A Gene Expression Programming Approach for Vehicle Body Segmentation and Color Recognition

Brenda C. Solari Berno, Lucas A. Albini, Vinícius C. Tasso, César M. Vargas Benítez, Heitor S. Lopes
Programa de Pós Graduação em Engenharia Elétrica e Informática (CPGEI)
Universidade Tecnológica Federal do Paraná (UTFPR)
Av. 7 de setembro, 3165 - 80230-901, Curitiba (PR), Brazil
Email: {brendab, lucasalbini, vintas}@alunos.utfpr.edu.br,
{cesarvargasb, hslopes}@utfpr.edu.br

***Abstract*—Color recognition in vehicles is a topic widely discussed in the literature given that color is one of the key features that define a vehicle identity. Recently, the need to recognize and describe a vehicle's appearance in traffic surveillance has grown in demand, as a result of the need for efficient traffic monitoring systems. In this work, we present a different approach to recognize the predominant color of a vehicle, with minimal computational resources when compared to other methods. The goal is to segment the car body from the original image and, then, recognize the predominant color in the segmented image. To accomplish the segmentation task, we use Genetic Expression Programming (GEP) to evolve a mathematical expression to filter the original image leaving only the body of the vehicle. Another objective of this work is to create a dataset, annotated at the pixel level, for car body segmentation. Our results showed that the proposed approach was efficient for vehicle color recognition, possibly for a real-time implementation.**

***Keywords*—Image Segmentation; Evolutionary Computation; Genetic Expression Programming; Color recognition;**

I. Introduction

Smart cities have a substantial concern about technology and the way citizens interact with the city, to bring well-being to the population and improve security, urban mobility, and sustainability. In most Brazilian capitals, there is a vast interest in safety and mobility improvements regarding the increasing number of vehicles circulating on the streets.

These cities are investing more and more in security systems, including surveillance cameras which, over time, has far outnumbered the people available to monitor them [1]. Furthermore, monitoring performed by humans has disadvantages such as inefficiency and high costs [2], when compared to autonomous systems. This is a significant problem because, besides these challenges, security footage is often analyzed reactively instead of proactively. This is to say that footage is first captured, opposite to being analyzed in real-time.

As cities grow, the number of vehicles increase and, as consequence, also increase accidents, traffic jams, and traffic violations. Therefore, traffic management systems, particularly in medium and large cities, are facing increasingly complex challenges. Therefore, research on active traffic surveillance, i.e. monitoring and managing traffic flow, has attracted much attention recently [3].

Obtaining vehicle information from images is a widely discussed topic in the literature as presented in [4], [5] and [6]. Within this context, for some applications, color recognition is one of the key challenges. A system that can efficiently and accurately recognize the color of a vehicle can be essential for systems regarding public and traffic safety. Possible applications are the localization of stolen vehicles, identification of vehicle registration fraud or illegal alterations to a vehicle.

To classify a car image regarding its color, several difficulties have to be overcome. Although it may look like a simple task for the human eye, it is a fundamental problem in computer vision, since the apparent color of a vehicle changes as a function of time, space, and light conditions [7].

To represent a color, many color spaces can be used, as shown with detail in [8]. The RGB (Red, Green and Blue) histogram is a common choice because of the wide range of applications. Despite not being robust enough to accurately represent colors in complex scenes [9], e.g with extreme lighting conditions, it was the color space of choice as a proof of concept.

There are several different approaches in the literature for color recognition of objects in images. In recent years, with the advancement of deep learning methods and Convolutional Neural Networks (CNN), excellent results were obtained [10]. However, these methods have limitations such as requiring high processing power, especially during the training steps, and difficulty to be embedded in online surveillance systems.

This paper proposes a method for improving the color recognition of vehicles by firstly segmenting the vehicle body. To do this, an evolutionary computation algorithm, known as Gene Expression Programming (GEP), was used. To train this algorithm, a new dataset was manually created through web scrapping techniques. For each vehicle image, there is an equivalent mask representing the desirable result for that image. This dataset includes car images in a variety of angles, colors and illumination situations.

The contributions of this work are threefold. First, it presents a segmentation method that tackles the need to obtain large amounts of data to present satisfactory results for vehicle color extraction. To show this, the training dataset is composed of a hundred unique vehicle images. Second, the computational costs, primarily during training of the GEP algorithm, are reasonably lower than those required by deep learning techniques, often used in these scenarios. Also, the final filter generated by GEP is lightweight enough to be embedded in online applications. Lastly, the dataset that was created will be publicly available so that the interested researchers can use it to compare with other approaches.

