Traffic Sign Detection and Recognition Using OpenCV

Mrs. P. Shopa
Department of CSE
Agni College of Technology
Chennai, Tamil Nadu,India.
shopa1990@gmail.com

Mrs. N. Sumitha
Assistant Professor
Department of CSE
Agni College of Technology
Chennai , Tamil Nadu, India
sumitha.cse@act.edu.in

Dr. P.S.K Patra
Head of the Department
Department of CSE
Agni College of Technology
Chennai , Tamil Nadu, India
csehod@act.edu.in

***Abstract*— The aim of the project is to detect and recognize traffic signs in video sequences recorded by an on-board vehicle camera. Traffic Sign Recognition (TSR) is used to regulate traffic signs, warn a driver, and command or prohibit certain actions. A fast real-time and robust automatic traffic sign detection and recognition can support and disburden the driver and significantly increase driving safety and comfort. Automatic recognition of traffic signs is also important for automated intelligent driving vehicle or driver assistance systems. This paper presents a study to recognize traffic sign patterns using openCV technique. The images are extracted, detected and recognized by pre-processing with several image processing techniques, such as, threshold techniques, Gaussian filter, canny edge detection, Contour and Fit Ellipse. Then, the stages are performed to detect and recognize the traffic sign patterns. The system is trained and validated to find the best network architecture. The experimental results show the highly accurate classifications of traffic sign patterns with complex background images and the computational cost of the proposed method**.

***Keywords*— Traffic sign detection and recognition; OpenCV; HSV Algorithm; Gaussian Filter;**

I. INTRODUCTION

In traffic environments, Traffic Sign Recognition (TSR) is used to regulate traffic signs, warn the driver, and command or prohibit certain actions. A fast real-time and robust automatic traffic sign detection and recognition can support and disburden the driver, and thus, significantly increase driving safety and comfort. Generally, traffic signs provide the driver various information for safe and efficient navigation. Automatic recognition of traffic signs is, therefore, important for automated intelligent driving vehicle or driver assistance systems. However, identification of traffic signs with respect to various natural background viewing conditions still remains challenging tasks. The Traffic Sign Recognition Systems usually have developed into two specific phases. The first is normally related to the detection of traffic signs in a video sequence or image using image processing. The second one is related to recognition of these detected signs, which is deal with the interest of performance in artificial neural network. The detection algorithms normally based on shape or color segmentation. The segmented potential regions are extracted to be input in recognition stage. The efficiency and speed of the detection play important role in the system. To recognize traffic signs, various methods for automatic traffic sign identification

have been developed and shown promising results. Neural Networks precisely represents a technology that used in traffic sign recognition. One specific area in which many neural network applications have been developed is the automatic recognition of signs. The difficulties of traffic sign detection and recognition are involved with the performance of system in real time. High efficient algorithms and powerful performance hardware are required in the system. Furthermore, the environment constraint included lighting, shadow occlusion, air pollution, weather conditions (sunny, rainy, foggy, etc.) as well as the additional image distortions, such as, motion burl, vehicle vibration, and abrupt contrast changes possibly occur frequently in the actual system.

In recently studies, the detection and recognition of traffic signs have been developed in many research centers. A vision system for the traffic sign recognition and integrated autonomous vehicle was developed in part of European research project PROMETETHEUS at DAIMLER-BENZ Research Center. Moreover, many techniques have been developed for road sign recognition, for example, Pacheco et al. used special color barcodes under road sign for detecting road sign in vision-based system. This took a lot of time and resources. Genetic algorithm was also proposed by Aoyagi and Askura to identify road sign from gray-level images, but the limitation of crossover, mutation operator, and optimal solution are not guaranteed. Color indexing was propose by

