Template-based autonomous navigation in urban environments

Jefferson R. Souza, Daniel O. Sales, Patrick Y. Shinzato,
Fernando S. Osório and Denis F. Wolf
Mobile Robotics Laboratory, University of Sao Paulo (USP)
Av. Trabalhador Sao-Carlense, 400 - P.O. Box 668 - 13.560-970 - Sao Carlos, SP, Brazil
{jrsouza, dsales, shinzato, fosorio, denis}@icmc.usp.br
http://www.lrm.icmc.usp.br/

ABSTRACT

Autonomous navigation is a fundamental task in mobile robotics. In the last years, several approaches have been addressing the autonomous navigation in outdoor environments. Lately it has also been extended to robotic vehicles in urban environments. This paper focus in the road identification problem, which is an important capability to autonomous vehicle drive. Our approach is based on image processing, template matching classification, and finite state machines processing. The proposed system allows to train an image segmentation algorithm in order to identify navigable and non-navigable regions (inside/outside roads), generating as output the steering control for an Electric Autonomous Vehicle, that should stay following the road. Several experimental tests have been carried out under different environmental conditions to evaluate the proposed techniques.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence:Robotics**]: Autonomous Vehicles.

General Terms

Algorithms, Performance, Design, Experimentation.

Keywords

Robotic Vehicles Navigation, Template Matching, Finite State Machines, Trapezoidal Algorithm and Urban Environments.

1. INTRODUCTION

Research in mobile robotics has reached significant progress in the last 10 years. Part of them focus on autonomous navigation, which is a fundamental task in the area [8]. Lately, several works have been improving on navigation in outdoor environments. Competitions like DARPA Challenges [4] and ELROB [1] have been pushing the state of the art in autonomous vehicle control.

Figure 1: Test platform used in the experiments.

The most relevant results obtained in such competitions combine information obtained from a large number of complex sensors. Some approaches use five (or more) laser range finders, video cameras, radar, differential GPS, and inertial measurement units [4], [5]. Although there are several interesting applications for such technology, the cost of such systems is very high, which is certainly prohibitive to commercial applications.

In this paper we propose a vision-based navigation approach for urban environments, based on a low/medium cost platform. Our system uses a single camera to acquire data from the environment, to detect the navigable regions (roads), estimating the maneuvers that should be done in order to keep the vehicle in a safe path, and finally, controlling the vehicle. Fig. 1 presents our Electric Autonomous Vehicle (EAV) test platform.

The images are acquired and then processed using an ANN that identifies the road ahead of the vehicle. Based on the ANN results, a template-based algorithm classifies the image and identifies the action that should be taken by the EAV. After that, a Finite State Machine (FSM) is used to filter some input noise and reduce classification and/or control errors.

This paper is organized as follows. Section 2 presents the related works. Section 3 describes the proposed method. Section 4 shows the experimental results and discussion. Finally, section 5 presents the conclusions and future works.

2. RELATED WORKS

Dahlkamp et al. proposes a method for identifying drivable surfaces in unpaved and off-road terrains as found in the DARPA Grand Challenge robot race [4]. The advantages of this approach is the robustness obtained by combining sensor information from a laser range finder, a pose estimation system, and a color camera. This method first identifies a nearby patch of drivable surface, then takes the patch and constructs appearance models to find drivable surface outward into the far range. After the model (map) is built, a quadrangle shaped as a trapezoid is fit to the largest drivable region in front of the vehicle. Several pre-processing techniques are also used such: extracting close range road location from sensors invariant to lightning conditions, removing sky and shadow areas from the visual field, learning a visual model of the nearby road, scoring the visual field by the road model, selecting identified patches, constructing a sky-view drivability map and finally, incorporating the drivability map into a driving strategy.

Another interesting work developing an autonomous vehicle control system, by Shihavuddin et al., approaches the path map generation of an unknown environment using a proposed trapezoidal approximation of road boundary [6]. At first, a blind map of the unknown environment is generated in computer, then the image of the unknown environment is captured by the vehicle and sent to the computer using a RF transmitter module. After that, the image is preprocessed and the road boundaries are detected using the trapezoidal approximation algorithm. During this process, the vehicle operates independently avoiding all obstacles.

