Animal Detection from Traffic Scenarios Based on Monocular Color Vision

György Jaskó, Ion Giosan, Sergiu Nedevschi
Computer Science Department
Technical University of Cluj-Napoca, Romania
jgyuri2000@gmail.com, Ion.Giosan@cs.utcluj.ro, Sergiu.Nedevschi@cs.utcluj.ro

***Abstract*—This paper presents a system capable of detecting various large sized wild animals from traffic scenes. Visual data is obtained from a camera with monocular color vison. The goal is to analyze the traffic scene image, to locate the regions of interest and to correctly classify them for finding the animals that are on the road and might cause an accident. A saliency map is generated from the traffic scene image, based on intensity, color and orientation features. The salient regions of this map are considered to be regions of interest. A database is compiled from a large number of images containing different four-legged wild animals. Relevant features are extracted from these and are used to train Support Vector Machine classifiers. These classifiers provide an accuracy of above 90% and is used to predict whether or not the selected regions of interest contain animals. If one of the regions is classified as containing an animal, a warning can be signaled**.

Keywords—animal; car; street; road; detection; deer; support vector machine;

I. Introduction

The domain of driving assistance systems is one of great potential. Particularly in the last decade numerous companies, governments and research institutions have invested time, money and effort into the development of such systems, with the goal of enhancing or automating vehicles for safety and better driving. These safety measures include collision and accident avoidance, through either taking over the control mechanisms of the vehicle, or signaling the driver for possible dangers.

The field of computer vision is closely associated with driving assistance, visual sensors, e.g. cameras, are an essential part of data gathering from the environment. The type of visual data can range from infrared vision, providing information especially in low light conditions, to computer stereo vision, extracting 3D information from digital images by comparing images from multiple vantage points.

With the large amount of data becoming available through constant gathering from experimental results, machine learning and especially pattern recognition is often used for object classification, an important part of the surroundings analysis. The most common detection targets are vehicles, pedestrians, traffic signs, road surface, markings, etc. One area that would merit increased attention is the detection of wild animals in traffic. Roads often cut through the natural habitat of wildlife and without creating special crossings, wild animals often attempt to cross the road, leading to dangerous situations that endanger the safety of both humans and animals involved.

The severity of animal-vehicle collisions (AVCs) depends on multiple factors, vehicle speed and size of the animal amongst others. The most common and dangerous AVC is side collision with a large sized animal that attempts to cross the road, appearing spontaneously in front of the vehicle. Such animals include deer, moose, stag, reindeer, elk, etc. Governmental institutions, insurance companies and animal protection agencies have conducted studies to determine the impact of these accidents. In 2000, there were 1 million involved AVCs out of the 6.1 million reported lightweight motor vehicle collisions in the United States. Collisions with deer alone lead to 200 human deaths and $1.1 billion in property damage every year. In 2012, 1.23 million deer related accidents occurred during a one year period, causing on average over $3000 in property damage. Governments, insurance companies and drivers spend annually around $3 billion to reduce the number of accidents [2].

II. Related Work

Although the field of driving assistance and autonomous driving is in the center of focus of many research institutions, the area of animal detection is not always included amongst the goals. Some of the reasons why this topic is avoided includes lack of serious need for such detection (in mostly urban areas), lack of data or equipment, big variations amongst animals, and difficulty in testing such systems.

Volvo has implemented Large Animal Detection [8], which is part of their City Safety system. This software can identify large animals in front of the vehicle, viewed from the side while engaging in a normal movement pattern. Even this system however has its limitations, for example it cannot recognize animals seen from the front or behind, partially obscured large animals, large animals that run or move quickly, animals if the background contrast is poor or small animals.

Those that have attempted to find solutions have proposed different methods for this kind of problem. One category of solutions involve some kind of external sensor that communicates with the vehicles that are near their physical

locations. In this Vehicle-to-Infrastructure communication system the advantage is that it creates a network between the vehicles and the sensors, managing more complex coordination, providing data about traffic and other obstacles that are out of the line of sight of these vehicles. The major disadvantage is that the system is spatially bounded and requires external sensors. This type of system is proposed by Vishnu et al. in [3] which uses fixed cameras in intersections that communicate wirelessly with nearby vehicles. Their proposed architecture can be seen in Fig. 1.

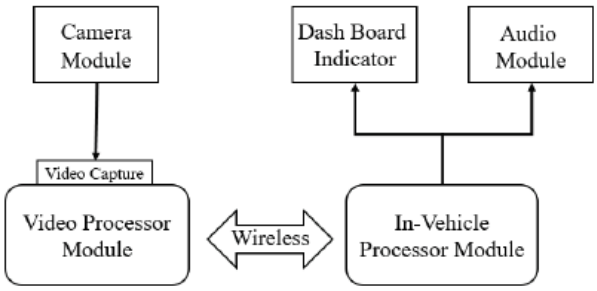


Fig. 1. Vehicle-to-Infrastructure communication [3]

Another category relies on an on board camera in the vehicle, that monitors the road ahead, and the software analyzes the video feed. In terms of hardware, the camera may have infrared vision such as in [4] proposed by Forslund and Bjarkefur, for detection in low light situations, fog or at night.

The algorithms used for image analysis vary. There have been many different approaches with different results that are applied not only in the context of vehicles, but facial recognition, such as Zhang et al. in [5] or Viola and Jones in [6] and other related fields. These involve some sort of image descriptor (Haar features, Local-Binary Pattern, Histogram of Oriented Gradients, etc.) and a classifier (AdaBoost, Support Vector Machine, etc.).

Yamanashi et al. present a segmentation method in [10] using variable regions of interest. They used saliency maps to determine high-salience positions and in combination with the scale invariant transform features, the foreground objects are separated from the background.

The most important metrics in animal detection are accuracy and speed. Creating a system that fulfills both of these metrics is difficult because they are usually in a trade-off relationship: increased accuracy results in reduced speed and vice versa.

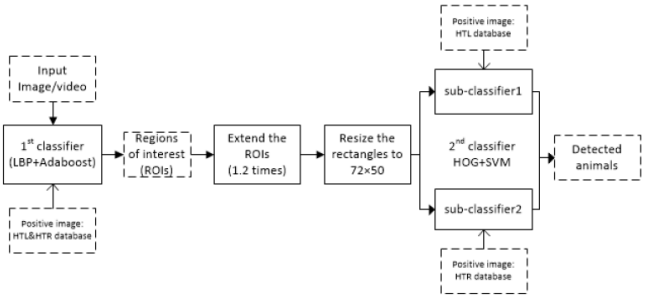


Fig. 2. Two stage classification system [7]

Mammeri et al. propose a compromise solution to this problem in [7] by creating a multi stage detection system that combines a first, speed oriented stage that offers fast selection of potential regions of interest, and a second, accuracy oriented stage that has the role of eliminating any false positive regions that have been selected by the first stage (see Fig. 2).

III. System Architecture

Since the scope of animal detection is too broad and vague, for our purposes, first it needs to be narrowed down and then made more specific.

The first decision to be made is the type of animals that we want to detect. The important aspects that have been considered are: prevalence and the danger posed. The size of the animal is usually correlated with the danger: small animals cause less harm upon impact than their larger counterparts and as a result they will be not included in the scope of this study. Some animals are regional, and thus only in specific cases should be paid attention to them. With these considerations, the selected animals must be of greater size and generally prevalent around the globe. Such animals include deer, elk, moose and horse.

The second decision was to determine the shape that we focus on. Unlike the case with pedestrians, the shape of an animal differs greatly when viewed from different perspectives, so they have to be treated separately. Research done by Volvo [8] shows that the most common shape that a vehicle is likely to encounter is the side view. Without special equipment, such as an infrared camera, detection becomes increasingly difficult as well as in the case of only partially visible animal. Therefore the presumptions that have been made about the context include good lighting conditions and a fully visible animal, having a normal movement pattern. Fig. 3 shows such examples of outlines.

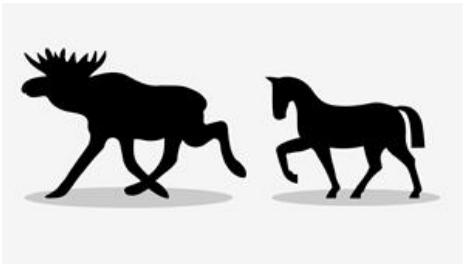


Fig 3. Elk and horse body outline from the side

When it comes to decisions made about the hardware, the type of camera used is the most important one. Stereo vision has many advantages, the additional information could be used to determine the distance of the vehicle from animal however it requires special equipment that is not always available. Monocular vision can provide enough data and is more available. Color vision is an obvious choice over black and white since valuable information can be extracted from the color data.

A. Hardware components

The system requires a computer for data processing and an input source that can be either a color camera with sufficiently large resolution, or from the computer's file system.

B. *Software components*

The software part is developed using the OpenCV library, which offers a great number of tools pertaining to image processing, pattern recognition and computer vision.

The architecture of the software will be based on the performance-accuracy compromise solution proposed in [7], employing a multi stage classification process.

The first software component is called Region of interest (ROI) detector and it gets as input the raw image or video data. Its role is to isolate any potential regions that might contain animals. This should be performed as fast as possible, eliminating the majority of the input, the parts that are considered irrelevant. The selected regions are forwarded to the second stage of the architecture.

The second component is a classifier and it gets its inputs from the output of the first component. The role of the classifier is to eliminate any possible false positives that were selected by the first component. Two parallel classifiers can be used simultaneously, one for left-to-right facing animals and one for the right-to-left facing ones. If a true positive is found by either classifier in this stage, a warning can be signaled.

Fig. 4 shows the conceptual software architecture of the system.

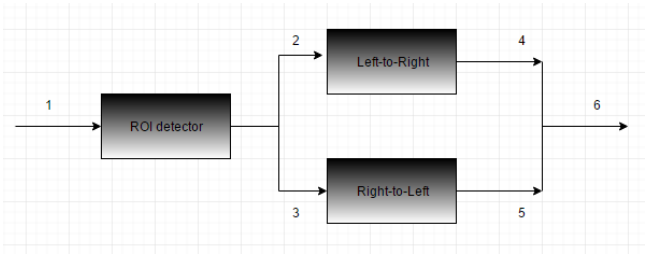


Fig. 4. Conceptual software architecture of the system:

1-Initial input
2-Regions of interest sent to first classifier
3-Regions of interest sent to second classifier
4-Responses from first classifier
5-Responses from second classifier
6-Final decision based on all responses

IV. ROI DETECTOR

The ROI detector is the first stage in the system. The used method was developed by Itti et al. in [9] with some modifications to better fit our purposes. The algorithm creates saliency maps (SM) of the scenes based on three features: intensity, colors and orientations. The regions with high saliency are subsequently considered for selection.

The input images can be of different sizes, large resolutions are scaled down to reduce the computational costs and increase the performance.

The first step is to obtain intensity, color and orientation maps. Each pixel from the intensity map I is computed as a weighted sum (see equation 1).

$$I = 0.3*r + 0.586*g + 0.114*b \quad (1)$$

where r, g and b represent the three color channels of the original image. Any intensity pixel that falls below the 10% threshold of the maximum value of the map is set to 0.

Four different color maps are created, for red, green, blue and yellow respectively, these are denoted by R, G, B and Y (see equations 2-5).

$$R = r - (g + b) / 2 \quad (2)$$

$$G = g - (r + b) / 2 \quad (3)$$

$$B = b - (r + g) / 2 \quad (4)$$

$$Y = (r + g) / 2 - |r - g| / 2 - b \quad (5)$$

The orientation maps are created for 0°, 45° , 90° and 145°, denoted by O(0), O(45), O(90) and O(145) respectively. Four different convolution kernels (see matrices 6-9) are used for generating the orientation maps.

$$O(0): \quad [-1 \quad 0 \quad 1] \quad (6)$$

$$O(45): \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (7)$$

$$O(90): \quad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad (8)$$

$$O(135): \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (9)$$

In case of negative convolution results, the absolute values are taken.

The next step is to create Gaussian pyramids from the previously obtained intensity, color and orientation maps. A pyramid is created from each of the maps (total of 9 maps), each having 8 levels, the lowest, level 0, being the original map. Such a pyramid is created by low-pass filtering and subsampling the image, resulting in vertical and horizontal reductions ranging from 1:1 (level 0) to 1:128 (level 7).

The elements of the resulting pyramids are labeled I(σ), R(σ), G(σ), B(σ), Y(σ), O(σ, θ), where σ ∈ [0..7] and θ ∈ {0, 45, 90, 145}.

After we obtained the pyramids, we must calculate the inter-level differences in each pyramid. From the intensity pyramid six difference maps are generated, labeled I(c, s) where c ∈ {2, 3, 4}, s = c + δ, where δ ∈ {3, 4} (see equation 10).

$$I(c, s) = | I(c) - I(s) | \quad (10)$$

From the color pyramids we need to extract 12 maps, 6 for the red/green chromatic and 6 for the blue/yellow opponency, called RG and BY (see equations 11-12).

$$RG(c, s) = | (R(c) - G(c)) - (G(s) - R(s)) | \quad (11)$$

$$BY(c, s) = | (B(c) - Y(c)) - (Y(s) - B(s)) | \quad (12)$$

For orientation, a total of 24 maps are computed, 6 for each angle (see equation 13).

$$O(c, s, \theta) = | O(c, \theta) - O(s, \theta) | \quad (13)$$

where θ ∈ {0, 45, 90, 145}.

The obtained maps need to be normalized in order to elevate a small number of strong peaks and suppress maps where a large number of such peaks exist. The normalizing operator N(.) performs the following operations on a map:

1. Calculates the average of the sum of local maxima from the image called $\overline{m}$
2. Find the value of the global maxima called M
3. Multiply all values by $(M - \overline{m})^2$

The normalization operator will be applied to all the 42 maps obtained. We reduce the resolution of the obtained maps to $1/16^{th}$ of the original to increase the performance. The normalized maps are summed up separately based on the features they describe to create 3 conspicuity maps, one for intensity, color and orientation respectively. These are denoted by $\bar{I}$,$\bar{C}$ and $\bar{O}$.

Each pixel from the final saliency map *S* is finally computed as an average (see equation 14).

$$S = 1/3 * (N(\bar{I}) + N(\bar{C}) + N(\bar{O})) \quad (14)$$

Fig. 5. Original image

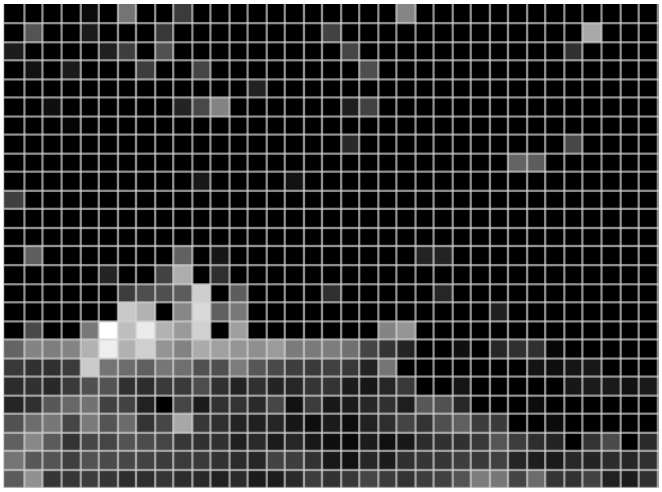


Fig. 6. Saliency map

Our experiments have shown, that eliminating areas of the saliency map that contain green predominantly increases the accuracy, since green areas are mostly vegetation, and certainly not the color of the animal that we are looking for.

Figure 5 shows an input image depicting a typical scenario, a deer on the road with vegetation in the background. Figure 6 shows the resulting saliency map. To be noted here, the large black area that is the suppressed vegetation and the higher intensity pixels mostly centered on the deer.

From the saliency map the regions of interest can be selected by centering a box around the pixel with the highest intensity. This process is repeated multiple times to select multiple regions. To prevent the selection of the same area multiple times, after each iteration the area around the highest intensity pixel is set to zero. Figure 7 shows such a resulting selection. The final resulting regions are sent to the second stage of the architecture to be classified.

Fig. 7. A Region of Interest

V. CLASSIFICATION

The second stage of the architecture consists in the classification of the regions of interests that are sent by the first stage. For this, two separate support vector machines are built, one is trained to classify animals facing right and the other is trained to classify animals facing left.

A database of images was assembled from different sources, containing scenarios where the animal is shown from the side. As for the type of animals being considered, the dataset contains 50% images of deer, 25% images of horses and 25% images of moose. The regions containing the animals are cropped out and resized to a standard 80*112 pixels format. Each image is mirrored vertically to obtain a symmetrical image that depicts the same animal but facing the other direction.

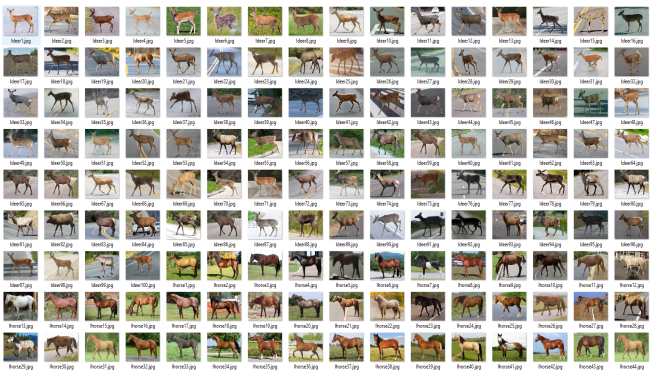


Fig. 8: Animal dataset

Figure 8 shows a snippet from the database containing left facing animals.

A set of negative images is also extracted from the traffic scenarios, these contain anything that is not an animal, from vegetation to empty road, from other vehicles to road signs. Figure 9 shows a snippet from this dataset.

Fig. 9: Negative images

For the feature extraction, histogram of oriented gradient descriptors were chosen, the comparisons done in [7] indicate it as being the most relevant in comparison to other descriptors.

The configuration of the descriptor is the following: cell size of 8*8 pixels, 2*2 cells per block and 9 bins for colors.

For an image of 80*112 pixels size, 4212 features are generated. This is generated for all the images, both negative and positive, and is used to train the support vector machines. The selected SVM is C-SVM type, classifying two classes with imperfect separation, with c = 2.5 penalty multiplier for outliers.

The two support vector machines that are used in the system are saved to files and can be any time loaded. The regions arriving from the first stage are resized to the standard 80*112 pixels size, their HOG descriptors are generated. The two SVMs classify these regions and if any of them detect an animal, that region is highlighted as containing an animal.

VI. Experimental Results

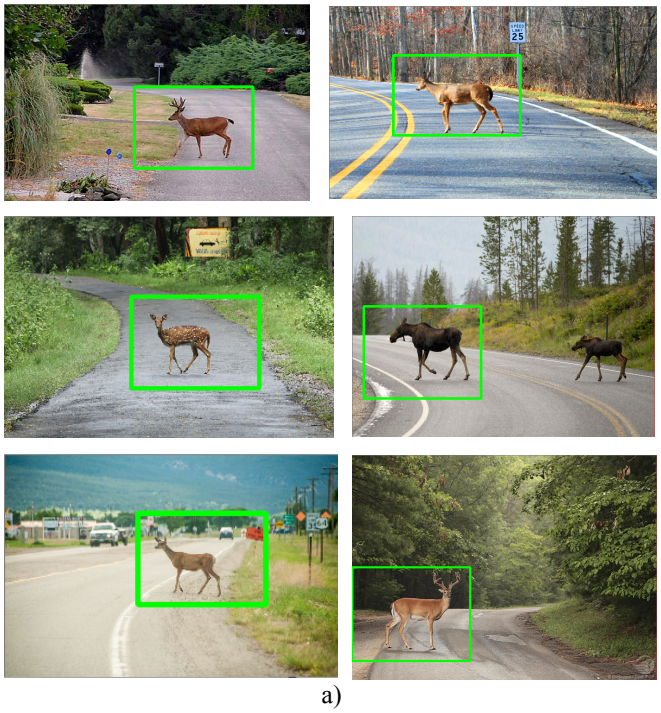
The support vector machines have been tested through the bootstrap aggregating technique. During the tests, 90% of the data set is used for training the SVM and the remaining 10% is used for testing. Both left and right classifiers were tested, both with positive and negative images. The first column shows the results for left facing images, second column for right facing images and the third one for negative images. A total of 10 tests were performed and results can be seen in Table I. On average, regions are classified correctly 95.5% of the time.

TABLE I. Performance of Support Vector Machines classifiers

Test Number	Performance metrics		
	Left facing animal (TP rate)	Right facing animal (TP rate)	Regions with no animals (TN rate)
1	100%	100%	100%
2	100%	95%	95%
3	100%	100%	95%
4	100%	100%	95%
5	95%	95%	95%
6	95%	90%	90%
7	85%	85%	100%
8	90%	85%	100%
9	100%	100%	80%
10	100%	100%	100%

Typical successful results (true positives and true negatives) are presented in Fig. 10, where the selected region of interest encompasses the whole animal with minimal extra regions that are between the animal and the border.

The experiments have shown that classification works well as long the regions of interest contains the whole animal that is viewed from the side, but does not include a too broad border between the animal and the edge of the box.



a)

b)

Fig. 10: Successful classification: a) true positives; b) true negatives

VII. Conclusions

Following the experimental results, it can be said that the proposed method for detecting animals works reasonably well within the scope of its purposes. The strengths of the system are the highly accurate classifiers that perform well even though they were trained using a limited dataset. The results can be improved by better quality training data in larger quantities and by creating more specialized classifiers.

For future work, the region of interest detection can be improved upon. The biggest weakness of the algorithm is the static size of region delimiter box. By creating dynamically changing box sizes, the risk of including only parts of the animal or including too much background can be reduced. Other classifiers using Neural Networks will be developed and their results will be compared.

REFERENCES

[1] A.F. Williams and J.K. Wells - "Characteristics of Vehicle-Animal Crashes in Which Vehicle Occupants Are Killed" Traffic Injury Prevention volume 6(1), pp. 56–59, 2005.

[2] State Farm - "Top Five States for Deer-Related Collisions". Available: https://www.statefarm.com/retirees/news/top-states-for-deer-collisions (2015-11-18).

[3] S. Vishnu, Ullas Ramanadhan, Nirmala Vasudevan and Anand Ramachandran – "Vehicular Collision Avoidance Using Video Processing and Vehicle-to-Infrastructure Communication", in Proceedings of International Conference on Connected Vehicles and Expo (ICCVE), pp. 387-388, 2015.

[4] David Forslund and Jon Bjarkefur - "Night Vision Animal Detection", in Proceedings of IEEE Intelligent Vehicles Symposium, pp. 737-742, 2014.

[5] L. Zhang, R. Chu, S. Xiang, S. Liao, S.Z. Li - "Face Detection Based on Multi-Block LBP Representation", in Advances in Biometrics. ICB 2007. Lecture Notes in Computer Science vol. 4642, Springer, Berlin, Heidelberg.

[6] Paul Viola , Michael J. Jones - "Robust Real-Time Face Detection", International Journal of Computer Vision vol. 57(2), pp. 137–154, 2004.

[7] Abdelhamid Mammeri, Depu Zhou, Azzedine Boukerche, and Mohammed Almulla - "An Efficient Animal Detection System for Smart Cars using Cascaded Classifiers", in Proceedings of IEEE International Conference on Communications (ICC), pp. 1854-1859, 2014.

[8] Volvo cars support - "Large animal detection". Available: http://support.volvocars.com/au/Pages/article.aspx?article=47cd9bf796d878dec0a801517dc39eb9 (2017-07-21).

[9] L. Itti, C. Koch, E. Niebur - A model of saliency-based visual attention for rapid scene analysis, in Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, issue 11, pp. 1254-1259, 1998.

[10] Ayaka Yamanashi, Hirokazu Madokorol, Yutaka Ishiokal, and Kazuhito Satol - "Visual Saliency Based Segmentation of Multiple Objects Using Variable Regions of Interest", in Proceedings of 14th International Conference on Control, Automation and Systems (ICCAS), pp. 88-93, 2014.