IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 13, Issue 4, Ver. I (Jul.-Aug. 2018), PP 85-97
www.iosrjournals.org

Automated System for Detection of Vehicle Features

Navdeep Kumar
*[*1]ECE Department, Ch. Devi Lal State Institute of Engineering & Technology, Panniwala Mota,(Sirsa),*

Abstract: *Automation is the process of detecting traffic rule violation and recognition of the vehicle. Violating the traffic rule is also a problem of great concern, with so many violations taking place, it would be humanely impossible to scrutinize each and every violation and bring the defaulter to justice.*
The above problem can be basically broken down into three parts. The first would be to detect the vehicle which has violated the traffic rule, which is an independent problem in itself. The second part would be to recognize the vehicle on the basis of extraction of the number plate. The third would be to identify different features of the vehicle like the color of the vehicle, its class and further the corresponding model which would help in the precise recognition of the vehicle. In this paper, we have concentrated our efforts on the third part of the problem. Algorithms have been developed and implemented for independent recognition of the vehicle on the basis of its class, model and color. Karhunen-Loeve transform in combination with Support Vector Machines has been used to obtain information about class of the vehicle. An algorithm has been developed and implemented to detect the color of the vehicle by means of modification of the color map of the image and by using RGB histograms. Finally Scale Invariant Feature Transform (SIFT and PCA-SIFT) has been made use of foridentifying the vehicle model. Very high recognition rates is being obtained in all the cases.
Key Words: *KL Transform, SHIFT, PCA-Shift,Vehicle class Identification*

I. Introduction

The automated extraction of different features of the vehicle is of considerable interest because of its potential applications to areas of controlling the traffic violations, automatic payment of toll fee, identification of stolen vehicles etc. A lot of previous approaches rely on the number plate to obtain complete information about the vehicle like its type, model and its color. A more robust approach to identify a vehicle would be to design algorithms to independently recognize vehicle type and model directly from the cameraimages.

Vehicle Make and Model Recognition (MMR) techniques provide an important functional enhancement to automatic vehicle identification systems that have traditionally been based solely on Automatic Number Plate Recognition (ANPR). A robust vehicle MMR technique would be capable of recognizing vehicle makes and models under varying environment and capture conditions. The idea is to use feature-based approaches to first identify distinct features of different makes and models, which are then matched under controlledcriteria.

1.1 ProblemDefinition

The objective of this paper is to develop a system to automatically identify different features of the vehicle, which would help in precisely identifying it with good results and the updated simulations with optimization of outputs by using SIFT and PCA.
Algorithms have been developed along with the GUI for the following tasks:
1. Class Identification: This part deals with broadly classifying a vehicle in the image as to be a large car, a small car, a bike or a scooter.
2. Model Identification: Once the vehicle has been categorized as per its class, it can be tested on a smaller set of specific database to obtain information about itsmodel.
3. Color Identification: Color of a vehicle is the first thing which comes to notice when you see it. So, color identification would be a vital process in identifying thevehicle.

1.2 Literature review

Robust and reliable vehicle detection from images acquired by a moving vehicle has been an interesting problem of research for decades. A lot of approaches have been developed and used in the past in this regard. One of the pioneers in the field of feature extraction and detection were Hiroshi Murase and Shree. K. Nayar [5] who conceptualized Principal Component Analysis for object recognition and poses estimation. The recognition problem was formulated as one of matching appearance rather than shape. Appearance-based methods learn the characteristics of the vehicle class from a set of training images, which capture the variability in vehicle appearance, which in turn helps in accurate identification of the vehicle. Zehang Sun, George Bibis

and Ronald Miller used Gabor filters [11] [12] specially optimized for the task of vehicle detection. Here the problem was identified to be one of feature selection. Various other vehicle detection approaches have also been reported in the computer vision literature. Bertozziet
al. [13], and Zhao et al. [14] used stereo-vision-based methods (e.g., inverse perspective mapping) to detect vehicles and obstacles. In Matthews et al. [15], PCA was used for feature extraction and neural networks for detection. Goerick et al. [16] used a method called Local Orientation Coding to extract edge information and neural networks for vehicle detection. Betke et al.
[17]used motion and edge information to hypothesize the vehicle locations and template- matching for detection. In Schneiderman et al. [18], the statistics of both object appearance and "non- object" appearance were represented using the product of two histograms with each histogram representing the joint statistics of a subset of wavelet coefficients and their position on the object. Papageorgiou et al. [19] proposed a general object detection scheme using wavelets andSVMs.

1.3 Brief overview of thealgorithms

1.3.1 Vehicle classidentification

An algorithm has been developed and implemented to classify the vehicle broadly into a particular class so that it can be analyzed later for model identification. Here, we have outlined a technique to automatically learn the three dimensional objects from their two dimensional views. [5] The appearance of an object is the combined effect of its shape, reflectance properties, and poses in the scene and the illumination conditions. Recognizing objects from brightness images is therefore more a problem of appearance matching rather than shape matching. The main problem is to compress the large image set to a low dimensional representation of object appearance.