The algorithm proposed in [6] generates maps with accurate road position. The issue with this approach is the dependency of the camera tilt angle, because the vehicle moves through of the trapezium and reaches the next approximated trapezium having a previously tilt angle.

3. PROPOSED METHOD

In this work, a vision-based navigation system is proposed (Fig. 2). Our approach is composed by 3 steps. In the first step an image is obtained and the road is identified using ANN classification (Fig. 3(c)). In the second step, a template matching algorithm is used to identify the geometry of the road ahead of the robot (straight line, soft turn, or hard turn). In the third step, a FSM is used to filter noisy inputs and any classification error, defining the action that the robot should take to keep on road. These steps will be described in the next sub-sections.

3.1 Image Processing Step

In the work [7], the performance of an Artificial Neural Networks (ANNs) was evaluated when applied into a road identification task. Based in the results obtained, a system composed by four Multilayer Perceptron (MLP) ANNs was proposed to identify the navigable region in urban spaces (Fig. 3 (a)). The result of this ANNs output combination is a navigability map (as shown in Fig. 3 (c)). The brighter is the area, the more likely it is to be navigable. The main advantage of this approach is that one can train the ANNs to identify different types of navigable and non-navigable regions (e.g. pavemented, non-pavemented roads, sidewalks).

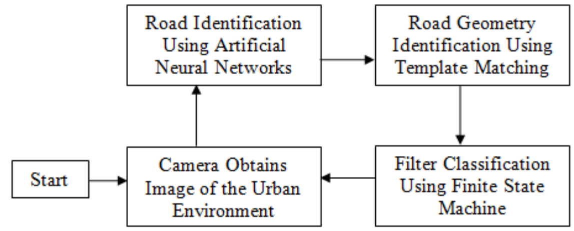


Figure 2: Representation of the proposed method.

Initially, the image processing step divides the image into blocks of pixels and evaluates then as single units. Several features are calculated for each block, such as: pixel attributes like red, green, blue (RGB) average, image entropy and others features obtained from this collection of pixels (region block). In the grouping step, a frame with (MxN) pixels resolution was sliced in groups with (KxK) pixels. Suppose an image represented by a matrix I of size (MxN). The element *I(m,n)* corresponds to the pixel in row m and column n of image, where ($0 <= m < M$) and ($0 <= n < N$). Therefore, group *G(i,j)* contains all the pixels *I(m,n)* such that $((i*K) <= m < ((i*K)+K))$ and $((j*K) <= n < ((j*K)+K))$. This strategy has been used to reduce the amount of data, allowing faster processing.

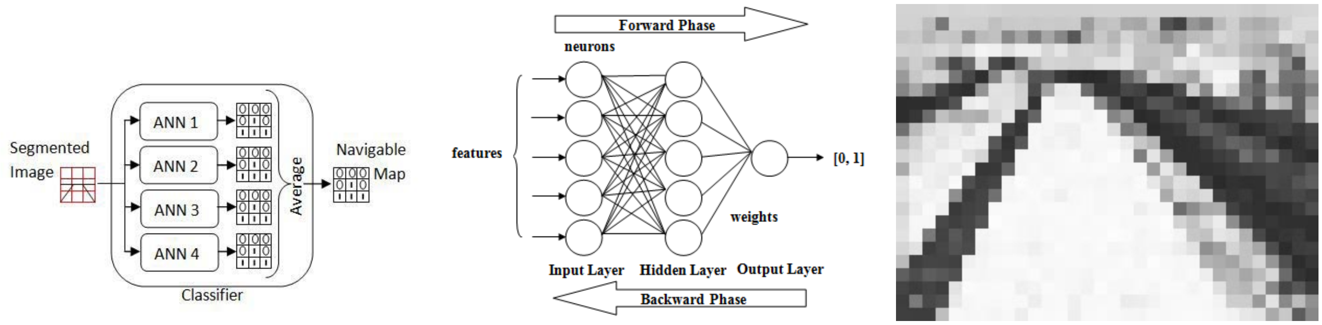
Once the block is processed, their attributes are used as inputs of the ANNs. Each ANN is used to classify the block considering their attributes (output 0 to non-navigable and output 1 to navigable). In this work, each ANN contains an input layer with four or five neurons (according to the image input features (see Table 1), one hidden layer with five neurons, and the output layer has only one neuron (binary classification). However, after the training step, the ANN return real values, between 0 and 1, as outputs. This real value can be interpreted as the classification certainty degree of one specific block. The main difference between the four ANNs is the set of image attributes used as input for each one. All these sets of attributes (see Table 1) are calculated during the block-segmentation of the image. The choice of these attributes was based on the results presented in [7].

