PEOPLE DETECTION IN CLUTTERED SCENES

by

Serdar Öztürk

B.S. in Electrical and Electronics Engineering, Boğaziçi University, 1999

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2010

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my thesis supervisor Prof. Bülent Sankur for his encouragement to me to return and complete my graduate education, and for his invaluable guidance and support throughout the preparation of this thesis.

I would also like to thank my thesis co-supervisor Dr. Ceyhun Burak Akgül for his significant advice and contribution throughout this study.

I would also like to thank Bilgin Eşme, who was with us at the first stages of this study, for his valuable contribution to this thesis.

Finally, I would like to express my appreciation to my wife Eylem, for her endless support and patience. I thank to my little son Utku for giving me happiness and joy.

ABSTRACT

PEOPLE DETECTION IN CLUTTERED SCENES

In this thesis, we have performed people detection in cluttered scenes. The people search operation in an image is performed by sliding a detection window and converting the content of each window to a feature vector. Dense feature representation of the detection window is obtained by dividing it into overlapping blocks and extracting local features of the blocks. These block features are concatenated to form the combined feature vector of the detection window. Feature vectors are obtained from windows with people and not containing people (negative samples), and used to train a linear SVM classifier.

We have studied various types of features to use for people detection. First, we have performed people detection using Histogram of Oriented Gradients (HOG) using various combinations of values of HOG feature extraction parameters like block sizes, gradient operators, HOG bin numbers and normalization methods. In addition to HOG features, we have also studied other features like Gabor energies, block orientation vectors, skin color, projection profiles and cluster distances. In order to increase the performance and the reliability of the detection algorithm, various fusion techniques are applied at data, feature and decision levels. For example, HOG based detector scores are fused with Gabor based detector scores and improved detection scores are obtained. Also, same type detectors are varied by changing detector parameters and the detection scores of these detectors are combined. The performance of the algorithms is measured using different parameters and configurations, and results are compared using Detection Error Tradeoff plots.

ÖZET

KARMAŞALI SAHNELERDE İNSAN BULUNMASI

Bu tez çalışmasında karmaşalı sahnelerde insan bulma çalışması gerçekleştirilmiştir. İnsan arama işlemi imge içinde uygun boyutta bir arama penceresi örtüşmeli şekilde kaydırılarak ve her bir pencere içeriği belli bir öznitelik vektörü ile ifade edilerek yapılmıştır. Arama penceresi de yoğun öznitelik gösterimi elde etmek amacıyla örtüşen hücrelere bölünmüş ve her hücre için yerel öznitelikler elde edilmiştir. Bu yerel öznitelikler art arda eklenerek arama penceresine özgü öznitelik vektörü elde edilmiştir. İnsanlı ve insansız arama pencerelerinin öznitelik vektörleri ile doğrusal Destek Vektör Makinesi (DVM) sınıflandırıcısı eğitilmiştir

İnsan bulmada kullanılmak üzere çeşitli öznitelikler ve yöntemler üzerinde çalışılmıştır. Öncelikle Yönlü Gradyan Histogramı (YGH) betimleyicileri kullanılarak imgeler içindeki insanlar saptanmaya çalışılmıştır. YGH'ya dayalı insan bulucu algoritma hücre büyüklüğü, gradyan işleci, histogram sele sayısı ve düzgeleme yöntemi gibi parametrelerin farklı değerleri ile sınanmıştır. YGH'a ek olarak Gabor enerjileri, hücre yön vektörleri, ten rengi, izdüşüm profilleri ve topak uzaklıkları gibi başka öznitelikler üzerinde de çalışılmıştır. İnsan bulucu algoritmaların başarımını ve güvenilirliğini artırmak amacıyla farklı insan bulucuların sonuçları tümleştirilmiştir. Tümleştirme işlemi aynı tipte olan, ancak farklı parametreler kullanılarak tasarlanmış insan bulcuların sonuçları üzerinde de uygulanmıştır. İnsan bulucu algoritmaların başarımları Sezim Hata Ödünleşim çizimleri kullanılarak karşılaştırılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

$\mathbf{w}$	Normal vector perpendicular to SVM hyperplane
$\mathbf{x}_i$	p-dimensional real vector
y_i	Class of the vector $\mathbf{x}_i$ (either 1 or -1)
C	SVM penalty factor
$\mathbf{v}_{ij}$	Feature vector of the cell at i^{th} vertical and j^{th} horizontal position
$\mathbf{V}$	Feature vector of the detection window
h_{cell}	Height of a cell
w_{cell}	Width of a cell
H_{win}	Height of the detection window
W_{win}	Width of the detection window
w_{cell}	Width of a cell
x_s	Horizontal cell step size
y_s	Vertical cell step size
BB_{dt}	Bounding box of the detection window
BB_{gt}	Bounding box of the ground truth
$\mathbf{G}_X$	Horizontal gradient of the image
$\mathbf{G}_Y$	Vertical gradient of the image
$\mathbf{G}$	Norm of the gradient of the image
f	Radial frequency of the Gabor filter
ξ_i	Degree of SVM misclassification
$\mathbf{\Theta}$	Orientation of gradients
$\psi_{f,\theta}$	2-Dimensional Gabor filter
θ_n	Orientation of the Gabor filter
λ	Wavelength of the Gabor filter
CCTV	Closed-Circuit Television
DET	Detection Error Tradeoff
FPPI	False Positive Per Image
HOG	Histogram of Gradients

LSUV	Linear Scaling to Unit Variance
LSUR	Linear Scaling to Unit Range
OCR	Optical Character Recognition
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
SIFT	Scale Invariant Feature Transformation
SVM	Support Vector Machine
VCA	Video Content Analysis

1. INTRODUCTION

Detection of people in images and video sequences has attracted considerable attention in recent years. It is a very challenging task, but it has a wide variety of applications, such as video surveillance, content-based image retrievals and driver-assistance systems

1.1. Applications of People Detection

In the field of image analysis, face detection is already very common. Most digital photo cameras use it to correctly focus and to adjust the white balance for portrait images. For other types of images, this is not yet possible and person detection can help to extend the possibilities. Person detection can also be used for a smarter image search. Automatically retrieved knowledge about the presence or absence of people would be valuable information.

Increasing interest in robust person detection algorithms is also coming from the visual surveillance community. The large number of surveillance cameras makes the screening of the footage both time consuming and expensive. In fact, it is virtually impossible to monitor all cameras in a non-automatic fashion. Person detection and tracking algorithms can help to watch an area and alert security guards or annotate the footage.

Another field of application in this context is behavior analysis. Commercial applications, for example, detect persons in front of a shop window and determine whether customers stop and look at the products. For human-robot interaction it is necessary to infer the exact position of a person in order to avoid collisions or to be able to properly assist the person.

People detection is also used in some modern CCTV cameras, where the control of the cameras is linked to a computer so that people can be tracked semi-automatically. The

technology that enables this is often referred to as VCA (Video Content Analysis), and is currently being developed by many technological companies around the world. The current technology enables the systems to recognize if a moving object is a walking person, a crawling person or a vehicle. What the system can do is basically identifying where a person is, whether he is moving and whether he is a person or for instance a car. Based on this information the system developers implement features such as blurring faces or "virtual walls" that block the sight of a camera where it is not allowed to film. It is also possible to provide the system with rules, such as "sound the alarm whenever a person is walking close to that fence" or in a museum "set the alarm if a painting is taken down from the wall".

VCA can also be used for forensic after the video has been recorded. It is then possible to search for certain actions within the recorded video. For example, if you know a criminal is driving a yellow car, you can set the system to search for yellow cars and the system will provide you with a list of all the times where there is a yellow car visible in the picture. These conditions can be made more precise by searching for "a person moving around in a certain area for a suspicious amount of time", for example if someone is standing around an ATM machine without using it.

In crowds the system can be used to detect anomalies, for instance a person moving in the opposite direction to the crowd, which might be a case in airports where passengers are only supposed to walk in one direction out of a plane, or in a subway where people are not supposed to exit through the entrances.

VCA also has the ability to position people on a map by calculating their position from the images. It is then possible to link many cameras and track people through a building; this can also be done for forensic purposes where a person can be tracked between cameras without anyone having to analyze many hours of film.

It is also possible to integrate face recognition capability to people detection systems. Then it becomes possible to determine a person's identity without alerting him that his identity is being checked and logged. The systems can check thousands of faces in a database in a second. The combination of CCTV and facial recognition has been tried as a

form of mass surveillance, but has been ineffective because of the low discriminating power of facial recognition technology and the very high number of false positives generated. This type of system has been proposed to compare faces at airports and seaports with those of suspected terrorists or other undesirable entrants.

Nowadays, in order to take active safety measures for pedestrians, visual pedestrian detection from a moving vehicle is also a popular research area. In the past, most of the measures against accidents with pedestrian were passive. Passive pedestrian safety measures involve vehicle structures (e.g. bonnet, bumper) that expand during collision in order to minimize the impact of the pedestrian leg or head hitting the vehicle. Passive pedestrian safety measures are constrained by the laws of physics in terms of ability to reduce collision energy and thus injury level. The aim of active video-based driver assistance systems is to detect dangerous situations involving pedestrians ahead of time, allowing the possibility to warn the driver or to automatically control the vehicle (e.g. braking). Such systems are particularly valuable when the driver is distracted or visibility is poor. Vision-based pedestrian detection is already a difficult problem. In addition to those difficulties, for a moving vehicle it is a more challenging task since usually pedestrians stand typically far away from the camera and therefore appear rather small in the image, at low resolution. Also there is no possibility to use simple background subtraction methods [1] (such as those used in surveillance applications) to obtain a foreground region containing the people. And finally, there are hard real-time requirements and stringent performance criteria for such systems.

1.2. Challenges in People Detection

People detection is a challenging task since appearance of people can vary greatly. The source of these appearance variations are manifold and will be explained briefly.

1.2.1 Variable Appearance and Clothing

Unlike some other objects with a predefined appearance, people do not share a common appearance. They may wear clothing with different color and texture. A man

wearing a black suit will appear very different from a woman wearing colorful dress. Therefore, pixel values itself is not a promising feature to characterize a human. Body shape information can be a better option than pixel values. However, there are also other variation sources which alter body shapes as well. For example, handbags or backpacks may cause changes in the appearance of the body shape and therefore make the detection process more difficult. Another source of variation is caused by physical properties of people. For example, a fat and short person may have a very different appearance than a slim and tall person.

Figure 1.1. Variable people appearances

1.2.2 Different Scales

Variations in scales of people also make the detection task more difficult. Most detection algorithms learn the object model on a fixed training scale. The training model contains information only from this scale and misses details which would be available in a higher resolution. Therefore, training scale should be selected carefully so that it is enough to capture the important details of the target objects. However, detection of small objects is still a problem since target object details may not be present at lower resolutions.

Figure 1.2. People at different scales

1.2.3 Body Articulations

Body articulations make detection of pedestrians particularly challenging as the object model has to be flexible enough to allow deformations of the shape. For example, while a pedestrian is walking, arm and leg positions get continuously changed and the resulting appearance of the pedestrian gets different at each time. Therefore, varying body articulations make pedestrian detection particularly challenging compared to rigid object detection task.

Figure 1.3. Different body articulations

1.2.4 Background Clutter

Background structures can also have adverse effect on the people detection task. These structures may accidentally be similar to a human's shape and therefore distract the detection algorithm. Most challenging background structures are highly textured regions and vertical edge structures. Posters, dummies and other human-like objects at the background may also distract the detection algorithm. Motion blob segmentation and background modeling approaches are often used to eliminate the effect of background clutter. However, these approaches are simple and effective only when the camera is stationary.

Figure 1.4. Background clutter

1.2.5 Viewpoint Changes

In-plane and out-of-plane rotations may cause the object viewpoint to change. Usually in-plane rotations are easier to handle since they do not affect which part of the object is visible. For the people detection task, in-plane rotation does not play an important role. Out-of-plane rotations may make the detection task very difficult if the aspect ratio of the object is significantly dependent upon the position of the camera. However, for most of the pedestrian detection tasks, the aspect ratio is less dependent on viewpoint and as the scale gets coarser and coarser, the influence of viewpoint differences decreases. For example, at a very coarse scale, sometimes it may be hard to tell whether a person is heading left or right.

Figure 1.5. Pedestrians from different viewpoints

1.2.6 Occlusion

Occlusions of body parts often occur when people overlap, carry accessories or when they are stay behind other objects. Occlusion is a very difficult problem to tackle especially for holistic people detection algorithms. Many object detection algorithms can not deal effectively with partial occlusion. For global approaches it is unclear how to deal with the missing data and how the detection score is influenced by the occluded portion of the object. The missing body parts often result in detections with poor recognition scores and these scores are often outweighed by false positive detections on the background. Parts-based detection approaches are more successful than holistic detection approaches on dealing with partial occlusions.

Figure 1.6. Pedestrians with partial occlusions

1.2.7 Illumination and Shading

Illumination changes and shading are one of the major difficulties for people detection applications. Average pixel values can be rather bright or dark depending on the lighting conditions. Overexposure or underexposure may hide certain details of the pedestrian as shown in Figure 1.7 below.

Figure 1.7. Illumination and shading effects

1.3. Outline

In Chapter 2, the main people detection approaches and some of the previous works will be summarized. In Chapter 3, we present an overview of our people detection algorithm, the normalization methods used, Support Vector Machine (SVM) classifier and our evaluation methodology. In Chapter 4, our training and test pedestrian image dataset is introduced. In Chapter 5, details of the people detection algorithm based on Histogram of Oriented Gradients (HOG) and the results of the algorithm are discussed. In Chapter 6, details of the people detection algorithm based on Gabor energies and the results of the algorithm are presented. In Chapter 7, skin color detection and its complementary role is explored. In Chapter 8, we investigate if projection profiles can be used to improve people detection. In Chapter 9, we explore an alternative to HOG, called combined orientation of block gradients and test its detection performance. In Chapter 10, people detection based on cluster distances will be presented. In Chapter 11, summary and conclusion of the thesis will be provided and possible feature works will be discussed.

2. BACKGROUND ON PEOPLE DETECTION

In this Chapter, people detection approaches and some of the previous works on people detection will be presented. Person and pedestrian detection has received much attention in the computer vision literature. Historically, many pedestrian recognition approaches have been developed in the field of video surveillance or intelligent vehicles. More recently, contributions also come from other communities.

There are also some survey and benchmarking studies recently published on pedestrian detection. Dollar and Wojek [2] have introduced the Caltech Pedestrian Dataset, which contains richly annotated video, recorded from a moving vehicle, with challenging images of low resolution and frequently occluded people. On this data set, they have benchmarked several promising detection systems, providing an overview of state-of-the-art performance and a direct, unbiased comparison of existing methods. They have also analyzed common failure cases in order to identify future research directions for the field.

Recently, Enzweiler and Gavrilla [3] have performed an extensive survey on pedestrian detection. They have presented main components of a pedestrian detection system and the underlying models. They have also presented several state-of-the-art pedestrian detection algorithms and performed detection experiments using these algorithms on a test sequence consisting of 21,790 images. They have presented the detection results using DET curves by plotting Detection Rate vs. False Positives per Frame. Typical false positives of the systems they have experimented are shown in Figure 2.1 below. We can see that most errors occur in local regions with strong vertical structures.

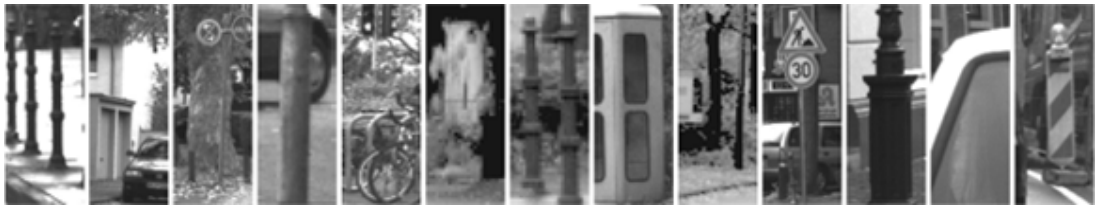

Figure 2.1. Typical false positives of people detection systems

2.1. People Detection Approaches and Related Works

There are numerous detection algorithms proposed in the literature. When we consider detection in a single image, these algorithms can be grouped into two leading approaches. The first approach uses a single detection window analysis and the other uses a parts-based approach. Within each approach, there are different features and different classifiers proposed in the literature. These approaches will be introduced briefly.

2.2.1 Holistic Detection Approach

In this approach, detectors try to find the whole human body within the image. Some methods extract global features like edge templates [4], some others use local features like HOG descriptors. For example, Gavrila and Philomin [5] extracted edges in the image and matched them with shape templates. This method is an example of holistic detector using global features.

An example of holistic detector using local feature is defined in [6], where Dalal and Triggs used the single window approach with a dense HOG representation which appeared to be successful for object representation. This method uses the fact that the shape of an object can be well represented by a distribution of local intensity gradients or edge directions. They divided the image in small spatial cells and calculated the histogram of edge orientations over all pixels within a cell.

Papageorgiou and Poggio [7] used Haar wavelets of three different orientations as feature descriptors and employed SVM as a classifier. In the traditional wavelet transform, wavelets do not overlap and they are shifted by the size of the support of the wavelet. In order to achieve better spatial resolution and a richer set of features, they overlapped wavelets by shifting ¼ size of the support of each wavelet. As a result, they obtained an overcomplete dictionary of wavelet features.

Zhu and Avidan [8] presented an algorithm to significantly speed up people detection using HOG descriptor method. They used HOG descriptors in combination with the cascade of rejecters algorithm normally applied with great success to the problem of face detection. Also, instead of using only blocks of uniform size, they introduced blocks that vary in size, location, and aspect ratio. In order to isolate the blocks best suited for people detection, they applied the AdaBoost algorithm to select those blocks to be included in the rejecter cascade. In their experimentation, their algorithm achieved comparable performance to the original Dalal and Triggs [6] HOG based algorithm, but operated at speeds up to 70 times faster.

Suard and Rakotomamonjy [9] 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 normal SVM 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 stereovision

Holistic detection approach can easily be affected by complex background clutter and produce high rate of false alarms. Another drawback of a holistic detector is that it may fail to detect people when occlusion occurs. In order to detect highly occluded people, parts-based detectors [10], [11] are mostly preferred.

2.2.2 Parts-Based Detection Approach

In this approach, people are modeled as a collection of body parts like head, legs, arms etc. At the first stage, the detector searches the image for human body parts and then these part hypotheses and the relationships between the parts are joined to form the best assembly of whole human body hypotheses. In order to find body parts, local features are extracted from the image. Edgelet features [12], [13] and the orientation features [14] can be given as examples of local features used for parts detection. Parts-based approach may

provide more robustness to occlusion; however it may not be suitable for detecting low resolution people.

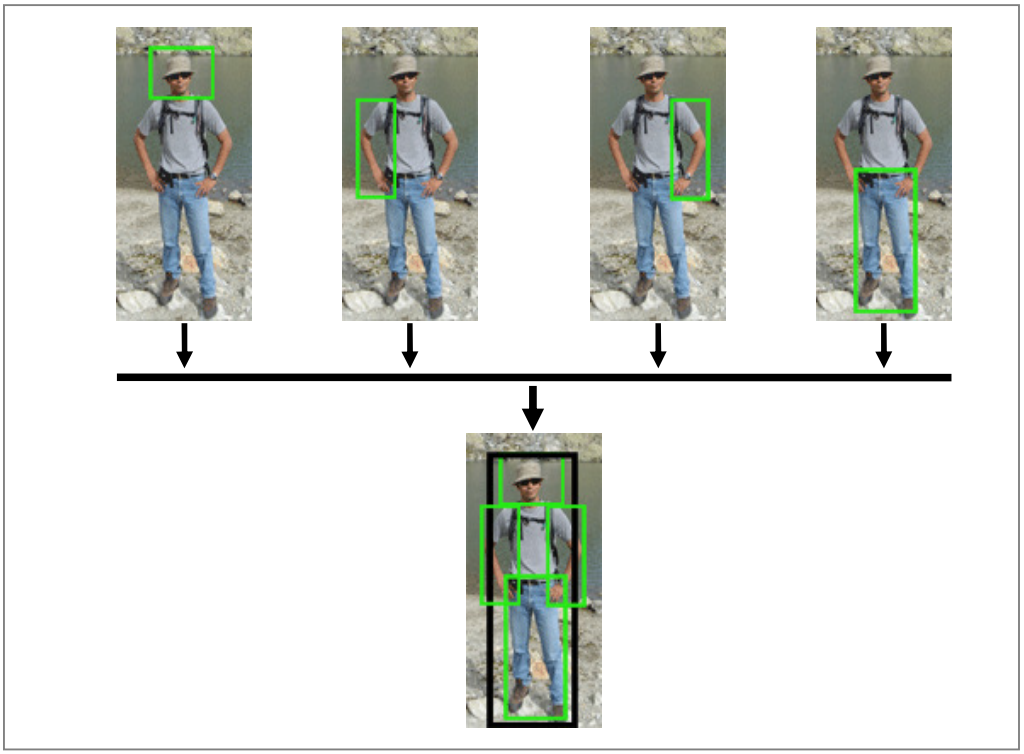

Figure 2.2. Diagrammatic description of parts-based detectors

Wu and Nevatia [12] used multiple part detectors for full body, head-shoulder, torso and legs and presented a novel edge-based feature called edgelet. They have shown that edgelet features perform better than Haar-wavelet features in their boosting framework. In their method, part hypotheses are aggregated in a probabilistic formulation with a Gaussian assumption.

2.2.2 People Detection Approaches in Video

When people detection is performed on video sequences, it is possible to use the motion information as a strong cue in addition to human appearance based features. The inclusion of motion features increases the performance by an order of magnitude relative to a similar static detector. If the camera is fixed, then background subtraction is another commonly used approach for people detection in video sequences. The adoption of a static

camera greatly simplifies the problem because only the presence of motion already provides a strong cue for people presence.

A person detector that incorporates motion descriptors has been proposed by Viola [15]. They build a human detector for static-camera surveillance applications, using generalized Haar wavelets and block averages of spatiotemporal differences as image and motion features and a computationally efficient rejection chain classifier trained with AdaBoost feature selection.

Gavrila and Munder [16] have proposed a multi-cue vision system for the real-time detection and tracking of pedestrians from a moving vehicle. The detection component involves a cascade of modules, each utilizing complementary visual criteria to successively narrow down the image search space, balancing robustness and efficiency considerations. They have tightly integrated the consecutive modules: (sparse) stereo-based ROI generation, shape-based detection [17], texture-based classification and (dense) stereo-based verification. For example, shape-based detection activates a weighted combination of texture-based classifiers, each attuned to a particular body pose.

3. OVERVIEW OF THE DETECTION ALGORITHM

In order to detect people in an image, the most relevant features need to be extracted from the image. There are various approaches used for image feature extraction. One of them is Sparse Local Representation, which is based on local descriptors of relevant local image regions. In this approach, a sparse set of salient image points which are also called key points are determined first. Then, local features at these key points are extracted and used in detection. Therefore, the overall performance of the detector is highly dependent on the reliability and accuracy of the key point finder. In this approach, mostly Scale Invariant Feature Transformation (SIFT) and shape context descriptors are used for key point detection. As an example, Agarwal and Roth [18] have built a vocabulary of parts that can be used to represent target objects, then transformed and represented each image in terms of parts from this vocabulary.

A second approach, which is used in this project, is “Dense Representation”. In this representation, image features are extracted pixel-wise over the entire image or detection window. Then these features are combined into a high-dimensional descriptor vector which can be used for classification or detection. Usually, this representation is based on image intensities, gradients or higher order differential operators.

3.1. People Search in Images

In order to detect people in images, we use a detection window. The size of the detection window has the same size as images in the training database. We slide this window over the whole image in an overlapping manner. At each position, dense feature representation of the window is obtained and this obtained feature vector is fed into the learned classifier for “people / non-people” labeling.

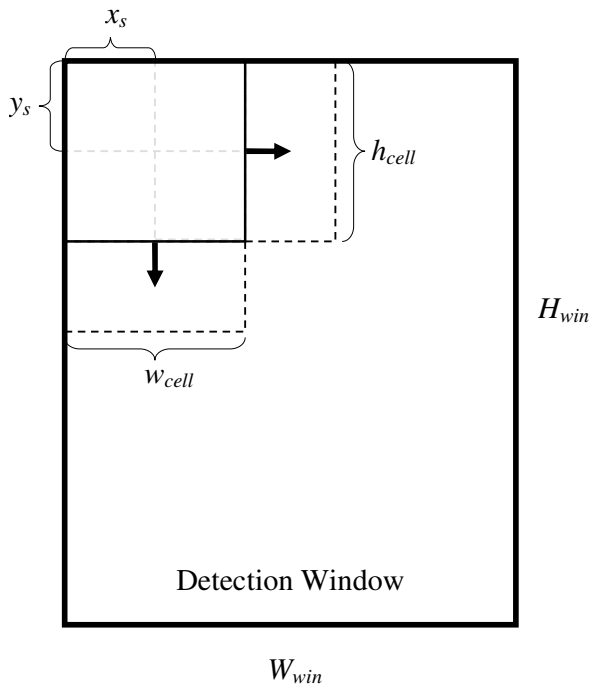

Figure 3.1. Dense representation of detection window

Dense feature representation of a detection window is obtained as described in Figure 3.1. The detection window is divided into overlapping cells and for each cell, local features are extracted. Then these local features are concatenated to form the feature vector of the detection window. Let $\mathbf{v_{ij}}$ be the feature vector of the cell at the i^{th} vertical and j^{th} horizontal cell position, then the feature vector $\mathbf{V}$ of the detection window is obtained as follows:

Definitions:

M, N : Number of horizontal and vertical cell positions

x_s, y_s : Horizontal and vertical cell step sizes in pixels

w_{cell}, h_{cell} : Width and height of cells in pixels

W_{win}, H_{win} : Width and height of the detection window in pixels

$$M = floor(\frac{W_{win} - w_{cell}}{x_s}) + 1 \qquad N = floor(\frac{H_{win} - h_{cell}}{y_s}) + 1$$

where *floor(x)* operation returns the value of *x* rounded downwards to the nearest integer

If $\mathbf{v}_{ij}$ is a D dimensional feature vector, then the dimension of the feature vector $\mathbf{V}$ of the detection window can be calculated as $N \times M \times D$ as follows:

$$\mathbf{V} = \{[\mathbf{v}_{11}\mathbf{v}_{12}...\mathbf{v}_{1M}][\mathbf{v}_{21}\mathbf{v}_{22}...\mathbf{v}_{2M}]...[\mathbf{v}_{N1}\mathbf{v}_{N2}...\mathbf{v}_{NM}]\}$$

In this project, 48x112 size detection windows are used. For example, if we use 12x12 size cells, $D = 9$, $x_s = 6$ and $y_s = 6$, we obtain a 1071 dimensional feature vector $\mathbf{V}$.

$$M = floor(\frac{48-12}{6})+1 = 7 \qquad N = floor(\frac{112-12}{6})+1 = 17$$

Total number of cells $= MxN = 119$

Dimension of $\mathbf{V}$ $= MxNxD = 1071$

Our training database contains people and non-people images of size $W_{win} \times H_{win}$. Feature vector $\mathbf{V}$ of each training image is extracted and labeled as people and non-people. Then a binary classifier is trained using these labeled vectors. During detection process, we slide the detection window over the whole image, and at each window position, we extract the feature vector $\mathbf{V}$ of the detection window, fed into the binary classifier and obtain people/non-people label for that window position. In this project we have used Support Vector Machine (SVM) as a binary classifier. The details of this classifier will be explained in Section 3.3.