Well-known image compression technique known as Karhunen-Loeve transform was used for this purpose. A learning database of 337 vehicle images consisting of 203 medium-build cars, 68 large-build cars, 41 bikes and 25 scooters belonging to 19 models overall was used to construct 4 different eigen spaces [5]. The eigenvectors form an orthogonal basis for representing individual images in the set. Though a large number of eigenvectors may be required for very accurate reconstruction of the object image, only a few are generally sufficient to capture the significant appearance characteristics of an object. From the perspective of machine vision, the eigenspace has a very attractive property. It is optimal in correlation sense: if any two images from the set are projected to eigenspace, the distance between the corresponding points in the eigenspace is a measure of similarity of theimages.

1.3.2 Vehicle coloridentification

The color of an image is characterized by its color space. A large number of color spaces have been defined, RGB, HSV, YCbCr to name a few. Therefore to identify a color, we just need to identify its color space and then build the corresponding color histogram to obtain the values of the three components, which constitute the colorspace.
Most of the images we are dealing with belong to the RGB color space, therefore we need to build the Red, Green and the Blue histogram of the dominant color in the image to obtain the RGB component values which can be compared with the RGB values of the standard colors to obtain the vehicle color information, correct to the shade.
We used a look up table of 110 colors, corresponding to different shades of Red, Blue, Yellow, Black, Gray, Green etc. The region of interest whose color needs to be identified was extracted using a GUI program (SelectROI) and its bitmap was reduced from 256 colors to 3 colors for a more convenient identification. The RGB values obtained were matched to the values in the look up table by means of the Euclidean distance.

1.3.3 Vehicle Modelidentification

Earliest algorithm developed and implemented consisted of projecting the images onto aeigen space built from the vehicles of a specific class [5]. But, the normal distance correlation did not work in this regard since the images were highly correlated.
FurtherHu"smomentinvariants[6],FlusserandSuk"saffinemomentinvariants[7]andOrientationdependantmodified Hu"smomentinvariants[8]werealso used as a basis for identifying the vehicle model, but they were not found to be sufficiently successful in segregating thevehicles.
Therefore, a more robust classifying mechanism in the form of Support Vector Machines was sought for. SVM [2] [3] [4] was introduced by Boser, Guyon and Vapnik and has been greatly developed ever since for the purpose of pattern recognition. The challenge is to detect and exploit complex patterns in the data set and classify them on basis of the same by introducing a notion of similarity. The Kernel methods, of which SVM is a part, exploit information about the inner products between data items. The data items considered are projected onto a dot-product space and a separating hyperplane is constructed based on the data so as to separate objects

belonging to two different classes. The initial algorithm mainly consisted of a combination of SVD and SVM, wherein SVD was used to obtain the defining features of the vehicle under consideration. A total of 273 images were used in the learning phase to train the SVM using the open source LibSVM[1] module, which was developed by Chih-Chung Chang and Chih-Jen Lin. The algorithm was tested for 176 test images and a recognition rate of67%wasobtained.Otheralgorithmsconsistingof Moment invariants with SVM, both Hu‟s and Affine were also developed and experimented, but the recognition rate in these cases was low.

Our final algorithm involves SIFT, Scale Invariant Feature Transform, developed by David Lowe [9] and its variant, PCA-SIFT [10], developed by Yan Ke, Carnegie Mellon University and Rahul Sukthankar of Intel Research, Pittsburgh. SIFT is a four step procedure wherein we generate points of interest or keypoints from the image by building a scale-space combined with the process of finding the local maxima of a set of difference of Gaussian (DoG) images. A 128-dimensional feature vector is then obtained for each key point by assigning orientations to each key point and taking a histogram of a 16 x 16 patch around the keypoint. The feature vectors thus obtained are used for mapping the similarity of the images in question and depending on the number of matches obtained, the test image is considered to be a match or a mismatch to the original image. A particular disadvantage of this method is the time required to make a match in case of a large database likeours. To alleviate this disadvantage the PCA-SIFT was used. Here the first three steps are the same as that of SIFT, but in the final step, instead of generating the feature vector from a histogram, it is generated by performing Principal Component analysis (PCA) on a 41 x 41 patch around the keypoint, and a 36 dimensional vector is obtained. By doing so, the time required for a match is reduced by 40% and the classification also becomes more robust. For example, a total of 203 images of cars belonging to 12 different models were used as the database. Each test image was matched with all the images in the database and the image in the database with the largest number of matches was obtained and its corresponding model was output as the model of the test vehicle.