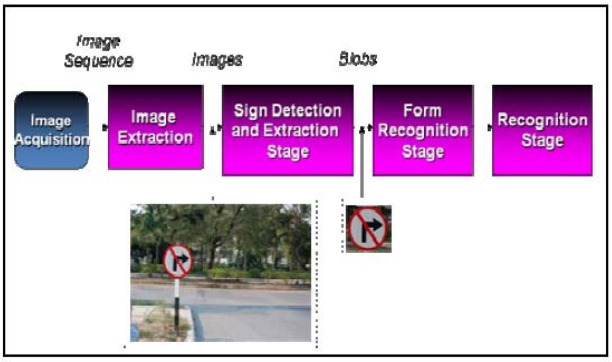
ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

Lalonde and Li to approach identifying road sign, unfortunately, the computation time was not allowed to be accepted in the complex traffic scenes. This paper is proposed to develop the real implementation using in intelligent vehicle.

The main objective is to reduce the search space and indicate only potential regions for increasing the efficiency and speed of the system. A higher robust and faster intelligent algorithm is required to provide the necessary accuracy in recognition of traffic signs. In the detection phase, the acquisition image is pre-processed, enhanced, and segmented according to the sign properties of color and shape. The traffic sign images are investigated to detect potential pixel regions which could be recognized as possible road signs from the complex background. The potential objects are then normalized to a specified size and input to recognition phase. This study investigates only to circle and hexagonal shape objects because these shapes normally present in many types of traffic signs. Multilayer Perceptron (MLP) with respect to back propagation learning algorithm is an alternative method to approach the problem of recognizing sign in this work.

II. SYSTEM DESCRIPTION

The first section is Image Extraction and Sign Detection and Extraction parts. The video images have been taken by a video camera, and Image Extraction block is the responsible for creating images. The Sign Detection and Extraction Stage extracts all the traffic signs contained in each image and generates the small images called blobs. Each blob will be performed by Form Recognition Stage to be valuable parameters input to Artificial Neural Networks in Recognition Stage which is the final part. Then the output of traffic sign recognition will be presented. The description of the traffic sign recognition system can be explained into Traffic Sign Pre-processing Stage and Recognition Core. For Traffic Sign Pre-processing Stage, it is divided in two parts: Sign Detection and Extraction and Form Recognition Stage.



A. Traffic Sign Detection

This stage is the image processing procedure. Image input from video sequence which is the natural background viewing image fed into the system. The image data is read both in color, and black and white mode Due to the black and white mode image is the base image that used to find the threshold of this image, this threshold is the criterion to change image from black and white to binary image. Moreover, the binary image is used to find contour and the interested region later on. Before the black and white image change to binary, the technique of smooth image with Gaussians filter and Canny edge detection to enhance image. Therefore, it shows that the smooth technique has potential to enhance the image to obtain the required region.

Figure 1.2 Image extracted

According to obtaining the binary image, it is processed to retrieve contours by find contour function for binary image and to return the number of retrieved contours which stored in the chain format. The OpenCV library uses two methods to represent contours. The first method is called the Freeman method or the chain code. For any pixel all its neighbors with numbers from 0 to 7 can be enumerated. The 0-neighbor denotes the pixel on the right side, etc. As a sequence of 8-connected points, the border can be stored as the coordinates of the initial point, followed by codes (from 0 to 7) that specify the location of the next point relative to the current one.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