The paper is presented as follows: Section II presents some theoretical aspects about the Genetic Expression Programming (GEP) paradigm; Section III presents a brief review of related works about color recognition and image segmentation; Section IV details the methods used in this work; Section V reports the experiments and results obtained. Finally, Section VI presents the conclusion and future works.

II. Genetic Expression Programming - GEP

GEP is an evolutionary algorithm which solves problems by evolving of computer programs [11], represented as variable-sized gene expression strings. Throughout generations, GEP evolves solutions by applying operators such as mutation and crossover. Differently from the traditional Genetic Programming approach created by Koza [12], GEP introduces two concepts: genotype and phenotype. Those concepts were borrowed from Genetic Algorithms (linear chromosomes of fixed length) and Genetic Programming (expression trees), respectively.

GEP chromosomes are composed of one or more genes (in this case, connected by a linking operator), where each gene is composed of two parts: the head (h) and the tail (t). An example of the representation of a GEP chromosome is the following:

[**+,Q,-,/,b,*,a,c,Q,b**,a,a,b,a,a,b,b,a,a,a,b]

where the set of terminals is $\{a, b\}$ and a set of functions is $\{+, -, /, *, Q\}$.

The head (bold section in the above example) always starts with a function and can contain either functions or terminals. The tail is composed exclusively of terminals. The tail length is given by:

$$t = h(n_{max} - 1) + 1 \tag{1}$$

where n_{max} is the maximum arity (number of function arguments). This chromosome can be converted to an expression tree (Fig. 1), and can be decoded as:

$$\left(\frac{a}{c}\right)^2 + [b - (a^2 * b)] \tag{2}$$

The flowchart of the GEP algorithm is shown in Fig. 2. First, an initial population is randomly created and, then, the operations determined by the current structure, such as mutation or crossover, are executed. To evaluate these individuals, the fitness function is calculated to determine the performance of all individuals of the current generation. It is at this point that the algorithm can either proceed to the next generation, through reproduction, in an attempt to improve the results, or terminate itself and keep the current generation.

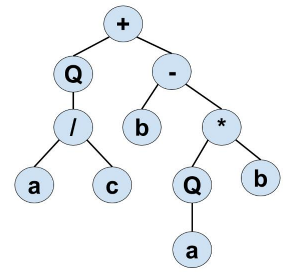


Fig. 1. Example of Expression Tree (ET).

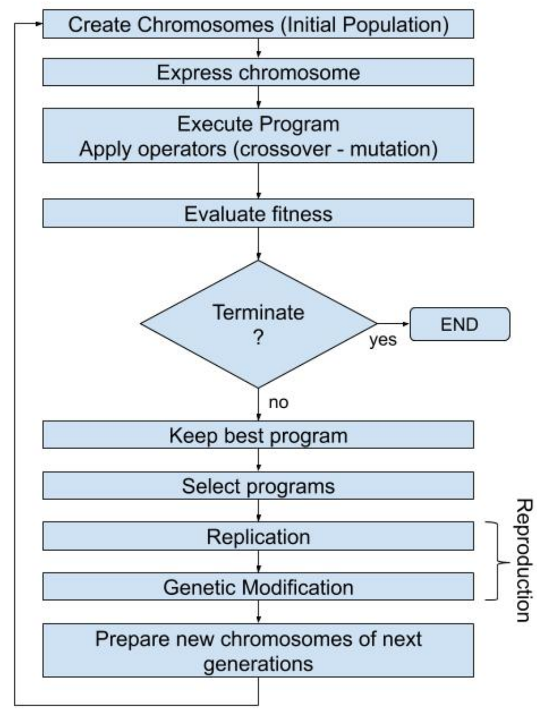


Fig. 2. Flowchart of the GEP algorithm.

III. Related Works

Color, as a notable quality of a vehicle, can prove to be reliable and important information in a variety of computer vision applications. Due to the potential practical applications, vehicle color recognition in natural scenes has raised interest by the research community and there is a rich collection of works related to this topic.

The work presented by [13], in an approach to detect vehicles from images, uses vehicle colors to detect possible vehicle candidates. [14] proposes a concept where, inside a region of interest in an image, regions of the front profile of a vehicle are detected to extract its colors. The color extraction approach used in [15] is similar to the method presented in this work in that each pixel color value is taken into account.

More recent work of [16] propose a technique to deal with distortions caused by different light conditions by means of color corrections.