1.4 GUI Development using GUIDE

The main reason GUIs are used is because it makes things simple for the end-users of the program. If GUIs were not used, people would have to work from the command line interface, which can be extremely difficult and frustrating. GUIDE, the MATLAB Graphical User Interface development environment, provides a set of tools for creating GUIs. These tools greatly simplify the process of laying out and programminga GUI.

When you open a GUI in GUIDE, it is displayed in the Layout Editor, which is the control panel for all of the GUIDE tools. The Layout Editor enables you to lay out a GUI quickly and easily by dragging components, such as push buttons, pop-up menus, or axes, from the component palette into the layout area. The following picture shows the Layout Editor.

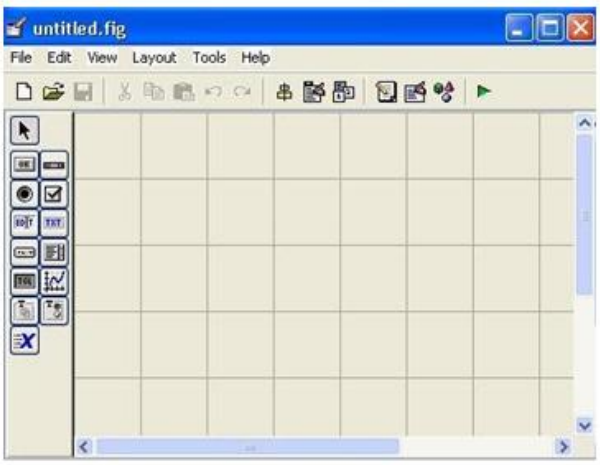

Fig 1.1: *GUIDE Layout editor*

Oncewelayoutyour GUI andseteachcomponent‟ sproperties, using the tools in the Layer. Layout Editor, we can program the GUI with the M- file Editor. Finally, when we press the Run button on the toolbar, the functioning GUI appears outside the Layout Editor Window. A GUI created during the project is as shown

Fig 1.2: *An example GUI developed for vehicle class identification*

Here, the layout consists of a text editor where we enter the path of the image to be loaded and then pressthe "Loadtheimage"button.Theimageinquestion would be loaded and displayed in the space provided.Wethenpushthe"GettheClass"button,which would perform the Eigen analysis of the image and display the class in the static textboxprovided.

II. Vehicle Class Identification Using Karhunen Loeve Transform And Svm

The problem of automatically learning object models for recognition is one of the present day challenges in image processing and computer vision. First we look at the methods that have been used to address this problem and hence provide a justification for our method. We have to devise a strategy to recognize 3D object using features such as object shape and illumination and not just geometrical features. [5] Hereit is being madeuseofthe "appearance".Also,our 3D system should have the ability to recognize the object from any possibleviewpoint.

2.1 Overview of Object Recognition Techniques One of the primary goals of an intelligent system is to recognize objects in an image and thereby compute their poses in the three dimensional scene. [5] For a vision system to be able to recognize objects, it must have models of objectsstoredin its memory. Such recognition has wide implications from visual inspection to autonomous navigation. In this age of the internet, it can also be used to recognize novel objects and thus incorporate it with the existing databases.

2.1.1 Approaches for Representation and Recognition ofvehicles

The main difficulty involved in the recognition of vehicles from their images is the loss of information between the vehicle and its 2D image and the viewpoint dependence of the image. The developing recognition scheme would involve establishing methods of recovering 3D information from 2D and defining measure or index which remains unchanged with respect to viewpoint. Therefore the issues of representation and features identification become crucial to the problem of recognition of vehicles. They assume importance because it is possible to maintain a database, which has every view of the vehicle. The various representation schemes to obtain the models of the vehicle give rise to different recognition frameworks. Broadly, every recognition scheme can be classified as either viewer centric or object centric. Object centric representation do not illustrate the dependency of a vehicle's appearance on viewpoint, therefore requires extensive image characterizing. It is widely used in computer graphics.

Viewer centered representations are used in vision and that is what we are concerned about. Here locations in the 3D space are specified with respect to an origin in the viewing position. The appearance of a vehicle is made more explicit by viewer centered representations and each of the vehicle is represented by a number of views. Some of the viewer centered representations which have been widely in use include aspect graphs, moment invariants [6] [7] [8] and appearance based methods [5]. In this project we are mainly concerned with the appearance based recognition because of its inherent advantages and its compatibility with the principal component analysis.

In appearance based approach, the recognition system is formulated as that of matching appearance rather than shape. Appearance of a vehicle in a 2D image depends on its shape, reflectance properties, and illumination conditions. For each model, a large set of images is obtained by automatically varying pose and

illumination, as other properties are intrinsic characteristics of the vehicle. This image set is compressed to obtain a low dimensional subspace called the eigenspace. Given an image of an unknown vehicle, the recognition system projects the image to the eigenspace. The object is recognized based on the manifold it lies upon.

2.1.2 Appearance based approach

Traditional image recognition techniques have emphasized the use of geometric models for recognition. However, appearance based approaches are more general in scope. The appearance of an object is a combined effect of its shape, reflectance properties, poses in the scene, and the illumination conditions. Recognizing objects from brightness images is therefore more a problem of appearance matching rather than shape matching. This observation is intrinsic to our work. While shape and reflectance are intrinsic properties that do not change for any rigid object, pose and illumination vary from scene to scene. Our approaches are one that of acquiringacompactmodeloftheobject'sappearance under different poses and illuminationconditions.

The aim is to encode appearance information of a set of 3D object in an optimal fashion. Image compression can be done in a number of ways. Most notable of them is the KL transform which is regarded as the most optimal of all energy compaction methods. But it has not been commercially used because of the dependence of the basis on the data. Other methods include Discrete Cosine Transform (DCT), Hadamard Transform etc. However for our application we have chosen the KL transform even though it is computationally intensive because it gives the least reconstruction error. That is tosay,ifwetakethe„n"mostsignificanttransformcoefficients in case of DCT or any other transform, andcompareitwiththatofKLT,the„n"issmallestincase of KLT. Hence the reason for using KL Transform. Also KL transformation is optimal for representing the features of a class ofobjects.

2.2 Mathematical tool for Encoding Appearance Information

Appearance data is a large data of image information. One has to extract the relevant information from this. In this section, we shall look at various tools that can be used to encode appearance information.

In our approach, the basis functions in the discrete Karhunen Loeve Transform [9] are obtained by solving the algebraic eigenvalue problem

$$\Delta=\Phi^T\Sigma\Phi \qquad (1.1)$$

where $\sum$ is the covariance matrix of the data, Φ is the eigenvector matrix of $\sum$ and Δ is the corresponding diagonal matrix of eigenvalues. The unitary matrix Φ defines a coordinate transform that results in making the data uncorrelated and makes explicit the invariant subspace of the matrix.

Most commonly it is being used KLT with reduced dimensions which identify the largest eigenvectors for projectingdata:

$$y = \Phi^T_M x \qquad (1.2)$$

Where Φ_M is a sub-matrix of Φ containing the principal eigenvectors. For recognition, the testimage is projected onto the most significant eigenvectors and the nearest neighbour is found using the Euclidean distancenorm.

2.1.1 Eigenvalues and Eigenvectors

Eigenvalues λ_K are defined as all roots of the

equation $|A-\lambda_K|=0$.Eigenvectors as all solutions of the equations $A\Phi_K$

$=\lambda_K\Phi_K$ *(1.3)*

such that $\Phi_K \neq 0$.

2.2.2 KL Transform

The KL transform [5] was originally introduced as a series expansion for continuous random process by Karhunen and Loeve. For random sequences the Hotelling transform first studied that was called the method of

principal components, which is the discrete equivalent of KL series expansion. For a real N x 1 random vector u, the basis vectors of the KL transform are given by the orthonormal eigenvectors of its autocorrelation matrix R, thatis

$$R\Phi_K = \lambda_K \Phi_K \qquad 0 \le k \le N-1 \qquad (1.4)$$

The KL transform of u is defined as $\Phi^{*T}u$ and the inverse transform is

$$u = \Phi v = \sum_{k=0}^{N-1} v(k)\Phi_k \qquad (1.5)$$

where Φ_k is the kth column of Φ. Also for a Hermitian matrix R, Φ reduces R to its diagonal form, thatis,

$$\Phi^{*T} R \Phi = \Lambda = Diag\{\lambda_1, \lambda_2, \ldots, \lambda_N\} \qquad (1.6)$$

We often work with the covariance matrix rather than the autocorrelation matrix. With μ = E (u), we have

$$R_o = \mathrm{cov}[u] = E[(u-\mu)(u-\mu)^T] = E[uu^T] - \mu\mu^T = R - \mu\mu^T \qquad (1.7)$$

If the vector μ is known, then the eigen matrix of Ro determines the KL transform of the zero mean random process u- μ. In general the KL transform of u and u- μ need not be identical. Note that the KL transform depends on the second order statistics of The Data

2.1 Recognition scheme using KL Transform

Object learning requires the acquisition of large image sets and the computationally intensive process of finding eigenvectors. However, learning is typically done offline and hence can afford to be slow. In contrast, recognition is often subject to severetimeconstraints. Hence, anefficient solution is required.