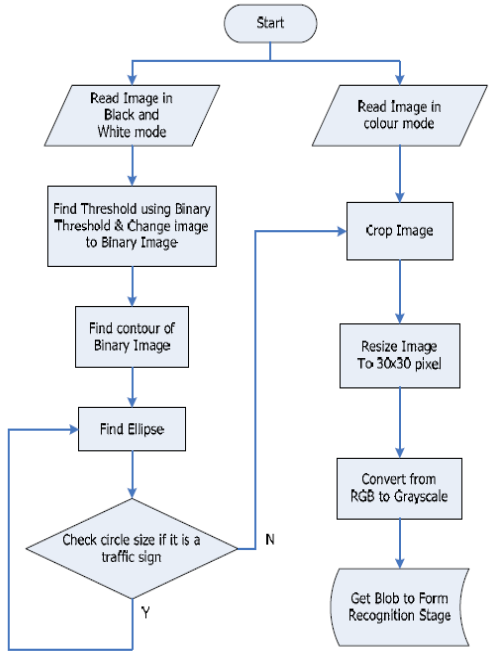


Figure 1.3 Block Diagram of pre-processing technique

B. *Traffic Sign Recognition*

Road sign recognition is a significant and essential part in intelligent vehicle navigation system (IVNS). The texts and logo embedded in a road sign usually contain much useful information such as guided direction and current traffic situations. However, it is a hard task to find and extract road signs directly from a single natural image due to complex background, variable light condition, appearance degeneration, perspective effects caused by camera, and so on. Recognition of road signs is a challenging problem that has engaged the attention of the Computer Vision community for more than 30 years. The first study of automated road sign recognition was reported in Japan in 1984. Since then, a large number of methods have been developed for road sign detection and identification. Since road signs' color is normally highly contrasted to their backgrounds and rectangle is often designed for road sign, previous approaches can be divided into two categories, that is, color-based and shape-based methods.

III. TRAFFIC SIGN DETECTION

Traffic Sign Detection (TSR) is a driver support function which can be used to notify and warn the driver which restrictions may be effective on the current stretch of road. Examples for such regulations are ‘speed limit zones’ or ‘no-overtaking’ indications. The system can help the driver to maintain a legal speed, obey local traffic instructions, or urban restrictions. The system recognizes and interprets various traffic signs, both fixed signs on the road side and variable LED signs overhead, using vision-only information and therefore signs which may be obscured by other vehicles or trees may not be recognized. Here the algorithm based on its core competence in vehicle and pedestrian detection. The algorithm shares the attention, classification and tracking framework of those other modules and uses the robust classifiers developed in those applications, trained on different examples. The algorithm used to detect is the HSV algorithm and for image extraction Gaussian filter is used.

A. *HSV Algorithm*

The traffic sign is detected by using the HSV algorithm. HSV are the two most common cylindrical-coordinate representations of points in an RGB color model. The two representations rearrange the geometry of RGB in an attempt to be more intuitive and perceptually relevant than the cartesian (cube) representation. Developed in the 1970s for computer graphics applications, HSV are used today in color pickers, in image editing software, and less commonly in image analysis and computer vision. HSV stands for *hue*, *saturation*, and *value*, and is also often called HSB (*B* for *brightness*). However, while typically consistent, the definition are not standardized, and any of these abbreviations might be used for any of these three or several other related cylindrical models.

In each cylinder, the angle around the central vertical axis corresponds to "hue", the distance from the axis corresponds to "saturation", and the distance along the axis corresponds to "lightness", "value" or "brightness". Note that while "hue" in HSL and HSV refers to the same attribute, their definitions of "saturation" differ dramatically. Because HSL and HSV are simple transformations of device-dependent RGB models, the physical colors they define depend on the colors of the red, green, and blue primaries of the device or of the particular RGB space, and on the gamma correction used to represent the amounts of those primaries. Each unique RGB device therefore has unique HSL and HSV spaces to accompany it, and numerical HSL or HSV values describe a different color for each basis RGB space.[1]