Table 1: Input attributes of the ANNs (average = av, entropy = ent and energy = en).

ANNs	Input attributes
ANN1	R av, H av, H ent and V ent
ANN2	R av, B av, H av, V ent, HSV en
ANN3	B av, S ent, V ent, S en and HSV ent
ANN4	R av, G av, H av, H ent and V ent

Fig. 3 shows the classifier structure composed by 4 ANNs. It also presents the topology of the ANNs and a frame after being processed using the image processing method.

After obtaining the four outputs of the ANNs referring to each block, the classifier calculates the average of these values to compose a single final output value. These values representing each block obtained from the original image form together the navigability map matrix (Fig. 3(c)). This



(a) (b) (c)

Figure 3: Classifier structure (a), ANN topology (b) and Image processed (c)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(a)

0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(b)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(c)

Figure 4: The formats of the templates matching are: left (a), center (b) and right (c).

matrix is used to locate the most likely navigable region.

It is important to mention that the ANN is previously trained using supervised examples of navigable and non-navigable regions interactively selected by the user over a set of initial images. After that, the trained ANN is integrated into the vehicle control system and used as the main source of information to the autonomous navigation control system.

3.2 Template Matching Step

After obtain the ANN classification, 7 different road templates are placed over the image in order to identify the road geometry. One of them identify a straight road ahead, two identify a straight road in the sideways, two identify soft turns, and two identify hard turns. Each template is composed by a mask of 1s and 0s (Fig. 4). Each value of the mask is multiplied by the correspondent value into the navigability matrix (values obtained from the ANN classification of the correspondent blocks of the image). The total score for each template is the sum of products. The template that obtains the higher score is selected as the best match of the road geometry.

3.3 Finite State Machine Step

The developed FSM uses as input the result of the template matching step: a classification for the road detected in each captured frame. This classification is defined by the template which better fit the matrix and its position. The developed FSM is composed by 5 states (Straight Road = SR, Soft Left Turn = SLT, Soft Right Turn = SRT, Left Turn = LT and Right Turn = RT) as shown on Table 2.

Fig. 5 presents a partial scheme of the FSM with 2 intermediate transitions to illustrate how the transition between two states happens. Transition "a" represents classification of an input as a transition in direction of State 2, and "b" classification as a transition in direction of State 1.

Table 2: Finite state machine presenting the five states.

States	Associated Action
SR	Keep steering wheel at center position
SLT	Smoothly turn the steering wheel to left
SRT	Smoothly turn the steering wheel to right
LT	Fully turn the steering wheel to left
RT	Fully turn the steering wheel to right

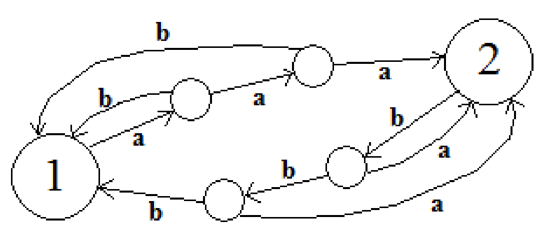


Figure 5: Transition between 2 states with 2 intermediate states.