2.3.1 Constructing the Eigen space

While constructing the pattern matrix, we first normalize all the images to the same size and also subtract the background from the image, as we always have the background information, so that the decision is not biased. Each image is scanned column wise and is converted into a unique k x k column vector.

$$p = [p_{11}, p_{12}, \ldots, p_{1k}, p_{21}, \ldots, p_{kk}]^T \qquad (1.5)$$

The unnormalised pattern matrix is then generated as

$$P^* = \begin{pmatrix} p'^1_{11} & \cdots & p'^n_{11} \\ \vdots & \ddots & \vdots \\ p'^1_{kk} & \cdots & p'^n_{kk} \end{pmatrix}_{nxk} \qquad (1.6)$$

The unnormalised pattern matrix is then subjected to type I normalization with respect to mean wherein the mean vector is generated as

$$m_i = \sum_{k=1}^{n} p_{ij}^{k}$$

The resulting mean vector $\underline{m}$is then subtracted from the unnormalised pattern vectors to obtain the normalized pattern vectors and hence the normalized pattern matrix.

$$\underline{p^i} = \underline{p'^i} - \underline{m} \qquad (1.8)$$

$$P = \begin{pmatrix} p^1_{11} & \cdots & p^n_{11} \\ \vdots & \ddots & \vdots \\ p^1_{kk} & \cdots & p^n_{kk} \end{pmatrix}_{nxk} \qquad (1.9)$$

The covariance matrix is hence obtained as

$$A = \frac{1}{n} PP^T \qquad (1.10)$$

$$m_i = \sum_{k=1}^{n} p_{ij}^{k}$$

The resulting mean vector $\underline{m}$is then subtracted from the unnormalised pattern vectors to obtain the normalized pattern vectors and hence the normalized pattern matrix.

$$\underline{p^i} = \underline{p'^i} - \underline{m} \qquad (1.8)$$

$$P = \begin{pmatrix} p^1_{11} & \cdots & p^n_{11} \\ \vdots & \ddots & \vdots \\ p^1_{kk} & \cdots & p^n_{kk} \end{pmatrix}_{nxk} \qquad (1.9)$$

The covariance matrix is hence obtained as

$$A = \frac{1}{n} PP^T \qquad (1.10)$$

The covariance matrix is hence subjected to eigen decomposition to obtain the corresponding classification if the database we are concerned with was completely linear. Generally, this is not the case. Therefore, to avail the advantages ofgeneralized classification, a more robust method was usedknown as the Support Vector Machines [2] [3].

2.4.1 Introduction to Support Vector Machines

One of the fundamental problems in learning is that suppose we are given two classes of objects, a new object arrives and we have to assign it to one of the two classes. The problem can be mathematically stated asfollows:

$$(x_1,y_1)\ldots(x_n,y_n) \in \chi \qquad x\{1,-1\} \qquad (2.15)$$

Here $\chi$is some non-empty set from which the patterns $x_i$are taken and the corresponding $y_i$is called the output which is either 1 or -1 in this case. This type of pattern recognition is called binary pattern recognition. These patterns χcould be anything, could even be sheep which need to be classified. In our case, these patterns would be images.

eigenvalues and the eigenvectors, which are then arranged in non-decreasing order of eigenvalues.

$$A=U^{-1}\Lambda U \qquad (1.11)$$

where U is the eigenvector matrix and Λis the diagonal matrix of eigenvalues arranged in ascending order of magnitude. The eigenvectors are orthogonal to each other which are further orthonormalized. A set of L eigenvectors corresponding to themost significant eigenvalues are then taken to be the basis of the eigenspace and all the database and test images are projected onto this space to yield a set of projection coefficients which are then used for comparison.

2.4 Support VectorMachines

Correlation between the feature vectors generated using KL Transform would have sufficed for

thatis,afunctionthat,giventwopatternsxandx',returns a real number characterizing their similarity. Aparticularsimilaritymeasureofthissortthatisof particular mathematical appeal is the *dot product.*

$$\langle x, x' \rangle := \sum_{i=1}^{n} [x]_i [x]'_i \qquad (2.17)$$

Here $[x]_i$ denotes the ith entry of x. The dot product has a notion of length of the vector.

In order to use the dot product as a similarity measure, we need to represent the patterns as vectors in some dot product space H . To this end, we use a map

$$\Phi: \chi \to H$$
$$x \to \Phi(x) \qquad (2.18)$$

However our choice of Φ is not restricted to the dot product space. Some of the input sequences may need

In order to study the problem of learning, however, we need an additional type of structure. In learning, we want to be able to generalize to unseen data points. In the case of pattern recognition, this means that given some new pattern x, we want to predict the corresponding y. By this we mean, loosely speaking, that we choose y such that (x, y) is in some sense similar to the training examples (3.1). To this end, we need notions of *similarity* in$\chi$and in$\{1, -1\}$. Let us considera similarity of the form

$$k : \chi x \chi \to R$$
$$(x, x') \to k(x, x') \qquad (2.16)$$

exists an optimal hyperplane, distinguished by the maximum margin of separation between any training point and the hyperplane. It is the solution of

$$\max_{w \in H, b \in R} \min \{ \sqrt{\langle x - x_i, x - x_i \rangle} \mid x \in H, \langle w, x \rangle + b = 0, i = 1 \ldots\ldots\ldots m \} \qquad (2.20)$$