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

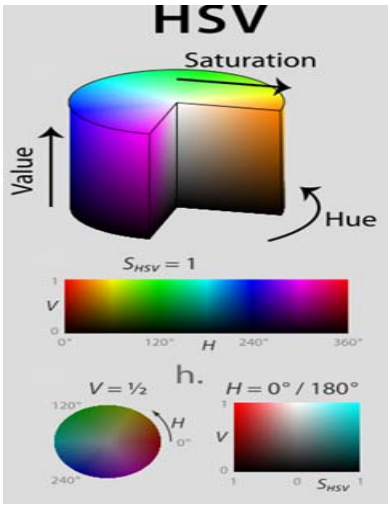


Figure 4.1 HSV

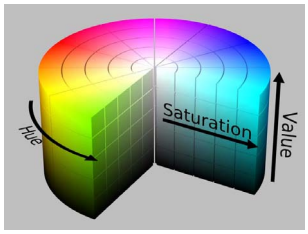


Figure 4.2 Cylindrical Geometries of HSV

HSV are cylindrical geometries, with hue, their angular dimension, starting at the red primary at 0°, passing through the green primary at 120° and the blue primary at 240°, and then wrapping back to red at 360°. In each geometry, the central vertical axis comprises the *neutral*, *achromatic*, or *gray* colors, ranging from black at lightness 0 or value 0, the bottom, to white at lightness 1 or value 1, the top. In both geometries, the additive primary and secondary colors – red, yellow, green, cyan, blue, and magenta – and linear mixtures between adjacent pairs of them, sometimes called *pure colors*, are arranged around the outside edge of the cylinder with saturation 1; in HSV these have value 1 while in HSL they have lightness ½. In HSV, mixing these pure colors with white – producing so-called *tints* – reduces saturation, while mixing them with black – producing *shades* – leaves saturation unchanged. In HSL, both tints and shades have full saturation, and only mixtures with both black and white – called *tones* – have saturation less than 1.

The definitions of saturation – in which very dark (in both models) or very light (in HSL) near-neutral colors, for instance ■ or □, are considered fully saturated – conflict with the intuitive notion of color purity, often a conic or bi-conic solid is drawn instead (fig. 3), with what this article calls *chroma* as its radial dimension, instead of saturation. Confusingly, such diagrams usually label this radial dimension "saturation", blurring or erasing the distinction between saturation and chroma.[2] As described below, computing chroma is a helpful step in the derivation of each model. Because such an intermediate model – with dimensions hue, chroma, and HSV value takes the shape of a cone or bicone, HSV is often called the "hexcone model"

The *hue* is the proportion of the distance around the edge of the hexagon which passes through the projected point, originally measured on the range [0, 1) but now typically measured in degrees [0°, 360°). For points which project onto the origin in the chromaticity plane (i.e., grays), hue is undefined. Mathematically, this definition of hue is written piecewise:[21]

$$H' = \begin{cases} \text{undefined}, & \text{if } C = 0 \\ \frac{G-B}{C} \bmod 6, & \text{if } M = R \\ \frac{B-R}{C} + 2, & \text{if } M = G \\ \frac{R-G}{C} + 4, & \text{if } M = B \end{cases}$$

$$H = 60° \times H'$$

Sometimes, neutral colors (i.e. with $C = 0$) are assigned a hue of 0° for convenience of representation. HSV models scale the chroma so that it always fits into the range [0, 1] for every combination of hue and lightness or value, calling the new attribute *saturation* in both cases (fig. 14). To calculate either, simply divide the chroma by the maximum chroma for that value or lightness.

$$S_{HSV} = \begin{cases} 0, & \text{if } C = 0 \\ \frac{C}{V}, & \text{otherwise} \end{cases}$$

B. Gaussian Algorithm

A Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales—see scale space representation and scale space implementation.

Mathematically, applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function. This is also known as a two-dimensional Weierstrass transform.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

The Gaussian blur is a type of image-blurring filter that uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image. The equation of a Gaussian function in one dimension is

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

in two dimensions, it is the product of two such Gaussians, one in each dimension:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. When applied in two dimensions, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the center point. Values from this distribution are used to build a convolution matrix which is applied to the original image. Each pixel's new value is set to a weighted average of that pixel's neighborhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receive smaller weights as their distance to the original pixel increases. This results in a blur that preserves boundaries and edges better than other, more uniform blurring filters; see also scale space implementation.

Gaussian blurring is commonly used when reducing the size of an image. When downsampling an image, it is common to apply a low-pass filter to the image prior to resampling. This is to ensure that spurious high-frequency information does not appear in the downsampled image (aliasing). Gaussian blurs have nice properties, such as having no sharp edges, and thus do not introduce ringing into the filtered image.

Low-pass filter: Gaussian blur is a low-pass filter, attenuating high frequency signals. Its amplitude Bode plot (the log scale in the frequency domain) is a parabola.