The full FSM was designed with 5 states, which is obviously more complex to represent, but the partial scheme presented in Fig. 5 still remains representative of the full

FSM scheme. A transition between the current state to any other state occurs only after detecting a sequence of "n + 1" identical inputs leading to the new state, where "n" is the established number of intermediate states. The number of intermediate states varies according to the noise level in the images and navigability matrix representing the road, and also depends of the frame rate. In a system based on a high image acquisition frame rate, more misclassified inputs could be generated per time unit, so more intermediate states can be needed to discard some of these bad/noisy inputs. Environments with low level of input noise are associated to a smaller number of intermediate states.

4. EXPERIMENTAL RESULTS

The experiments were performed using the EAV shown in Fig. 1, an electric vehicle capable of autonomous navigation in a urban road, equipped with a VIDERE DSG video camera and a Roboteq AX2580 motor controller for steering control, and a Garmin 18X-5Hz GPS. The GPS was used only to register the log of the vehicle trajectory, and it was not used as input of the system. The image acquisition resolution was set to (320 x 240) pixels. The ANNs were executed using Fast Artificial Neural Network (FANN) [2], which is a free open source neural network library which implements multilayer artificial neural networks in C. For the development of the image acquisition, image processing and template matching algorithms, we used the OpenCV [3] library that has been widely used in computer vision algorithms. Finally, the finite state machines and other complementary developments where also made in the C Language.

The results presented here describe two experiments performed with our electric vehicle [1]. In the first one, the vehicle should follow the road pavement (asphalt surface), including straight, soft turn and hard turn path segments. In the second experiment the vehicle should navigate into a narrow straight path segment, following a surface composed by a red pavement (red bricks). Both trajectories have been logged and can be seen in Fig. 6. The first experiment is the longer path, and the second one is the shorter straight path, both described in terms of their absolute GPS coordinates. In both cases the vehicle was able to keep itself in the road successfully, following the road path defined by the region previously indicated and trained as the navigable surface.

In order to obtain a quantitative analysis of the system performance, 30 representative frames of the defined states for this problem have been manually classified and compared to the outputs of the proposed algorithms. As shown in Table 3, satisfactory results were obtained (67.4% overall correct classification result). Most errors are due to small variations in the classification, and occurred between neighbor states/templates: left turn (LT) and soft left turn (SLT); right turn (RT) and soft right turn (SRT).

Fig. 7 shows an example of the proposed identification method that resulted in 15 total state transitions detections (considering only the present frame), during a total sequence of 60 frames. The FSM proposed reduced this 15 state transitions to only 4 effective state transitions. One example of a wrong frame discarded by the FSM is showed in Fig. 7 (c), where the current state was kept. Even when traversing a crosswalk (marked with strips painted on the road) the vehicle could successfully keep moving in the correct direction.

Fig. 8 presents in details an example of straight, soft left and right turns, and left and right turns being accurately identified by our system, with successfully navigation inside the safe region defined by the pavement paths (navigable regions). Fig. 9 shows an example of a straight path being accurately identified by our system, with successfully navigation following the path defined by the red bricks surface.

[1]Experiment videos available in the Internet:
1st. Experiment - http://www.4shared.com/video/u8ObE9is/
2nd. Experiment - http://www.4shared.com/video/HRZIdqr7/

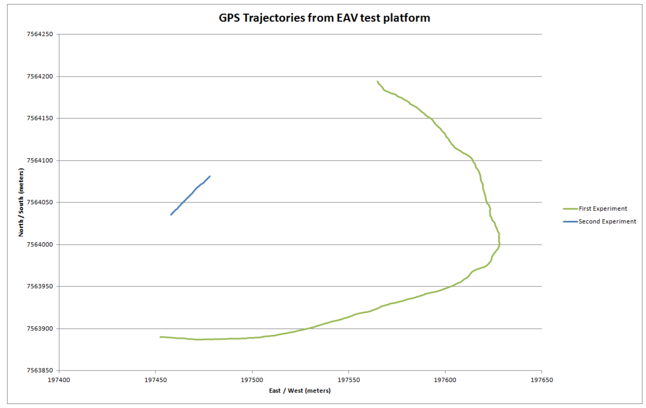


Figure 6: GPS trajectories from test platform.