The optimal hyperplane is as shown in the corresponding figure

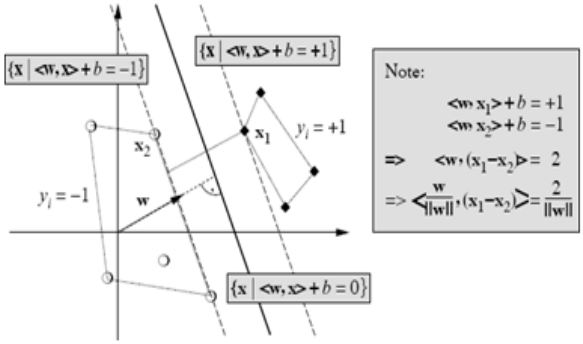

Fig 2.1: *Optimal Hyperplane for a binary classification*

a more robust similarity measure, in which case, there is a possibility of the map Φ being non-linear too. This is particularly true in case of images.

2.4.2 Choosing ahyperplane

Once we have mapped the data onto the dot product space, we formulate the similarity features between the input sequences and construct a hyperplane which separates the data into a set of classes. A hyperplane in a dot product space is defined as

$$\langle w,x \rangle + b = 0 \quad \text{where}$$

$$w \in \mathrm{H} \text{ and } b \in R \qquad (2.19)$$

corresponding to decision function $f(x) = \mathrm{sgn}(\langle w,x \rangle + b)$ for a binary case. This method of classification is based on the fact that among all hyperplanes separating the data, there

$$\frac{\delta}{\delta b} L(w, b, \alpha) = 0$$
$$\frac{\delta}{\delta w} L(w, b, \alpha) = 0 \qquad (2.23)$$

leading to

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$
$$and \qquad (2.24)$$
$$w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

The solution vector thus has the expansion in terms of a subset of training vectors with a non-zero α_i. All such vectors which correspond to the solution (2.24) are called *Support Vectors*. These support vectors lie on the margin of thehyperplane.

A further extension of the above theory involves modification of the above problem for non-separable data sets in which case the constraint equation is made flexible by introducing

In order to construct the hyperplane, we need to solve

$$\underset{w \in H, b \in R}{minimize} \tau(w) = \frac{1}{2}\langle w, w \rangle \qquad (2.21)$$

subjectto $y_i(\langle w, x_i \rangle + b) > 1 \text{ forall} i = 1 \ldots\ldots n$

A solution to the problem involves theuseofLagrange'smultipliers.Theproblem canbe defined asfollows

$$L(w,b,\alpha) = \frac{1}{2}\langle w, w \rangle - \sum_{i=1}^{m} \alpha_i (y_i(\langle w, x_i \rangle + b) - 1) \qquad (2.22)$$

The Lagrange L has to be minimized with respect to the primal variables w and b and maximized with respect to the dual variables α_i. It means the derivative of L with respect to w and b must vanish. Mathematically

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{m} \alpha_i y_i k(x, x_i) + b) \qquad (2.27)$$

2.5 Procedure forClassification

2.5.1 Algorithm forClassification

a) A set of 4 eigenspaces were constructed corresponding to 203 small cars, 68 large cars, 41 bikes and 25 scootersrespectively.

b) Classification was done on a binarybasis, i.e. test images were divided into two classes corresponding to four wheelers andtwo wheelers, the four wheelers correspondingto small and big cars and two wheelers corresponding to bikes and scooters.

c) The test images corresponding to four wheelers were projected on the eigenspaces corresponding to small cars and big cars and the feature vectors were stored. Similarly, the feature vectors for test images corresponding to two wheelers were

slack variables, that is.

$$y_i(\langle w, x_i \rangle + b) > 1 - \xi_i \qquad (2.25)$$

and hence minimizing the function

$$\underset{w \in H, b \in R}{\text{minimise}} \tau(w) = \frac{1}{2}\langle w, w \rangle + C\sum_{i=1}^{m} \xi_i \qquad (2.26)$$

where C>0 determines the trade off between margin maximization and training error minimization. In this case the optimizing hyperplane is found out to be non-linear, so a kernel trick is made so as to provide a pseudo mapping of the data sets onto a higher dimensional space wherein the separating hyperplane becomes linear. This is done by incorporating a kernel *k(x, xi)*, due to which the decision functions changeto

calculated by projecting them onto the eigenspaces constructed using bikes and scooters.

d) The feature vector of each testimage was compared with the feature vectors ofeach image in the small car database by means of Euclidean distance and the minimum distance was noted. Similar procedure was followed in the case of large cars and another minimum distance was noted. The two distances were compared and the vehicle was assigned a class which corresponded to the smaller distance of the two.

a) For classification using SVM, a combined pattern matrix consisting of the feature vectors corresponding to the database images of small cars and large cars was constructed and was used to train the SVM. The feature vectors corresponding tosmall
cars were labeled as belonging to class 1 and those corresponding to large cars were labeled as belonging to class 2.

b) A matrix consisting of feature vectors of 165 test images with arbitrary labels was used for classification using the trained SVM. Initially the labels were provided as per their class for verification purposes i.e. if the test imagewasalargecar,itwaslabeled2and
hence.

