Night Time Car Recognition Using MATLAB

Author: Hazim Hamza

Supervisor: Prof. Paul Whelan

***Abstract*— This paper presents a car recognition system by locating and segmenting tail lights in the night-time road environment. Numerous approaches towards car recognition during day time have been implemented in the past. However, the features of cars during daytime are seldom available when it's dark due to lack of lighting and other conditions. Unlike previous work in the area, this system employs HSV color thresholding for segmenting the red regions (brake lights) and capturing certain object features of the segmented parts. These features are used to train and classify the different classes of lights for different car models. For this, the machine learning based approach – Support Vector Machines (SVMs) is incorporated. Satisfactory results were obtained and they show that the SVM technique is effective for classification**

***Index Terms*— Computer Vision, Image processing, SVMs, Hsv color space, car recognition.**

I. Introduction

This paper is focused on classification of vehicles in night time road driving environment. Recent statistics suggest that night conditions are an important area of scrutiny for road security [2]. The use of Computer Vision techniques has gone a long way in aiding with road safety and security. The appearance of vehicles during night time is strikingly different when compared to its daylight counterpart as several attributes come into the picture, such as environment lighting, color of vehicles, reflection of light on the body of vehicles, etc. Thus, an entirely different image processing approach is absolutely essential when it comes to dealing with night time road environment.

This application targets on segmenting the tail lights of vehicles and classifying them. When vehicles are viewed from behind at night, they are primarily visible by the red color rear facing brake lights. All car models have their own peculiar physical and structural features which make them distinctive from each other and the appearance of brake lights is one of those features. Hence, the representation of different shapes and designs of these rear facing lights enables us to determine the model/manufacturer of the car during night time. This application can play an important role in road traffic police stealth monitoring and may be installed in secluded and rural areas where lighting conditions are scarce. Preferably, the system could be installed on a police vehicle for classifying the preceding vehicles and can be extremely beneficial for stealth missions, surveillance, security enforcement, ticketing vehicle without human interruption, etc. For this application to be most efficient, ideally a tweaked camera is to be used which filters only the red lights, thereby eliminating image bleeding. In this report only the software aspects are discussed and the hardware implementation part is out of scope of this project.

II. Prior work

The different techniques commonly employed for vehicle detection under daylight conditions have comprehensively been reviewed before in [2][1]. Although detection of vehicles during night time has been reviewed in the past, a system which actually determines the make of a car hasn't been implemented yet. As mentioned earlier, most of the features engaged in daylight car detection have limited use at dark conditions [2]. Vehicle shadows, vertical and horizontal edges, and corners are almost impossible to detect in darkness, making the rear facing brake lights as the most compelling preceding vehicle features in dark surroundings. For lamp detection, it is common to begin with some sort of thresholding [2]. Thresholding based on Grayscale or brightness is common to start with [2][3-6]. For color thresholding, the most common approach is to use the red-green-blue (RGB) color space [7-11]. Chung-che Wang et. al [12] proposed a vision- based driver assistance system to enhance the drivers safety in the nighttime [13]. The proposed system performs both lane detection and vehicle recognition (as detecting whether the light segmented is that of a vehicle or not) [13]. Ronan O' Malley et. al [2] have discussed the need for a system to avoid or mitigate forward collisions during night time by presenting an algorithm for forward collision detection at night using a visual camera [13]. The technique filters out red and white colors in the HSV space where white regions adjacent to red regions are searched for symmetrical pairs and aspect ratio constraints are applied to resulting bounding boxes, thereby producing detected rear target vehicle lights [13].

III. Night time car recognition implementation

The algorithm of this program comprises of two main parts:

- Segmenting the tail lights to extract certain prominent features
- Training the extracted features using Support Vector Machines.

The program is developed in MATLAB with the help of a third party tool called libsvm for classification.

