CHAPTER-1
INTRODUCTION

1.1 Overview

Due to the rise in the popularity of automobiles over the last century, road accidents have become cause of fatalities. Road safety was even recognized as an area of significant interest in Global Environmental Policy deliberations at the Rio+20 UN Conference on Sustainable Development on June 20-22, 2012. A clear link was made between Road Safety and Sustainable Development as well.

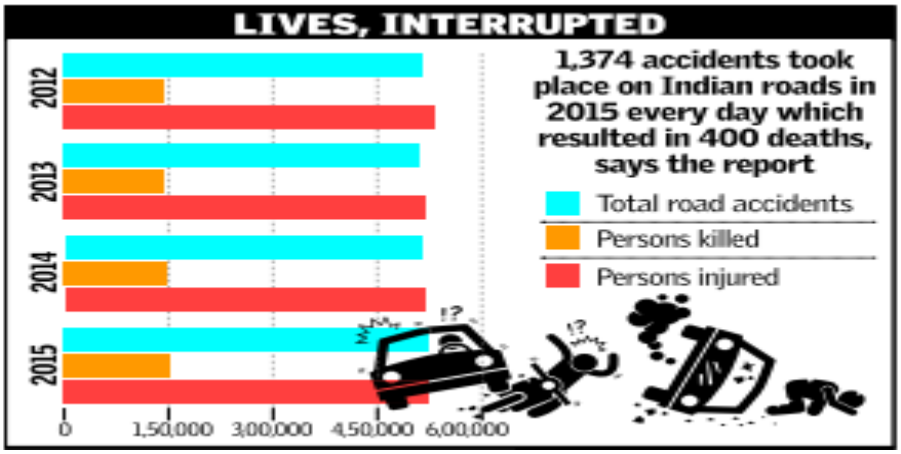
In 2010, the governments of the world declared 2011–2020 as the "Decade of Action for Road Safety". They invited the World Health Organization to prepare a report as a baseline to assess the state of global road safety at the onset of the decade, and to be able to monitor progress over the period of the decade. The unanimous support for this "Decade of Action for Road Safety" from member states indicated a growing awareness that the devastating scale of road traffic injuries is a global public health and development concern.

The Global Status Report on road safety 2013 shows that 1.24 million people were killed on the world's roads in 2013. This is unacceptably high. Road traffic injuries take an enormous toll on individuals and communities as well as on national economies. Middle-income countries, which are motorizing rapidly, are the hardest hit.

Road traffic injuries are the eighth leading cause of death globally, and the leading cause of death for young people aged 15–29. More than a million people die each year on the world's roads, and the cost of dealing with the consequences of these road traffic crashes runs to billions of dollars. Current trends suggest that by 2030 road traffic deaths will become the fifth leading cause of death unless urgent action is taken.

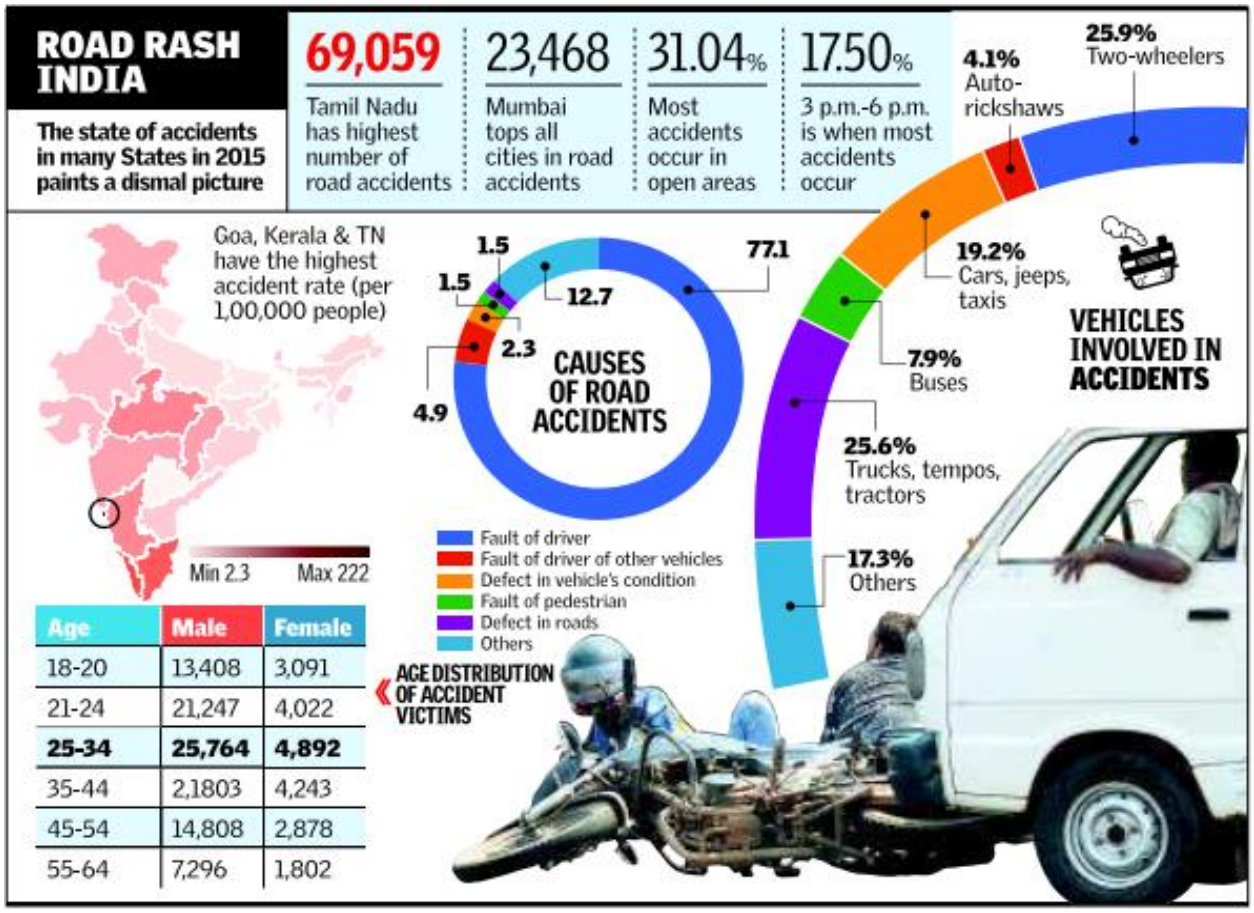
Data on road accidents in India are collected by Transport Research Wing (TRW) of Ministry of Road Transport & Highways, in terms of 19 items with the format devised under the Asia Pacific Road Accident data (APRAD)/Indian Road Accident Data (IRAD) project- of United Nations Economic and Social Commission for the Asia and the Pacific (UN-ESCAP), and are obtained from Police Departments of States/Union Territories in India.

In India, 16 people died every hour in 2014, approx 385 road accident deaths per day and over 1.41 lakh deaths during the year. The maximum number of fatalities was reported on the roads in Delhi with 2,199 deaths during the year as per National Crime Records Bureau.



An official report released by Union Road Transport and Highways Minister Shri Nitin Gadkari on 10 June' 2016, said 1.46 lakh people were killed in road accidents in India in 2015- an increase of five per cent from 2014.

As per the report, road accidents as a whole rose 2.5 per cent during 2015 to 5.01 lakh or 1,374 accidents every day, claiming 400 lives. The report said a majority (54.1 per cent) of those killed in road accidents during 2015 were in the age group of 15-34. Thirteen States, including Tamil Nadu, Maharashtra, Madhya Pradesh, Karnataka, Kerala and Uttar Pradesh, accounted for the highest number of accidents. Among cities, while Mumbai had the highest number of accidents (23,468), Delhi saw the most number of deaths (1,622) in road accidents. Also, drivers' fault was responsible for 77.1 per cent of the accidents, deaths and injuries, mainly because of over-speeding, the report noted. The data also reveals more than half of those killed in the productive age group of 15 to 34, pointing to calamitous loss of young lives and significantly adversely affecting Indian economy as well.



Source: Ministry of Road Transport & Highway, Govt of India

Very recently (10th June, 2016) the Ministry of Road Transport & Highway, Govt of India has decided to form the "National Road Safety and Traffic Management Board" through an executive order after it failed to push the Road Safety Bill owing to the logjam in Parliament reacting to the alarming condition on road safety coming out in public domain. The National Road Safety and Traffic Management Board will be an advisory body mandated to advise on rules and regulations, road safety and road engineering. The body is likely to be chaired by a former Road Transport Secretary. Also it is emphasised that the need of hour is to have a scientific accident investigating agency that looks at composite factors including drivers' fault, bad road design and failure of civic agencies to maintain infrastructure while fixing the responsibility for accidents.

Both the scientific community and the automobile industry have contributed to the development of different types of protection systems in order to improve traffic safety. Initially, improvements consisted of simple mechanisms like seat belts, but then more complex devices, such as antilock braking systems, electronic stabilization programs, and airbags, were developed.

Over the last decade, research has moved towards more intelligent on-board systems that aim to anticipate accidents in order to avoid them or to mitigate their severity. These systems are referred to as- Advanced Driver Assistance Systems (ADASs), as they assist the driver in marking decisions, provide signals in possibly dangerous driving situations, and execute counteractive measures. A particular type of ADAS is the Pedestrian Detection Systems (PDSs). Therefore, a perfect on-board Pedestrian Detection System, referred to as PDS, must detect the presence of people, stationary or moving, on the way of the vehicle and react according to the risk of running over the pedestrian. The action performed by PDS in case a pedestrian appears right before the moving vehicle and is likely to get harmed includes: warning the driver in advance, or apply braking action, or deploy external airbags, perform an evasive manoeuvre or else. It is also necessary that the entire system works well without disturbing the driver needlessly in normal situations, if there is no risk at all. Moreover, such a system should work well independent of the time, road, and weather condition. Additionally, the cost of the pedestrian detection module should be relatively small compared to the total cost of the vehicle.

Accident statistics indicate that 70 percent of the people involved in car-to-pedestrian accidents were in front of the vehicle, of which 90 percent were moving. Pedestrian detection before the impact (either long or short term) is crucial given that the severity of injuries for the pedestrian decreases with speed of the crashing vehicle. Thus, any reduction in the speed can drastically reduce the severity of the crash. According to a research conducted by "Society of automotive engineers", pedestrians have a 90% chance of surviving to car crashes at 30 km/h or below, but less than 50% chance of surviving to impacts at 45 km/h or above.

After reading a range of literature related to pedestrian detection systems and other ADAS related application, it is found that the primary requirement is that of a robust feature set that allows the human form to be discriminated cleanly, even in cluttered backgrounds under difficult illumination.

It is clear that the topic differs from general human detection systems, such as surveillance applications or human-machine interfaces, for which some simplifications can be implemented. It is also found that locally normalized Histogram of Oriented Gradient (HOG) descriptors provide excellent performance relative to other existing feature sets including wavelets. The HOG descriptors are reminiscent of edge orientation histograms, Scale Invariant Feature Transform (SIFT) descriptors and shape contexts, but they are computed on a dense grid of uniformly spaced cells and they use overlapping local contrast normalizations for improved performance.

A further improvement Relative Discriminative Histogram of Oriented Gradients (RDHOG) based on the HOG descriptor is used to represent the difference between the central block and the surrounding blocks of the target. After combining the features of HOG and RDHOG, the descriptor could describe the shape of the target and discriminate between the centre and borders. Thus, the weighting contributions of the features can be improved by combining the conventional HOG features and RDHOG features which enhances the descriptive ability of the central block and the surrounding blocks.

1.2 Objective of a Pedestrian Detection System using Moving Camera:

The main objective of Pedestrian Detection System (PDS) is to detect the presence of both stationary and moving people in a specific area of interest around the moving host vehicle -in order to warn the driver in advance, and perform braking actions, perform an evasive manoeuvre or else, if a collision is likely to occur. The topic differs from general human detection systems, such as surveillance applications or human-machine interfaces, because in such cases use of a static camera allows the use of background subtraction techniques. Therefore, "Moving camera" here means, it is mounted on any moving object so background keep changing.

As for this specific PDS application, pedestrians must be identified in highly dynamic scene that too with a camera mounted on a vehicle in motion, which further complicates tracking and movement analysis. Since both the pedestrian and camera are in motion in this case, it requires a more efficient method to cope up with the challenges offered in such dynamic cases and correctly identify the pedestrians from a moving vehicle.

1.3 Inspiration for this Project

• Road traffic injuries are the eighth leading cause of death globally, and the leading cause of death for young people aged 15–29.

• The Global status report on road safety 2013 shows that 1.24 million people were killed on the world's roads in 2013.

• In India, 16 people died every hour in 2014, approx 385 road accident deaths per day and over 1.41 lakh deaths during the year.

•The maximum number of fatalities was reported on the roads in Delhi with 2,199 deaths during the year as per National Crime Records Bureau.

• Having studied and learnt concepts behind Artificial Neural Network, Suppost Vector Machine (SVM) classifiers, the working of HOG and other object detection and tracking related techniques; the application of these concepts in the problems of practical nature being confronted every now and then.

1.4 Challenges associated with Pedestrian Detection System:

Advanced Driver Assistance Systems (ADASs), and particularly Pedestrian Detection Systems (PDSs), have become an active research area aimed at improving traffic safety. Due to the varying appearance of pedestrians (e.g., different clothes, changing size, aspect ratio, and dynamic shape) and the unstructured environment, it is very difficult to cope with the demanded robustness of this kind of system. The major challenge of PPSs is the development of reliable on-board pedestrian detection systems.

Generally speaking, a Visual Processing System needs to function well under a wide range of visibility conditions covering over-cast sky, strong highlights, low visibility due to inclement weather, wide dynamic range of imaging conditions, change of context, day-time and night-time driving. On top of that, the class of pedestrians is particularly challenging for a number of reasons:

• The image space variability of the class is very large as pedestrians appear in various poses, clothing and various articulations of body parts. The articulation of body parts also makes the process of tracking a pedestrian along an image sequence somewhat challenging.

• Pedestrians are found mostly in city traffic conditions where the background texture (from surrounding manmade structures, other vehicles poles and trees) form a highly cluttered environment.

• The background clutter covers both shape (texture) and depth. If in an open roadway a pedestrian would stand out using depth disparity cues (such as by using stereopsis), depth cues are unlikely to be useful for segmenting out pedestrians in city traffic due to the heavy disparity clutter.

• Pedestrians occupy a narrow image strip and from a distance may look similar to many background objects such as trees, poles, parts of parked vehicles, narrow windows and openings, and so forth.

• Laterally moving pedestrians form an important subclass for which motion measurements form a powerful cue. However, parts of moving vehicles (in slow traffic) also generate inward motion signals and motion-based segmentation from a moving platform is still a difficult problem especially in an environment rich with other moving structures.

1.5 Histogram of Oriented Gradients (HOG)

The Histogram of Oriented Gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image.

The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions.

The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient “directions” is compiled. The descriptor is then the concatenation of these histograms.

For improved accuracy, the local histograms can be contrast normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions. Moreover, as Dalal and Triggs (1) discovered, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization permits the individual body movement of pedestrians to be ignored so long as they maintain a roughly upright position. The HOG descriptor is thus particularly suited for human detection in images.

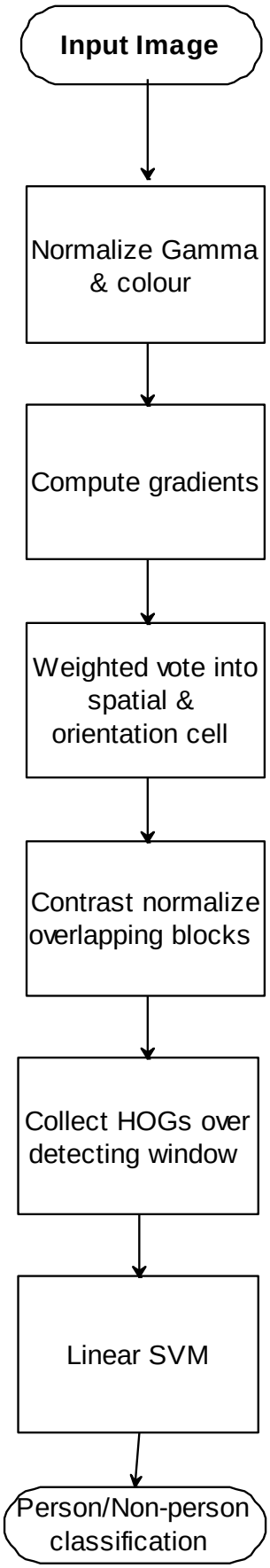


Fig. 1.1: An overview of HOG feature extraction and object detection chain.

1.5.1 HOG Steps:

(i). Gradient Computation:

The first step of calculation in many feature detectors in image pre-processing is to ensure normalized colour and gamma values. However, this step can be omitted in HOG descriptor computation, as the ensuing descriptor normalization essentially achieves the same result. Image pre-processing thus provides little impact on performance.

Instead, the first step of calculation is the computation of the gradient values. The most common method is to apply the 1-D centred, point discrete derivative mask in one or both of the horizontal and vertical directions. Specifically, this method requires filtering the colour or intensity data of the image with the following filter kernels:

Dalal and Triggs (2) tested other, more complex masks, such as the 3x3 Sobel mask or diagonal masks, but these masks generally performed poorer in detecting humans in images. They also experimented with Gaussian smoothing before applying the derivative mask, but similarly found that omission of any smoothing performed better in practice.

To find the horizontal and vertical gradients, convolve the image with kernels [1 0 1] and $[-1\ 0\ 1]^T$.

(ii). Orientation Binning:

The second step of calculation is creating the cell histograms. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. The cells themselves can either be rectangular or radial in shape, and the histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is "unsigned" or "signed". In (1) it was found that unsigned gradients used in conjunction with 9 histogram channels performed best in their human detection experiments. As for the vote weight, pixel contribution can either be the gradient magnitude itself, or some function of the magnitude. In tests, the gradient magnitude itself generally produces the best results. Other options for the vote weight could include the square root or square of the gradient magnitude, or some clipped version of the magnitude. Therefore, in a nutshell after computing gradients find out the orientations for each pixel

gradients and then do binning for these into 9 classes from 0° to 180° with an interval of 20°. Number of pixels in a particular class constitutes the weight for that particular bin

(iii). Descriptor blocks:

To account for changes in illumination and contrast, the gradient strengths must be locally normalized, which requires grouping the cells together into larger, spatially connected blocks. The HOG descriptor is then the concatenated vector of the components of the normalized cell histograms from all of the block regions. These blocks typically overlap, meaning that each cell contributes more than once to the final descriptor. Two main block geometries exist: rectangular R-HOG blocks and circular C-HOG blocks.

R-HOG blocks are generally square grids, represented by three parameters: the number of cells per block, the number of pixels per cell, and the number of channels per cell histogram. In (1) for the human detection experiment, the optimal parameters were found to be 8x8 cell blocks of 16x16 pixel cells with 9 histogram channels. Moreover, they found that some minor improvement in performance could be gained by applying a Gaussian spatial window within each block before tabulating histogram votes in order to weight pixels around the edge of the blocks less. The R-HOG blocks appear quite similar to the Scale Invariant Feature Transform (SIFT) descriptors; however, despite their similar formation, R-HOG blocks are computed in dense grids at some single scale without orientation alignment, whereas SIFT descriptors are usually computed at sparse, scale-invariant key image points and are rotatedS to align orientation. In addition, the R-HOG blocks are used in conjunction to encode spatial form information, while SIFT descriptors are used singly.

Circular HOG blocks (C-HOG) can be found in two variants: those with a single, central cell and those with an angularly divided central cell. In addition, these C-HOG blocks can be described with four parameters: the number of angular and radial bins, the radius of the centre bin, and the expansion factor for the radius of additional radial bins. It was found that the two main variants provided equal performance, and that two radial bins with four angular bins, a centre radius of 4 pixels, and an expansion factor of 2 provided the best performance in their experimentation. Also, Gaussian weighting provided no benefit when used in conjunction with the C-HOG blocks. C-HOG blocks appear similar to shape context descriptors, but differ strongly in that C-HOG blocks contain cells with several orientation channels, while shape contexts only make use of a single edge presence count in their formulation.

(iv). Block Normalization:

In last step block normalization need to be done. There are four different methods for block normalization: L2-norm, L2-hys, L1-norm and L1-sqrt.

$$L2-norm.,\ f = \frac{v}{\sqrt{\|v\|^2 + e^2}}$$

L2-hys: L2-norm followed by clipping (limiting the maximum values of v to 0.2) and renormalizing it.

$$L1-norm: \frac{v}{(\|v\|_1 + e)}$$

$$L1-sqrt:\ f = \sqrt{\frac{v}{\|v\|_1 + e}}$$

(v). Support Vector Machine (SVM) Classifier

The final step in object recognition using histogram of oriented Gradient descriptors is to feed the descriptors into some recognition system based on supervised learning. The Support Vector Machine (SVM) Classifier is a binary classifier which looks for an optimal hyperplane as a decision function. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. Once trained on images containing some particular object, the SVM classifier can make decisions regarding the presence of an object, such as a human, in additional test images.

1.5.2 Why HOG preferred over SIFT?

SIFT is typically used for matching local regions in two images for purposes of alignment / reconstruction / structure from motion, though it has been used for recognition as well .SIFT computes the gradient histogram only for patches) around specific interest points obtained by taking the DoG's (Difference of Gaussians) in the scale space.

Whereas, HOG is typically used in a Sliding Window fashion in object detection systems. HOG is computed for an entire image by dividing the image into smaller cells and summing up the gradients over every pixel within each cell in an image.

In short, HOG and SIFT are quite having similar gradient orientation, histogram computation etc techniques, difference being HOG as more global compared to SIFT.

1.6 Artificial Neural Network

Artificial neural network (ANN) is a mathematical model for predicting system performance (i.e., system output) inspired by the structure and function of human biological neural networks. The ANN is developed and derived to have a function similar to the human brain by memorizing and learning various tasks and behaving accordingly. It is trained to predict specific behaviour and to remember that behaviour in the future like the human brain does. Its architecture also is similar to human neuron layers in the brain as far as functionality and inter-neuron connection.

In a ANN, the operations are organized into a multilayered feed-forward network with four layers: Input layer, Hidden layer, Pattern layer/Summation layer, Output layer.

When an input is present, the first layer computes the distance from the input vector to the training input vectors. This produces a vector where its elements indicate how close the input is to the training input. Inside the neurons, the main mathematical calculations occur to process the inputs and provide the proper outputs. Like a biological neuron in the real brain, the neuron in the hidden layer receives and sends a line of values from the previous layer and to the next layer, respectively. The received and sent values to and from the neuron differ depending on the weight value of the channel (i.e., line) that carries the value to and from the

neuron. The weight of the channel means the value that is multiplied by the carried value (i.e., multiplying the weights value by the coming value from the previous neuron) before passing the result to the next neuron. The weight value changes by changing the intended task to perform; its value is decided by learning and memorizing doing that task.

The second layer sums the contribution for each class of inputs and produces its net output as a vector of probabilities. Finally, a complete transfer function on the output of the second layer picks the maximum of these probabilities, and produces a 1 (positive identification) for that class and a 0 (negative identification) for non-targeted classes.

Input layer

Each neuron in the input layer represents a predictor variable. In categorical variables, N-1 neurons are used when there are N number of categories. It standardizes the range of the values by subtracting the median and dividing by the inter-quartile range. Then the input neurons feed the values to each of the neurons in the hidden layer.

Pattern layer

This layer contains one neuron for each case in the training data set. It stores the values of the predictor variables for the case along with the target value. A hidden neuron computes the Euclidean distance of the test case from the neuron's centre point and then applies the RBF kernel function using the sigma values.

Summation layer

For PNN networks there is one pattern neuron for each category of the target variable. The actual target category of each training case is stored with each hidden neuron; the weighted value coming out of a hidden neuron is fed only to the pattern neuron that corresponds to the hidden neuron's category. The pattern neurons add the values for the class they represent.

Output layer

The output layer compares the weighted votes for each target category accumulated in the pattern layer and uses the largest vote to predict the target category.

A Perceptron takes several binary inputs, and produces a single binary output:

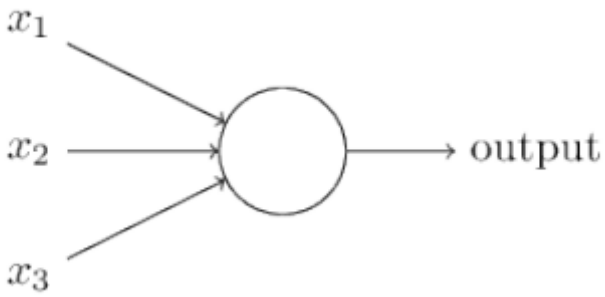


Fig. 1.2: A general Perceptron representation.

In the example shown the Perceptron has three inputs, x_1, x_2, x_3. In general it could have more or fewer inputs. It includes a simple rule proposed to compute the output. For this, weights, w_1, w_2, w_3…, real numbers expressing the importance of the respective inputs to the output are introduced. The neuron's output, or, is determined by whether the weighted sum is less than or greater than some threshold value. Just like the weights, the threshold is a real number which is a parameter of the neuron. To put it in more precise algebraic terms:

$$output = \begin{cases} 0.....if \sum_j w_j x_j \leq threshold \\ 1.....if \sum_j wjxj > threshold \end{cases}$$

1.6.1 The McCulloch-Pitts Neuron

This vastly simplified model of real neurons is also known as a Threshold Logic Unit. It has following components:

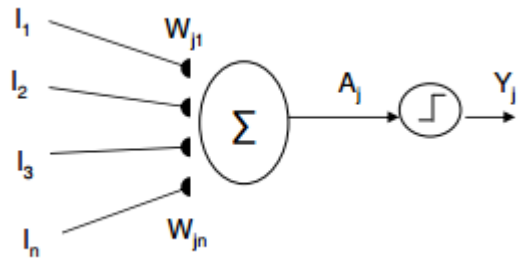


Fig. 1.3: Simplified model of neuron

1. A set of synapses (i.e. connections) brings in activations from other neurons.

2. A Processing unit sums the inputs, and then applies a non-linear activation function.

3. An Output line transmits the result to other neurons.

How the Model Neuron Works:

• Each input I_i is multiplied by a weight w_{ji} (synaptic strength)

• These weighted inputs are summed to give the activation level, A_j

• The activation level is then transformed by an activation function to produce the neuron's output, Y_i

• W_{ji} is known as the weight from unit i to unit j

– $W_{ji} > 0$, synapse is excitatory

– $W_{ji} < 0$, synapse is inhibitory

• Note that I_i may be

– External input

– The output of some other neuron

The McCulloch-Pitts Neuron Equation

We can now write down the equation for the output Y_j of a McCulloch-Pitts neuron as a function of its inputs Ii:

$Y_j = \text{sgn}(\sum I_i - \theta)$; where θ is the neuron's activation threshold.

When $Y_j = 1$, if $\sum I_k \geq \theta$, $Yj = 0$, if $\sum I_k < \theta$

In this way, the McCulloch-Pitts neuron is an extremely simplified model of real biological neurons. Nevertheless, they are computationally very powerful. One can show that assemblies of such neurons are capable of universal computation.

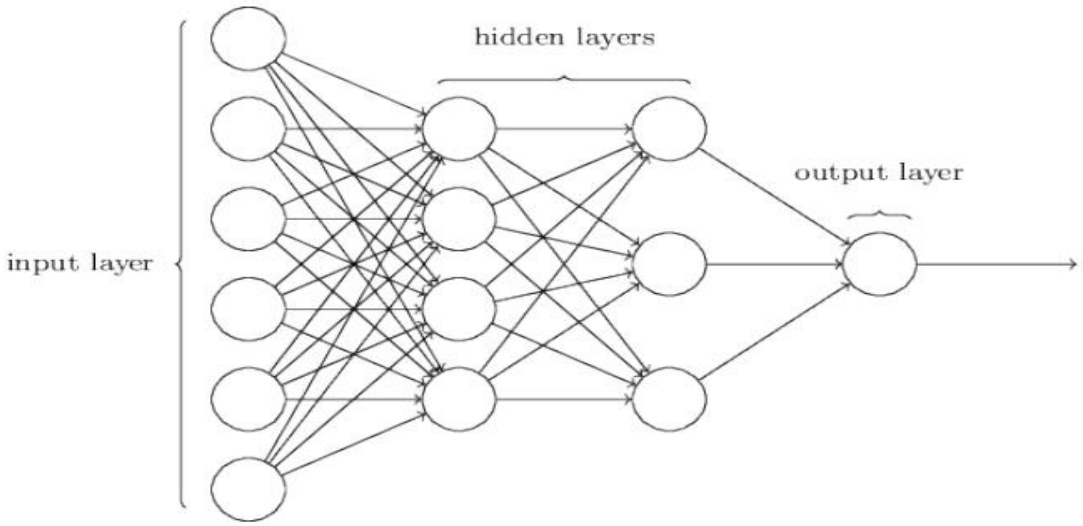


Fig. 1.4: General representation of PNN

In this network, the first column of perceptrons - what we'll call the first layer of perceptrons - is making three very simple decisions, by weighing the input evidence. Now for the perceptrons in the second layer, each of those perceptrons is making a decision by weighing up the results from the first layer of decision-making. In this way a perceptron in the second layer can make a decision at a more complex and more abstract level than perceptrons in the first layer. And even more complex decisions can be made by the perceptron in the third layer. In this way, a many-layer network of perceptrons can engage in sophisticated decision making. Such multiple layer networks are sometimes called Multilayer Perceptrons or MLPs. And when the output from one layer is used as input to the next layer, such networks are

called feed-forward neural networks and here there are no loops in the network –and the information is always fed forward, never fed back. However, there are other models of artificial neural networks in which feedback loops are possible. These models are called recurrent neural networks.

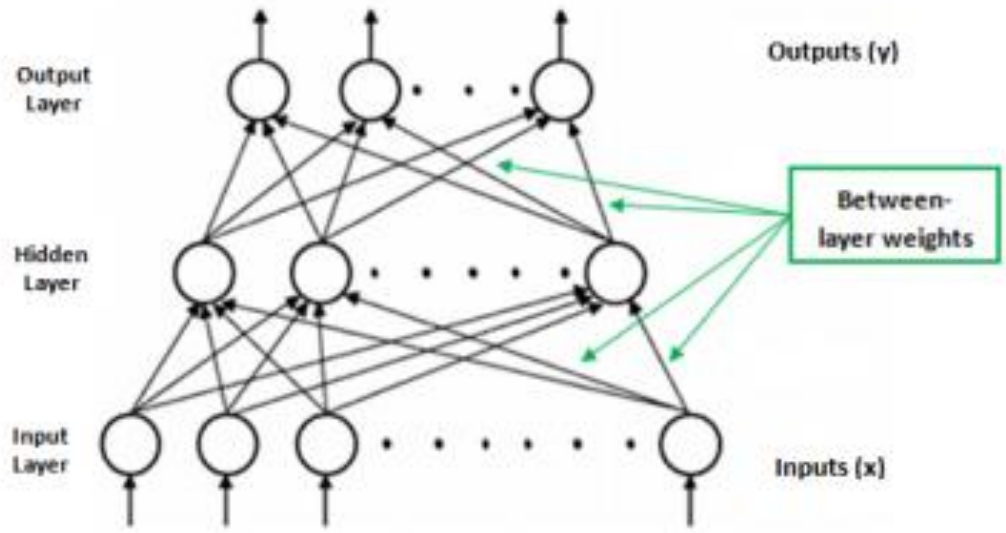


Fig. 1.5: Feed forward neural network representation

In Figure 1.5, x represents the vector of the network's inputs, and y is a vector of the network's output. ANN is a kind of unconstrained optimization problem, where the neuron weight's values are design variables to be found. The minimization function is the Mean Square Error (MSE), which is the difference between the network predicted outputs and the exact training outputs. The following shows the optimization formula for the general ANN:

$$\text{Find: } W_i \in R^{N.}$$

$$MSE = \sum_{i}^{n} (T_i - Y_i)^2$$

In the above formula, N represents the number of hidden neurons, and W_i is the i^{th} weight value in the "between layer weights." Many models and algorithms are available in ANN, ranging from basic models that could consist of single input, hidden, and output layers to more complex multi-layer ones. The complexity of ANN depends on the problem to be solved. The complexity of such a problem depends on the number of inputs that the network needs to handle and create relationships between and the number of outputs it needs to predict. The network works as a multi-dimensional curve fit (i.e., regression curve) for the system inputs. The hidden layer determines the function f(x) that expresses the training data, while the neurons determine the dimension of the function (n) and the variable coefficients $[a_0,a_1,a_2,a_3,\ldots\ldots a_n]$.

The system inputs represent $[x=x_1,x_2,x_3\ldots\ldots x_R]$ for (R) the number of inputs. As stated previously, more complex problems need more neurons in the hidden layer and, rarely, more hidden layers, because the literature indicates that a single hidden layer ANN is able to predict any practical nonlinear system (Bishop et al., 1995).

1.6.2 Learning Processes in Neural Networks

A neural network, is distinguished by the ability of the network to learn from its environment, and to improve its performance through learning. The improvement in performance takes place over time in accordance with some prescribed measure.

A neural network learns about its environment through an iterative process of adjustments applied to its synaptic weights and thresholds. Ideally, the network becomes more knowledgeable about its environment after each iteration of the learning process.

There are three broad types of learning:

1. Supervised learning (i.e. learning with an external teacher)

2. Unsupervised learning (i.e. learning with no help)

3. Reinforcement learning (i.e. learning with limited feedback).

1.6.3 Training Process in Neural Networks

Generally, ANN applications fall into the categories of data clustering, classification, or regression. Data clustering creates relationships between fed inputs and separates them into different clusters based on their similarities. In classification, inputs are assigned to their class among different classes. Data regression means creating a curve that passes and fits between training data sets. There are two types of training processes: supervised and unsupervised. The supervised training includes a set of training data where both input and output are known. The network is trained to create the proper output by combining the inputs in the proper way.

1.6.4 Common Types of Artificial Neural Networks

Many types of ANN have been developed to be used for many applications. Even for the same type, there are ANNs that differ in transfer functions and training approaches. Thus, selecting the most appropriate ANN type for a specific problem is not trivial. The two main types of ANN that are used specifically to solve regression problems are

(i). Feed-Forward Neural Network (FFNN)

(ii). Radial-Basis Neural Network (RBNN).