In [17], [18], [19] and [20], different thresholding methods were used, which consists in classifying the pixels of an image into two classes: foreground and background. The gray level of each pixel is compared with a threshold value (or several values, for multilevel classification). If the gray level of a given pixel is above the threshold, it is classified as foreground, and in the same way to the background. The last step was to binarize the output image with white pixels as foreground and black pixels as background.

Felzenszwalb et al. [21] proposed an approach for image segmentation based on graphs, where the image is divided into regions and then represented by a graph. The vertices are the elements to be segmented, and the edges connect the pairs of neighboring vertices, each edge has a corresponding weight, which represents the dissimilarity between two vertices (regions).

More recently, Convolutional Neural Networks (CNN) become very appealing for image segmentation due to their ability to extract a hierarchy of increasingly complex features. For that, in [22], [23], [24] and [25] adapt classification networks as AlexNet, the VGG net, and Faster R-CNN into fully convolutional networks and transfer their learned representations to the task of image segmentation.

Other approaches with Genetic Programming variants were proposed as well. In [26], [27], and [28] different set of terminals and functions were used to create filters to segment images for specific applications. In these cases, the set of generated filters is highly dependent on the type of data.

IV. Methods

The first step in this work was to find suitable images of vehicles under different light conditions to create a dataset. Those images were treated manually to segment the car body from the rest of the image. This was the general procedure for creating the segmentation dataset used in the training and test steps. The next step was to use GEP for evolving filters capable of segmenting the images in foreground (the car body) and the background (all the rest in the image). Later, the images of the test set were used to access the quality of segmentation. Finally, an algorithm for color extraction in the images was used. The Fig. 3 presents a block diagram of the steps of the method proposed, explained in the following section.

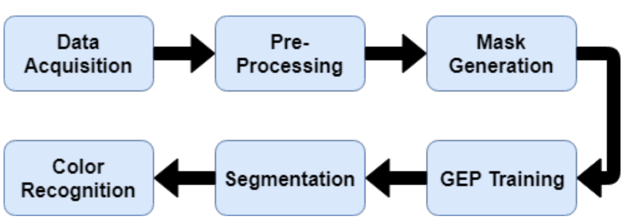
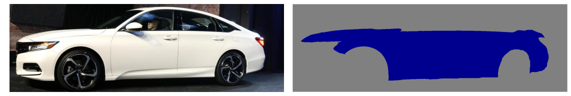


Fig. 3. Methodology



(a) Original (b) Mask

Fig. 4. Example of training dataset. (b) is the mask that is desired on the GEP output.

A. Data Acquisition and Preprocessing

For data acquisition, a Python web scrapping script was used. Images of vehicles were obtained from the internet and manually annotated. The dataset consists of 100 images for training and 30 images for testing. Both, training and testing, datasets have 10 classes each, with the same number of samples. Each class is a predominant color with variants (dark, regular, light). The full dataset, including the segmentation masks, is available for research purposes [1].

The first step of preprocessing was to remove images that came in the wrong or corrupted format. The next step was to remove excessive background in the images using Yolo-v3 [29] to crop only a single vehicle in the image.

B. Mask Generation

The last step of data preparation was to manually segment the training and testing dataset. This was accomplished by using the Pixel Annotation Tools[2], an example of that is shown in the Fig. 4.

C. GEP Training

In GEP it is necessary to define the following elements for modeling the problem:

- A set of 16 terminals, shown in Table I
- A set of 27 functions (operators), shown in Table II
- Fitness function

First, the terminals set was based on [26], which contains different operations over the original image in different sizes of kernels (e.g. 3 x 3). Some of these operations were inspired in the convolutions performed by CNNs.

In real-world situations, the brightness of the metal in car images may cause the color to be confused with reflected tones. To address this problem the filters shown in Table I were applied to the *IM_0* image (the original image), transformed first from RGB to HSV and, then, to gray-scale.

The functions set used here, as suggested by [26], was based on simple algebraic operations, such as addition or subtraction, between two images. The output of these operations is another image of the same size.

Filters, such as maximum, minimum, median and standard-deviation, were implemented taking different sizes of kernels. For instance, in Figure 5 it is shown a 3 x 3 kernel, meaning that a 3 x 3 pixels window slides through the image with a

[1]Available at https://labic.utfpr.edu.br/datasets/data/UTFPR-CBS/UTFPR-CBS.tar.gz

[2]Source code available at: https://github.com/abreheret/PixelAnnotationTool.

stride of 1 pixel and the respective operation is executed in that region. The output image from each operator has to be the same size, such that we can apply a zero padding of size 1. This means that the input image will be completed with zeros around the border, then this would result in an image of the same size.

