (4).

In this paper, we tackle the problem of when the vehicle should stop under the autonomous approach. More precisely, we examine a machine learning system, which mimics human vision, to detect stop signs and estimate the distance to them based purely on individual images. The hope is that such a system can be designed and trained to perform as well as, if not better, than a human in real-time while using acceptable computing resources. Importantly, we explore overcoming the traditional roadblock of training such a system: the lack of sufficiently large amounts and variations of properly labeled training data, by harvesting examples of labeled driving scenes from virtual environments, in this case, Grand Theft Auto 5 (5–7).
3 RELATED WORK

In recent years, a wide variety of computer vision and machine learning techniques are used to achieve high rates of traffic sign detection and recognition. Loy and Barnes use the symmetric nature of sign shapes to establish possible shape centroid location in the image and achieve a detection rate of 95% (8). In (9), the generalization properties of SVMs are used to conduct traffic sign detection and recognition. Results show the system is invariant to rotation, scale, and even to partial occlusion with an accuracy of 93.24%. Another popular technique uses color thresholding to segment the image and shape analysis to detect signs (10, 11). A neural network trained to perform classification on thresholded images obtains an accuracy of around 95% (10, 11). Lastly, another method employs both single-image and multi-view analysis, where a 97% overall classification rate is achieved (12).

Research on localization of traffic signs has gained attention more recently. In (13), the authors describe a real-time traffic sign recognition system along with an algorithm to calculate the approximate GPS position of traffic signs. There is no evaluation of accuracy regarding the calculated position. Barth et al. presents a solution for localization of a vehicle's position and orientation with respect to stop sign controlled intersections based on location specific image features, GPS, Kalman filtering and map data (14). Based on this method, a vehicle starting 50 m away from the target intersection can stop within a median distance of 1 cm of the stop line. A panoramic image-based method for mapping road signs is presented in (15). The method uses multiple images for localization and is able to localize 85% of the signs correctly. Accuracy of the calculated position is not reported.

Similarly, Timofe et al. (12) uses multiple views to locate signs in 3 dimensions with 95% of the signs located within 3 m of the real location and 90% of signs located within 50 cm. Their system uses 8 roof mounted cameras and runs at 2 frames per second. While they do discuss potential for a real time system running at 16 frames per second, they do not report localization accuracy (12). Theodosis et al. used a stereo camera for localizing a known size stop sign by mapping the relative change of pixels to distance. The accuracy of the calculated position is not presented in the results (16). Welzel et al. introduced and evaluated two methods for absolute traffic sign localization using a single color camera and in-vehicle Global Navigation Satellite System (GNSS). The presented algorithms in (17) are able to provide a reliable traffic sign position with accuracy between 0.2 m and 1.6 m within the range of 7 m to 25 m from the stop sign.

3.1 Direct Perception

Researchers achieved reasonable accuracy in localization with dependency on additional sensors such as GNSS or under certain weather and time conditions. Within the autonomous approach to autonomous driving, specific systems can be categorized based on the world model the system constructs. Classical categories include behavior reflex and mediated perception (18). Behavior reflex (19) approach uses learning models which internalize the world model. Pomerleau used this method to map images directly to steering angles (20). Mediated perception (10, 18, 21) uses several learning models to detect important features and then builds a world model based on these features. For example, a mediated perception system would use a vehicle detector, a pedestrian detector, and a street sign detector to find as many objects as it can in the driving scene and then estimate the location of all of these objects. As (19) points out, behavior reflex has diffi-
cultures handling complex driving situations while mediated perception often creates unnecessary complexity by extracting information irrelevant to the driving task.

With consideration for the shortcoming of these approaches, Chen et al. (19) proposed a direct perception approach. Direct perception creates a model of the world using a few specific indicators which are extracted directly from the input data. In (19), the computer learns a transformation from images generated in the open source virtual environment TORCS (22), to several meaningful affordance indicators needed for highway driving. These indicators include distance to the lane markings and the distances to cars in current and adjacent lanes. Chen et al. showed this approach works well enough to drive a car in a virtual environment, and generalizes to real images in the KITTI dataset (23), and to driving on US 1 near Princeton. We believe that an autonomous driving system can be designed with the camera playing the role of the human eye in a direct perception approach.