(i). Feed-Forward Neural Network (FFNN)

Feed-Forward Neural Network (FFNN), which is shown in Figure 1.5, is one of the most common and first developed types of ANN. Inputs are included in the input layer, which is shown in the figure as a set of circles. The inputs enter the hidden layer by the neuron weights that are shown in the figure. The hidden neurons are represented as circles each inside with a Sigmoidal transfer function. The output layer receives the outputs of the hidden layer neurons by another set of neuron weights. Inside each neuron in the output layer, there is linear transfer function, shown in the same figure, to provide the final results (outputs). Generally, the Sigmoidal and linear transfer functions are used on the hidden and output layers, respectively, when the problem is a regression type.

FFNN is widely used because of its use in applications in both classifications and regression problems. The advantages of using FFNN are as follows:

1. Generalizing system prediction at any input or extrapolating off-grid training space. After the network is trained, it will be able to predict any new input, even those out of the training limits.
2. Working well for many applications, especially curve fitting of the time series data (i.e., data that come in different times and values).

FFNN, however, has some limitations that constrain using it for some applications. These limitations include the following:

1. It could be highly inaccurate because of local minima solution that comes from optimization. Usually, FFNN has more neurons in its hidden layer than other types of ANN. So, a local optimization solution is more likely to occur in FFNN.
2. It experiences training time and memory issues during the training process because it has more neurons to be optimized.

Therefore, these limitations exclude FFNN as an option in some applications when the number of the training cases and/or inputs and outputs are large. It is also excluded when high accuracy is required for system performance.

(ii). Radial-Basis Neural Network (RBNN)

Figure 1.6 shows the Radial Basis Neural Network (RBNN), which is another type of ANN that is widely used in various applications. Besides the input and output vectors, the network consists of one hidden layer and one outputs layer.

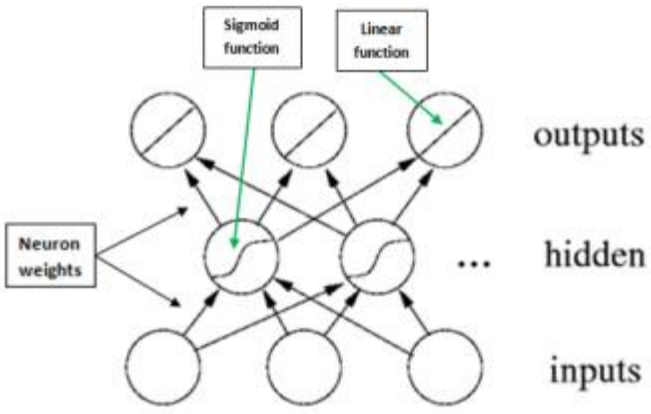


Fig. 1.6: Radial basis neural network general representation.

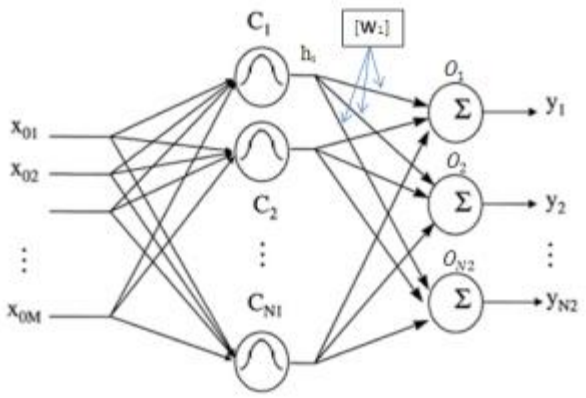


Fig. 1.7: Radial basis neural network (detailed representation)

In the figure 1.6, x= [x_{01}, x_{02}, …….,x_{0M}] represents inputs of the network, [C_1, C_{12},….,C_{N1}]are the neurons of the hidden layer, [W_1]is the vector of weights at the first neuron in the hidden layer (called line weights), and [y_1, y_2,.., y_{N2}] represent the network's outputs.

In the figure, x= [x_{01}, x_{02}, …….,x_{0M}] provides the input for each neuron in the hidden layer, labelled C_1 in the figure. In this case, the neurons are an essential radial basis function, hence the name radial basis neural network. All of the neurons collectively constitute the hidden layer. The hidden layer has N_1 neurons , [C_1, C_{12},….,C_{N1}]. Inside each hidden neuron $C_i(x)$, there is a radial transfer function that produces output h_i. The output h_i is multiplied with weight vector W_i to produce hidden output vector A_i. The dimension of the weight matrix, as shown in equation above, and hidden output matrix A is N_2xN_1. Each row of W and A is referred to as a weight and hidden output vector associated with a corresponding neuron. The output layer has a number of neurons, labelled O_1 in the figure, equal to number of outputs [y_1, y_2,.., y_{N2}].Inside each output neuron $O_i(A_i)$, the output is calculated by taking the sum of the received lines A_i , which represents a column of matrix A.

$$W=\begin{pmatrix} w11 & \cdots & w1n2 \\ \vdots & \ddots & \vdots \\ wn11 & \cdots & wn1n2 \end{pmatrix} \qquad h= [h_1, h_2, \ldots..h_{N1}] \qquad A=h^T.W$$

RBNN is trained by solving the optimization problem, and then find: W_{N1XN2}

$$MSE = \sum_{i}^{n} (T_i - Y_i)^2$$

In the above formula, T_i represents the i^{th} training output, y_i is the predicted output from the network. Note that y is a function of W, as shown in equation above. The training starts with the first iteration with one hidden neuron (N_1=1). Then, N_1 is incremented by 1 each time before the next iteration. The optimization stops once the MSE equals a small value (almost zero).

Like FFNN, the RBNN is used for all applications in both classification and regression problems. RBNN is specifically used in certain applications owing to certain specific advantages as:

1. It provides highly accurate results within the limits of the training space (i.e., inside the domain of the training values).

2. There are no local minima problems. The network does not optimize to local minimum solutions because the number of hidden neurons is optimized automatically in the training process. Thus, the optimal solution is obtained in terms of the number of neurons and the network weight matrix W.

3. There are no computational time and computer memory problems, especially when there are a large number of input/output training sets, because the network does not have a large number of neurons and weights. The weight values to be optimized exist only on the output side of the hidden layer, while FFNN has weights in both sides.

4. It is also found by experience that RBNN is the best type of ANN for high dimensional regression models.

Although RBNN has powerful prediction capabilities, it has some limitations, as follows:

1. The network parameter (Gaussian width) is determined heuristically, which could produce poor results.

2. It cannot predict points that are out of training grid space. The network cannot provide accurate outputs when the input is outside the range of training data (i.e., no extrapolation).

1.6.5 Advantages of ANN over SVM:

1. ANNs can have multiple outputs while SVMs have only one output.

2. n-ary classifiers in SVMs have to be trained one by one whereas in ANNs they can be trained in one go.

3. Neural networks make more sense because they are one whole whereas SVMs are isolated systems. This helps if the outputs are inter-related.

For example, if the goal was to classify hand-written digits, ten support vector machines would do. Each support vector machine would recognize exactly one digit, and fail to recognize all others. Since each handwritten digit cannot be meant to hold more information than just its class, it makes no sense to try to solve this with an artificial neural network.

However, suppose the goal was to model a person's hormone balance (for several hormones) as a function of easily measured physiological factors such as time since last meal, heart rate, etc ... Since these factors are all inter-related, artificial neural network regression makes more sense than support vector machine regression.

- ANN are extremely powerful computational devices. Moreover, massive parallelism makes them very efficient.
- ANN can learn and generalize from training data – so there is no need for enormous feats of programming.
- They are particularly fault tolerant.
- They are very noise tolerant – so they can cope with situations where normal symbolic systems would have difficulty.
- In principle, they can do anything a symbolic/logic system can do, and more.

ANN is fast and accurate because after the training process is completed, optimization and time-consuming calculations are no longer needed. So, the network outputs are predicted directly for the provided inputs based on what it has learned to predict for a specific system.

CHAPTER-2

LITERATURE SURVEY

In spite of the difficulties and numerous other challenges, pedestrian detection continues to be an active research area in computer vision. Because of falling expenses and the expanded effectiveness of computational power, vision-based techniques remain popular for a variety of applications including: adaptive driver assistance systems, collision avoidance, monitoring traffic scenes etc. Numerous approaches have been proposed in this regard. There are detection methods proposed in the literature, which use monocular cameras, stereo cameras, and active sensors, such as radar and infrared to capture the object of interest for specific application.

In pedestrian detection framework, a feature pattern learnt by a classifier is exhaustively searched in the full image. Distinguishing human bodies taking into account appearance is a great deal more troublesome than identifying other rigid objects as cars or faces. Human bodies are non-rigid, and highly articulated. This implies that we have to deal with a high range of different poses and postures. Additionally, in human detection it is not possible to take advantage of some specific textures and colour information due to the variability of worn cloths.

2.1 Various Pedestrian Detection approaches used are as follows:

(i). Holistic detection:

In this, the detectors are trained to search for pedestrians in the video frame by scanning the whole frame. The detector would detect a pedestrian, if the image features inside the local search window meets certain criteria. Some methods employ global features such as edge template as in (1), others use local features like histogram of oriented gradients (HOG) descriptors (2). The main challenge associated with this approach is that the performance of such detectors can be affected by background clutter and occlusions.

(ii). Part-based detection

In case of part based detection, the pedestrians are modelled as collections of several parts. Part hypotheses are firstly generated by learning local features, which include edgelet as in (3) and orientation features as in (4). In the subsequent steps, these part hypotheses are then joined to form the best assembly of existing pedestrian hypotheses. Despite the fact that this methodology is appealing, yet part location itself is a troublesome undertaking. Implementation of this approach, in fact follows a standard procedure for processing the image data that consists of first creating a densely sampled image pyramid, computing features at each scale, performing classification at all possible locations, and finally performing non-maximal suppression to generate the final set of bounding boxes.

(iii). Patch-based detection

Leibe et al (5) proposed an approach combining both the detection and segmentation with the name-Implicit Shape Model (ISM) in which a codebook of neighbourhood or local appearance is learned during the training process. In the detecting process, extracted local features are used to match against the codebook entries, and each match casts one vote for the pedestrian hypotheses. Final detection results can be obtained by further refining those hypotheses. The advantage of this approach is that comparatively only a smaller number of training images are required.

(iv). Motion-based detection

When the conditions permit (fixed camera, stationary lighting conditions,and so forth), background subtraction can help to detect pedestrians. Background subtraction classifies the pixels of video streams as either background, where no motion is detected, or as foreground where motion is detected. This procedure highlights the silhouettes/outlines (the connected components in the foreground) of every moving element in the scene, including people. Subsequently algorithms have been developed by researchers to analyze the shape of these silhouettes in order to detect the humans.

Since the methods that consider the silhouette as a whole and perform a single classification are, in general, highly sensitive to shape defects, a part-based method splitting the silhouettes in a set of smaller regions has been proposed to decrease the influence of defects.

To the contrary of other part-based approaches, these regions do not have any anatomical meaning. This algorithm has been extended to the detection of humans in 3D video streams.

(v). Detection using multiple cameras.

Fleuret et al.[6] suggested a method for integrating multiple calibrated cameras for detecting multiple pedestrians. In this approach, the ground plane is partitioned into uniform, non-overlapping grid cells, typically with size of 25 by 25 (cm). The detector produces a Probability Occupancy Map (POM), which in turn provides an estimation of the probability of each grid cell to be occupied by a person. Given two to four synchronized video streams taken at eye level and from different angles, this method can effectively combine a generative model with dynamic programming to accurately follow up to six individuals across thousands of frames in spite of significant occlusions and lighting changes. It can likewise derive metrically accurate trajectories for each one of them.