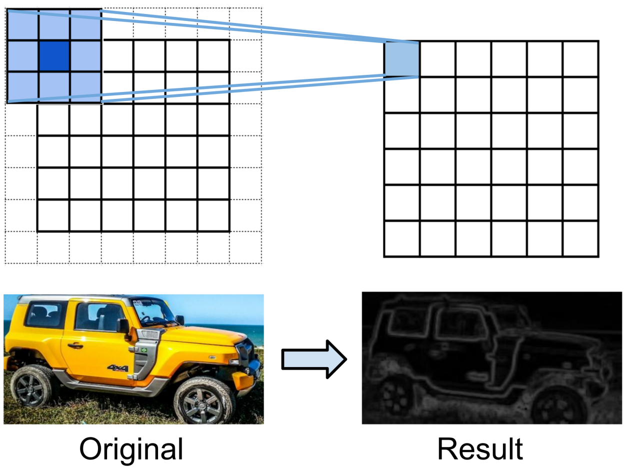


Fig. 5. Filters

The objective function, which measures the quality of a solution, is defined in Equation 3, which represents the Mean Squared Error (MSE) between two images. Therefore GEP will attempt to minimize this value over generations.

$$ObjFun = MSE(Y, Y_p) = \frac{\sum (Y - Y_p)^2}{height * width} \qquad (3)$$

where Y is the original mask and Y_p is the mask generated by the algorithm.

Equation 3 is calculated for each image and summed over all test images, thus representing the overall fitness of a given individual.

The GEP implementation was based on the Geppy framework[3] and the main control parameters were: Population size (500), Number of generations (80), Number of individuals used for the tournament selection (3), Number of genes in the chromosome (2), and Linking operator (ADD). The control parameters were experimentally adjusted by observing the evolution of the algorithm. The mutations selected were extracted from the general operators of Geppy.

[3]Geppy website: https://geppy.readthedocs.io

Symbol	Description
IM_0	Original image
IM_1	3 x 3 mean filter
IM_2	5 x 5 mean filter
IM_3	7 x 7 mean filter
IM_4	3 x 3 standard-deviation filter
IM_5	5 x 5 standard-deviation filter
IM_6	7 x 7 standard-deviation filter
IM_7	3 x 3 maximum filter
IM_8	5 x 5 maximum filter
IM_9	7 x 7 maximum filter
IM_10	3 x 3 minimum filter
IM_11	5 x 5 minimum filter
IM_12	7 x 7 minimum filter
IM_13	3 x 3 median filter
IM_14	5 x 5 median filter
IM_15	7 x 7 median filter

TABLE I
TERMINALS SET USED BY GEP

Operator	Description
ADD(A,B)	Add images A and B
SUB(A,B)	Subtract image B from A
MUL(A,B)	Multiply image A by B
DIV(A,B)	Divide image A by B
MAX2(A,B)	Maximum, pixel-by-pixel, of images A and B
MIN2(A,B)	Minimum, pixel-by-pixel, of images A and B
ADDC(A)	ADD a constant C to all pixels of image A
SUBC(A)	Subtract a constant C from all pixels of image A
MULC(A)	Multiply all pixels of image A by a constant C
DIVC(A)	Divide all pixels of image A by a constant C
SQRT(A)	Square root of image A
LOG(A)	Natural logarithm of image A
MAX_3X3(A)	3 x 3 maximum filter of image A
MAX_5X5(A)	5 x 5 maximum filter of image A
MAX_7X7(A)	7 x 7 maximum filter of image A
MIN_3X3(A)	3 x 3 minimum filter of image A
MIN_5X5(A)	5 x 5 minimum filter of image A
MIN_7X7(A)	7 x 7 minimum filter of image A
MEDN_3X3(A)	3 x 3 median filter of image A
MEDN_5X5(A)	5 x 5 median filter of image A
MEDN_7X7(A)	7 x 7 median filter of image A
MED_3X3(A)	3 x 3 mean filter of image A
MED_5X5(A)	5 x 5 mean filter of image A
MED_7X7(A)	7 x 7 mean filter of image A
STDV_3X3(A)	3 x 3 standard-deviation filter of image A
STDV_5X5(A)	5 x 5 standard-deviation filter of image A
STDV_7X7(A)	7 x 7 standard-deviation filter of image A

TABLE II
FUNCTION SET USED BY GEP.

The size of the input images was set with 300 of width and the height was variable between 100 and 200, so as to accelerate the evolution process.

D. Segmentation

GEP generates an expression, such as Equation 4, in each generation during the evolution process. At the end of the process, the best combination of operators and terminals is chosen to accomplish the desired task, in this case, car segmentation.

$$add(div(IM\_1, IM\_0), min\_(IM\_4, add(IM\_3, IM\_1))) \qquad (4)$$

Such expression is in the preorder form (also known as "Polish notation"), since the operator comes before their operands. Such expression can be easily transformed into a tree for better human comprehension. The application of the expression to segment an image is straightforward: image terminals are created from the original image and, then, operators are applied to the terminals according to the expression. The final output image is binarized in white and black tones. The final output image is a mask to extract only the car body of the image.

E. Color Recognition

The color recognition method chosen consists of extracting each pixel that composes a given image and comparing its RGB value with predefined colors. A total of ten colors were defined, each one with three variants (dark, regular, light), as shown in Fig 6. Each color is represented by three RGB values, each of which represents a different tone of that color. Aiming at classifying pixels in the image into the known colors categories, the Euclidean distance between the RGB value of the pixel and each known color tone was calculated. The smaller distance will determine the closest color tone to that pixel and therefore, the pixel color. The Euclidean distance between colors q and p is obtained by

$$d(q,p) = \sqrt{\sum_{i=1}^{3}(q_i - p_i)^2} \qquad | \; i \in \{red, green, blue\} \quad (5)$$

After all pixels are labeled according to the known colors, the predominant color of the image is assigned as the vehicle color.

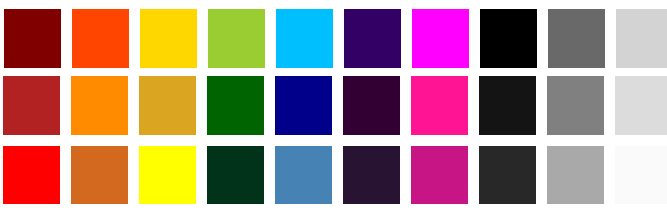


Fig. 6. Colors and tones defined. From left to right: Red, Orange, Yellow, Green, Blue, Purple, Pink, Black, Gray, White.

It is important to notice that this method is prone to error when there is noise and irrelevant information, e.g. when the background represents a larger portion of the image than the vehicle itself. However, as later sections will discuss, despite being an extraordinarily simple approach, when combined with GEP segmentation, the results are quite promising.

F. Baseline

In order to evaluate the result of the proposed approach, it is necessary to establish baselines for comparison, regarding the final objective of color recognition. Figure 7 synthesizes the four approaches, detailed below:

- Approach 1: Recognition of car color in the original set of collected images, i.e. without filtering or editing. The goal is to define a baseline to compare to the other approaches.
- Approach 2: Recognition of car color in images after cropping, i.e. removal of irrelevant background material. This approach performs, generally, a very subtle change to the input image given its simplicity. The objective is to analyze how much improvement can be accomplished by discarding most of the background, enclosing the car in a rectangular frame.
- Approach 3: Car segmentation using a pre-trained neural network for vehicle color extraction and to get a better reading of the colors that represent the car. The goal with this approach is to establish the performance obtained with the state of art method. The Mask-RCNN [25] neural network was trained specifically to approach the vehicle color recognition problem and it is widely used in the literature.
- Approach 4: Car body segmentation using a filter created by GEP. This approach aims to evaluate the performance of the proposed method regarding the previous ones.

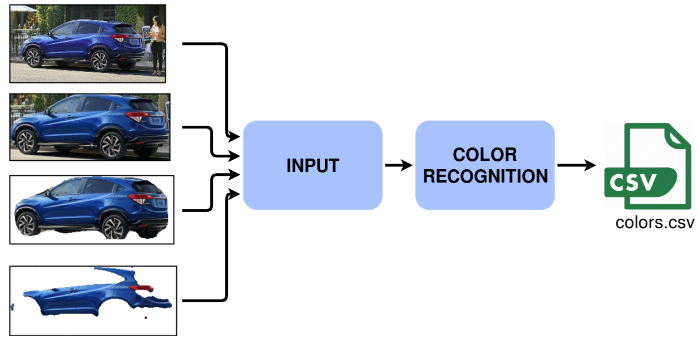


Fig. 7. Different approaches to color recognition of the vehicle in the image.

V. Experiments and Results

A. Segmentation

At the end of the evolution process, the tree with best performance for segmenting the car body in all the test images was the one presented in Figure 9. Each leaf represents the result of a particular filter (Table I) applied to the original image, whereas each node represents the mathematical operations (Table II) applied to its children nodes.

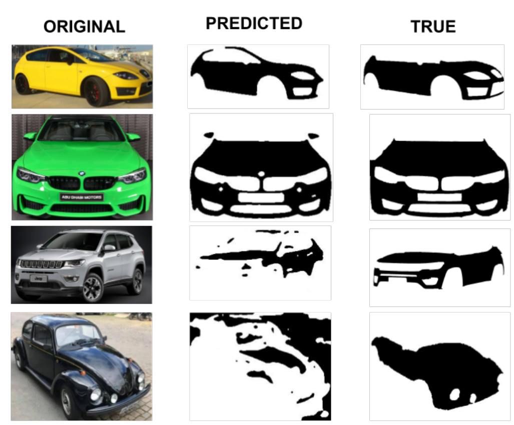


Fig. 8. Some results with test dataset. The left image was the original input given to GEP. The center image is the corresponding output from our method. The right image is the ground truth using manual segmentation.

To enable the reader to understand the overall flow of information in the tree, at each node the resulted image from the operations was drawn. Notice that, in some operations, such as the LOG, the output image was almost completely white and, to facilitate visualization, these images were modified in

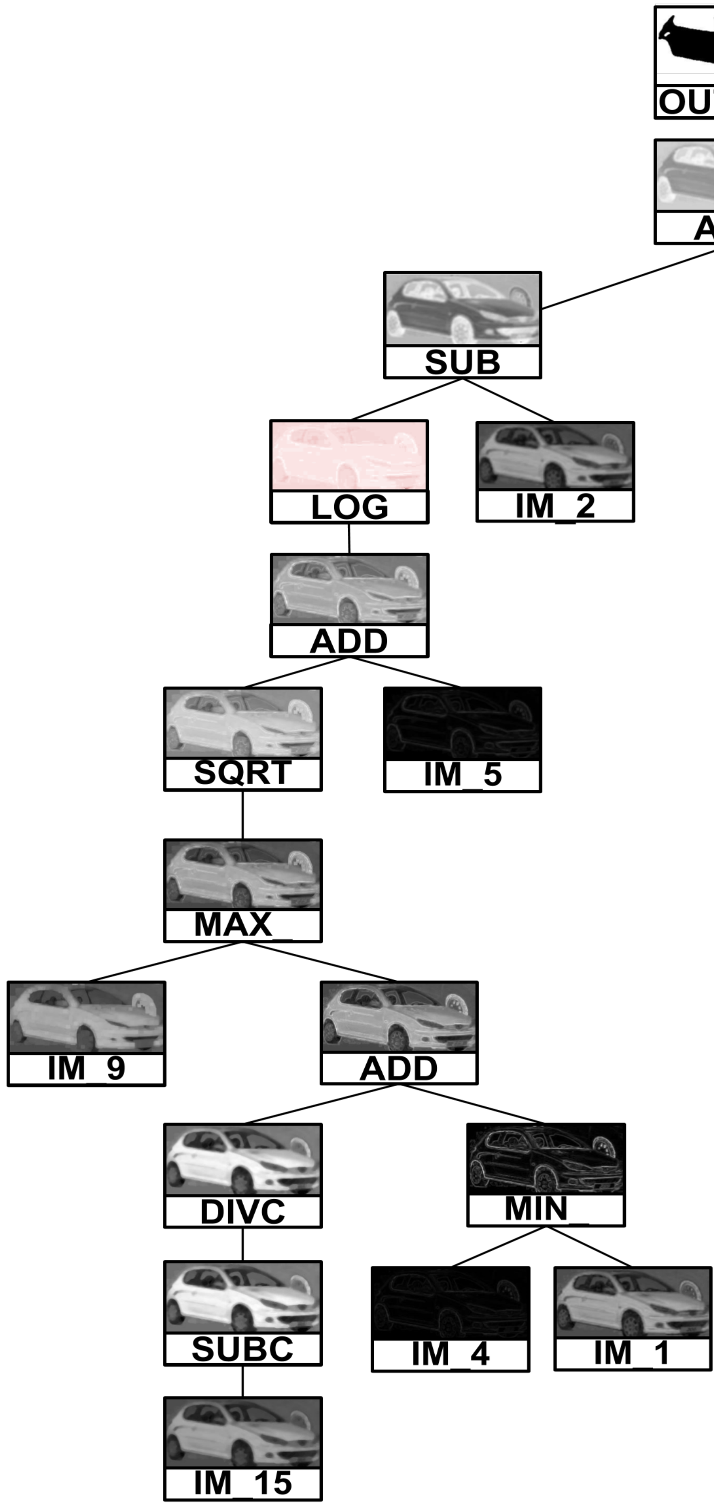
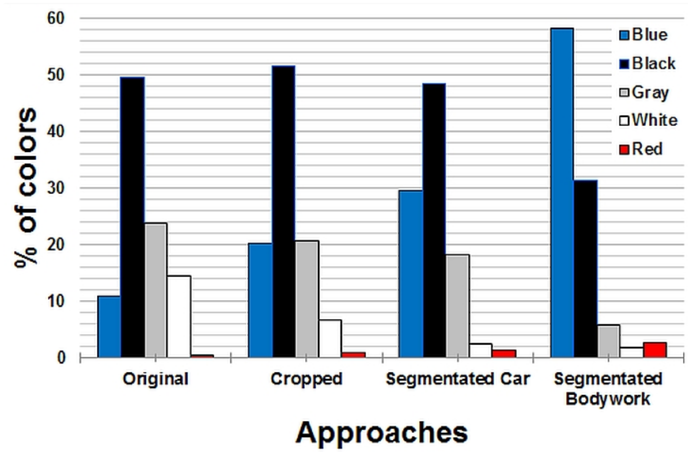


Fig. 9. The final tree generated by GEP. The images shown are the actual inputs and outputs at each processing step, for a given car image.

this representation. The final operation of the tree is ADD, then the result was binarized to obtain the OUTPUT image. This is the tree used in the next steps for testing.

Figure 8 shows four sample images of the testing set, together with the output segmentation provided by the tree, and the ground truth. There are some small errors in the segmented images, possibly due to the small dataset used. There is, also, a noticeable decrease of performance of the the GEP based approach for black and gray cars, when compared with colored cars. A possible cause for this drawback is the fact that all

(a) Original image



(b) Percentage Of colors

Fig. 10. Example of color recognition of a car in a real-world scene. The bar chart presents a visual comparison of the percentage of the five most representative colors detected between four different approaches.

images are converted to grayscale causing the darker shades of color, e.g dark and gray, to be confused with the background of the scene.

B. Color Recognition

Figure 10a shows an image of a car in a complex street background. For this example, Figure 10b shows a bar graph comparing the baselines mentioned in Section IV-F with the GEP-based approach. We observe that the only method in which color recognition hits the predominant color was the body segmentation method proposed in this work, when compared with the other approaches.

VI. Conclusion

Color recognition in vehicles is a difficult problem to solve given that irrelevant and potentially misleading information comes with the image. This paper presents an approach for the segmentation of vehicle body in order to recognize the predominant color, a problem not so frequently addressed in the literature.

For a real-world scene, it is automatic for humans to disregard glasses, wheels, taillights, licence plate, rubber/plastic trims and other elements of a car. Also, light reflections usually do not mislead humans in identifying the predominating color of a car. Notwithstanding, this is a hard task for a computational system based on image processing.

The proposed method was capable of achieving good results for images with colored vehicles and performed much better than the other baseline approaches presented. Some problems raised in images with black, gray and white cars, when our method was not able to fully segment the car body. Notwithstanding, the part segmented can be used for color recognition within some error margin.

Future works should focus on increasing the performance of the color extractor, using, for instance, an approach for clustering image colors before comparing them with the predefined ones. However, we believe that the most important improvement to overcome the drawbacks of the method is to create a much larger and richer annotated dataset for training GEP. In order to assess the robustness of the approach for similar problems, future works could also include experiments with datasets for different applications of image segmentation. Finally, we believe that the proposed approach is promising for future integration with online public safety and traffic control systems.

Acknowledgments

L.A. Albini and V.C. Tasso thanks CNPq for the PIBIC scholarship, B.C.S. Berno thanks UTFPR for the Msc. scholarship. H.S. Lopes thanks CNPq for the research grants no. 311778/2016-0 and 423872/2016-8, and Fundação Araucária for the research grant PRONEX 042/2018. Special thanks to NVIDIA Corp. for the donation of the Titan-Xp GPU boards used in this work.

References

[1] M. Gutoski, "Learning and transfer of feature extractors for automatic anomaly detection in surveillance videos," MSc. Dissertation, Graduate Program in Engineering and Computer Science, Federal University of Technology Paraná – UTFPR, 2018.

[2] N. Li, X. Wu, D. Xu, H. Guo, and W. Feng, "Spatio-temporal context analysis within video volumes for anomalous-event detection and localization," *Neurocomputing*, vol. 155, pp. 309–319, 2015.