3.2. Normalization of Feature Vectors

Normalization is one of the most important tools utilized by automatic recognition systems to obtain better performance. Ideally, the same range of values for each input feature is desired in order to minimize any bias for one feature over another. Data normalization is especially useful when the input feature values are on widely different scales. Normalization puts numeric feature values on the same scale and prevents features with a large original scale from biasing the solution. Normalization also minimizes the likelihood of overflows and underflows. In addition, Support Vector Machine classifier also requires the normalization of input data.

In this project, we have experimented different normalization methods over the input data on a local scale by normalizing feature vectors of each block or on a global scale by normalizing the whole feature vector of the detection window. The following normalization techniques are applied:

- L1-Normalization: The feature vector **x** is normalized using the following transformation:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_1} \quad \text{where } \mathbf{x} = [x_1 x_2 \ \ldots \ x_n] \text{ and } \|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

- L2-Normalization: The feature vector **x** is normalized using the following transformation:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \quad \text{where } \|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

- Linear Scaling to Unit Variance (LSUV): In this technique, each feature component x is transformed to a random variable with zero mean and unit variance by applying the following transformation:

$$\tilde{x} = \frac{x - \mu}{\sigma}$$

where σ is the sample mean and μ is the sample standard deviation of the feature calculated over the training set.

- Linear Scaling to Unit Range (LSUR): Given a lower bound l and an upper bound u for a feature component x, the following transformation is performed in order to map all x values to [0,1] range.

$$\tilde{x} = \frac{x - l}{u - l}$$

During application of LSUV and LSUR normalization methods, first σ, μ, u and l statistics of training data are calculated, and during detection process, test data is normalized using the statistics of the training data. Let $\mathbf{X}_T$ be the matrix containing the training data feature vectors, this matrix can be expressed as follows:

$$\mathbf{X}_T = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \mathbf{...} \\ \mathbf{x}_p \\ \mathbf{y_1} \\ \mathbf{y_2} \\ \mathbf{...} \\ \mathbf{y}_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1k} \\ x_{21} & x_{22} & \dots & x_{2k} \\ \dots & \dots & \dots & \dots \\ x_{p1} & x_{p2} & \dots & x_{pk} \\ y_{11} & y_{12} & \dots & y_{1k} \\ y_{21} & y_{22} & \dots & y_{2k} \\ \dots & \dots & \dots & \dots \\ y_{n1} & y_{n2} & \dots & y_{nk} \end{bmatrix}$$

where each row is a feature vector of an image in the training data set, $\mathbf{x_i}$ is the feature vector of the i^{th} positive training image, $\mathbf{y_i}$ is the feature vector of the i^{th} negative training image, p is the number of positive training images, n is the number of negative training images and k is the dimension of a feature vector. Then σ, μ, u and l statistics of training data are calculated for each column of $\mathbf{X}_T$ data matrix. As an example, the calculation of $\boldsymbol{\mu}$ is performed as follows:

$$\boldsymbol{\mu} = [\mu_1 \mu_2 ... \mu_k], \quad \mu_j = \frac{1}{p+n}\left(\sum_{i=1}^{p} x_{ij} + \sum_{i=1}^{n} y_{ij} \right) \text{ where } \mu_j \text{ is the mean of } j^{th} \text{ column.}$$

3.3. Support Vector Machine (SVM)

In order to perform people detection, we need a trainable classifier which learns to differentiate between people and non-people patterns. Currently, most of the people detection systems use SVM as a classifier; hence in this project we have also selected SVM as the classifier.

A Support Vector Machine (SVM) is a binary classifier which performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories. SVM has strong regularization properties, which provides the generalization of the model to new data. The quality of generalization and ease of training of SVM is far beyond the capabilities of other traditional methods like neural networks. SVM can model complex, real-world problems such as text and image classification, hand-writing recognition, and bioinformatics and biosequence analysis.

One of the main attractions of using SVMs is that they are capable of learning in high-dimensional spaces with very few training examples. They accomplish this by minimizing a bound on the empirical error and the complexity of the classifier, at the same time. SVM classification is based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. SVM finds the "support vectors" that define the separators giving the widest separation of classes. In practice, this is determined through solving a quadratic programming problem.

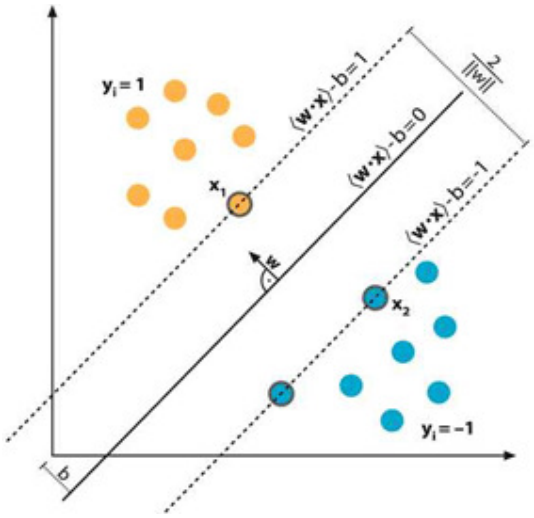

Figure 3.2. Support vector machine

The illustration of SVM is shown in Figure 3.2, where $\mathbf{x_i}$ is a *p*-dimensional real vector, y_i is either 1 or -1, indicating the class to which the point $\mathbf{x_i}$ belongs to and $\mathbf{w}$ is a normal vector which is perpendicular to the hyperplane. Any hyperplane can be written as the set of points $\mathbf{x}$ satisfying

$$\mathbf{w}\cdot\mathbf{x}-b=0$$

We want to choose the $\mathbf{w}$ and b to maximize the margin, or distance between the parallel hyperplanes that are as far apart as possible while still separating the data. These hyperplanes can be described by the equations $\mathbf{w}\cdot\mathbf{x}-b=1$ and $\mathbf{w}\cdot\mathbf{x}-b=-1$

In this project, we have used a linear soft margin SVM provided in [19]. Soft margin SVM has a modified maximum margin idea that allows for mislabeled examples. If there exists no hyperplane that can split the "yes" and "no" examples, the Soft Margin method chooses a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. This method introduces slack variables, ξ_i, which measure the degree of misclassification of the datum $\mathbf{x_i}$. The objective function of SVM is then increased by a function which penalizes non-zero ξi, and the optimization becomes a trade-off between a large margin and a small error penalty. If the penalty function is linear, the optimization problem becomes:

$$\min_{\mathbf{w},\xi}\left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i\right\} \text{ subject to } y_i(\mathbf{w}\cdot\mathbf{x}_i - b) \geq 1 - \xi_i,\ \xi_i \geq 0$$

For Soft Margin SVM, it is critical to choose a proper value for C, the penalty factor. If it is too large, we have a high penalty for non-separable points and we may store many support vectors and overfit. If it is too small, we may have underfitting.

3.4. Evaluation Methodology

In order to compare the performances of different detectors, performances should be quantified and visualized. At each position of the detection window, SVM classifier produces a decision value, where larger values indicate higher confidence that a pedestrian is present at the tested location. For object detection tasks, there are three different approaches: Detection Error Tradeoff (DET), Receiver Operating Characteristics (ROC) and Recall-Precision curves. These curves are drawn by adjusting the SVM decision threshold value. Evaluation starts from the lowest threshold value and then we progressively increase the threshold until we reach the highest possible score. At each threshold point, we calculate the parameters such as number of false positives, recall rate or precision rate. Each evaluated threshold provides a point on the curve.

Traditionally, the ROC curve has been widely used for comparison of detector performances. Generally, false alarm rate is plotted on the horizontal axis, while correct detection rate is plotted on the vertical. However, detection tasks can be viewed as a tradeoff between two error types: missed detections and false alarms. In the DET curve we plot false alarm rate on the horizontal axis and miss rate on the vertical axis. DET curves allow easier observation of system contrasts. There may be some special points on the DET curve, which may correspond to a fixed false alarm rate, fixed missed detection rate or perhaps a performance objective for an evaluation. Confidence intervals or a confidence box around such points can be included.