2.2 Various Foreground Segmentation techniques used are:

Foreground Segmentation, which is also referred to as "Candidate generation", extracts regions of interest (ROI) from the image to be sent to the classification module, avoiding as many background regions as possible. Specific segmentation module (e.g. exhaustive scanning) is of momentous significance not only to reduce the number of candidates, but also to avoid scanning regions like the sky. The key to this stage is to avoid missing pedestrians; otherwise, the subsequent modules will not be able to correct the error. In describing this module, the term pedestrian size constraints (PSC) is used , which refers to the aspect ratio, size, and position that candidate ROIs must fulfill to be considered to contain a pedestrian.

For any object in an image, interesting points on the object can be extracted to provide a feature description of the object. This description, extracted from a training image, can then be used to identify the object when attempting to locate the object in a test image containing many other objects. To perform reliable recognition, it is important that the features extracted from the training image be detectable even under changes in image scale, noise and illumination. Such points usually lie on high-contrast regions of the image, such as object edges.

Another important characteristic of these features is that the relative positions between them in the original scene shouldn't change from one image to another. For example, if only the four corners of a door were used as features, they would work regardless of the door's position; but if points in the frame were also used, the recognition would fail if the door is opened or closed. Similarly, features located in articulated or flexible objects would typically not work if any change in their internal geometry happens between two images in the set being processed. Therefore, in practice we need such feature descriptors that detect and use a much larger number of features from the images, which in turn also reduces the contribution of the errors caused by these local variations in the average error of all feature matching errors.

The simplest candidate generation procedure is an Exhaustive Scanning approach that selects all of the possible candidates in an image according to pedestrian size constraints, without explicit segmentation. The exhaustive scan is typically used in general human detection systems, e.g., image retrieval, whereas PPSs tend to use some kind of segmentation. In fact, the latter can take advantage of some application prior knowledge (e.g., it is not necessary to search the top area of the image) so that the number of ROIs to process can be greatly reduced.

Detecting pedestrians by their body heat is attractive, certainly when considering images shot on a cold winter night, where the pedestrians stick out as white regions before a black background. However, the situation is less appealing on sunny summer days, when there is an abundance of additional hot spots. In the latter case, one needs to resort to a similar set of detection techniques as in the visible domain.

Other interesting approaches for pedestrian detection that have been proposed, amongst them most works have pursued a learning-based approach, bypassing a pose recovery step and describing human appearance directly in terms of simple low-level features from a Region of Interest (ROI).

Background subtraction, which obtains the foreground (moving) pixels by subtracting the current input image from a background image is one of the most efficient methods for change detection, and it performs well when separating the object or region of interest from the background, say a pedestrian or a moving vehicle. Standard background subtraction, as frequently used in surveillance applications, is unsuitable because of the moving camera used

here. Therefore the viable alternatives include sliding windows, detection of independently moving objects, and stereo-based obstacle detection.

In applications such as video surveillance and monitoring, since the location of the persons in the scene provides useful information to activity recognition and understanding, it becomes necessary to properly identify the object of interest(which is, the pedestrian in our case) in real-time. Therefore, to enable the processing of the vast amounts of data provided by camera, the volume of data must first be reduced through pedestrian detection methods so that one can focus only on regions of interest to solve problems such as person tracking, face recognition, person re-identification, action and activity recognition, aiming at the scene analysis and understanding.

Several challenges are faced by the pedestrian detection. Among them are changes in illumination and person's appearance, pose variations, low quality of the acquired data, and the small size of the pedestrian in the image, making the detection process harder. However, besides all these challenges, the majority of applications require high performance and reliable detection results, which increases the need for efficient and accurate pedestrian detection approaches.

The sliding window approach shifts ROI windows of all possible sizes, at all locations over the images while performing feature extraction and pattern classification. This approach in combination with powerful classifiers i.e. Mohan et al., (13) is currently computationally too intensive for real-time application. In the Sliding window technique, detector windows at various scales and locations are shifted over the image. The computational costs are often too high to allow for real-time processing. Significant speedups can be obtained by either coupling the sliding window approach with a classifier cascade of increasing complexity or by restricting the search space based on known camera geometry and prior information about the target object class. These include application-specific constraints such as the flat-world assumption, ground-plane-based objects and common geometry of pedestrians, e.g., object height or aspect ratio. In case of a moving camera in a real-world environment, varying pitch can be handled by relaxing the scene constraints or by estimating the 3-D camera geometry online.

One of the well-known implementations of the Sliding Window algorithm is proposed by Dalal et al. (2), in which a scale pyramid is first constructed and then a fixed window of 64 × 128 pixels is used to scan each one of the scales.

As in (20), proposes a novel approach to optimize pedestrian detection methods based on sliding windows. The idea is to perform a random filtering in the image to select a very small number of detection windows and discard the remaining ones. In the method used, it does not perform any type of processing for the discarded windows, providing a significant speed-up.

Due to the random nature of the choice, the selected windows might be slightly dislocated of the person's body, which need to be fixed before presenting them to a classifier. Therefore, a regression, referred to as location regression, is executed to each detection window to adjust its location in the image.

The experimental evaluation, performed using the publicly available INRIA Pedestrian Dataset (21) (which is widely employed to pedestrian detection evaluation), demonstrates that it is possible to discard a large number of detection windows and still achieve accurate results. In addition, the employment of the location regression has shown to be very effective on correcting the detection window locations. As a consequence, a significant reduction on computational cost can be achieved.

However, Viola et al. (8) demonstrated an efficient variant of the sliding window technique, which involves a detector cascade using simple appearance and motion filters (similar to the Haar-wavelets). Simpler detectors, with a smaller number of features, are placed earlier in the cascade and more complex detectors later. At each detector stage, AdaBoost by Duda et al., (9) incrementally selects those features with the lowest weighted error on the training set, until a user-supplied correct and false-detection rate is achieved on a validation set.

Another approach for obtaining ROIs is by stereo vision. Zhao and Thorpe et al (7) obtain a foreground region by clustering in the disparity space. Some other techniques to obtain initial object hypotheses employ features derived from the image data. Besides approaches using Stereo Vision methods; Object Motion has been used as an early cueing mechanism.

Surveillance approaches using static cameras often employ background subtraction. Generalizations to moving cameras mostly assume translatory camera motion and compute

the deviation of the observed optical flow from the expected ego-motion flow field. This approach typically assumes translatory camera motion and detects deviations in the optical flow field from the expected background motion.

Another technique employs interest-point detectors to recover regions with high information content based on local discontinuities of the image brightness function that often occur at object boundaries.

2.3 Various Classification approaches used are:

Classification is defined as 'ability of machine to classify a certain object into a category or class on the basis of features obtained from object'. And features are those characteristics of an object that differentiate it from other objects. For example, if we have to classify a pet animal as a cat or a dog, so on the basis of its features like shape, size and colour we can classify its either as a cat or dog.

Classification is usually of two types: Supervised and Unsupervised classification. In supervised learning first a model is trained, but in unsupervised learning model learns by itself. In supervised learning we trained classifier according to the given features of object to be classified known as positive feature set and features of those objects which are different from object to be classified and which could appear in background of that object, these features are known as negative feature set. So by using positive and negative feature set classifier will able to model a decision boundary that can classify an object in two classes. Where as in unsupervised learning classifier uses an algorithm based on clustering, in which it makes cluster of similar kind of features. Neural networks, logistic regression and support vector machine are supervised classification techniques, k-mean clustering and mean-shift clustering are unsupervised classification methods.

The object classification module receives a list of ROIs that are likely to contain a pedestrian. In this stage, they are classified as pedestrian or non-pedestrian aiming, with the goal of minimizing the number of false positives and false negatives. The approaches to object classification are purely 2D, and can be broadly divided into silhouette matching and appearance. After a set of initial object hypotheses has been acquired, further verification

(classification) involves pedestrian appearance models, using various spatial and temporal cues. Following a rough categorization of such models into generative and discriminative model, introduce delineation in terms of visual features and classification techniques. In both the generative and discriminative approaches to pedestrian classification, a given image (or a sub-region thereof) is to be assigned to either the pedestrian or non-pedestrian class, depending on the corresponding class posterior probabilities. The main difference between generative and discriminative models is how posterior probabilities are estimated for each class.

(i) Silhouette matching.

The simplest approach is the binary shape model, presented by Broggi et al.(10), in which an upper body shape is matched to an edge modulus image by simple correlation after symmetry-based segmentation. A more sophisticated approach is the Chamfer System, a silhouette-matching algorithm proposed by Gavrila et al (11). This system consists of a hierarchical template-based classifier that matches distance transformed ROIs with template shapes in a coarse-to-fine manner.

(ii) Appearance.

The methods included in this group define a space of image features (also known as descriptors), and a classifier is trained by using ROIs known to contain examples (pedestrians) and counterexamples (non-pedestrians).

Following a holistic approach (i.e., target is detected as a whole), Gavrila et al(11) proposed a classifier that uses image gray-scale pixels as features and a neural network with local receptive fields as the learning machine that classifies the candidate ROIs generated by the Chamfer System. Zhao and Thorpe(7) used image gradient magnitude and a feed forward neural network. Papageorgiou and Poggio(1) introduced the so-called Haar wavelets (HWs) as features to train a quadratic support vector machines (SVMs) with front- and rear viewed pedestrians. Haar wavelets compute the pixel difference between two rectangular areas in different configurations, which can be seen as a derivative at a large scale. Viola and Jones(8) proposed AdaBoost cascades (layers of threshold-rule weak classifiers) as a learning algorithm to exploit Haar-like features (the original HWs plus two similar features, for surveillance-oriented pedestrian detection.

Dalal and Triggs(2) presented a human classification scheme that uses SIFT-inspired features, called histograms of oriented gradients (HOGs), and a linear SVM as a learning method. An HOG feature also divides the region into k orientation bins (in this case, k ¼ 9), but instead of computing the ratio between two bins, they define four different cells that divide the rectangular feature. In addition, a Gaussian mask is applied to the magnitude values in order to more heavily weight the centre pixels, and the pixels are interpolated with respect to pixel location within a block (both factors disallow the use of the integral image). The resulting feature is a 36-D vector containing the summed magnitude of each pixel cells, divided into 9 bins.

2.4 Classifier Architectures.

The main things that are important for Pedestrian Classification are: Image features extraction and Learning Classifier. Features extraction includes extracting the most relevant information from the available images. We want to find out an optimal image pixel representation that can underline differences between pedestrians and non-pedestrian images.

Discriminative classification techniques aim at determining an optimal decision boundary between pattern classes in a feature space. Feed-forward multilayer neural networks implement linear discriminant functions in the feature space in which input patterns have been mapped nonlinearly, e.g., by using the previously described feature sets. Optimality of the decision boundary is assessed by minimizing an error criterion with respect to the network parameters, i.e., mean squared error. In the context of pedestrian detection, multilayer neural networks have been applied particularly in conjunction with adaptive local receptive field features as nonlinearities in the hidden network layer. This architecture unifies feature extraction and classification within a single model.

Support Vector Machines (SVMs) have evolved as a powerful tool to solve pattern classification problems. In contrast to neural networks, SVMs do not minimize some artificial error metric but maximize the margin of a linear decision boundary (hyperplane) to achieve maximum separation between the object classes. Regarding pedestrian classification, linear SVM classifiers have been used in combination with various (nonlinear) feature sets.

AdaBoost (9) , which has been applied as automatic feature selection procedure (see above), has also been used to construct strong classifiers as weighted linear combinations of the selected weak classifiers, each involving a threshold on a single feature. To incorporate nonlinearities and speed up the classification process, boosted detector cascades have been introduced by Viola et al(8). Motivated by the fact that the majority of detection windows in an image are non-pedestrians, the cascade structure is tuned to detect almost all pedestrians while rejecting non-pedestrians as early as possible. AdaBoost is used in each layer to iteratively construct a strong classifier guided by user-specified performance criteria. During training, each layer is focused on the errors the previous layers make. As a result, the whole cascade consists of increasingly more complex detectors. This contributes to the high processing speed of the cascade approach, since usually only a few feature evaluations in the early cascade layers are necessary to quickly reject non-pedestrian examples.

Once ROIs have been established, different combinations of features and pattern classifiers can be applied to make the distinction between pedestrian and non-pedestrian. For example, Broggi et al. (10) employed vertical symmetry features. Zhao and Thorpe (7) apply a high-pass filter and normalize the ROI for size, thereafter applying a feed-forward neural network. Papageorgiou and Poggio (1) pioneered the use of Haar-wavelet features in combination with a Support Vector Machine (SVM).

Component-based approaches have been utilized to reduce the complexity of pedestrian classification. Shashua et al. (2004), for instance, extract a feature vector from each of 9 fixed sub-regions. Other approaches attempt to directly identify certain body parts.

Mohan et al. (13), extended the work of Papageorgiou and Poggio (1) to four component classifiers for detecting heads, legs, and left/right arms separately. Individual results are combined by a second classifier, after ensuring proper geometrical constraints. Additional attempts have been made towards reducing classification complexity by manually separating the pedestrian training set in non-overlapping sub-sets (i.e. based on pedestrian heading direction).

2.5 Verification/ Refinement:

Many systems contain one step that verifies and refines the ROIs classified as pedestrians. The verification step filters false positives, using criteria that do not overlap with the classifier, while the refinement step performs a fine segmentation of the pedestrian (not necessarily silhouette oriented) to provide an accurate distance estimation or to support the subsequent tracking module. For refinement, one essential algorithm that provides one detection per target is non-maximum suppression. Assuming that classifiers provide a peak at the correct position and scale of the target and weaker responses around it, Dalal (2) makes use of mean shift to find the minimum set of ROIs that best adjust to the pedestrians in the image.

The designed system for object detection by C. Papageorgiou et al.(1) is based on local multi-scale oriented intensity differences using Haar wavelet transform. It uses SVM(support vector machine) classifier for classification. And the method proposed addresses the problem of object and pattern detection in static images of unconstrained, cluttered scenes. The system makes use of a new representation that describes an object class in terms of a large set of local oriented intensity differences between adjacent regions; this representation is efficiently computable as a Haar wavelet transform. Images are mapped from the space of pixels, to that of a database of Haar wavelet features that provides a rich description of the pattern. This representation is able to capture the structure of the class of objects that is to be detected, while ignoring the noise inherent in the images. The example-based learning approach used here implicitly derives a model of an object class by training a support vector machine classifier using a large set of positive and negative examples. However, the system has to search the whole image at multi-scales for pedestrians. This would be an extremely computationally expensive procedure, and it may cause multiple responses from a single pedestrian.

Another method proposed by Zhao-Thorpe et al.(7) presented a fast and robust algorithm for detecting pedestrians in a cluttered scene from a pair of moving cameras. This is achieved through stereo-based segmentation and neural network-based recognition. The algorithm includes three steps. First, the image is segmented into sub-image object candidates using disparities discontinuity. Second, the sub-image object candidates are merged and split into sub-images, that satisfy pedestrian size and shape constrains. Third, intensity gradients of the

candidate sub-images are used as input to a trained neural network for pedestrian recognition. The experiments on a large number of urban street scenes demonstrated that the proposed algorithm: 1) can detect pedestrians in various poses, shapes, sizes, clothing, and occlusion status; 2) runs in real-time; and 3) is robust to illumination and background changes.

In the proposed system, a neural network trained with the back-propagation algorithm is used to discriminate pedestrians from other objects. The system used shape features instead of motions cues to detect both moving and stationary pedestrians. Since neural networks can express highly nonlinear decision surfaces, they are especially appropriate to classify objects presenting high degree of shape variability. In this system, the trained neural network implicitly represents the appearance of pedestrians in various poses, postures, sizes, clothing, and occlusion situations; it performs pedestrian detection in real time.

Method proposed by Viola-Jones et al. (8) describes a pedestrian detection system that integrates image intensity information with motion information. In this, a detection style algorithm that scans a detector over two consecutive frames of a video sequence. The detector is trained (using AdaBoost) to take advantage of both motion and appearance information to detect a walking person. Past approaches have built detectors based on appearance information, but ours is the first to combine both sources of information in a single detector. The implementation described runs at about 4 frames/second, detects pedestrians at very small scales (as small as 20×15 pixels), and has a very low false positive rate. The detection approach builds on the detection work of Viola and Jones. Novel contributions of the method proposed includes: i) development of a representation of image motion which is extremely efficient, and ii) implementation of a state of the art pedestrian detection system which operates on low resolution images under difficult conditions (such as rain and snow).

Further improvements in this method were made by: Gavrila et al. (11) whereby the author took a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has been used in a practical real-time pedestrian detection system. The system uses a generic two-step approach for efficient object detection.

In the first step, contour features are used in a hierarchical template matching approach to efficiently “lock” onto candidate solutions. The shape matching is based on Distance Transforms. By capturing the object’s shape variability by means of a template hierarchy, and

using a combined coarse-to-fine approach in shape and parameter space, this method achieves very large speed-ups compared to a brute-force method. They also measured gains of several orders of magnitude.

The second step utilizes the richer set of intensity features in a pattern classification approach to verify the candidate solutions (i.e. using Radial Basis Functions).

Next in method proposed by Shashua et al.(12) wherein classification on single frame was performed using a novel scheme of breaking down class variability by repeated training of simple classifiers on training set clusters. Along-with, it performs multi-frame approval process by using properties such as gait patterns, motion analysis, parallax, classifier consistency, tracking quality. Together with a shift-invariant local description of image sub-regions and discriminant integration using Adaboost , a powerful classifier is obtained that outperforms the leading approaches .

One of the key points made in this work is the observation that it is not realistic to expect a reasonable system level performance using single-frame classification only. The path from single-frame to system level performance must include the integration of additional cues measured over time (dynamic gait, motion parallax, stability of re-detection measures), situation specific features (such as leg positions at certain poses), and most importantly via building up additional object categories consisting of vehicles (both in motion and stationary) and stationary background structure such as poles, trees, guardrails, lane markings and so forth.

The experimental results of the system so far indicates that for some of the functions (such as inward moving pedestrian detection) the performance is satisfactory for daytime and normal weather conditions, and for the remaining functionalities the gap which remains is relatively small for meeting a daytime normal weather specification.

In (14) author presented a subsystem for the recognition stage of pedestrian detection system. This module is based on use of histogram of oriented gradients (HOG) combined with Support Vector Machines (SVM) classifier and works on both infrared and daylight images. This paper details filtering subsystem for a tetra-vision based pedestrian detection system. The complete system is based on the use of both visible and far infrared cameras; in an initial phase it produces a list of areas of attention in the images which can contain pedestrians. This

list is further refined using symmetry-based assumptions. Then, this results is fed to a number of independent validators that evaluate the presence of human shapes inside the areas of attention. Histogram of oriented gradients and Support Vector Machines are used as a filter and demonstrated to be able to successfully classify up to 91% of pedestrians in the areas of attention.

Further a paper by Bing-Fei Wu et al(18). presents a relative discriminative histogram of oriented gradients (HOG) (RDHOG)-based particle filter (RDHOG-PF) approach to traffic surveillance with occlusion handling. Based on the conventional HOG, an extension known as RDHOG is proposed, which enhances the descriptive ability of the central block and the surrounding blocks. RDHOG-PF can be used to predict and update the positions of vehicles in continuous video sequences. RDHOG was integrated with the particle filter framework in order to improve the tracking robustness and accuracy. To resolve multi-object tracking problems, a partial occlusion handling approach is addressed, based on the reduction of the particle weights within the occluded region. RDHOG-PF can determine the target by using the feature descriptor correctly, and it overcomes the drift problem by updating in low-contrast and very bright situations.

In (19), the author presented a novel mixture-of-experts framework for pedestrian classification with partial occlusion handling. The framework involves a set of component-based expert classifiers trained on features derived from intensity, depth and motion. Occlusions of individual body parts manifest in local depth- and motion discontinuities. In the application phase, a segmentation algorithm is applied to extract areas of coherent depth and motion. Based on the segmentation result, occlusion-dependent weights are determined for the component-based expert classifiers to focus the combined decision on the visible parts of the pedestrian. To handle partial occlusion, expert weights are computed that are related to the degree of visibility of the associated component. This degree of visibility is determined by examining occlusion boundaries, i.e. discontinuities in depth and motion. Occlusion-dependent component weights allow focusing the combined decision of the mixture of-experts classifier on the un-occluded body parts.

To optimize pedestrian detection methods, in (20) the author proposed a novel approach that performs a random filtering supported by the Maximum Search Problem theorem to select a very small number from all possible detection windows. Although the random filtering is able

to select regions that capture every person on an image, some windows can cover only parts of a person, diminishing the accuracy. To solve that, a regression is applied to adjust the windows to the person's location. The computational cost reduction comes from the fact that the proposed approach does not need to perform any processing while selecting windows, differently from cascades of rejection that must evaluate at least simple features for every window. The experiments performed using a pedestrian detection based on Partial Least Squares show that the approach is effective in both accuracy and computational cost reduction.

In (26) author put forward an approach for real-time person tracking in crowded and/or unknown environments using multi-modal integration. We combine stereo, colour and face detection modules into a single robust system, and show an initial application in an interactive, face-responsive display. Dense, real-time stereo processing is used to isolate users from other objects and people in the background. Skin-hue classification identifies and tracks likely body parts within the silhouette of a user. Face pattern detection discriminates and localizes the face within the identified body parts. Faces and bodies of users are tracked over several temporal scales: short-term (user stay's within the field of view), medium-term (user exits/re-enters within minutes), and long term (user returns after hours or days). Short-term tracking is performed using simple region position and size correspondences, while medium and long-term tracking are based on statistics of user appearance. We discuss the failure modes of each individual module, describe our integration method, and report results with the complete system in trials with thousands of users. To increase reliability, some systems integrate multiple cues such as stereo, skin colour, face, and shape pattern to detect pedestrians. However, skin colour is very sensitive to illumination changes; face detection can only identify pedestrians facing the camera. These systems further prove that stereo and shape are more reliable and helpful cues than colour and face detection in general situations.

Further developments in HOG based feature extraction techniques after Dalal (2) put forward his method were:

(A) As part of the 2006 European Conference on Computer Vision (ECCV), Dalal and Triggs teamed up with Cordelia Schmid to apply HOG detectors to the problem of human detection in films and videos. They combined HOG descriptors on individual video frames with their newly introduced internal motion histograms (IMH) on pairs of subsequent video

frames. These internal motion histograms use the gradient magnitudes from optical flow fields obtained from two consecutive frames. These gradient magnitudes are then used in the same manner as those produced from static image data within the HOG descriptor approach. When testing on two large datasets taken from several movies, the combined HOG-IMH method yielded a miss rate of approximately 0.1 at a 10-4 false positive rate.

(B) At the Intelligent Vehicles Symposium in 2006, F. Suard, A. Rakotomamonjy, and A. Bensrhair (16) introduced a complete system for pedestrian detection based on HOG descriptors. Their system operates using two infrared cameras. Since human beings appear brighter than their surroundings on infrared images, the system first locates positions of interest within the larger view field where humans could possibly be located. Then support vector machine classifiers operate on the HOG descriptors taken from these smaller positions of interest to formulate a decision regarding the presence of a pedestrian. Once pedestrians are located within the view field, the actual position of the pedestrian is estimated using stereo vision.

(C) At the IEEE Conference on Computer Vision and Pattern Recognition in 2006, Qiang Zhu, Shai Avidan, Mei-Chen Yeh, and Kwang-Ting Cheng presented an algorithm to significantly speed up human detection using HOG descriptor methods. Their method uses HOG descriptors in combination with the cascading classifiers algorithm, normally applied with great success to face detection. Also, rather than relying on blocks of uniform size, they introduce blocks that vary in size, location, and aspect ratio. In order to isolate the blocks best suited for human detection, they applied the AdaBoost algorithm to select those blocks to be included in the cascade. In their experimentation, their algorithm achieved comparable performance to the original Dalal and Triggs algorithm(2), but operated at speeds up to 70 times faster.

Chapter-3

Proposed Method

The system is composed by several of the proposed algorithms, making use of concepts introduced and working over the shortcomings of these methods mentioned in the literature survey as well. It makes the best use of existing methods by suitably employing them, with changes made in them as per the requirement of our project, so as to get best possible results and efficient detection of the pedestrian so as to avoid collision.

1. For Pedestrian Segmentation "sliding window" based approach will be used, as the camera is moving, then in such moving systems case the prevalent methods for segmentation like background subtraction will get failed. The sliding window approach shifts ROI windows of all possible sizes and scales at all locations over the images while performing feature extraction and pattern classification.

2. For feature extraction, HOG (Histogram of Oriented Gradients) based features descriptor is favourable because it is independent of intensity and based on change in intensity levels i.e. gradients or edges.

3. Based on the conventional HOG, an extension known as RDHOG is used; this enhances the descriptive ability of the central block and the surrounding blocks. RDHOG feature descriptors are computed by comparing HOG features of each block with the central block.

4. Neural network Classifier modelled on training and subsequent testing decides whether the region cropped by sliding window is in actual containing a pedestrian or not. So here basically the classifier is trained for four classes. These four classes correspond to four different view of pedestrian like: front view, side view (left or right) and rear view.

5. For training, features of positive and negative images are extracted. Positive images are pedestrian images and negative images are images of possible background objects.

The Algorithm proposed consists of following steps:

3.1. Database formation

As we are using neural network classifier for object segmentation, so before recognize any image you need to do training for a classifier. For example, if object to be tracked is a human then we need to train our classifier for different views of human. So we need database for different views of object to be tracked. Usually a database consists of positive and negative set of images. Positive set consist of images of object to be tracked and negative set consist of images of all other objects that could appear in background of the object to be tracked.

Fig 2.1: Sample of front view database

Fig 3.2: Sample images of back view database

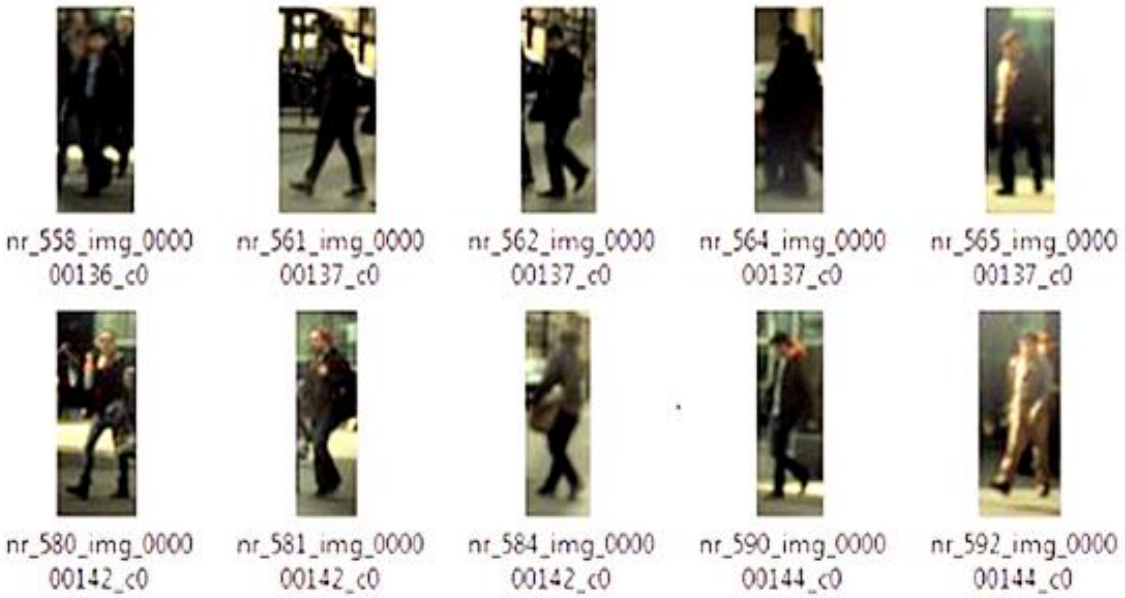


Fig 3.3: Sample images for left view database

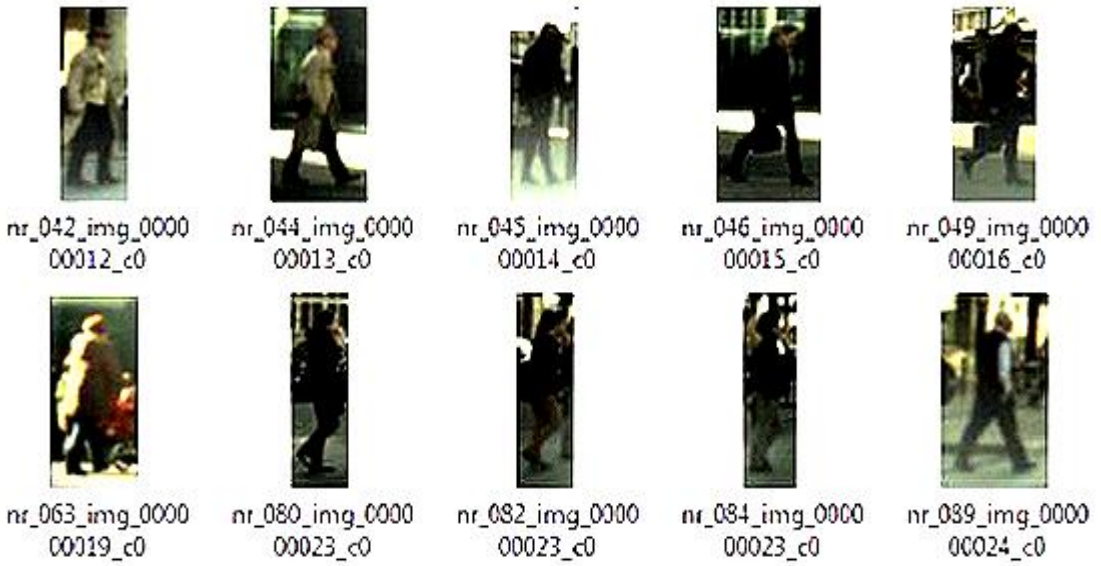


Fig 3.4: Sample images of right view database.

3.2. Classifier Training

For training a classifier we need feature descriptor set for both positive and negative set of images. Ideally we need such a feature descriptors which should converge well for a single class and should be invariant of surrounding changes. So we are using HOG feature descriptors for training negative and positive images of dataset. Here we have used neural network based classifier for classifier training. In neural network classifier we have used two variants:

1. Radial Basis Function Neural Network (RBFNN) based classifier,

2. Probabilistic Neural Network (PNN) classifier. But in implementation in main algorithm we preferred Probabilistic neural network classifier because it gives exact 0 and 1 as output class, as required for proper classification as per our requirement. But in RBFNN classifier, we get values between 0 to 1, so it is difficult to decide threshold value to classify it in a particular class.

3.3. Feature Descriptor calculation

Feature descriptors we have used to characterize an object are HOG [5] feature descriptors and its variant RDHOG [18] feature descriptors. These feature descriptors are well efficient and robust and also less computationally expensive in compare to features descriptors like SURF and SIFT. And these are also invariant to changes like scale, rotation and colour. HOG abbreviated as histogram of oriented gradients and RDHOG as relative discriminative HOG.

3.3.1. HOG Feature Descriptors

To calculate HOG feature descriptors first we resize our object to 32 by 32 pixels size. Then divide that in 9 overlapping blocks of 16 by 16 pixels size. Then each block is further divided into 4 cells of 8 by 8 pixels size. Then for each cell gradient is calculated, gradient id defined as change in intensity levels in particular direction. Then orientation for those gradients will be calculated and that orientation is further classified in to nine classes. That means we are constructing histogram of nine bins with a step size of 20 degree as range of orientation value

is 0 to 180 degree. So for each cell we are getting a vector of length 9, which means we will get a feature vector for 32 by 32 pixels image of length 324.

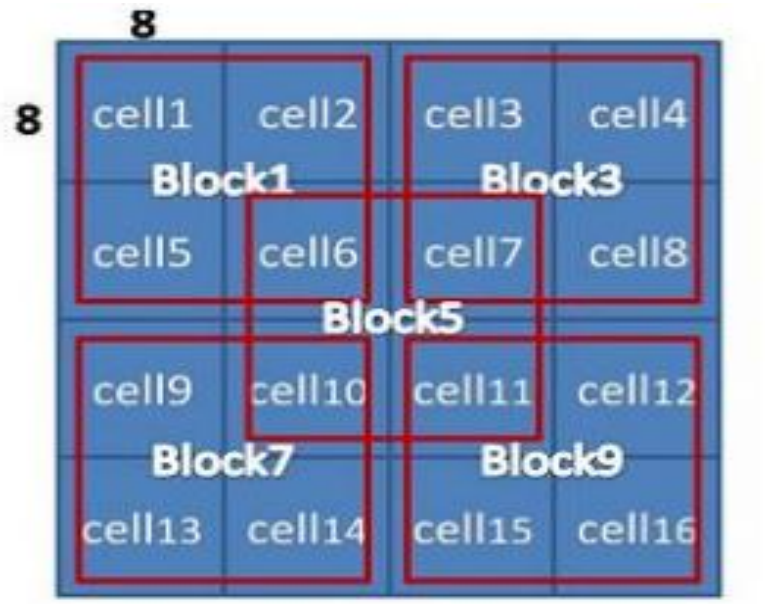


Fig 3.5: Block and Cell division in HOG and RDHOG feature descriptors

3.3.2. RDHOG Feature Descriptors

After getting 324 length HOG feature descriptor, now we will compute RDHOG feature descriptor by comparing HOG features of each block with the central block, as there are 9 blocks so 8 comparisons will be done. So we will get RDHOG feature vector of length 288.

3.3.3 HOG + RD-HOG Feature descriptor Steps:

1. Resize the object to 32 by 32 pixels size, and then divide that in 9 overlapping blocks of 16 by 16 pixels size. Then each block is further divided into 4 cells of 8 by 8 pixels size.

2. Gradient computation: Gradient is the change in intensity levels in particular direction. To find the horizontal and vertical gradients, convolve the image with kernels [-1 0 1] and $[-1\ 0\ 1]^T$.

3. Orientation binning: After computing gradients find out the orientations for each pixel gradients and then do binning for these into 9 classes from 0° to 180° with an interval of 20°. Number of pixels in a particular class constitutes the weight for that particular bin. So, as for the vote weight the pixel contribution can be the gradient magnitude itself or the square root or square of the gradient magnitude. So for each cell we are getting a vector of length 9, which means we will get a feature vector for 32 by 32 pixels image of length 324.