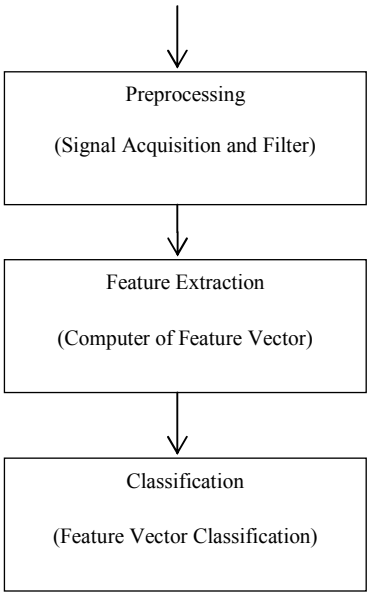
Implementation: A Gaussian blur effect is typically generated by convolving an image with a kernel of Gaussian values. In practice, it is best to take advantage of the Gaussian blur's separable property by dividing the process into two passes. In the first pass, a one-dimensional kernel is used to blur the image in only the horizontal or vertical direction. In the second pass, another one-dimensional kernel is used to blur in the remaining direction. The resulting effect is the same as convolving with a two-dimensional kernel in a single pass, but requires fewer calculations.

Gaussian smoothing is commonly used with edge detection. Most edge-detection algorithms are sensitive to noise; the 2-D Laplacian filter, built from a discretization of the Laplace operator, is highly sensitive to noisy environments. Using a Gaussian Blur filter before edge detection aims to reduce the level of noise in the image, which improves the result of the following edge-detection algorithm. This approach is commonly referred to as Laplacian of Gaussian, or LoG filtering.

IV. Traffic Sign Recognition

Road sign recognition research has been around since the mid 1980's. High variance of sign appearance has made the detection and recognition of road signs a computer vision problem over which many studies have lately been performed. Road signs use particular colors and geometric shapes to attract drivers' attention. However, the difficulty in recognizing road signs is largely due to the following reasons: (1) Colors may fade after long exposure to the sun. Moreover, paint may even flake or peel off, and signs may get damaged. (2) Air pollution and weather conditions may decrease the visibility of road signs. (3) Outdoor lighting conditions vary from day to night and may affect the apparent colors of road signs. (4) Obstacles, such as trees, poles, buildings, and even vehicles and pedestrians, may occlude or partially occlude road signs. (5) Video images of road signs often suffer from blurring in view that the camcorder is mounted on a moving vehicle.

In a typical pattern recognition classifier consist in three modules:



ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

Preprocessing: in this module we go to process our input image, for example size normalize, convert color to BN…

Feature extraction: in this module we convert our image processed to a characteristic vector of features to classify, it can be the pixels matrix convert to vector or get contour chain codes data representation.

Classification module get the feature vectors and train our system or classify an input feature vector with a classify method as knn.

Each image we get is pre-processed and then convert the data in a feature vector we use. After processed and get train data and classes we then train our model with this data, in our sample we use knn method then:

```
knn=new CvKNearest( trainData, trainClasses, 0, false, K );
```

Then we now can test our model, and we can use the test result to compare to another methods we can use, or if we reduce the image scale or similar. There are a function to create the test in our basicOCR class, test function.This function get the other 500 samples and classify this in our selected method and check the obtained result.Test use the classify function that get image to classify, process image, get feature vector and classify it with a find_nearest of knn class. This function we use to classify the input user images:

V. Conclusion

This study discussed Traffic Sign Recognition (TSR) using OpenCV application. The images were pre-processed in stages with image processing techniques, such as, threshold technique, Gaussian filter, canny edge detection, Contour and Fit Ellipse. Then, these stages were performed to recognize the traffic sign patterns. The main reason to select this method is to reduce the computational cost in order to facilitate the real time implementation. The first strategy is to reduce the number of MLP inputs by pre-processing the traffic sign image, and the second strategy is to search for the best network architecture which reduced complexity by selecting a suitable error criterion for training. The system were trained with training data set, and validated with validating data set to find the best network architecture. The cross-validation technique was implemented with training data set, validating data set, and test set. The experiments show consistency results with accurate classifications of traffic sign patterns with complex background images. The processing time in each frame of image is provided which is satisfied to apply in the real application.

