End-to-End Learning of Vision-Based Obstacle Avoidance for Off-Road Robots

Yann LeCun*, Eric Cosatto, Jan Ben, Urs Muller
The Courant Institute, New York University
Net-Scale Technologies, Morganville, NJ

Navigation and obstacle avoidance for autonomous vehicles is a subject of active research. The wide diversity of the appearance of potential obstacles, and the variability of the surroundings, lighting conditions, and other factors, make the problem extremely challenging.

Recent efforts have attacked the problem by relying on a multiplicity of sensors, including laser range finders, Radars, and stereo cameras with dedicated hardware for depth map extraction. While active sensors make the problem considerably simpler, there seems to be an interest from potential users for purely passive systems that rely exclusively on camera input. Since active sensors tend to be rather bulky and power hungry, using only cameras allows levels of miniaturizations that are not otherwise possible.

Avoiding obstacle but relying solely on camera input requires solving a highly complex vision problem: mapping pixels from two (or more) cameras to a steering angle. The conventional approach is to look for vertical edge features, extract a depth map using one of the numerous methods for stereo matching, and make a steering decision based on the depth map thereby extracted. Unfortunately, the extreme variability of lighting conditions, and the highly unstructured nature of off-road obstacles (e.g. vegetation), make this approach all but impractical.

We demonstrate an *end-to-end* learning approach to this problem. The system was trained to directly map stereo images to steering angles by emulating the behavior of a human driver.

The robot is shown in figure 1. It is a 4-wheel drive truck platform, roughly 50cm in length. It contains two video cameras with analog video transmitters. The video is transmitted to a remote computer that processes the video and sends steering and throttle control to the truck via radio.

The training data was collected by having a human driver control the robot by watching the video camera output. Training sequences were collected from a wide variety of terrains, weather conditions, lighting conditions, and obstacle types (in parks and backyards). In all the training sequences, the robot was driven in straight line towards an obstacle. As the robot approached the obstacle, the human driver would veer to the left or to the right to avoid the obstacle. Data was recorded from the two cameras in YUV format at 320x240 pixels resolution, and 30 frames per second. The throttle and steering angles from the human driver were also recorded. Approximately 3,300 sequences with 70 frames each were collected, for a total of 225,000 frames. The training consisted of roughly 100,000 frames.

The learning systems consisted of a large convolutional neural network with 6 input planes of size 149x58 pixels. The six planes respectively contained the Y, U and V components for the left camera, and the Y, U and V components for the right camera. The images were formed by subsampling the raw 320x240 image by 2 horizontally and by 4 vertically. The first layer consisted of 6 feature maps connected to various combinations of the input maps through 3x3 kernels. This was followed by an averaging/subsampling layer (3x4 subsampling ratio), a convolutional layer with 24 feature maps and 5x3 kernels, a subsampling layer with 5x3 ratio, and convolutional layer with 100 feature maps and 4x9 kernels, and finally, an output layer with two outputs. The two outputs respectively coded for "turn left" and "turn right" signals. The network had 3.15 Million connections and about 72,000 trainable parameters. All the parameters were trained in supervised mode so as to minimize the discrepancy between the network output and the human-provided steering angle.

Figure 2 shows two snapshots from test sequences with the input images (top), the steering angle produced by the network (the light blue bar in the dark blue box), and the internal state of the network from the six input planes (left) to the two outputs (right).

The system's output is computed on a frame by frame basis with no memory of the past. While using multiple frames would seem like a logical thing to do, this is difficult to do in supervised mode because a close approximation of the desired steering angle is easily derived from the input data in training conditions, but not in testing conditions: the current steering angle can be easily extracted from multiple frames.

While good subjective performance was obtained on the test sequences driven by the human (with the system acting as a back-seat driver), the best test was to let the system drive the robot in real time.

The robot was test driven by the trained system on various terrains and through various obstacles. A complete propagation through the network took roughly 60ms on a standard Linux PC configuration, resulting in an overall performance of about 10 frames processed (and 10 steering angles produced) per second.

Objective measures of performance are difficult to come by, but subjective testing showed an excellent ability for the system to detect and avoid obstacles with a very simple, reactive, state-less driver. Videos of the system in action will be shown.

Figure 1: The off-road robot.

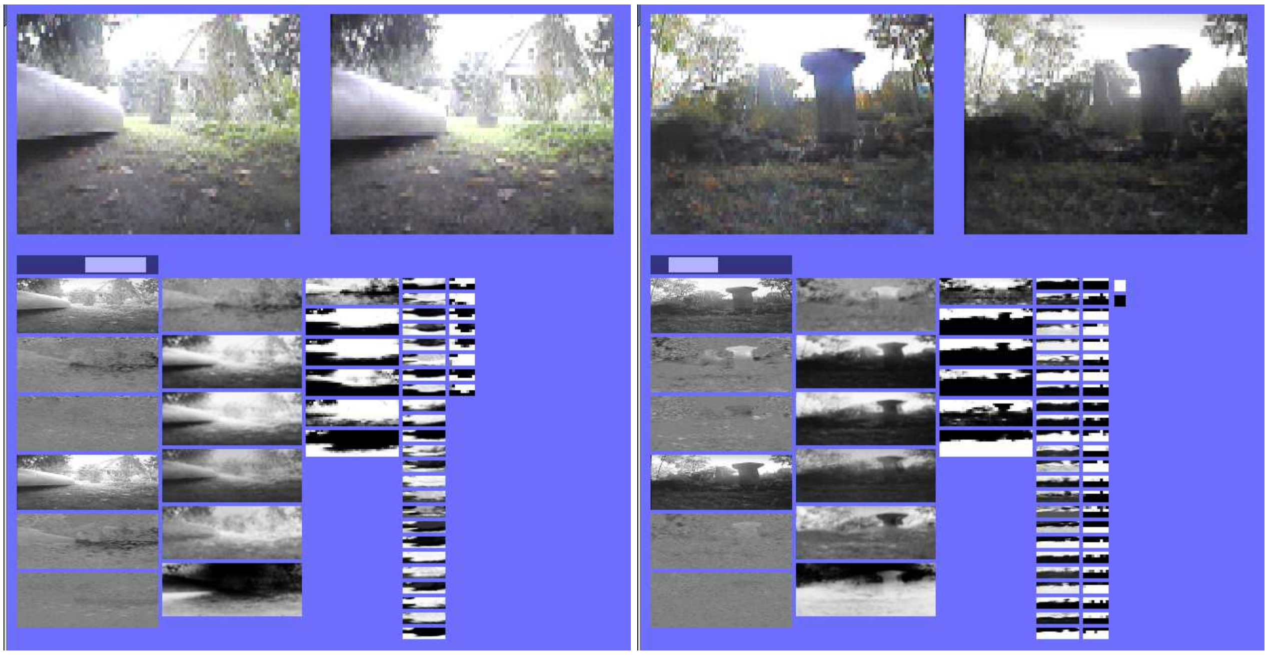

Figure 2: Examples of input images, network states, and computed steering angles.