Table 3: Results of the supervised classification of the trapezoidal algorithm with 30 frames for each state used in first experiment.

States	SR	LT	RT	SLT	SRT
SR	**93.4%**	0.0%	0.0%	6.6%	0.0%
LT	20.0%	**53.4%**	0.0%	26.6%	0.0%
RT	13.3%	0.0%	**83.4%**	0.0%	3.3%
SLT	46.6%	0.0%	0.0%	**53.4%**	0.0%
SRT	46.6%	0.0%	0.0%	0.0%	**53.4%**

5. CONCLUSION AND FUTURE WORKS

Autonomous vehicle navigation is a very important task in mobile robotics. This paper presented a vision-based system which can be trained to identify the road and navigable regions using ANNs, template matching classification and FSM. Our approach was evaluated using an Electrical Vehicle tested in outdoor road following experiments. The vehicle was able to navigate autonomously in this urban environment. Our quantitative analysis also obtained reasonable results for this road identification task.

As future works, we plan to evaluate other classification methods and decision making algorithms. We also planning to integrate camera and LIDAR information in order to better deal with obstacles, bumps and depressions in the road.

6. ACKNOWLEDGMENTS

The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), and FAPESP doctoral grant (process 2009/11614-4).

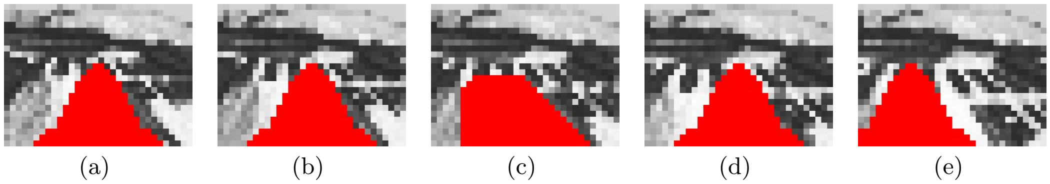


Figure 7: Identification by the proposed method in the Straight Road state (a), (b), (d) and (e). It suggests the Soft Left Turn state (c).

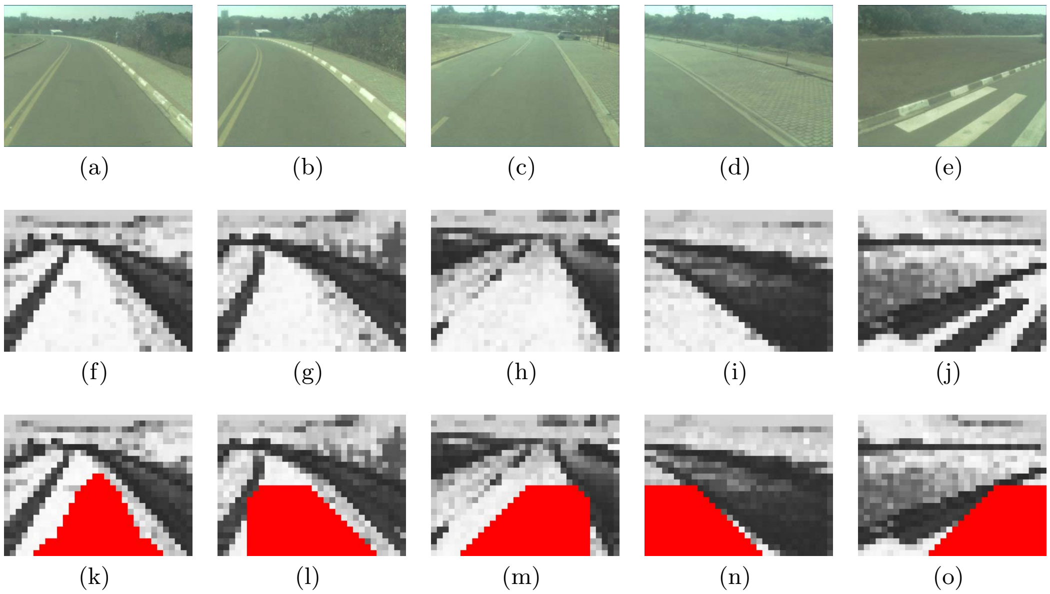


Figure 8: Image classification results using the test platform. Scene 1: Original Image (a), (b), (c), (d) and (e). Scene 2: Identification by ANNs (f), (g), (h), (i) and (j). Scene 3: Output of the proposed method (k), (l), (m), (n) and (o).