[3] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang, "Hierarchical and Networked Vehicle Surveillance in ITS: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 25–48, 2017.

[4] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.

[5] C. Marina Martinez, M. Heucke, F. Y. Wang, B. Gao, and D. Cao, "Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666–676, 2018.

[6] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 377–391, 2008.

[7] J. W. Hsieh, L. C. Chen, S. Y. Chen, D. Y. Chen, S. Alghyaline, and H. F. Chiang, “Vehicle color classification under different lighting conditions through color correction,” *IEEE Sensors Journal*, vol. 15, no. 2, pp. 971–983, 2015.

[8] M. W. Schwarz, J. C. Beatty, C. O. KlA, J. C. Beatty, and m. W. Schwarz, “An Experimental Comparison of RGB, YIQ, LAB, HSV, and Opponent Color Models,” *ACM Transactions on Graphics*, vol. 3511, no. 2, pp. 123–158, 1987.

[9] C. Hu, X. Bai, L. Qi, P. Chen, G. Xue, and L. Mei, “Vehicle Color Recognition With Spatial Pyramid Deep Learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2925–2934, 2015.

[10] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] C. Ferreira, “Gene expression programming: a new adaptive algorithm for solving problems,” *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001.

[12] J. R. Koza, *Genetic Programming: on the programming of computers by means of natural selection*. Cambridge, USA: MIT Press, 1992.

[13] L.-W. Tsai, J.-W. Hsieh, and K.-C. Fan, “Vehicle detection using normalized color and edge map,” *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 850–64, 2007.

[14] X. Zhan and C. Jing, “Vehicle color extraction based on first sight window,” *Proc. 1st Int. Conference on Information Science and Engineering (ICISE)*, pp. 1503–1506, 2009.

[15] K. J. Kim, S. M. Park, and Y. J. Choi, “Deciding the number of color histogram bins for vehicle color recognition,” in *Proc. of the 3rd IEEE Asia-Pacific Services Computing Conference (APSCC)*. IEEE, 2008, pp. 134–138.

[16] P. Chen, X. Bai, and W. Liu, “Vehicle color recognition on urban road by feature context,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2340–2346, 2014.

[17] M. Kallergi, K. Woods, L. P. Clarke, W. Qian, and R. A. Clark, “Image segmentation in digital mammography: comparison of local thresholding and region growing algorithms,” *Computerized Medical Imaging and Graphics*, vol. 16, no. 5, pp. 323–331, 1992.

[28] Y. Liang, M. Zhang, and W. N. Browne, “Multi-objective genetic programming for figure-ground image segmentation,” in *Proc. Australasian Conference on Artificial Life and Computational Intelligence*. Springer, 2016, pp. 134–146.

[18] R. Adollah, E. Francis, M. Mashor, and N. Harun, “Bone marrow image segmentation based on multilevel thresholding,” in *Proc. Int. Conference on Biomedical Engineering (ICoBE)*. IEEE, 2012, pp. 457–461.

[19] J. Kubicek, M. Penhaker, K. Pavelova, A. Selamat, R. Hudak, and J. Majernik, “Segmentation of MRI data to extract the blood vessels based on fuzzy thresholding,” in *New Trends in Intelligent Information and Database Systems*. Springer, 2015, pp. 43–52.

[20] M. Frucci and G. S. di Baja, “From segmentation to binarization of gray-level images,” *Journal of Pattern Recognition Research*, vol. 1, pp. 1–13, 2008.

[21] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[22] A. Brilhador, M. Gutoski, L. T. Hattori, A. S. Ináacio, A. E. Lazzaretti, and H. S. Lopes, “Classification of weeds and crops at the pixel-level using convolutional neural neural networks and data augmentation,” in *Proc. of the 6th IEEE Latin American Conference on Computational Intelligence (LA-CCI)*. IEEE, 2019, pp. 1–6.

[23] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, “Brain tumor segmentation with deep neural networks,” *Medical Image Analysis*, vol. 35, pp. 18–31, 2017.

[24] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proc. of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[26] H. A. Perlin and H. S. Lopes, “A genetic programming approach for image segmentation,” in *Computational Intelligence in Image Processing*. Springer, 2013, pp. 71–90.

[27] R. Poli, “Genetic programming for feature detection and image segmentation,” in *Proc. AISB Workshop on Evolutionary Computing*. Springer, 1996, pp. 110–125.

[29] J. Redmon and A. Farhadi, “Yolo-v3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.