In my future enhancement, the detection and recognition of the traffic signs are merged together and implemented in hardware to notify the vehicle driver about the traffic signs.

References

1) R. Vicen-Bueno, R. Gil-Pita, M.P. Jarabo-Amores and F.L´opez-Ferreras, “Complexity Reduction in Neural Networks Applied to Traffic Sign Recognition”, *Proceedings of the 13th European SignalProcessing Conference*, Antalya, Turkey, September 4-8, 2005.
2) R. Vicen-Bueno, R. Gil-Pita, M. Rosa-Zurera, M. Utrilla-Manso, and F.Lopez-Ferreras, “Multilayer Perceptrons Applied to Traffic Sign Recognition Tasks”, *LNCS 3512, IWANN 2005, J. Cabestany, A. Prieto, and D.F. Sandoval (Eds.)*, Springer-Verlag Berlin Heidelberg 2005, pp.865-872.
3) H. X. Liu, and B. Ran, “Vision-Based Stop Sign Detection and Recognition System for Intelligent Vehicle”, *Transportation Research Board (TRB) Annual Meeting 2001*, Washington, D.C., USA, January7-11, 2001.
4) H. Fleyeh, and M. Dougherty, “Road And Traffic Sign Detection And Recognition”, *Proceedings of the 16th Mini - EURO Conference and 10th Meeting of EWGT*, pp. 644-653.
5) S. Kang, N. C. Griswold, and N. Kehtarnavaz, “An Invariant Traffic Sign Recognition System Based on Sequential Color Processing and Geometrical Transformation”, *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation* Volume , Issue , 21-24 Apr 1994, pp. 88 – 93.
6) M. Rincon, S. Lafuente-Arroyo, and S. Maldonado-Bascon, “Knowledge Modeling for the Traffic Sign Recognition Task”, Springer Berlin / Heidelberg Volume 3561/2005, pp. 508-517.
7) Y. Fang, C. S. Fuh, P. S. Yen, S. Cherng, and S. W. Chen, “An Automatic Road Sign Recognition System based on a Computational Model of Human Recognition Processing”, *Computer Vision and Image Understanding*, Vol. 96 , Issue 2 (November 2004), pp. 237 – 268.
8) Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, T. Koehler, “A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information”, *Proceedings of the 2005 IEEE Intelligent Vehicles Symposium* , Las Vegas, USA., June 6 - 8, 2005.
9) L. Pacheco, J. Batlle, X. Cufi, “A new approach to real time traffic sign recognition based on colour information”, *Proceedings of the Intelligent Vehicles Symposium*, Paris, 1994, pp. 339–344.
10) Y. Aoyagi, T. Asakura, “A study on traffic sign recognition in scene image using genetic algorithms and neural networks”, *Proceedings of the IEEE IECON Int. Conf. on Industrial Electronics, Control, and Instrumentation*, Taipei, Taiwan, vol. 3, 1996, pp. 1838–1843.
11) M. Lalonde, Y. Li, Detection of Road Signs Using Color Indexing, Technical Report CRIM-IT-95/12-49, Centre de Recherche Informatique de Montreal. Availablefrom: publications.html, 1995.Intel Corporation, “Open Source Computer Vision Library,” Reference Manual, Copyright © 1999-2001, Available: www.developer.intel.com Laurence Fausett, “Fundamentals of Neural Networks Architectures, Algorithms, and Applications”, Prentice Hall Upper Saddle River, New Jersey 1994.
12) G. Schwarzer, W. Vach, and M. Schumacher, “On the Misuses of Artificial Neural Networks for Prognotic and Diagnostic Classification in Oncology,” *Statistics in Medicine,* 2000, Vol. 19, pp. 541-561.
13) S. Kumar, “Neural Networks A Classroom Approach”, Mc Graw Hill 2005.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE