2012 Intelligent Vehicles Symposium
Alcalá de Henares, Spain, June 3-7, 2012

Fast Visual Road Recognition and Horizon Detection Using Multiple Artificial Neural Networks

Patrick Y. Shinzato, Valdir Grassi Jr, Fernando S. Osorio and Denis F. Wolf

Abstract— **The development of autonomous vehicles is a highly relevant research topic in mobile robotics. Road recognition using visual information is an important capability for autonomous navigation in urban environments. Over the last three decades, a large number of visual road recognition approaches have been appeared in the literature. This paper proposes a novel visual road detection system based on multiple artificial neural networks that can identify the road based on color and texture. Several features are used as inputs of the artificial neural network such as: average, entropy, energy and variance from different color channels (RGB, HSV, YUV). As a result, our system is able to estimate the classification and the confidence factor of each part of the environment detected by the camera. Experimental tests have been performed in several situations in order to validate the proposed approach.**

I. INTRODUCTION

Visual road recognition, also known as "lane detection", "road detection" and "road following", is one of the desirable skills to improve autonomous vehicles systems. As a result, visual road recognition systems have been developed by many research groups since the early 1980s, such as [1] [2] [3]. Details about these and others works can be found in several surveys [4] [5] [6] [7].

Most work developed before the last decade was based on certain assumptions about specific features of the road, such as lane markings [8] [9], geometric models [10] and road boundaries [11]. These systems have limitations and in most cases they showed satisfactory results only in autonomous driving on paved, structured and well-maintained roads. Furthermore they required favorable conditions of weather and traffic. Autonomous driving on unpaved or unstructured roads, and adverse conditions have also been well-studied in the last decade [12] [13] [14] [15]. We can highlight developed systems for the DARPA Grand Challenge [16] like focusing on desert roads.

One of the most representative works in this area is the NAVLAB project [3]. Systems known as ALVINN, MANIAC and RALPH were also developed by the same research group. Among these systems, the most relevant reference for this paper are ALVINN and MANIAC because they are also based on artificial neural networks (ANN) for road recognition.

This work was not supported by any organization

P. Y. Shinzato, F. S. Osorio and D. F. Wolf is with Mobile Robotic Laboratory, Institute of Mathematics and Computer Science, University of Sao Paulo - ICMC-USP, Sao Carlos, SP, Brazil shinzato@icmc.usp.br, fosorio@icmc.usp.br, denis@icmc.usp.br

V. Grassi Jr. is with Dept. of Electrical Engineering, So Carlos School of Engineering, University of Sao Paulo (EESC-USP), Sao Carlos, SP, Brazil vgrassi@usp.br

The idea of ALVINN [17] consists of monitoring a human driver in order to learn the steering of wheels while driving on roads on varying conditions. This system, after several upgrades, was able to travel on single-lane paved and unpaved roads and multi-lane lined and unlined roads at speeds of up to 55 mph. However, it is important to emphasize that this system was designed and tested to drive on well-maintained roads like highways under favorable traffic conditions. Beyond those limitations, the learning step takes a few minutes [17] and the authors mention that when is necessary a retraining then this is a shortcoming [18]. According to [19], the major problem of ALVINN is the lack of ability to learn features which would allow the system to drive on road types other than that on which it was trained.

In order to improve the autonomous control, MANIAC (Multiple ALVINN Networks In Autonomous Control) [19] has been developed. In this system, several ALVINN networks must be trained separately on their respective roads types that are expected to be encountered during driving. Then the MANIAC system must be trained using stored exemplars from the ALVINN training runs. If a new ALVINN network is added to the system, MANIAC must be retrained. Both systems trained properly, ALVINN and MANIAC, can handle non-homogeneous roads in various lighting conditions. However, this approach only works on straight or slightly curved roads [12].