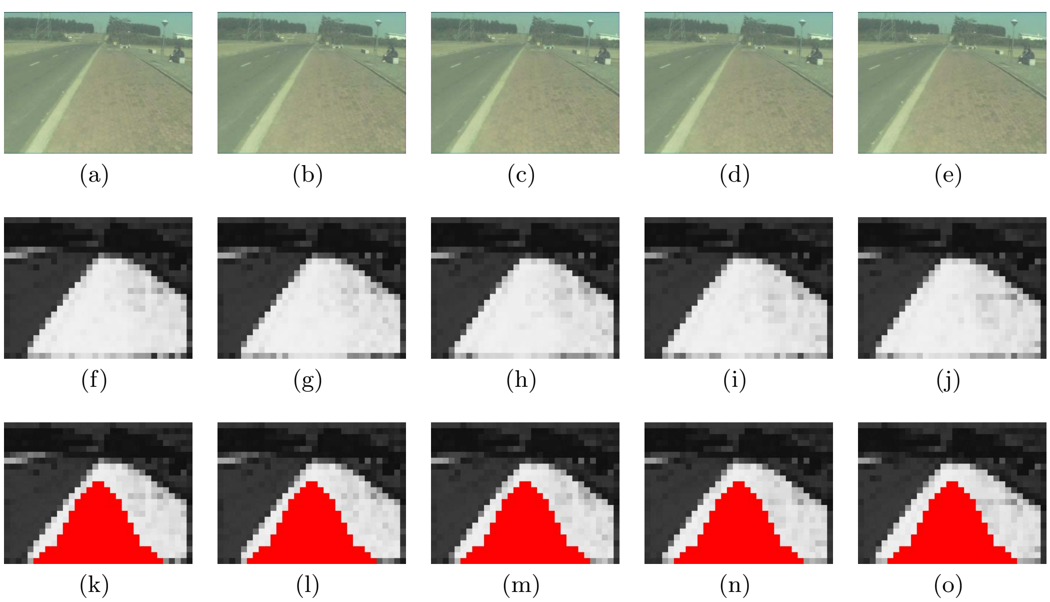


Figure 9: Image classification results using the test platform. Scene 1: Original Image (a), (b), (c), (d) and (e). Scene 2: Identification by ANNs (f), (g), (h), (i) and (j). Scene 3: Output of the proposed method (k), (l), (m), (n) and (o). We obtained these results in second experiment.

7. REFERENCES

[1] European land-robot (elrob), 2010. http://www.elrob.org/, Access on 18 june.

[2] Fast artificial neural network library (fann), 2010. http://leenissen.dk/fann/, Access on 02 june.

[3] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008.

[4] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In G. Sukhatme, S. Schaal, W. Burgard, and D. Fox, editors, *In Proceedings of the Robotics Science and Systems Conference*, 2006.

[5] A. Petrovskaya and S. Thrun. Model based vehicle tracking in urban environments. *IEEE International Conference on Robotics and Automation*, 2009.

[6] A. S. M. Shihavuddin, K. Ahmed, M. S. Munir, and K. R. Ahmed. Road boundary detection by a remote vehicle using radon transform for path map generation of an unknown area. *International Journal of Computer Science and Network Security (IJCSNS)*, 8(8):64–69, 2008.

[7] P. Y. Shinzato and D. F. Wolf. A road following approach using artificial neural networks combinations. *Journal of Intelligent and Robotic Systems*, 2010.

[8] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L. E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. Van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley, the robot that won the darpa grand challenge. *Journal of Field Robotics*, 2006.