In people detection literature, some authors compare per-window performances as opposed to the per-image measures frequently used in object detection. The typical assumption is that better per-window scores will lead to better performance on entire images; however, in practice per-window performance can fail to predict per-image performance. Some disadvantages and side effects of per-window analysis are provided in reference [8]. In this project, we have preferred per-image analysis for comparing performances of different detectors. We use DET curves by plotting miss rate on the vertical axis versus false positives per-image (FPPI) on the horizontal axis on log-log scale. On the DET plot, lower curves indicate better performance. In the pedestrian detection benchmarking work presented in [8], 1 False Positive per Image (FPPI) is suggested as a reference point for comparisons of pedestrian detection tasks. Similarly, we have also selected 1 FPPI as a reference point for our comparisons.

3.4.1 Definitions of Terms Used in Evaluation

- *True Positive:* Human is correctly identified as human
- *False Positive:* Non-human is incorrectly identified as human
- *False Negative:* Human is incorrectly identified as non-human (Which is also called as "miss")
- *True Negative:* Non-human is correctly identified as non-human
- $Recall = \dfrac{\text{Number of True Positives}}{\text{Number of Total Positives}}$

- $Precision = \dfrac{\text{Number of True Positives}}{\text{Number of True Positives + Number of False Positives}}$
- *Miss Rate* = 1 – Recall
- *FPPI:* False Positive Per Image

3.4.2 Detection Criteria

A tight bounding box (BB) is drawn for every pedestrian within test images. The position, width and height information of these bounding boxes are recorded as ground-truth data. A detection is accepted as true positive if detection window bounding box BB_{dt} and a ground truth bounding box BB_{gt} overlap sufficiently

$$\frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})} > 0.5$$

3.4.3 Sample Curves

In Figure 3.3, the detection performance of a detector is visualized by three different curve types.

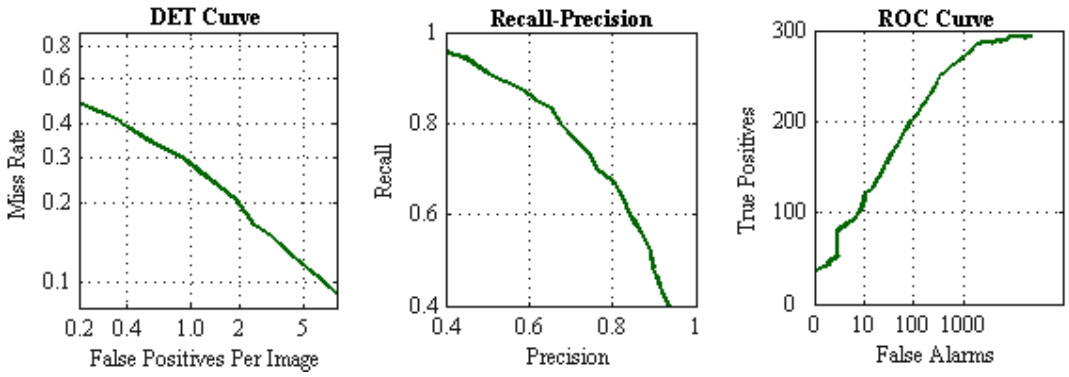

Figure 3.3. Performance evaluation curves for a sample detector

4. IMAGE DATABASE

Our image database is constructed by mostly using the images from MIT Pedestrian Database [20] and INRIA Pedestrian Database [21]. MIT database contains 924 positive images of size 64x128 pixels. MIT database does not contain negative images; that is images from the negative class, all images contain either a front or a back view of a centered, standing person in city scenes. The range of poses is relatively limited, and the people are normalized to have approximately the same size in each image. The subjects are always upright and standing. Each image is scaled to the size 64x128 and aligned so that the person's body is in the center of the image. The height of these people is such that the distance from the shoulders to the feet is approximately 80 pixels. In contrast to MIT images, INRIA database contains both positive and negative images in various sizes; furthermore these images contain more variations in terms of body articulations and poses. INRIA database contains 1218 negative and 614 positive images in various sizes.

Our training database contains 3340 positive images and 9404 negative images. This is made up of 924 positive training images from MIT database and the rest extracted from the INRIA database. All images have been scaled to size 48x112. Some examples from the positive training image database are shown in Figure 4.1. INRIA database contains negative images, but their size vary and larger than the size 48x112. Therefore, our negative training images are produced by sampling INRIA negative images with randomly placed 48x112 size windows. Some sample images from the negative training database are given in Figure 4.2

Figure 4.1. Examples from positive training image database

Figure 4.2. Examples from negative training image database

Our test database contains 133 images in various sizes. Most of the images are obtained from INRIA test images and some from Internet. In total, there are 300 people in our test images. The bounding boxes of people in test images are set manually as ground-truth human positions. Some sample images from the test image database are given in Figure 4.3

Figure 4.3. Examples from the test image database

Test database contains many variations in terms of background clutter, occlusion, scale differences, viewpoint, illumination, shading and body articulations. Some examples of these variations can also be seen from the sample images given in Figure 4.3. Although the scales of test images are aligned so that the size of people becomes close to images in the training database, there are still slight size variations. As evidence of these variations, the histogram of widths and heights of bounding boxes of people in test images are given in Figure 4.4 below.

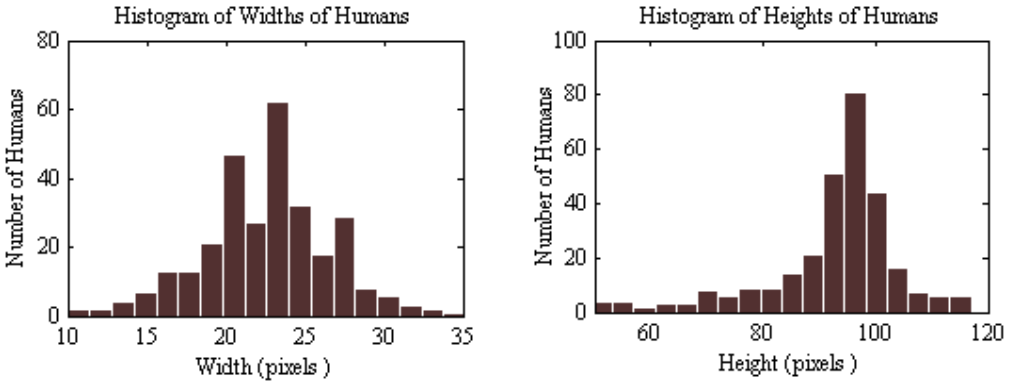

Figure 4.4. Histograms of widths and heights of bounding boxes of people in test images

5. PEOPLE DETECTION USING HOG DESCRIPTORS

For effective people detection, we need a feature set that will provide high inter-class variability and low intra-class variability. In this approach, we used HOG descriptors within test windows as features and then applied the detection process outlined in Chapter 4.

5.1. Histogram of Oriented Gradients (HOG)

The main idea underlying the Histogram of Oriented Gradients descriptor is that object appearance and shape within a local image can be described by the distribution of intensity gradients or edge directions. This descriptor can be implemented by dividing the image (the detection window) into appropriately dimensioned cells, and for each cell compiling a histogram of gradient orientations. The combination by concatenation of these histograms then becomes the feature vector of the detection window.

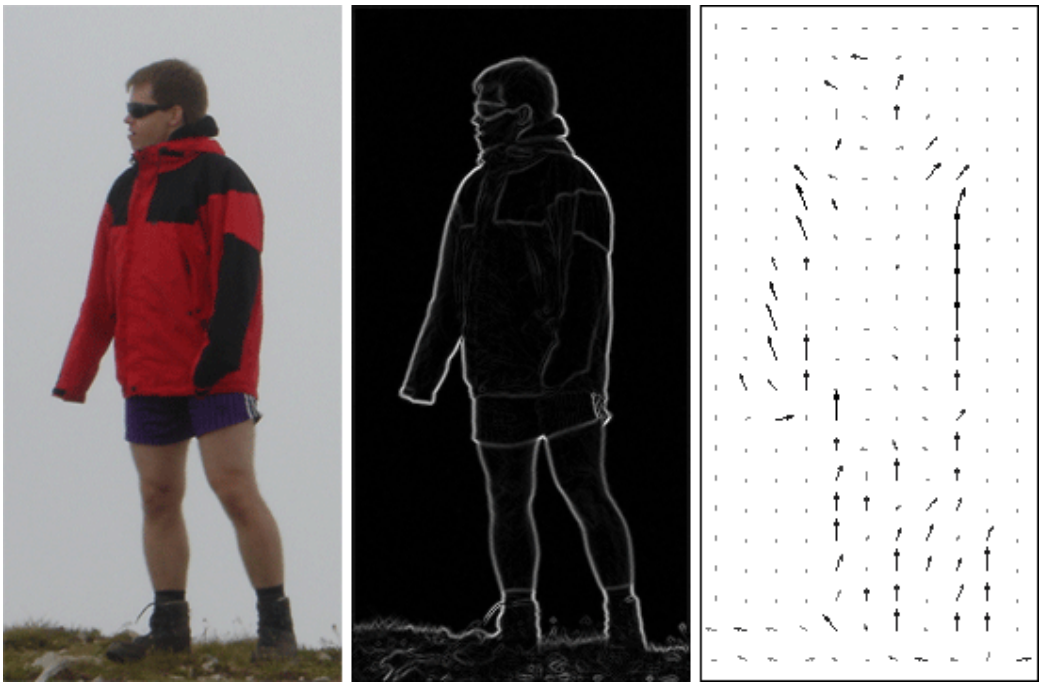

Figure 5.1. Original RGB image, gradient image and gradient orientations of blocks

In Figure 5.1, a sample color image is first converted to grayscale, then its gradients are computed and finally gradient magnitude and orientation of each pixel are calculated. The rightmost image shows the main orientation of each 16x16 cell and the length of the orientation vector is proportional to the gradient norm.