A. Support Vector Machines (SVMs)

Support Vector Machine is a pattern classification and regression technique based on the mathematical foundations of statistical learning theory, which was first proposed by Vapnik in 1992 [14]. The primitive training principle of Support Vector Machines is to find an optimal hyper-plane to separated two classes linearly. The optimal hyper-plane is formed in such a way as to minimize the expected classification error for unseen test samples [15]. For binary classification, the training data are labeled {xi, yi}, where i=[1…n], yi belongs to {-1,1}, xi belongs to $\Re^d$, where d is the dimension of the vector, and n is the number of training vector [15]. The classification of a new pattern x can be obtained by solving the decision function f(x) as shown in equation below, where α_i is the Lagrange multipliers and b is the bias:

$$f(x) = sign\left(\sum_{i=1}^{n} y_i \alpha_i (x.x_i) + b \right)$$

However, in many cases, the data cannot be separated by a linear function. A solution is to map the input data into a higher dimension feature space $\phi(x)$ where classification can be performed by linear SVMs [16]; the decision function now being expressed as [16]:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b$$

where x is the input vector to be classified and K() is the kernel function. There are four types of kernels viz. [16]

- Linear: $K(x_i, x_j) = x_i^T x_j$
- Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- Radial basis function (RBF): $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$
- Sigmoid $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

where γ , r and d are kernel parameters.

For classification problems the optimal hyper plane is not able to separate the input vectors completely, so different classification types have been proposed for SVMs [15]. The most common one is known as C-SVC (C-Support vector classification). Given training vectors $x_i \in \Re^d$, C-SVC solves the following problem for binary classification $y_i \in \{-1,1\}$:

$$Minimize_{\xi, w, b} \left[\frac{1}{2} \langle w^T w \rangle + C \sum_{i=1}^{l} \xi_i \right] [15]$$

where w is the optimal hyper plane, stack variable ξ, allows some data to be miss-classified, and C $(C \in [0, \infty])$ is a priori constant [15].

The process of recognizing the tail lights of cars are divided into two phases: Training and Testing.

1. Training

I incorporate four different models for training purposes and each of these models is classified separately. The first model takes two features of the brake lights as training data: The number of lights and the aspect ratio of the cropped image formed by considering only the lights at the extremities. I found that RBF is the ideal type of kernel to be used for this model since the number of attributes is less compared to the number of instances. Both the training and testing data are scaled relatively in order to increase the accuracy.

The second model takes the pattern of the tail-lights as training data, as in the way the lights are distributed. Each of these cropped images is normalized to 30x360 pixels binary image, so the number of attributes per pattern image is 10800. Here the data is not scaled as it is binary and that is one of the reasons why the binary models are not combined with the first model.

The third and fourth model comprises of the contours of the extreme left and right lights, respectively. The centroid of these contours is marked as well, in order to increase efficiency. Each of these cropped images are then normalized to 75x75 pixels binary images, making the number of attributes per contour image as 5625. For the last three models, I discovered that the best classification result is obtained when training the SVM with C-SVM type, linear kernel and C=1 (since the number of features is way more abundant than the number of instances).

2. Testing

The testing image is classified in accordance with all four separate models, and four separate predicted labels are returned as the end result, respectively. The classified labels of the models are considered and the label that is most commonly occurring among these models is chosen as the final predicted label. If all four labels return different result, then it's highly possible that the model/type of the particular test vehicle is not in the database.

B. HSV Color Thresholding

The motive here is to filter only the red color component which will essentially remove all the non-vehicle light sources. As mentioned earlier, the most common technique uses the RGB color space where only the red channel of the RGB data is analyzed. However, this is inefficient because the bright non-red lights can also have a significant red-color component. The RGB color space is not ideal for the task of color thresholding, since it is difficult to set and manipulate color parameters due to a high correlation between the red, green and blue channels [2]. To overcome this issue, I make use of the HSV color space.

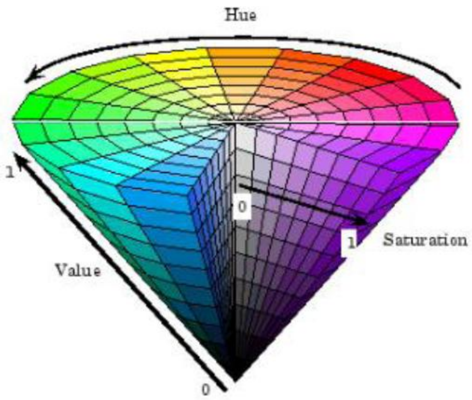

Fig 1: HSV color space

The HSV color space is based on three properties: Hue, Saturation and Value. Full color spectrum can be procured by manipulating these properties. The fundamental advantage of HSV is that each of these attributes directly corresponds to the basic color concepts, which makes it conceptually simple

[17]. HSV color space is much more intuitive than its RGB counterpart as it is easier to select continuous regions. The most important aspect pertaining to this application is the fact that the red color component is centered at Hue=0. Another important factor of this color space is that the white color component is the highest intensity parameter [18]. After a series of tries I found out that a Hue value between 0 to 9 degrees and between 342 to 0 degrees covers the entire range of red color, at least with regards to this application.

Fig 2: Input image

The next step is to eliminate all the non red color components (i.e. set the saturation property to zero) and then convert the resultant image to RGB type.

Fig 3: Converted RGB image after HSV manipulation

Next, the image is converted to gray scale and subtracted from the red channel result of the same RGB image as shown below

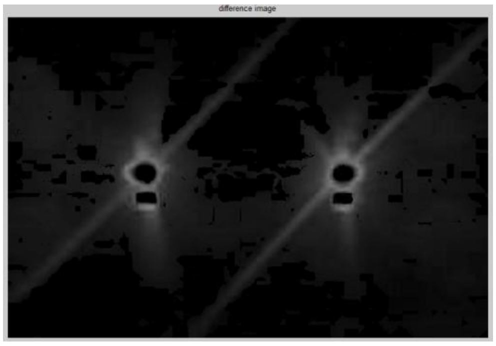

Fig 4: Difference Image

After that, a median filter is used to filter out the noise and then converted to a binary image. The holes are then isolated after a few image processing operations which gives us a white blob in place of the red lights as shown in figure below.

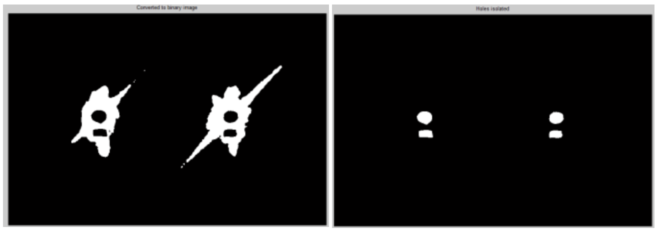

Fig 5: Resultant image converted to binary and then holes isolated.

C. *Method for determining the features of the first model*

The features used for the first model are: The number of lights and The Aspect ratio of the cropped image bounding only the tail lights situated at the extremities. The method of finding the former parameter is pretty straightforward. All we have to do is count the number of left over blobs.

The second parameter to be considered here is the aspect ratio. I take aspect ratio as a solid parameter because this parameter returns approximately similar values for same vehicles, regardless of the change in distance of the vehicle from the camera.

In order to find the aspect ratio, first the image is cropped by determining the horizontal and vertical (x, y) positions that bounds all of the white blobs in the image.

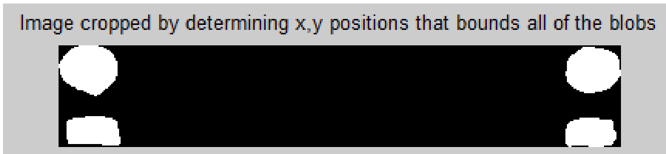

Fig 6: Cropped image

The next step is to retrieve only the lights at the extremities (i.e. the ones touching the left and right boundaries), crop it again, and then find the aspect ratio of the resulting image. This is done because the tail lights of a vehicle can have two different appearances: one without brakes applied and the other with brakes applied.

Consider the case of the vehicle in the image above (Volkswagen vortex (Fig 2)). When the brakes are not applied, only the round lights are lit up and the ones at the bottom wouldn't come into the picture. Whereas when the brakes are applied all of the lights are profusely illuminated. So, we need only take the aspect ratio of the primary appearance of tail lights (i.e. without brakes applied), as both the appearances classify as the same car.

To obtain only the lights at the extremities, the morphological operation named Reconstruction by dilation is performed. First, two vertical white lines are drawn: one at the extreme left and the other at the extreme right of the cropped image (planting of seeds). The initial image (cropped image) is treated as the Mask image. The Marker image is point-wise minimum between a single pixel wide white border and the Mask image. Reconstruction by dilation is then applied to the Mask and Marker image.

Fig 7: The marker image

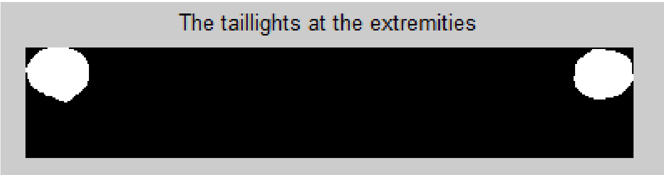

Fig 8: The resultant image after Reconstruction by Dilation

Next, the image is cropped once again to produce an output image which perfectly borders the blobs in the image in as compact a manner as possible. The aspect ratio (width/height ratio) is then found and taken as a parameter.

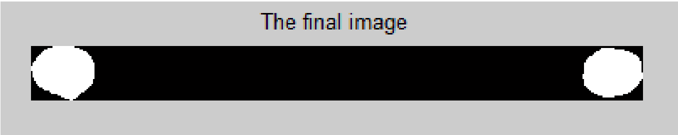

Fig 9: The final cropped image containing only the primary lights. Aspect Ratio = 0.0907

D. Method for determining the features of the second model

The second model deals with the way the rear lights are distributed. The initially cropped image is considered. A blank binary image of the same size is created (zeros). Then, a pixel (thickness=8) is drawn on the blank image at the relative positions of centroids detected for each of the individual blobs contained in the input image. The resulting image is considered as the pattern image.

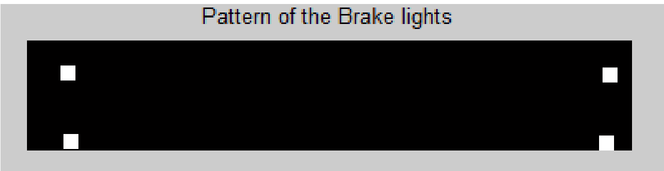

Fig 10: The Pattern of brake lights

This is then resized to 30x360 for classification purposes.

E. Method for determining the features of 3rd and 4th models

The third and fourth models represent the contours of the left and right extreme lights. The final image (i.e. the cropped image containing only the tail lights at extremities) is considered and the (x, y) positions are defined accordingly to retrieve only the left light. Next, the centroid of the only blob is found and set to black (zero). Finally the image is cropped again after performing a Sobel's edge detection. I included the centroid to make it more efficient for classification purposes, as the pattern (which includes the distance between the border pixels and centroid) will be taken into consideration.

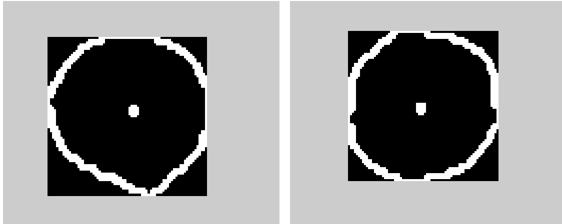

Fig 11: The contours of the left and right lights with centroid marked

The contour of the right light is found in the same manner except that the image is first mirrored and then cropped and then mirrored back again to its original state. The images are then resized to 75x75 for classification purposes.

IV. SAMPLE RESULTS

For classification, the parameters are first defined in the data and then they are sorted as a sparse matrix. This is done for the first model only, as it does not contain any null values.

```
1,4,0.072445
2,4,0.072289
3,4,0.07027
4,3,0.085533
5,6,0.087083
6,3,0.099387
7,2,0.10761
8,4,0.059582
9,4,0.060035
10,6,0.097396
11,7,0.09841
12,3,0.091969
13,2,0.090426
14,4,0.12912
15,5,0.11845
16,8,0.086869
17,3,0.12132
18,8,0.10856
19,8,0.086164
20,7,0.10625
21,3,0.096774
22,2,0.13158
23,3,0.14351
24,2,0.10437
25,5,0.14186
26,3,0.076768
27,2,0.070907
28,3,0.092426
29,2,0.065705
30,4,0.077485
31,2,0.059701
32,2,0.08418
33,3,0.083214
34,5,0.10603
```

Fig 12: Unsorted features

```
1 1:4 2:0.072445
2 1:4 2:0.072289
3 1:4 2:0.07027
4 1:3 2:0.085533
5 1:6 2:0.087083
6 1:3 2:0.099387
7 1:2 2:0.10761
8 1:4 2:0.059582
9 1:4 2:0.060035
10 1:6 2:0.097396
11 1:7 2:0.09841
12 1:3 2:0.091969
13 1:2 2:0.090426
14 1:4 2:0.12912
15 1:5 2:0.11845
16 1:8 2:0.086869
17 1:3 2:0.12132
18 1:8 2:0.10856
19 1:8 2:0.086164
20 1:7 2:0.10625
21 1:3 2:0.096774
22 1:2 2:0.13158
23 1:3 2:0.14351
24 1:2 2:0.10437
25 1:5 2:0.14186
26 1:3 2:0.076768
27 1:2 2:0.070907
28 1:3 2:0.092426
29 1:2 2:0.065705
30 1:4 2:0.077485
31 1:2 2:0.059701
32 1:2 2:0.08418
33 1:3 2:0.083214
34 1:5 2:0.10603
```

Fig 13: Sorted sparse features

The data is then scaled to the range [0, 1]. The first model of the test data is also scaled in the range [0, 1] *relative* to the training data. This is essential to obtain optimal results.

```
(18,1)        1.0000
(19,1)        1.0000
(20,1)        0.8333
(21,1)        0.1667
(23,1)        0.1667
(25,1)        0.5000
(26,1)        0.1667
(30,1)        0.3333
(33,1)        0.1667
(34,1)        0.5000
 (1,2)        0.1533
 (2,2)        0.1514
 (3,2)        0.1273
 (4,2)        0.3092
 (5,2)        0.2479
 (6,2)        0.4743
 (7,2)        0.5723
 (9,2)        0.0054
(10,2)        0.4506
(11,2)        0.4626
(12,2)        0.3859
(13,2)        0.3675
(14,2)        0.6870
(15,2)        0.7014
(16,2)        0.3251
```

Fig 14: A part of Data in the range [0, 1] (type 34x2 sparse double)

For the second, third and fourth models, the data are not converted to sparse matrix because it is in binary format and we do not want to get rid of the zeros.

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

Fig 15: Example: A part of the pattern data

Sample output:

the predicted labels for model1..
34

the predicted labels for model2..
11

the predicted labels for model3..
34

the predicted labels for model4..
34

In case of the above, label 34 is the most frequent one; hence the car will be classified as label 34.

Sample Output (GUI)

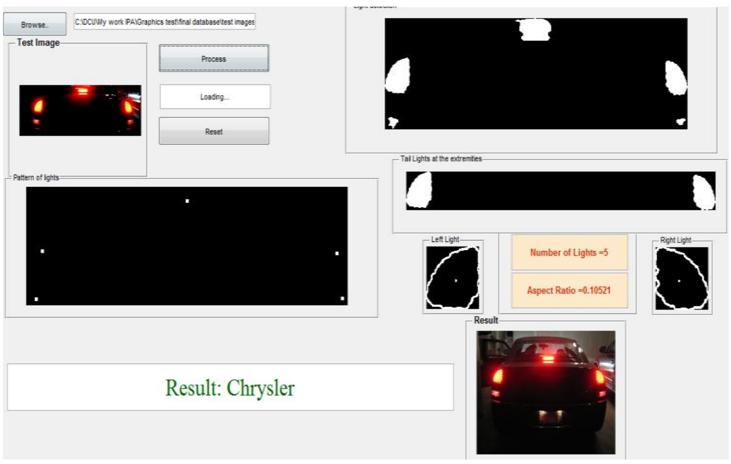

Fig 16: Sample GUI output

V. Analysis

I observed that better robust results are obtained when the different parameters are separated as different models (rather than combining all of them together). Classification can be rendered more precise if some sort of mechanism was introduced where probability estimates of all four models could be combined and processed to give one accurate result. There is a need to expand the database with the two different appearances of each vehicle as mentioned earlier.

VI. Conclusion

The approach of Night time car recognition has been successfully proposed in this report. Nonetheless, there is a lot of room for improvement before it could be brought out into the real world. As previously stated, blending a hardware implementation with this application is very much necessary. A standard color camera hardware which incorporates a lower static exposure could be used to essentially eliminate color bleeding. Furthermore, the development of an embedded operation of this application which utilizes automatic exposure control, thereby enabling itself to dynamically establish an optimum appearance of tail lights, may be considered for future works.

References

[1] "Traffic safety Basic Facts 2007-Motorways,"Eur. Road Safety Observatory, 2007.

[2] Ronan O'Malley, Edward Jones, *Member, IEEE*, and Martin Glavin, *Member, IEEE* -Rear-Lamp Vehicle Detection and Tracking in Low-Exposure Color Video for Night Conditions

[3] N. Alt, C. Claus, and W. Stechele, "Hardware/software architecture of an algorithm for vision-based real-time vehicle detection in dark environments," in *Proc. Conf. Design, Autom. Test Eur.*, 2008, pp. 176–181.

[4] S.-C. Kim and T.-S. Choi, "Fast vision-based vehicle detection algorithm using recognition of light pattern," in *Proc. Appl. Digital Image Process. XXIII*, 2000, vol. 4115, pp. 654–661.

[5] S.-Y. Kim, S.-Y. Oh, J.-K. Kang, Y.-W. Ryu, K.-S. Kim, S.-C. Park, and K.-H. Park, "Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 2173–2178.

[6] P. Alcantarilla, L. Bergasa, P. Jimenez, M. Sotelo, I. Parra, D. Fernandez, and S. Mayoral, "Night time vehicle detection for driving assistance lightbeam controller," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 291–296.

[7] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Mach. Vis. Appl.*, vol. 12, no. 2, pp. 69–83, Aug. 2000.

[8] Y.-L. Chen, "Nighttime vehicle light detection on a moving vehicle using image segmentation and analysis techniques," *WSEAS Trans. Comput.*, vol. 8, no. 3, pp. 506–515, Mar. 2009.

[9] M.-Y. Chern and P.-C. Hou, "The lane recognition and vehicle detection at night for a camera-assisted car on highway," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2003, vol. 2, pp. 2110–2115.

[10] R. Sukthankar, "Raccoon: A real-time autonomous car chaser operating optimally at night," in *Proc. IEEE Intell. Vehicles Symp.*, Jul. 1993, pp. 37–42.

[11] C.-C.Wang, S.-S. Huang, and L.-C. Fu, "Driver assistance system for lane detection and vehicle recognition with night vision," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3530–3535.

[12] Ming-Yang Chem , Ping-Cheng Hou, National Chung Cheng University, Min-Hsiung, Chia-Yi, Taiwan, The Lane Recognition and Vehicle Detection at Night for A Camera-Assisted Car on Highway, Proceedings of the *1003* IEEE, International Conference on Robotics & automation, Taipei. Taiwan, September 14-19, 2003.

[13] Prof. V.B. Sutar, Dr. Mrs. L. S. Admuthe , Department of electronics and telecommunication engineering, TEI, Rajwada, Ichalakaranji, India: Night time vehicle detection and classification using Support vector machines.

[14] Vapnik V.N., "The nature of statistical learning theory", New York: Springer, 1995

[15] Than Bui Minh," An automatic licence plate recognition system based on Support Vector Machines (SVMs).

[16] Paul F Whelan (2011), "EE544: Computer Vision Course Notes", Centre for Image Processing & Analysis, Dublin City University.

[17] http://en.wikipedia.org/wiki/HSL_and_HSV

[18] Ronan O'Malley Connaught Automotive Research Group Department of Electronic Engineering National University of Ireland, Galway ISVCS 24th July 2008- Vehicle detection at night using tail light detection.

[19] Paul F Whelan (2011) "VSG Image Processing and Analysis (VSG IPA) Toolbox", Technical Report, Centre for Image Processing & Analysis, Dublin City University.

[20] Paul F Whelan (2011) "VSG Image Processing and Analysis (VSG IPA) Matlab Toolbox - Software"", http://www.cipa.dcu.ie/code.html, Centre for Image Processing & Analysis, Dublin City University.

[21] Paul F. Whelan and D. Molloy (2000), "Machine Vision Algorithms in Java: Techniques and Implementation", Springer (London), 298 Pages. ISBN 1-85233-218-2.

[22] Paul F Whelan (2011), "EE425/EE453: Image Processing and Analysis Course Notes", Centre for Image Processing & Analysis, Dublin City University.