Other group that developed visual road recognition based on ANN was the Intelligent Systems Division of the National Institute of Standards and Technology [20]. They developed a system that make use of a dynamically trained ANN to distinguish between areas of road and nonroad. This approach is capable of dealing with nonhomogeneous road appearance if the nonhomogeneity is accurately represented in the training data. In order to generate training data, three regions from image were labeled as road and three others regions as nonroad, i.e., the authors made assumptions about the location of the road in the image, which causes problems in certain traffic situations. Additionally, this system works with the RGB color channel that suffers a lot of influence in the presence of shadows and lighting changes in the environment. A later work [21] proposed dynamic location of regions labeled as road in order to avoid these problems. However, under shadows situations, the new system becomes less accurate than the previous one because the dynamic location does not incorporate the road with shadow information in the training database.

In this work, we present a visual road detection system that uses multiple ANN, similar to MANIAC, in order to improve the robustness. However, our ANN learns colors and textures from sub-images instead of all road appearance like ALVINN does. In our system, each ANN receives different image features as input. Features like averages, entropy, energy and variance from differents color channels (RGB, HSV, YUV) from sub-images. In the final step of the algorithm, we combine a set of ANN's outputs to generate only one classification for each sub-image. Also, we estimate the height of the horizon line in image in order to improve the classification of all image. Another detail about this classification is that it provides a confidence factor for each sub-image classification that can be used by any control system. Unlike [20], our system does not need to be retrained all the time because the generalization capacity of our system is better than theirs. Therefore, our system does not require to make assumptions about the location of the road in the image.

II. System Description

The system's goal is to identify the road region on an image obtained by a video camera attached to a vehicle. To accomplish this task, our system is divided into four stages composed by different algorithms and parameters. The first stage, called **Features Generation**, transforms the image into set of sub-images and generates image features for each one. These features are used in the second and third stages called **Road Identification** and **Horizon Identification** to detect the road and the horizon line respectively. After that, into the last stage called **Combination**, our system combines the two *identifiers results* to provide a matrix that is the final "Visual Navigation Map" (VNMap). A control algorithm uses VNMap in order to control the vehicle autonomously. After executing an action, the system captures another image from the environment and returns to the first stage. The Fig. 1 shows how the system works.

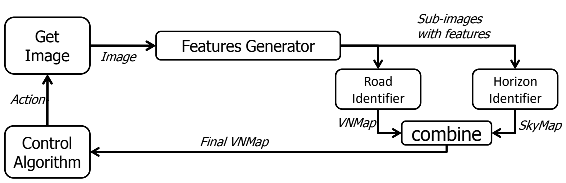
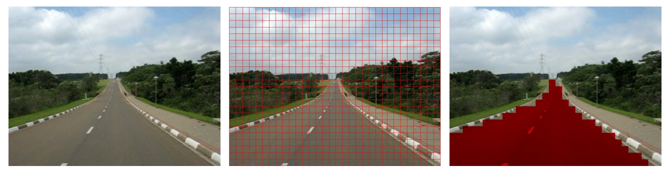


Fig. 1. The System Architecture: Given an image, it is transformed into a set of sub-images that will be separately classified by two types of identification. Combining all outputs from road and horizon Identifiers, our system provides the VNMap that is used by a control algorithm.

A. Features Generation

This stage transforms an image into a set of sub-images and generates image-features for each of them. More specifically, an image I with size of $(M \times N)$ pixels is decomposed in many sub-images with $(K \times K)$ pixels, as shows Fig. 2(a) which is transformed into Fig. 2(b). This subdivision is described as follows: The element $I(m,n)$ corresponds to the pixel in row m and column n of the image, where $(0 \leq m < M)$ and $(0 \leq n < N)$. Therefore, sub-image (i,j) is represented by group $G(i,j)$ that contains all pixels $I(m,n)$ such that $(iK \leq m < (iK+K))$ and $(jK \leq n < (jK+K))$. For each group, many image features are generated. These features will be used by road and horizon identifiers that determine whether the group, or sub-image, belongs to a particular class or not. If the sub-image is classified as belonging to a specific class, e.g. *road class*, then all pixels from that group are considered as belonging to the same class. Fig.2(c) shows sub-images belonging to road class painted in red. This strategy has been used to reduce the amount of data, allowing faster processing and obtaining information like texture from sub-images.