4. After getting 324 length HOG feature descriptor, now we will compute RDHOG feature descriptor by comparing HOG features of each block with the central block.

As there are 9 blocks, so 8 comparisons will be done.

$$RDb_j(i) = b_5(i) - b_j(i),\ \ j = 1 \sim 9, j \neq 5$$

where $b_j(i)$ is the i^{th} bin of the j^{th} block, $b_5(i)$ is the i^{th} bin of the central block, and $RDb_j(i)$ is the i^{th} RDHOG bin of the j^{th} block. So we will get RDHOG feature vector of length 288. Hence, the total number of elements in the descriptor is 612 after combining HOG features and RDHOG features.

5. Block normalization:

There are four different methods for block normalization: L2-norm, L2-hys, L1-norm and L1-sqrt.

3.4 Architecture of Pedestrian Detection System using Moving Camera

The following modules are proposed for splitting the architectures of pedestrian detectors for ADAS, listed according to the processing pipeline order:

- Pre-Processing
- Canny Edge detection to obtain gradient image.
- Use of Sliding Window approach to scan the frame.
- Apply thresholding on the basis of gradient density.
- Compute the HOG features of the segmented image.
- Apply the trained Neural Network Classifier, to select the segmented HOG features and check if frame contains pedestrian or not.

3.4.1 Procedural Implementation Steps:

1. Open the video in which pedestrian is to be detected, construct the video object of the video file and read number of frames in video. These frames now contain images run after another in a video sequence.

2. Now perform RGB to Gray Conversion of the image.

3. Compute the Canny edge to obtain the Gradient Image.

4. Scan the image using Sliding Window concept column-wise and then row-wise in a loop fashion so as to cover entire frame.

5. Now apply the threshold on the basis of gradient density, wherein the cropped segment having sufficient edge points are saved.

6. Compute the HOG features of the segmented image.

7. Apply the Neural Network Classifier, to select the segmented HOG features. The classifier is trained over a set of INRIA Pedestrian detection dataset images.

8. Now insert the bounding box wherever a pedestrian is detected and number the detected pedestrian in the output frame.

3.5 Algorithm for Classifier Training, Testing and its subsequent application is as follows:

The entire method implementation has been divided into two parts:

1. Classifier Training: In this step, we will train the Probabilistic Neural Network (PNN) Classifier for different postures of pedestrian.

2. Online Pedestrian Detection: Here we will detect the location of pedestrian in a video.

3.5.1 Classifier Training

Step 1: Generating the Database

First prepare (generate) the database for positive and negative images.

Positives images comprises of images of pedestrian in 4 different postures:

1. Front, 2. Back, 3. Left, 4. Right.

And negative images consist of images of objects that could appear in background of a pedestrian.

Step 2: Generating the training features to train the classifier

After generating database, we have to generate training features of these images so as to train our classifier.

Step 3: Training the PNN classifier

After obtaining the features for all classes, we will train our PNN classifier.

3.5.2 Online Pedestrian Detection:

Step 1: Open MATLAB file containing the function meant for object detection and run the code in it, which will in turn ask us to select the video file in which pedestrian detection needs to be done.

Step 2: Visualizing the Output.

Step 3: Obtaining the Output as Video.

CHAPTER-4

RESULTS & DISCUSSION

In pattern recognition and information retrieval with binary classification, Precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance, that is, precision is "how useful the search results are", and recall is "how complete the results are".

In simple terms, high precision means that an algorithm returned substantially more relevant results than irrelevant, while high recall means that an algorithm returned most of the relevant results.

In a classification task, the precision for a class is the number of true positives (i.e. the number of items correctly labelled as belonging to the positive class) divided by the total number of elements labelled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labelled as belonging to the class).

Recall in a classification task is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labelled as belonging to the positive class but should have been).

In information retrieval, a perfect precision score of 1.0 means that every result retrieved by a search was relevant (but says nothing about whether all relevant cases were retrieved) whereas a perfect recall score of 1.0 means that all relevant cases were retrieved by the search (but says nothing about how many irrelevant cases were also retrieved).

Now finally applying the algorithm and using the codes run on MATLAB, the frames captured in real-time for the purpose of pedestrian detection using moving camera gives following results. The subsequent values of: True Positive (tp), False Negative (fn), False Positive (fp) help in determining the values of Recall and Precision, and comparative analysis for HOG based and Neural network based systems.

PARAMETER	HOG Based	Neural N/w based
True Positive (tp)	1437	1321
False Negative (fn)	220	336
False Positive (fp)	256	138
Recall	86.72	79.72
Precision	84.88	90.54

where the values for Precision, Recall and Accuracy are calculated using formulae given below:

$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

SCENE 1

FRAME 1

FRAME 2

FRAME 3

FRAME 4

FRAME 5

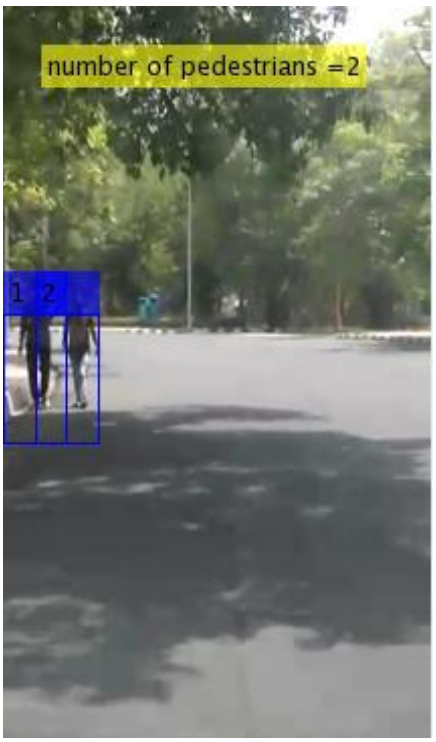
FRAME 6

FRAME 7

FRAME 8

SCENE 2

FRAME 1



FRAME 2

FRAME 3

FRAME 4

FRAME 5

FRAME 6

FRAME 7

FRAME 8

CHAPTER-5

CONCLUSION & FUTURE WORK

In order to detect the presence of pedestrian in the image frame, the segmentation is performed using Sliding Window method. As for our project application, wherein the camera is moving, then in such moving systems case the prevalent methods for segmentation like background subtraction will get failed. Therefore, with the help of sliding window technique which scans through the frame through different scales, segmentation of the object of interest, which is the pedestrian in our case, is done.

Added to that advantages offered by HOG such as better representation of human contour, invariance to illumination changes and small movements, and easy computation in constant time make it best suited for its application as a feature extractor.

In addition to that, Neural Network is preferred because single neural classifier can be used for training and classification of multiple classes. Also for large set of database, convergence is better in neural network based classifier.

LIMITATIONS

The limitation associated with our method used is that the size of our sliding window is not adaptable; it is constant throughout the image. Considerable speed up can be obtained if this window is made adaptable. Moreover, the computational complexity is high which further needs to be improved upon.

FUTURE WORK:

Following are some of the improvements which can be worked upon to considerably speed up the implementation and enhance the efficiency further:

• We can use Weighted-HOG feature descriptors.

• We can introduce tracking also using Particle Filter.

• We can use Adaptive Sliding Window size, in which window size can be varied accordingly.

• We can use CNN(Concurrent Neural Network) classifier instead of Simple Neural Network.

• We can use Contour-based approach for Segmentation part.

REFERENCES

1. C. Papageorgiou and T. Poggio, “A Trainable System for Object Detection,” Int’l J. Computer Vision, vol. 38, no. 1, pp. 15-33, 2000.

2. N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in IEEE Intl. Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.

3. Bo Wu and Ram Nevatia, "Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors"- 2005.

4. Mikolajczyk, K. and Schmid, C. and Zisserman, A. "Human detection based on a probabilistic assembly of robust part detectors"- 2005

5. B.Leibe, E. Seemann, and B. Schiele. "Pedestrian detection in crowded scenes"-2005.

6. Fleuret et al.[Multi-Camera People Tracking with a Probabilistic Occupancy Map -2008] IEEE Transactions on Pattern Analysis and Machine Intelligence.

7. L. Zhao and C. Thorpe, “Stereo and Neural Network-Based Pedestrian Detection,” IEEE Trans. on Intelligent Transportation Systems, 148-154, September 2000.

8. P. Viola, M. Jones, and D. Snow, “Detecting Pedestrians Using Patterns of Motion and Appearance,” Int’l J. Computer Vision, vol. 63, no. 2, pp. 153-161, 2005.

9. Adaboost -R.O.Duda, P.E.Hart and D.G.Stork, ‘Pattern Classification’, Johy Wiley, 2002.

10. A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, “Shape-Based Pedestrian Detection,” Proc. IEEE Intelligent Vehicles Symp., pp. 215-220, 2000.

11. D. Gavrila, “Pedestrian Detection from a Moving Vehicle,” Proc. European Conf. Computer Vision, vol. 2, pp. 37-49, 2000.

12. A. Shashua, Y. Gdalyahu, and G. Hayun, “Pedestrian Detection for Driving Assistance Systems: Single-Frame Classification and System Level Performance,” Proc. IEEE Intelligent Vehicles Symp., pp. 1-6, 2004.

13. A. Mohan, C. Papageorgiou, and T. Poggio, “Example-Based Object Detection in Images by Components,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 4, pp. 349-361, Apr. 2001.

14. A Pedestrian Detector Using Histograms of Oriented Gradients and a Support Vector Machine Classifier M. Bertozzi, A. Broggi, M. Del Rose, M. Felisa, A. Rakotomamonjy and F. Suard.

15. Markus Enzweiler, Dariu M. Gavrila, "Monocular Pedestrian Detection: Survey and Experiments", IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.31, no. 12, pp. 2179-2195, December 2009, doi:10.1109/TPAMI.2008.260

16. F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian Detection Using Infrared Images and Histograms of Oriented Gradients," Proc. IEEE Intelligent Vehicles Symp., pp. 206-212, 2006.

17. Qiang Zhu, Shai Avidan, Mei-Chen Yeh, and Kwang-Ting Cheng; IEEE Conference on Computer Vision and Pattern Recognition, 2006.

18. A Relative-Discriminative-Histogram-of-Oriented Gradients Based Particle Filter Approach to Vehicle Occlusion Handling and Tracking Bing-Fei Wu, Fellow, IEEE, Chih-Chung Kao, Student Member, IEEE, Cheng-Lung Jen, Student Member, IEEE, Yen-Feng Li, Ying-Han Chen, and Jhy-Hong Juang AUGUST 2014

19. M. Enzweiler, A. Eigenstetter, B. Schiele and D. M. Gavrila, "Multi-cue pedestrian classification with partial occlusion handling," Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, San Francisco, CA, 2010.

20. V. H. C. d. Melo, S. Leão, D. Menotti and W. R. Schwartz, "An Optimized Sliding Window Approach to Pedestrian Detection," Pattern Recognition (ICPR), 2014 22nd International Conference on, Stockholm, 2014, pp. 4346.

21. INRIA Pedestrian Dataset, MIT pedestrian database.

22. R.O.Duda, P.E.Hart and D.G.Stork, 'Pattern Classification', Johy Wiley, 2002

23. C. M. Bishop, 'Neural Networks for Pattern Recognition', Oxford University Press, Indian Edition, 2003.

24. C.M.Bishop, 'Pattern Recognition and Machine Learning', Springer, 2006.

25. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. ICCV '05 Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1

26. T. Darrell, G. Gordon, M. Harville, and J. Woodfill, “Integrated person tracking using stereo, color, and pattern detection,” And Haritauglu, D. Harwood, and L. S. Davis, “A real-time system for detecting and tracking people”