Overview of HOG feature extraction and detection process is described in Figure 5.2. In the following sections, the steps in this chain will be explained in more detail.

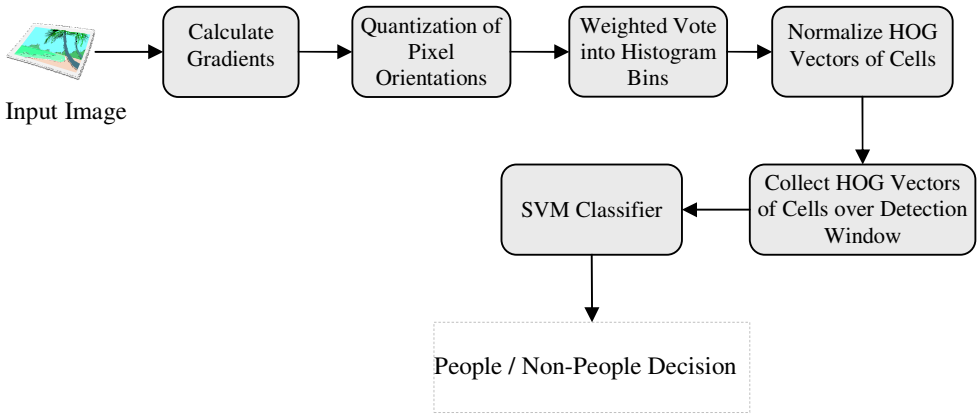

Figure 5.2. Overview of HOG based people detection algorithm

5.1.1 Gradient Calculation

First, horizontal and vertical gradients of the image are calculated. Two types of gradient operators are used in this project. In the first type, [-1 0 1] and $[-1\ 0\ 1]^T$ operators are used. In the second type, Sobel gradient operators are used, as shown in Figure 5.3. The input image is convolved with these masks to obtain gradient images $\boldsymbol{G_X}$ and $\boldsymbol{G_Y}$.

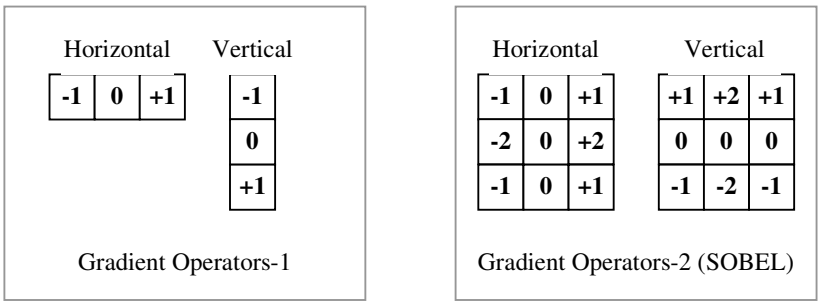

Figure 5.3. Gradient operators used in the project

The gradient norm and gradient orientations of image pixels are calculated using $\boldsymbol{G_X}$ and $\boldsymbol{G_Y}$ as follows:

$$|\mathbf{G}| = \sqrt{\mathbf{G}_X^2 + \mathbf{G}_Y^2}\ , \qquad \boldsymbol{\Theta} = \tan^{-1}\left(\frac{\mathbf{G}_Y}{\mathbf{G}_X}\right)$$

5.1.2 Quantization of Pixel Orientations and Voting into Histogram Bins

Orientation histogram bins are determined by evenly dividing the 0°-180° range or the 0°-360° range, depending on whether the gradient is "unsigned" or "signed". For unsigned gradients, we use the 0°-180° range. For example, when we use 0°-180° range, the 225° and 45° orientations are considered to be the same and the pixel's orientation is put into the histogram bin of 45°. More often the 0°-180° range is preferred since it results in better overall detection performance. As an example, if we select the number of orientation histogram bins as 9 and use the 0°-180° angle range, then we obtain orientation bins 20°apart as described in Figure 5.4. For example, 5th bin corresponds to angles between 80° and 100°.

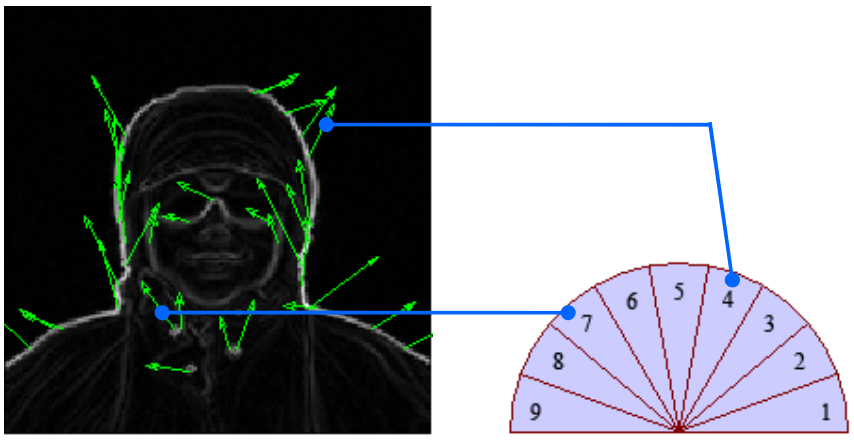

Figure 5.4. Quantization of gradients orientations for the choice of 9 bins

The next step is to estimate the orientation histogram for each block in the detection window. These blocks are made to overlap to obtain a denser HOG representation from the detection window. We used weighted voting scheme for constructing histogram of gradient orientations. Each pixel's vote is weighted by its gradient norm. As an illustration, the image in Figure 5.5 is divided into 16x16 blocks and histogram of gradient orientations is obtained for each block. As it can also be seen from the figure, histogram values of low gradient blocks are very small compared to blocks where edges are present.

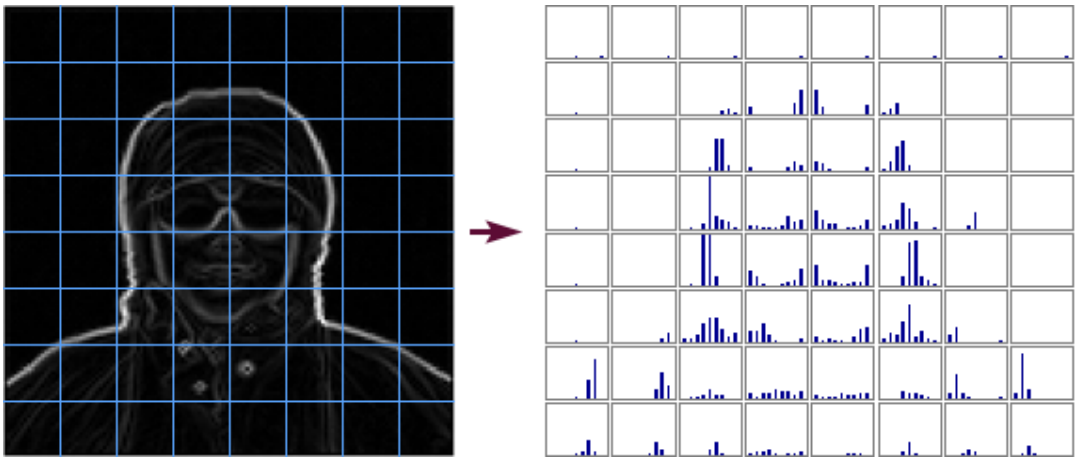

Figure 5.5. Histogram of gradient orientations for 16x16 image blocks

5.1.3 Normalization of HOG Vectors

In this project, we have experimented with different normalization methods of HOG histograms. First, one must decide whether to normalize HOG vectors of each block or on a global scale by normalizing the whole HOG feature vector within the detection window. Suppose that there exists *N* blocks in a detection window and let *k* be the HOG bin count, then feature vector **v** of the detection window can be expressed as follows:

$$\mathbf{v} = [\mathbf{x_1 x_2 \ldots x_N}] \text{ where } \mathbf{x_i} = [x_1 x_2 \ \ldots \ x_k] \text{ is the HOG vector of i}^{th} \text{ block}$$

Normalization techniques can be applied either on vectors $\mathbf{x_i}$ for each block separately or on the whole feature vector **v** of the detection window. Whether block-based or window-based we have compared the following four types of normalization: L1 normalization, L2 normalization, LSUV and LSUR. The effect of normalization techniques on the overall recognition performance is presented in section 5.2.

5.2. Detection Results using HOG Descriptors

In this section, we present people detection performance with HOG descriptors as well as the effect of different parameter values and techniques on the detection performance. Numerous detection experiments are performed using various combinations of block sizes, HOG bin numbers, block overlap ratios, normalization techniques and gradient operators.

5.2.1 Effect of Block Size

We have performed detection experiments using 8x8, 12x12 and 16x16 square blocks each with 50% block overlap ratio. Over both training and test images, we have observed that 12x12 block size performs better than other block sizes. With 8x8 blocks one can scan the image in more detail, but this size produces more false alarms. Also, for smaller blocks, there are more blocks per window, which results in a higher dimensional

feature vector, which slows down the scanning of the image with the detector. On the other hand, 16x16 blocks scan the image in less detail and therefore the miss rate increases.

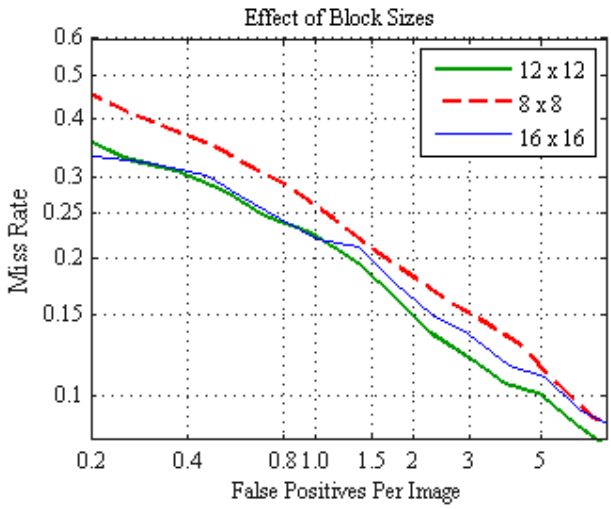

Figure 5.6. Effect of block size on the detection performance

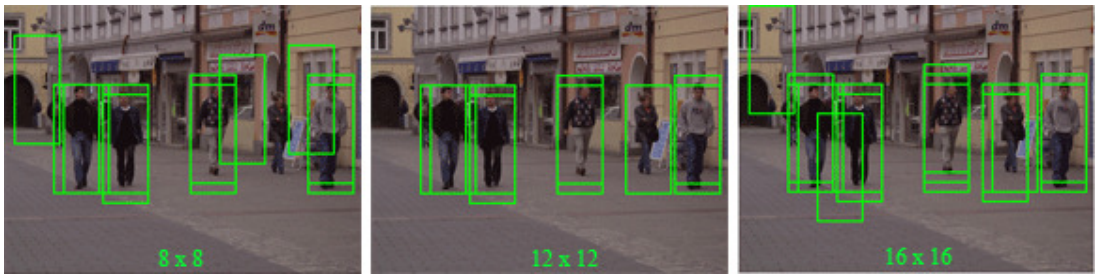

Figure 5.7. Effect of block size on a sample test image at 1 FPPI

5.2.2 Effect of HOG Bin Count

We have performed detection experiments by evenly dividing the 0°-180° range into various numbers of bins. Increasing the number of bins means higher orientation resolution at the cost of fewer hits per bin, hence lower reliability. In our experiments, the choice of 9 HOG bins produced the best detection performance. 9 HOG bins correspond to 20° orientation resolution. The effect of changing HOG bin number on the detection performance is given in Figure 5.8.

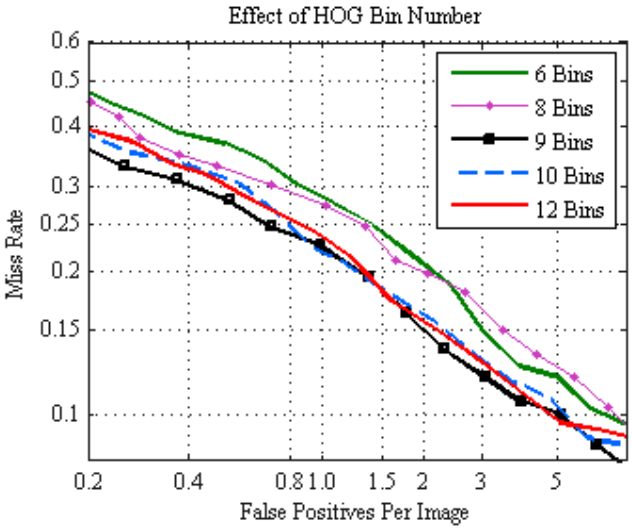

Figure 5.8. Effect of HOG bin number on the detection performance

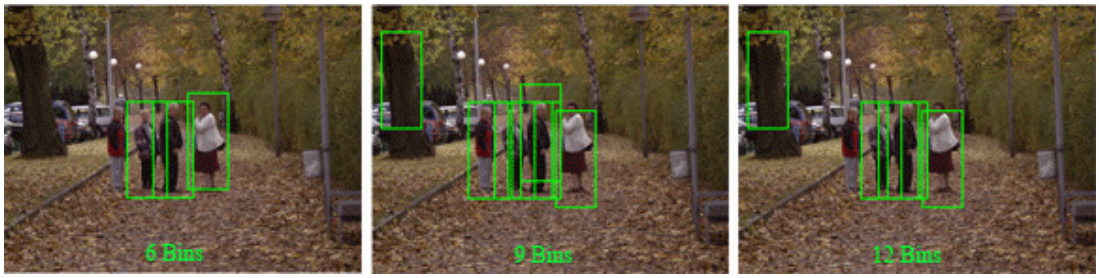

Figure 5.9. Effect of HOG bin number on a sample test image at 1 FPPI

5.2.3 Effect of Normalization

We have applied the normalization techniques described in section 3.2. Normalization can be applied locally for each block or globally for the whole detection window. In our experiments, L1-Normalization technique applied block-wise produced the best detection performance. Effect of normalization on the overall detection performance is given in Figure 5.10.

Both L1-Normalization and "Linear Scaling to Unit Variance (LSUV)" normalization of the HOG vector of the detection window produced very poor detection results; therefore they are not shown in the figure.

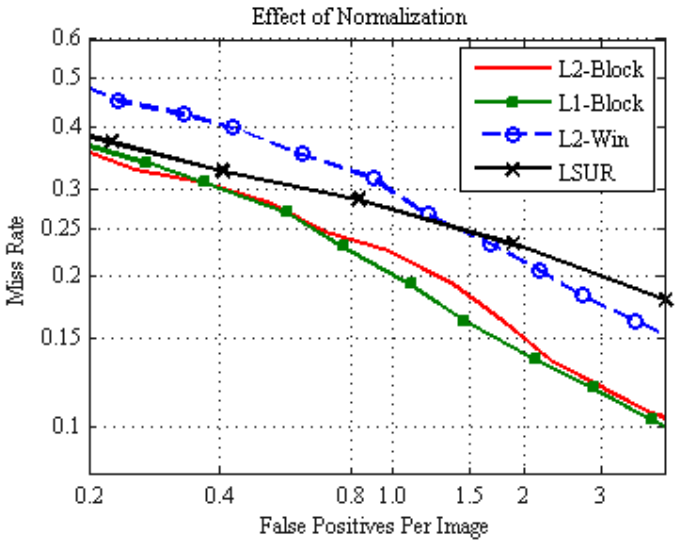

Figure 5.10. Effect of normalization on the detection performance

The meanings of legends in the figure are as follows:

L1-Block: L1-Normalization of HOG vectors of each block

L2-Block: L2-Normalization of HOG vectors of each block

L2-Win: L2-Normalization of the HOG vector of the detection window

LSUR: "Linear Scaling to Unit Range" normalization of the HOG vector of the detection window

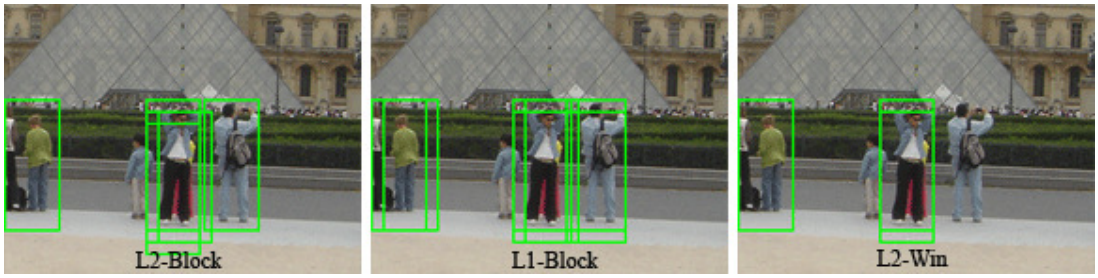

Figure 5.11. Effect of normalization on a sample test image at 1 FPPI

5.2.4 Effect of Gradient Operator

For gradient calculations [-1 0 1] gradient operator or Sobel operators are used. They produced similar results. However, as seen from the Figure 5.12, around 1 FPPI reference point, [-1 0 1] gradient operator performs slightly better than Sobel operator.