(a) (b) (c)

Fig. 2. In features generation stage, the image (a) is transformed into a set of sub-images, each one represented by a square in image (b). After the classification, we can obtain results like (c), where all pixels from a square receive the same classification. Red squares were classified as belonging to road class.

In this work, an image feature is a statistical measure computed for a group of pixels considering a specific color channel. We used the following statistical measures: mean, entropy, variance and energy. These measures can be associated with some channel from four different color spaces in order to define a feature: RGB, HSV, YUV and normalized RGB. The normalized RGB is composed by $(R/(R+G+B))$, $(G/(R+G+B))$, $(B/(R+G+B))$. The Table I shows all the combinations of statistical measures with color channels calculated by our system. The choice of these attributes was based on a feature selection method based on ANN and a previous work [22].

TABLE I

Features calculated by our system. Note that RN, GN, BN are RGB channels normalized.

Measure	Channels from several color spaces										
	R	G	B	H	S	V	Y	U	RN	GN	BN
Mean	×	×	×	×	×	×		×		×	×
Entropy				×	×	×	×			×	
Variance			×		×						
Energy					×					×	

B. Road and Horizon Identification

An identifier is responsible for classifying a sub-image as belonging or not to a specific class. It is composed by several different ANN, as shows Fig. 3. Each ANN uses some, not all, features generated by the previous stage as input. In other words, these ANN are distinguished by the combination of features used as input. The Section III will explain the ANN used here in more detail.

Our system uses two identifiers, one to detect road and other to detect horizon line. The major difference between these identifiers are the features and training database used by its ANN. Fig. 4 shows for the same image the difference

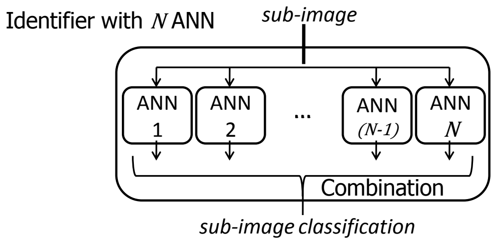
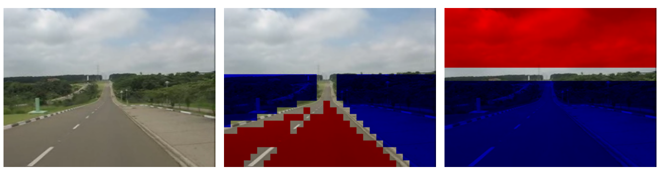


Fig. 3. An *identifier* is composed by multiple ANN. After combining all ANN outputs, is generated the classification for one sub-image. The classifications from all sub-images compose a *Map* that will be used by the system.

between the training data. Pictures (b) and (c) show sub-images with classification 1 (*detected*) painted in red and sub-images with classification 0 (*not detected*) painted in blue. This classification was manually defined in order to generate training database for both *identifiers*.



(a) (b) (c)

Fig. 4. Differences between training databases of *identifiers*, image (a) is original image, image (b) is classification for road training database and image (c) is classification for horizon training database. The pixels painted of red represent patterns that *identifier* must returns value "1". Pixels painted of blue represents patterns thar *identifier* must return "0".

The **Road Identifier** detects sub-images that represent road regions from image. In this stage, all the ANNs of the road identifier classifies each sub-image, and return a value ranging from 0 to 1. The classification of a sub-image is average of the results of all ANN outputs. After classifying all sub-images from an image, this stage generates a matrix Map of real numbers where $Map(i, j)$ is the classification of sub-image $G(i, j)$. Fig. 5 shows a sample of an image classified by this identifier, where the right image shows the VNMap in gray-scale - black represents non-road class, white represents road class and the gray represents the intermediate values.

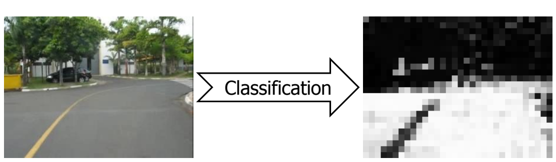


Fig. 5. Results from a classification sample from *road identifier*. Black represents non-road class, white represents road class and the gray represents the intermediate values.

The **Horizon Identifier** distinguishes sub-images above the horizon line from sub-images below it, as shows Fig. 6. This stage is similar to **Road Identification** but it uses different features, number of ANN and training database. After generating Map, it calculates the height of horizon line as follows:

$$height = \frac{\frac{X}{2} + Y}{W}, \tag{1}$$

where X is how many sub-images were classified with score higher than *threshold1* and less than *threshold2*, Y is how many sub-images were classified with score higher than *threshold2* and finally, W is the number of sub-images (columns) in one line of Map. These threshold values were defined empirically, where ($0 <$ *threshold1* $<$ *threshold2* < 1). This helps the system to distinguish X - that represents sub-images above horizon with low confidence - from Y - that represents sub-images above horizon with high confidence.

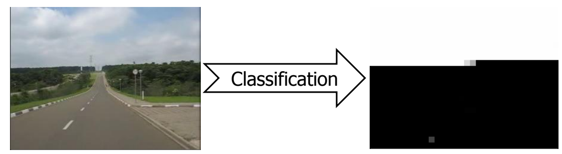


Fig. 6. Results from a classification sample from *horizon identifier*. Black represents non-sky class, white represents sky class and the gray represents the intermediate values.

After determining the position of the horizon, at the last stage our system updates the classification of all sub-images above the horizon line to zero in VNmap generated by **Road Identification**, as shows Fig. 7. This procedure improves the classification system and this VNmap can be used by some control algorithm. In order to improve system performance, we recommend the execution of the horizon identifier first, so that only the sub-images below horizon line will be submitted to the road identifier.

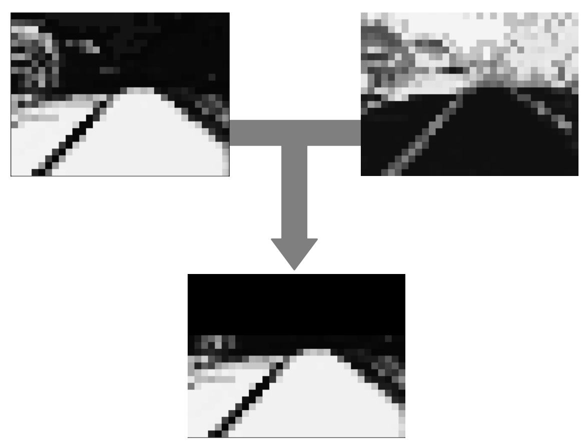


Fig. 7. Results from combination of classification from *horizon identifier* and *road identifier*. Note that the uncertainty above horizon was eliminated causing classification improvement.

III. Artificial Neural Network

The ANN used in our system consist of a multilayer perceptron (MLP) [23], which is a feedforward neural network model that maps sets of input data into specific outputs. We use the resilient propagation technique [24], which estimates the weights based on the amount of error in the output compared to the expected results.

The ANN topology consists in, basically, two layers, where the hidden layer has five neurons and the output layer has only one neuron, as shows the Fig. 8. We chose this topology based on previous evaluation [22]. The ANN was trained to return value 1.0 when receives patterns belonging to a determined class and to return 0.0 when receives patterns that does not belong to class. The input layer depends on features chosen for each ANN instance. Since the number of neurons is small, the training time is also reduced enabling the instantiation of multiple ANN.

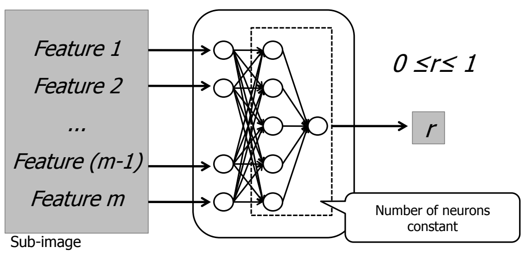


Fig. 8. ANN topology: The ANN uses some features, not all, to classify the sub-image between belonging to a class or not. The output is a real value ranging from 0.0 to 1.0, where if the value is closer to 1 then greater will be the confidence factor about belonging to class. If the value is closer to 0 then greater will be the confidence factor about not belonging.

Regarding ANN convergence evaluation, two metrics are frequently used: “MSE” and “Error Rate”. The MSE is “Mean-Square Error” and usually the training step stops when the “MSE” converges to zero or some acceptable value. However, a small mean-square error does not necessarily imply good generalization [24]. Also this metric does not provide how many patterns are missclassified, i.e., if the error is higher for some patterns or if the error is evenly spread for all patterns. Other way of evaluating the convergence is checking how many patterns were missclassified, or “Error Rate”. The problem of using this criteria is how to define an adequate threshold value to interpret the ANN output, i.e, given an ANN output varying from 0 to 1, determine whether this output belongs or not to the class being assessed.

In our work, in order to evaluate the convergence we propose a method that assigns a weight to the classification error, and compute a score for the ANN. More specifically, a greater weight is assigned to an ANN output with error of 0.1 than to output with error of 0.2. Therefore, a high score indicates few large errors or several small errors. The Equation 2 shows how to calculate the score:

$$score = \frac{\frac{1}{N*p(0)}\left(\sum_{i=0}^{hmax} h(i)*p(i)\right)+1}{2.0}, \tag{2}$$

where N is the number of evaluated patterns, $h(i)$ is the number of classifications with error varying from $\frac{i}{hmax}$ to $\frac{(i+1)}{hmax}$, *hmax* is the size of $h()$ and finally, $p()$ is the weight function. The value *hmax* determines a precision for the interpretation of ANN's output. In other words, for instance, if $hmax = 10$ then the output that has real value ranging from 0 to 1 will be divided into 10 classes of errors, one class for errors between 0 and 0.1, other class for errors between 0.1 and 0.2, and so on. Also, each class is directly related to a weight. To simplify the handling of the *score*, the weight always ranges from -1 to 1, thus the *score* ranges from 0 to 1. Where $score = 1$ means that all ANN outputs are correct. Therefore, the training step runs until *score* converges to 1 or some acceptable value.

IV. Experiments and Results

In order to validate the proposed system, several experiments were perfomed. Our setup for the experiments was a car equipped with an A610 Canon digital camera. The image resolution was (320×240) pixels. The car and camera were used only for data collection. In order to execute the experiments with ANNs, we used a Fast Artificial Neural Network (FANN) which is a C library for neural networks. The OpenCV library has been used in the image acquisition and to visualize the processed results from our system. The sub-image size used was $K = 10$, so each image has 768 sub-images.

Our system uses ten ANN during execution, six to *road identification* and the remaining four to *horizon identification*. In the training step, each ANN are trained five times until 5000 cycles for that our system select one with best score. In other words, our system selects 10 from 50 created ANN. The used *hmax* has value equals “10” and weight function $p()$ used was a linear function. The *road identification* uses only three sets of features as input, two ANN are instantiated for each set resulting six ANN. The sets of features used are:

- Mean of U, V, H and Normalized B; Entropy of H; Energy of Normalized G.
- Mean of V, G, U, R, H, Normalized B, Normalized G; Entropy of H and Y; Energy of Normalized G.
- Mean of U, Normalized B, V, S, H, Normalized G; Variance of B; Entropy of Normalized G.

The *horizon identification* uses four sets of features as input for ANN. The features sets used are:

- Mean of R, B and H; Entropy of V.
- Mean of R and H; Entropy of H and V.
- Mean of B; Entropy of S and V; Energy of S
- Mean of B; Entropy of V; Energy of S; Variance of S.

Several paths traversed by the vehicle were recorded using a video camera. These paths are composed by road, sidewalks, parking, buildings, and vegetation. Also, some stretchs presents adverse conditions such as dirt, traces of other vehicles and shadows (Fig. 9). Altogether, data were collected from eight scenarios. For each one, it was created a training database and an evaluating database. Also, we combined elements from these eight training data into a single database, and elements from the eight evaluating data into another single database. Thus, we tested the system with nine databases.

A. Evaluation using score

Score is a hit rate measure that varies from 0 to 1. In order to compare this method with another metrics well known,

Fig. 9. Samples of scenarios used in this work. Note the occurrence of shadows and how the colors of road changes in different lighting conditions.

we transforms the *score* into "error measure" calculating $(1 - score)$. The graphic from Fig. 10 shows the performance of our system in a boxplot format. Each column represents results from one evaluating database. To generate each column, the system was independently evaluated for each image from the same scenario. Among all image results, column shows minimun, quartile of 25%, median in blue, quartile of 75% and maximun.

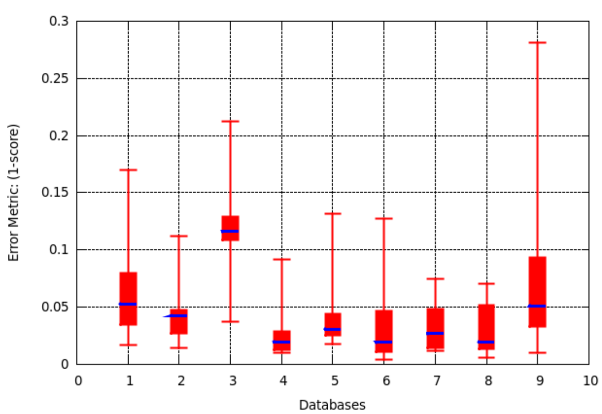


Fig. 10. Each column represents our system results from one evaluating database. Also, each column shows minimun, quartile of 25%, median in blue, quartile of 75% and maximun. The most important result that shows robustness of our system is the ninth column. This column shows that our system classifies different scenarios with good precision.

In general, the results were satisfactory. Except for the third scenario, all the medians were lower than 0.05 and based on analysis of quartiles, most of the images were classified with an error less than 0.1. The most important result that shows robustness of our system is the ninth column, because it represents the system performance for a database consisting of patterns of various scenarios under different conditions. Fig. 11 shows some results from the system trained with the database 9. We can see that our system has higher performance in different traffic and weather conditions, also different environments.

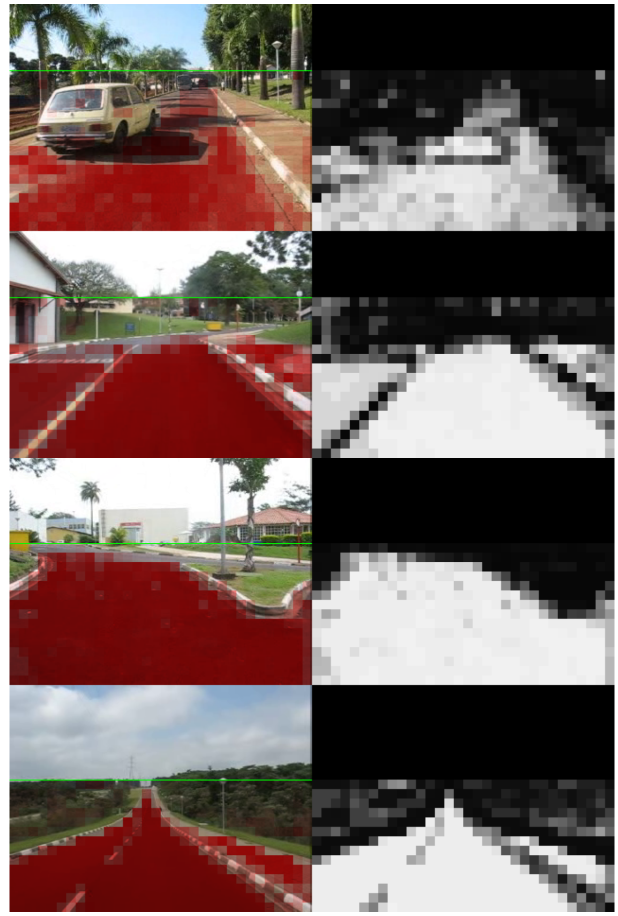


Fig. 11. Samples of results from our system when it is trained with database 9. This database has patterns from many different scenarios in different weather and traffic conditions. Our system was capable to differ road of non-road pixels. Also, in order to improve the classification, the green line shows where the system estimates the line horizon. All pixels above horizon line was non-road.