2.1.2 GUI for class identification andResults

A total of 165 images corresponding to small and large cars were used as test images. When classified using just the eigenspaces, 129 images were correctly identified as belonging to the class of small cars or large cars, which corresponded to a recognition rate of 78.18%. Two sets of GUI were developed, one for classifying four-wheelers, the other one for classifying two wheelers. The results with the corresponding GUI for the four different classes we have considered are as shownbelow:

Fig 2.2: GUI for four wheeler identification

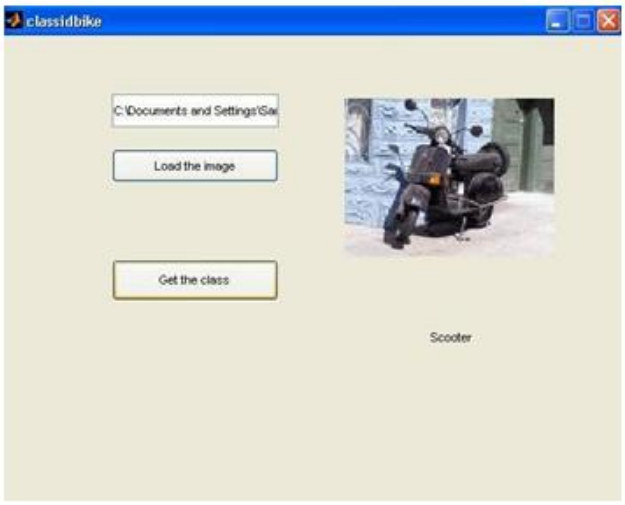

Fig 2.3: GUI for two wheeler identification

For the sake of applying SVM onto our problem, we have used **LibSVM**[1]**,** which are a set of libraries for Support Vector Machines implemented by Chih-Chung Chang and Chih-Jen Lin. **LibSVM**is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports multi-class classification. It basically supports four types of kernels: polynomial: (gamma*u'*v + coef0) ^degree, linear: u'*v, radial basis function: exp (-gamma*|u- v|^2), sigmoid: tanh (gamma*u'*v + coef0).

The procedure for training has been outlined in the documentation of **LibSVM**. First we scale both the training data and the test data in the required range, which provides the best classification, using the svm-scale functon and then train the resulting scaled matrix by means of svm-train with a suitable kernel and the kernel parameters. The commands for scaling and training and eventually predicting the class of the test images are as follows: svm-scale -l -1 -u 1 -s range train >train.scale (2.28) svm-scale -r range test>test.scale (2.29)

This scales the training data first in the range -1 to 1 andsavesthetrainingdetailinafile„range",whichis subsequently used to scale the test data.
svm-train -s 1 -t 1 -n 0.1 –r 0.1 -g 0.1 -e 0.02
train.scale (2.30)
-s selects the particular SVM, which is µ-SVM here, - n chooses the µ value, -g chooses the gamma value and –e chooses the epsilon value, -r corresponds to thevalueofthecoefficient.–tchoosesthekernel.„–t"can take values of 0,1,2,3 or 4, 0 corresponding to linear kernel, 1 to polynomial, 2 to radial basis and 3 to sigmoid kernel. The option 4 is for using a customized kernel. Once we have trained the SVM we apply the test.scale file to the SVM and classify the data using svm-predict. svm-predict test.scale train.scale.modeltest.predict (2.31) The test predict file holds the results after prediction. In our classification mechanism, we first scaled the training and the test data in the range of -5 to 5 and applied the SVM with kernel and kernel parameters as specified in eqn. (2.30). We could classify 140 out of 164 vehicles correctly as per their respective class resulting in an improved recognition rate of85.36%.

Apart from LibSVM, the Support Vector Machine toolbox for Matlab developed by Anton Schwaighofer was also used for our classification. This toolbox optionally makes use of a Matlab wrapper for an interior point code in LOQO style (Matlab wrapper by Steve Gunn, LOQO code by Alex Smola). Optimum classification was obtained in this case by the use of the RBF kernel with gamma of 0.5 and a C of 100. The results are as shown:

```
To get started, select MATLAB Help or Demos from the Help menu.

>> carid
Loading code from file code14-11
Starting the training.
This may take a while, since we need to train one SVM per bit ...
Training for code bit 1 (out of 14)
Training for code bit 2 (out of 14)
Training for code bit 3 (out of 14)
Training for code bit 4 (out of 14)
Training for code bit 5 (out of 14)
Training for code bit 6 (out of 14)
Training for code bit 7 (out of 14)
Training for code bit 8 (out of 14)
Training for code bit 9 (out of 14)
Training for code bit 10 (out of 14)
Training for code bit 11 (out of 14)
Training for code bit 12 (out of 14)
Training for code bit 13 (out of 14)
Training for code bit 14 (out of 14)
Classification result on the test set: 139 out of 165 correct
```

2.1 Conclusion

An algorithm involving the combination of Principal Component analysis and Support Vector machines was successfully developed for identification of the class of the vehicle. By the suitable choice of kernel and kernel parameters an identification rate of 85.34% was obtained using LibSVM and an identification rate of 85% was obtained using the Support Vector Machine toolbox for Matlab developed by Anton Schwaighofer. Once the class is identified, the next step can be identifying the vehicle"scolorandmostimportantly, its model.

2.7. Scope for future work

The classification of vehicles according to their class could be further extended to include six-wheelers like buses, trucks etc. The present algorithm could further be improved by using more robust feature extraction techniques like Multidimensional scaling, Independent Component Analysis etc.

PCA-SIFT combined with Support vector machines could provide an extremely robust pattern matching tool, but the only problem here is the excessive time required to organize the SIFT data in the format applicable to SVM. A possible solution would be to find a relation between the key points which would reduce their dimensionality while still keeping their properties intact.

Statistical methods could be developed for vehicle detection and model identification. Since the statistics of an image are unique, this factor could be used to generate feature vectors for each image which could distinctly represent the vehicle.

BIBLIOGRAPHY

[1]. Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available athttp://www.csie.ntu.edu.tw/~cjlin/libsvm

[2]. Bernhard Schölkopf and Alex Smola, "Learning With Kernels," MIT Press, Cambridge, MA,2002

[3]. R.-E. Fan, P.-H. Chen and C.-J. Lin, "Working set selection using the second order information for training SVM," Journal of Machine Learning Research 6, 1889-1918,2005.

[4]. SudeepNema, "An automatic system for extraction of different features of a moving vehicle", M.Tech dissertation, December2000.

[5]. Hiroshi Murase, Shree K. Nayar, "Visual learning and recognition of 3-D objects from appearance,"International Journal of Computer Vision, v.14 n.1, p.5-24, Jan.1995

[6]. Ming-Kuei Hu, "Pattern Recognition by Moment Invariants," PROC. IRE (Correspondence), vol 49, p. 1428, September1961

[7]. Flusser J and Suk T, "Pattern Recognition by Affine Moment Invariants," Pattern Recognition, 1993, Vol. 26, pp.167-174

[8]. Feng Pan and Mike Keane, "A new set of moment invariants for handwritten numeral recognition," International Conference on Image Processing (1), 1994, pp.154-158

[9]. David G. Lowe, "Distinctive Image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp.91-110

[10]. R Suthankar, Yan Ke, "PCA-SIFT: A more distinctive representation for local image descriptors," Computer Vision and Pattern Recognition, Vol.2 (2004), pp.II-506-II-513

[11]. Zehang Sun, George Bebis and Ronald Miller, "On-Road Vehicle Detecion Using Evolutionary Gabor Filter Optimization," IEEE Transactions on Intelligent Transportation Systems, Vol. 6, No.2, June 2005

[12]. Zehang Sun, George Bebis and Ronald Miller, "Quantized wavelet features and support vector machines for on-road vehicle detection," 7th Int. Conf. Control, Automation, Robotics Vision,Singapore, Dec. 2002, pp. 1641-1646

[13]. M. Bertozzi and A. Broggi, "Real-time lane and obstacle detection on the gold system," IEEE Intelligent Vehicle Symposium, pp. 213–218,1996

[14]. G. Zhao and Y. Shini‟chi, "Obstacle detection by vision system for autonomous vehicle," IEEE Intelligent Vehicle Symposium, pp. 31–36,1993

[15]. N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle detection and recognition in greyscale imagery," Control Engineering Practice, vol. 4, pp. 473–479,1996

[16]. C. Goerick, N. Detlev and M. Werner, "Artif̧cial neural networks in real-time car detection and tracking applications," Pattern Recognition Letters, vol. 17, pp. 335–343,1996

[17]. M. Betke, E. Haritaglu and L. Davis, "Multiple vehicle detection and tracking in hard real time," IEEE Intelligent Vehicles Symposium, pp. 351–356, 1996

[18]. H. Schneiderman and T. Kanade, "Probabilistic modeling of local appearance and spatial relationships for object recognition,"IEEE 2001 International Conference on Computer Vision and Pattern Recognition, pp. 45–51, 1998

[19]. C. Papageorgiou and T. Poggio, "A trainable system for object detection," International Journal of Computer Vision, vol. 38, no. 1, pp. 15–33,2000