4 LEARNING STOP SIGN RECOGNITION AND LOCALIZATION

Following the direct perception (19, 24) and deep learning (25) paradigms, we construct a deep convolutional neural network (CNN) (26) and train it to detect and locate stop signs using a training set of images and ground truths from a video game’s virtual world.

4.1 The Learning Model

Our direct perception convolutional neural network is based on the standard AlexNet architecture (26) with modifications based on (19). It is built in Caffe (27) and consists of 280 × 210 pixel input image, 5 convolutional layers followed by 4 fully connected layers with output dimensions of 4096, 4096, 256, and 2. The two final outputs are a 0/1 indicator which is one when a stop sign is detected and a continuous variable that reflects the estimated distance to the stop sign. This distance is in the range of 0m to 70 m. If no stop sign is detected, the distance is set to 70 m. We normalize both outputs to the range of [0.1, 0.9]. The model’s 68.2 million unknown coefficients are evolved to find the global minimum of the Euclidean loss function using a stochastic gradient decent method with an initial learning rate of 0.01, mini-batches of 64 images, and 300,000 iterations. We call the resulting solution the Long Range CNN. We also fine tune the model coefficients for short distances by training the final coefficients of Long Range CNN for another 100,000 iterations on examples of stops signs within 40 meters. The output range is redefined to 0.9 representing 40 meters. We refer to this solution as the Short Range CNN.

4.2 Grand Theft Auto 5 and Data Collection

Large learning models, such as the one we are using, tend to be able to correlate complicated high dimensional input to a small dimensional output given enough data. Depending on the application, collecting large datasets can be very difficult. Often the limiting factor, especially in datasets of images, is annotating the images with ground truth labels. For our purpose, we would need a person to indicate if a stop sign is in an image and measure the distance to that stop sign. While a part of this process can be automated by using measuring tools, such as in (28), this presents additional limitations. Sensors may not function in all weather conditions, such as rain or
fog, and their output may still need to be interpreted by a human before the desired ground truth is determined. We overcome this problem by using a video game called Grand Theft Auto 5 (GTA5).

Virtual environments have been used by (29) to create a system for construction 3D bounding boxes based on 2D images and (19) to learn distances to road lane markings and cars. For our application, GTA5 provides a rich road environment from which we can harvest vast amounts of data. GTA5 has a 259 square kilometer map (5) with a total population of 4,000,000 people. The game engine populates this world with 262 different types of vehicles (7), and 1,167 different models of pedestrians and animals. There are 77,934 road segments (6) which make up a road network of bridges, tunnels, freeways, and intersections in urban, suburban, rural, desert and woodland environments. Additionally, GTA5 has 14 weather conditions and simulates lighting conditions for 24 hours of the day, see Figure 1. Five models of traffic lights and many traffic signs, each according to US standards, are available. Visually, the scenes generated in GTA5 appear realistic and in proportion to the real world. Table 1 provides a side by side comparison of a driving scene with a stop sign at different distances in GTA5 and the real world.

Unlike previously used virtual worlds, GTA 5 is a closed source game. There is no out-of-the-box access to the underlying game engine. However, due to the game’s popularity, fans have hacked into it and developed a library of functions for interacting with the game engine. This is done by the use of scripts loaded into the game. Two tools are needed to write scripts for GTA
**TABLE 1**: Sample images of a stop sign in GTA5 and the real world at distances 70m, 60m, 40m, 20m, and 10m, from top to bottom.
The first tool is ScriptHook by Alexander Blade. It comes with a very useful trainer which provides basic control over many game variables including weather and time control. The next tool is a library called Script Hook V .Net by Patrick Mours which allows the use of C# and other .Net languages to write scripts for GTA 5.

To make the data collection more realistic in-game vehicle with a mounted camera is used; similar to (23). The user starts data collection by marking two points in the game as the start line, two points for the end line, and one point for the location where the stop sign model should be generated. The start line and the end line are perpendicular to the heading of the road. Additionally, the end line is next to the stop sign as it is used to compute the distance to it. The rest of the data collection process is automated and as follows. The test vehicle is spawned at a random point on the start line and driven by in-game AI to a random point on the end line. As the vehicle is driving, pictures and ground truth data are recorded at around 30 frames per second. Once the vehicle reaches the end line, the stop sign model is turned invisible and the vehicle travels the same route. This creates a set of examples with a clear distinction of what is a stop sign. Once the two passes are made, the hour and weather are changed and the vehicle is set to travel between another pair of random points. This is repeated for each of 24 hours and 5 weather conditions; extra sunny, foggy, rain, thunder, and clear. The process takes about 40 minutes to complete and generates about 80,000 properly labeled images.

The ground truth label for each image includes the indicator, the distance, the weather and the time. The weather and time are recorded based on the current state of the game. The indicator is set to 1 if a stop sign is visible in the image, otherwise it is set to 0 and the distance is set to 70m. When a stop sign is visible, the distance to it is computed as the distance between the center of the camera’s near clipping plane and the point on the ground next to the stop sign defined by the intersection of the end line and the heading of the vehicle. While this measure does not take into account the curvature of the road and thus is not necessarily the road distance to the stop sign, we chose this measure for three reasons. First, we did not find a way to compute the road distance in the game. Second, this measure is easier to interpret than the direct distance between the stop sign and the vehicle. Third, the actual difference in measurement is small.

This method collected a total of 1,426,190 images from 25 different locations. We used over one million images for training and the remainder for testing. Also created was a data subset for training and testing on examples of stop signs within 40 meters, as well as subsets for day and night. Training and testing were performed on completely different locations. The images are distributed almost uniformly across time and weather. The total time spent collecting these data was about 10 hours.

5 RESULTS

On GTA5 test examples, The Short Range CNN outperforms the Long Range CNN in both detection accuracy and absolute error (AE), the absolute value of the difference between the the network estimated distance and the ground truth distance. It achieves state-of-the-art comparable accuracy of 96.1% and 94.9% on ranges 0m to 10m and 10m to 20m with an average error in distance of 2.2m and 3.3m respectively and a 5.6% false positive rate. The performance of both

\(^1\)http://www.dev-c.com/gtavscripthookv/
\(^2\)https://github.com/crosire/scripthookvdotnet
CNNs degrades substantially beyond 20m, with about a 10 percentage points drop in accuracy for range 20m to 30m and 40 percentage points drop in accuracy for subsequent ranges. In general, both models perform poorly on our small real world dataset of 239 images. Although the Short Range CNN did achieve an accuracy of 96.2% on 0m to 10m range with 15% false positive rate. Please contact the authors for a fully report of error distributions and accuracy.

5.1 Long Range CNN

The Long Range CNN maintains above 90% accuracy within 20m of the stop sign with a high false positive rate of 31.3%, Table 2. In terms of estimating the distance to the stop sign, the Long Range CNN is on average off by 10.2m and 8.9m in the first two ranges, Table 2. Since the network outputs 70m when it believes there is not stop sign, it is instructive to see the mean absolute error when these values are removed. The Long Range CNN is off by 5.5m and 6.7m on average when it correctly detects a stop sign in the first two ranges. These errors are too high considering the proximity of the stop sign.

<table>
<thead>
<tr>
<th>Range</th>
<th>Accuracy</th>
<th>False Negative Rate</th>
<th>False Positive Rate</th>
<th>Mean AE (m)</th>
<th>Median AE (m)</th>
<th>Mean AE (m) when correct</th>
<th>Median AE (m) when correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>0m - 10m</td>
<td>0.903</td>
<td>0.097</td>
<td>na</td>
<td>10.2</td>
<td>4.0</td>
<td>5.5</td>
<td>3.8</td>
</tr>
<tr>
<td>10m - 20m</td>
<td>0.934</td>
<td>0.066</td>
<td>na</td>
<td>8.9</td>
<td>4.8</td>
<td>6.7</td>
<td>4.4</td>
</tr>
<tr>
<td>20m - 30m</td>
<td>0.801</td>
<td>0.199</td>
<td>na</td>
<td>14.6</td>
<td>9.4</td>
<td>9.2</td>
<td>7.0</td>
</tr>
<tr>
<td>30m - 40m</td>
<td>0.512</td>
<td>0.488</td>
<td>na</td>
<td>18.9</td>
<td>18.2</td>
<td>11.0</td>
<td>10.4</td>
</tr>
<tr>
<td>40m - 50m</td>
<td>0.489</td>
<td>0.511</td>
<td>na</td>
<td>18.4</td>
<td>19.4</td>
<td>19.9</td>
<td>20.8</td>
</tr>
<tr>
<td>50m - 60m</td>
<td>0.396</td>
<td>0.604</td>
<td>na</td>
<td>17.7</td>
<td>14.3</td>
<td>28.1</td>
<td>31.0</td>
</tr>
<tr>
<td>60m - 70m</td>
<td>0.501</td>
<td>0.499</td>
<td>na</td>
<td>25.5</td>
<td>19.6</td>
<td>45.1</td>
<td>49.1</td>
</tr>
<tr>
<td>&gt;70m</td>
<td>0.687</td>
<td>na</td>
<td>0.313</td>
<td>16.9</td>
<td>7.2</td>
<td>5.1</td>
<td>1.3</td>
</tr>
</tbody>
</table>

na = not applicable

5.2 Short Range CNN

The Short Range CNN maintains above 95% accuracy within 20m of the stop sign. It is 79.8% accurate within 20m to 30m, but effectively guesses the presence of stop signs beyond 30m. The false positive rate is low at 5.6%, Table 3. In terms of estimating the distance to the stop sign, the Short Range CNN is on average off by 2.2m and 3.3m in the first two ranges. Figure 2 shows the error distribution for the first range. This error deceases to 1.2m and 2.4m when only correctly identified examples are considered, Table 3. The Short Range CNN performs better during the day than at night, Table 4. The false positive rate is 3% during the day and 10.5% during the night. The error in distance estimates increases from day to night by 1m to 1.9m.
### TABLE 3: Performance of Short Range CNN on Short Range Test Set

<table>
<thead>
<tr>
<th>Range</th>
<th>Accuracy</th>
<th>False Negative Rate</th>
<th>False Positive Rate</th>
<th>Mean AE (m)</th>
<th>Median AE (m) when correct</th>
<th>Median AE (m) when correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>0m - 10m</td>
<td>0.961</td>
<td>0.039</td>
<td>na</td>
<td>2.2</td>
<td>0.9</td>
<td>1.2</td>
</tr>
<tr>
<td>10m - 20m</td>
<td>0.949</td>
<td>0.051</td>
<td>na</td>
<td>3.3</td>
<td>1.7</td>
<td>2.4</td>
</tr>
<tr>
<td>20m - 30m</td>
<td>0.798</td>
<td>0.202</td>
<td>na</td>
<td>4.7</td>
<td>3.4</td>
<td>3.1</td>
</tr>
<tr>
<td>30m - 40m</td>
<td>0.440</td>
<td>0.560</td>
<td>na</td>
<td>3.4</td>
<td>2.6</td>
<td>3.1</td>
</tr>
<tr>
<td>&gt; 40m</td>
<td>0.944</td>
<td>na</td>
<td>0.056</td>
<td>1.8</td>
<td>0.2</td>
<td>0.9</td>
</tr>
</tbody>
</table>

*na = not applicable*

### TABLE 4: Performance of Short Range CNN on Short Range Day and Night Test Sets

<table>
<thead>
<tr>
<th>Range</th>
<th>Accuracy</th>
<th>False Negative Rate</th>
<th>False Positive Rate</th>
<th>Mean AE (m)</th>
<th>Median AE (m) when correct</th>
<th>Median AE (m) when correct</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Day</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0m - 10m</td>
<td>0.980</td>
<td>0.020</td>
<td>na</td>
<td>1.6</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>10m - 20m</td>
<td>0.974</td>
<td>0.026</td>
<td>na</td>
<td>2.7</td>
<td>1.5</td>
<td>2.3</td>
</tr>
<tr>
<td>20m - 30m</td>
<td>0.858</td>
<td>0.142</td>
<td>na</td>
<td>4.1</td>
<td>2.9</td>
<td>2.7</td>
</tr>
<tr>
<td>30m - 40m</td>
<td>0.618</td>
<td>0.382</td>
<td>na</td>
<td>3.1</td>
<td>2.2</td>
<td>2.7</td>
</tr>
<tr>
<td>&gt; 40m</td>
<td>0.970</td>
<td>na</td>
<td>0.030</td>
<td>1.2</td>
<td>0.2</td>
<td>0.6</td>
</tr>
</tbody>
</table>

| **Night** |          |                     |                     |             |                             |                             |
| 0m - 10m  | 0.936    | 0.064               | na                  | 2.9         | 1.0                         | 1.6                         |
| 10m - 20m | 0.916    | 0.084               | na                  | 3.9         | 2.1                         | 2.7                         |
| 20m - 30m | 0.727    | 0.273               | na                  | 5.4         | 4.2                         | 3.5                         |
| 30m - 40m | 0.229    | 0.771               | na                  | 3.6         | 3.0                         | 4.2                         |
| > 40m     | 0.895    | na                  | 0.105               | 3.1         | 0.4                         | 1.5                         |

*na = not applicable*
5.3 Real World Data

On the single track of real world data, the Long Range CNN has a false positive rate of 65%. Distance estimation is equally poor with large absolute error even when the network correctly recognizes a stop sign. The Short Range CNN has a false positive rate of 15%, which make its high accuracy of 96.2% in 0m to 10m range significant. However, the network could not recognize the stop sign beyond 10m, Table 5. Even in the 0m to 10m range, the Short Range CNN has a high error in distance estimation of 8.8m on average, Table 5.

| TABLE 5: Performance of Short Range CNN on Real World Data |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Range       | Accuracy    | False Negative Rate | False Positive Rate | Mean AE (m) | Median AE (m) when correct | Median AE (m) when correct |
| 0m - 10m    | 0.962       | 0.038           | na                 | 8.8         | 7.4                  | 7.8                  |
| 10m - 20m   | 0.344       | 0.656           | na                 | 22.0        | 22.1                | 21.8                |
| 20m - 30m   | 0.000       | 1.000           | na                 | 13.5        | 12.9                | -                   |
| 30m - 40m   | 0.093       | 0.906           | na                 | 4.3         | 2.7                 | 11.2                |
| > 40m       | 0.850       | 0.150           | na                 | 4.1         | 0.2                 | 1.0                 |

na = not applicable, - = no results due to missing data

6 DISCUSSION

An important part of the discussion is visual observations made using a program for examining the output of the network in real time. Figure 3 shows the user interface of this program. The input image is visible in the upper right corner. On the left side is a visualization of the network output and ground truth. The blue rectangle and text show the location of the car and the distance as estimated by the network. The yellow rectangle is the true location of the vehicle, and the white
text shows the true distance to the stop sign. This program can be used to assess the robustness of the network while driving in the real world.

![FIGURE 3](image)

**FIGURE 3**: User interface of the program for observing the real-time output of the CNN. Blue rectangle and text show the output of the network. The yellow rectangle is the true location of the vehicle, and the white text shows the true distance to the stop sign.

### 6.1 The Use of a Virtual World

The use of GTA5 has three benefits. The first benefit is quick creation of a dataset of stop sign images with automatic ground truth annotation large enough to be used in training of large CNNs. The second benefit is the control of time and weather, adding variety to the dataset. The third benefit is improved performance due to the ability to take images of the same area with and without stop signs. In the first attempt to train a CNN, over a million images were collected by driving around the game looking for places with and without stops signs, and recording data at those random locations. The performance of this first long range CNN was around 55% for all distances on game data and 0% on the real world data. Employing a strategy of collecting data in the same location with and without a stop sign increased performance in the range of 0m to 20m to around 90% and 80% in the range 20m to 30m on game data, see Table 2. Performance also improved on real world data as well, see Table 3. It is very difficult if not impossible to obtain these benefits with real world data collection methods.

These unique and useful benefits beg the question of applicability of virtual worlds to real world situations. Due to time and technological constraints, this study is not able to definitively answer this question. On the real world data, the Long Range CNN has a false positive rate too high to be useful. However, the Short Range CNN has a false positive rate of 15%, and a 96%
accuracy for 0m to 10m, Table 5. This is promising, but the errors in distance for this network are high, Table 5. The reason for low accuracy may be the lamp post in front of the stop sign; the test images can be seen in Table 1. Both networks struggled with such occlusions in the game. In the 0m to 10m range where the Short Range CNN is very accurate, there may be a systematic error caused by differences in the game and real world camera models and data collection setups. Ultimately, more real world data is necessary to test the CNNs.

6.2 Effects of Distance

According to the test results, neither CNN can reliably detect a stop sign beyond 30 meters. With this approach and image resolution, the signal representing the stop sign is too small to be detected reliably. As seen in Table 1, at those distances the sign is barely visible to the human eye at this image resolution. Requiring the network to correlate these images causes noisy output at shorter distances as features the network learns from images in the far ranges are irrelevant to the task. The results from the Short Range CNN indicate this may be true. The false positive rate decreases by 26 percentage points and accuracy for 0 to 20 meters increases between 2 and 5 percentage points. The average error in distance deceases by 4 meters, Tables 2 and 3. Interestingly, the accuracy deceased in the ranges of 20m to 30m and 30m to 40m while the error in distance deceased on average 6m to 8m. This counterintuitive result requires further exploration.

6.3 Effects of Time and Weather

The CNNs work decisively better during the day. Table 4 details the differences in performance of the Short Range CNN during the day and night. The Long Range CNN has similar differences. There are two reasons which may be responsible for these differences. Red taillights might be increasing the false positive rate during the night. The test vehicle’s headlines do not illuminate the stop sign at distances further than 20 meters. Even at closer distances, the stop sign may not be illuminated making it difficult to see. It is possible to change the headlights in the game and a more realistic configuration should be added. In regards to weather, considering that the number of examples across all the weather conditions have been kept proportional across all the training and test sets, it does not appear that any one weather condition is particularly adverse.

6.4 Effects of Stop Sign Position and Other Observations

There are several important observations made while watching the network outputs in regard to the position and occlusion of stop signs. The network can detect a stop sign which is partially off the screen. However, it is not successful at detecting a stop sign which is partially occluded by a vehicle. The design of the dataset does not specifically plan occlusions. Perhaps this is a case which needs to be more represented in the dataset. Additionally, the network struggles to detect stop signs on the left side of the road, since they appear mostly on the right side. A source of some false positives are trees and poles in built environments and a source of some false negatives are pedestrians. All of these observations suggest that the dataset could be intelligently expended to increase accuracy.
7 CONCLUSION

We examined a deep learning direct perception approach to stop sign detection and distance estimation on individual images. We developed a method to automatically collect labeled data from Grand Theft Auto 5 and assembled a dataset of 1.4 million images with and without stop signs across different environments, weather conditions, and times of day. Our method can detect 95.5% of the stop signs within 20 meters of the vehicle with a false positive rate of 5.6% and an average error in distance of 1.2m to 2.4m on game data. Performance of the CNN degrades beyond 20m and is poor on real world data requiring further investigation.

This study provides several avenues for future work. One of the most important avenues is transferring the performance to the real world. This requires the collection of a large dataset of real world images with measurements. The performance of the model might be improved by expanding the dataset to include more occlusions and stop sign positions. It would also be interesting to explore the precise effects of including weather and time variations. Further research should explore the use of larger images to see if the performance improves at greater distances and the use of temporal information. Once the limits of this approach for stop signs are reached and are useful, the scope of the task should be generalized. Instead of just stop sign detection and localization, the task should be stop object detection and localization, where a stop object could be a stop sign, red and yellow traffic light, railroad crossing and maybe even a pedestrian, police officer or crosswalk.

8 ACKNOWLEDGMENTS

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References