B. Comparison

We evaluate our system with 3 error metrics described in this paper, "MSE", "Error Rate" and "1 - score". We compared our results with the ANN described in [21] that uses only one ANN with 26 input neurons composed by a RGB binary (24 features) and normalized position of sub-image (2 features). Table II shows the comparison between our system and theirs for each database.

The ANN from [21] was better than our system in all evaluations when the system was trained and evaluated with patterns from the same scenery (database 1 to 8). However, when systems were trained with database 9, that contains patterns from all scenarios, our system was much better than theirs, since their ANN failed to learn. We can see this clearly in the column "Error Rate" where [21] ANN missclassified 99.7%. In other words, their system is better than our system for short paths. For longer paths or paths where the light conditions changes, our system is better because their system must be retrained over the path. The retraining is only possible if the system makes assumptions about the location of the road in the image, which can cause problems in certain traffic situations, or if the road is located using other sensors. For our vision system, these assumptions are not necessary.

TABLE II

TABLE OF ERRORS: COMPARISON BETWEEN OUR SYSTEM AND [21] FROM NIST THAT ALSO USING ANN.

Scenery	MSE		Error Rate		$1 - score$	
	LRM	NIST	LRM	NIST	LRM	NIST
Data 1	0.032	0.026	0.106	0.038	0.061	0.030
Data 2	0.023	0.017	0.071	0.029	0.046	0.022
Data 3	0.075	0.042	0.253	0.071	0.131	0.055
Data 4	0.018	0.035	0.047	0.039	0.031	0.035
Data 5	0.022	0.022	0.062	0.041	0.043	0.031
Data 6	0.021	0.010	0.051	0.016	0.035	0.013
Data 7	0.020	0.026	0.050	0.058	0.032	0.041
Data 8	0.020	0.017	0.044	0.032	0.031	0.023
Data 9	0.045	0.248	0.118	0.997	0.074	0.449

V. CONCLUSION

Visual road recognition is one of the desirable skills to improve autonomous vehicles systems. We presented a visual road detection system that uses multiple ANN. Our ANN is capable to learn colors and textures instead of totally road appearance. Also, our training evaluation method is a more adequate assessment to the proposed problem, since many classifications with low degree of certainty lead to low score. Our system was compared with another system and it performed better for longer paths with sudden lighthing condition changes. Finally, the system classification provides confidence factor for each sub-image. This information can be used by a control algorithm.

In general, the results obtained in the experiments were relevant, since the system reached good classification results when the training step obtains good score. Furthermore, the set of features presented in this paper can be used in different road types and weather conditions. It may be possible to get a better classification if we add a preprocessing to reduce the influences of the shadows in the image and maybe use more ANN in the system. As a future work, we plan to integrate it with other visual systems like *lane detection* in order to improve the system in urban scenarios. We also plan to integrate our approach with laser mapping in order to make conditions to retrain the ANN without human intervention and without making assumptions about the image. Finally, as the system classifies each block independently, we intend to improve the processing efficience using a GPU.

ACKNOWLEDGMENT

The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), processes 573963/2008-9, 08/57870-9 and 2010/01305-1.

REFERENCES

[1] A. Broggi, M. Bertozzi, and A. Fascioli, “Argo and the millemiglia in automatico tour,” *Intelligent Systems and their Applications, IEEE*, vol. 14, no. 1, pp. 55 –64, January 1999.

[2] R. Gregor, M. Lutzeler, M. Pellkofer, K. Siedersberger, and E. Dickmanns, “Ems-vision: a perceptual system for autonomous vehicles,” in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, 2000, pp. 52 –57.

[3] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, “Vision and navigation for the carnegie-mellon navlab,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 3, pp. 362 –373, May 1988.

[4] F. Bonin-Font, A. Ortiz, and G. Oliver, “Visual navigation for mobile robots: A survey,” *Journal of Intelligent & Robotic Systems*, vol. 53, pp. 263–296, 2008, 10.1007/s10846-008-9235-4.

[5] G. Desouza and A. Kak, “Vision for mobile robot navigation: a survey,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 2, pp. 237 –267, Feb. 2002.

[6] M. Bertozzi, A. Broggi, and A. Fascioli, “Vision-based intelligent vehicles: State of the art and perspectives,” *Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 1–16, 2000.

[7] E. D. Dickmanns, “Vehicles capable of dynamic vision: a new breed of technical beings?” *Artif. Intell.*, vol. 103, pp. 49–76, August 1998.

[8] A. Takahashi, Y. Ninomiya, M. Ohta, and K. Tange, “A robust lane detection using real-time voting processor,” in *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on*, 1999, pp. 577 –580.

[9] M. Bertozzi and A. Broggi, “Gold: a parallel real-time stereo vision system for generic obstacle and lane detection,” *Image Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 62 –81, Jan. 1998.

[10] J. Crisman and C. Thorpe, “Unscarf, a color vision system for the detection of unstructured roads,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, April 1991, pp. 2496 – 2501.

[11] A. Broggi and S. Bert, “Vision-based road detection in automotive systems: A real-time expectation-driven approach,” *Journal of Artificial Intelligence Research*, vol. 3, pp. 325–348, 1995.

[12] C. Tan, T. Hong, T. Chang, and M. Shneier, “Color model-based real-time learning for road following,” in *Intelligent Transportation Systems Conference, 2006. ITSC ’06. IEEE*, September 2006, pp. 939 –944.

[13] F. Diego, J. Alvarez, J. Serrat, and A. Lopez, “Vision-based road detection via on-line video registration,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, September 2010, pp. 1135 –1140.

[14] J. Alvarez and A. Lopez, “Novel index for objective evaluation of road detection algorithms,” in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, October 2008, pp. 815 –820.

[15] M. Manz, F. von Hundelshausen, and H.-J. Wuensche, “A hybrid estimation approach for autonomous dirt road following using multiple clothoid segments,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 2410 –2415.

[16] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA Grand Challenge: The Great Robot Race*, 1st ed. Springer Publishing Company, Incorporated, 2007.

[17] D. Pomerleau, “Efficient training of artificial neural networks for autonomous navigation,” *Neural Computation*, vol. 3, no. 1, pp. 88–97, 1991.

[18] D. Pomerleau and T. Jochem, “Rapidly adapting machine vision for automated vehicle steering,” *IEEE Expert*, vol. 11, no. 2, pp. 19 –27, Apr. 1996.

[19] T. Jochem, D. Pomerleau, and C. Thorpe, “Maniac: A next generation neurally based autonomous road follower,” in *Proceedings of the International Conference on Intelligent Autonomous Systems*, February 1993.

[20] M. Foedisch and A. Takeuchi, “Adaptive real-time road detection using neural networks,” in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, October 2004, pp. 167 – 172.

[21] ——, “Adaptive road detection through continuous environment learning,” in *Applied Imagery Pattern Recognition Workshop, 2004. Proceedings. 33rd*, October 2004, pp. 16 – 21.

[22] P. Shinzato and D. Wolf, “A road following approach using artificial neural networks combinations,” *Journal of Intelligent & Robotic Systems*, pp. 1–20, 2010, 10.1007/s10846-010-9463-2.

[23] P. S. Churchland and T. J. Sejnowski, *The Computational Brain*. Cambridge, MA, USA: MIT Press, 1994.

[24] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*, 2nd ed. Prentice Hall, July 1998.