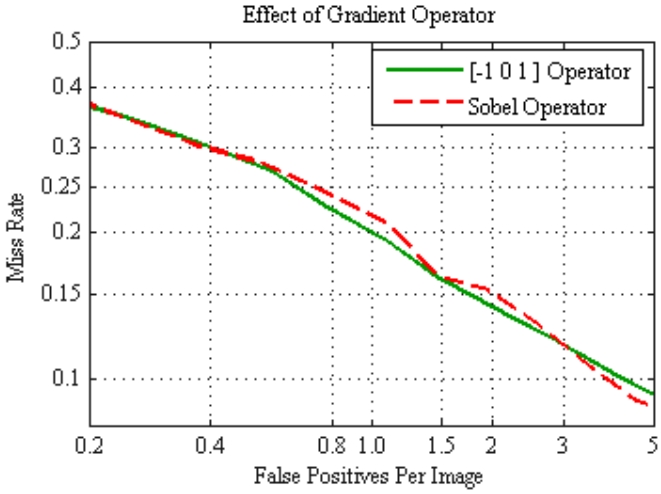

Figure 5.12. Effect of the choice of the gradient operator

5.2.5 Detection Results of the HOG Based Detector

In Figure 5.13, some sample false positive detection results at 1 FPPI by the HOG based detector are presented. From the figure, we can see that generally vertical structures and head-like objects are detected as people.

Figure 5.13. Various false detections by the HOG based detector

In Figure 5.14, some people which cannot be detected by the HOG based detector at 1 FPPI are shown. We can infer from the figure that viewpoint and scale changes, poor illumination, body articulations and partial occlusion are the main problems of the HOG based detector.

Figure 5.14. Various missed detections by the HOG based detector

In Figure 5.15, some detections performed by the HOG based detector at 1 FPPI are shown.

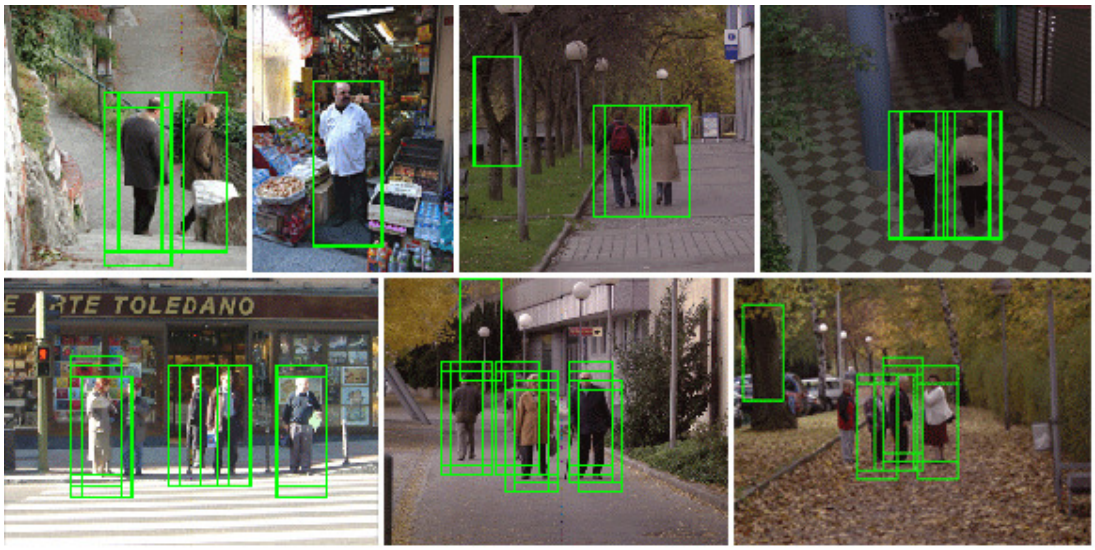

Figure 5.15. Sample detection results by the HOG based detector at 1 FPPI

5.3. HOG based Detection using Data and Score Fusion on R,G,B Channels

We have performed above people detection experiments exploring various combinations of HOG parameters like block sizes, gradient operators, HOG bin numbers,

and normalization methods. In order to increase the performance and the reliability of the detection algorithm we set to explore in the sequel, fusion techniques [22] applied at data, feature and decision levels. The performance of these algorithms is compared using Detection Error Tradeoff plots.

5.3.1 Detection on Grayscale Images

In this approach, first R,G,B color images are converted into grayscale images, then gradient calculation and HOG feature extraction tasks are performed on grayscale images. These results have already been reported in Section 5.2.

5.3.2 Fusion of R,G,B Channel Gradients

In this fusion approach, gradients are calculated on R,G,B channels separately. Then, these gradients are combined using Sum or Max fusion rule. Sum fusion rule is applied on each pixel by calculating equal-weighted sum of R,G,B gradient values. Max fusion rule is applied on each pixel by picking the largest gradient value of R,G,B gradient values. HOG features are extracted on the combined gradients.

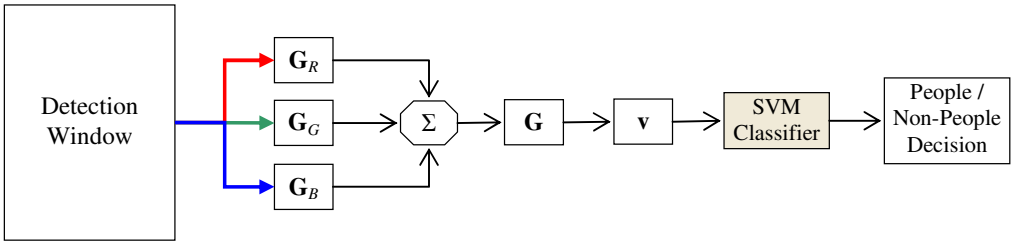

Figure 5.16. Fusion of R,G,B gradients

(**G** *:* Gradient, **v**: HOG Vector)

5.3.3 Fusion of HOG Vectors of R,G,B Channels

In this fusion approach, image gradients are calculated on R,G,B channels separately and HOG vectors are also calculated separately on each channel. The final HOG vector of the detection window is obtained by combining channel HOG vectors using equal-weight sum rule or by cascading channel HOG vectors. In the latter case, the HOG vectors become three times longer.

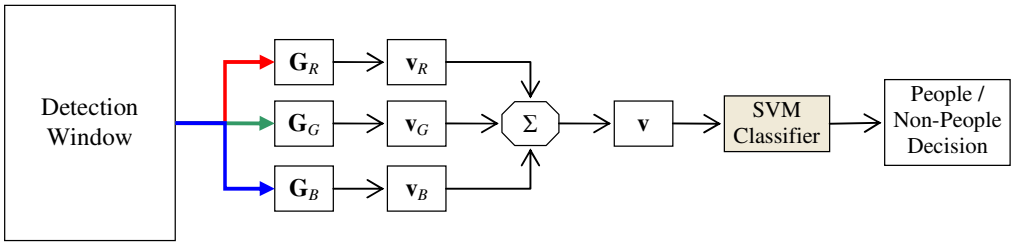

Figure 5.17. Fusion of HOG vectors of R,G,B gradients

(**G** *:* Gradient, **v**: HOG Vector)

5.3.4 Fusion of SVM Scores of R,G,B Channels

In this fusion approach, image gradients are calculated on R,G,B channels separately and HOG vectors are also calculated separately on each channel. For each channel, a separate SVM classifier is trained and during the detection phase, decision scores of these SVM classifiers are combined using various fusion techniques as described in Appendix-A. Among these fusion techniques, we have observed that Sum and Product fusion rules perform better than other fusion rules.

For each detection window, SVM classifier generates a decision value which indicates the distance of the feature vector to the decision boundary. Sum fusion rule is applied by summing these decision values. However, Product fusion rule cannot be applied directly on these decision values. It requires posterior probability of being human in the detection window. SVM tool converts decision values to posterior probabilities as described in [19].

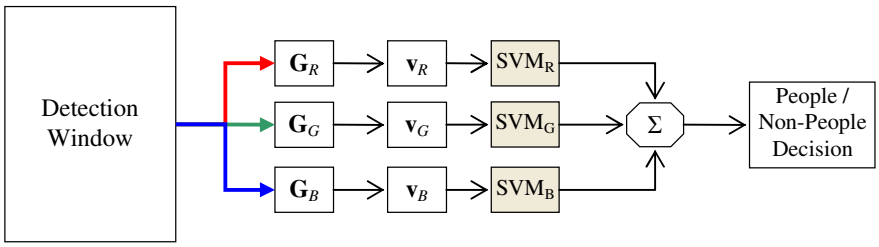

Figure 5.18. Fusion of SVM scores of R,G,B channels

(**G** *:* Gradient, **v**: HOG Vector)

5.3.5 Comparison of R,G,B Channel Fusion Results

The list of compared algorithms is given in Table 5.1 and detection performances of these algorithms are shown in Figure 5.19. We see from this figure that at 1 FPPI reference point, there is at most 3% detection performance difference between the algorithms. Therefore, we can conclude that fusion of color channels does not gain a significant advantage. Comparison of performance results indicate that the best choice is HOG based detector on grayscale images

Table 5.1 Compared R,G,B fusion algorithms

Algorithm	Description
Grayscale	Detection on Grayscale
GradMax	Fusion of R,G,B Gradients using Max Rule
GradSum	Fusion of R,G,B Gradients using Sum Rule
HOGSum	Fusion of HOG vectors of R,G,B Channels
SVMSum	Fusion of SVM Scores of R,G,B Channels using Sum Rule
SVMProd	Fusion of SVM Scores of R,G,B Channels using Product Rule

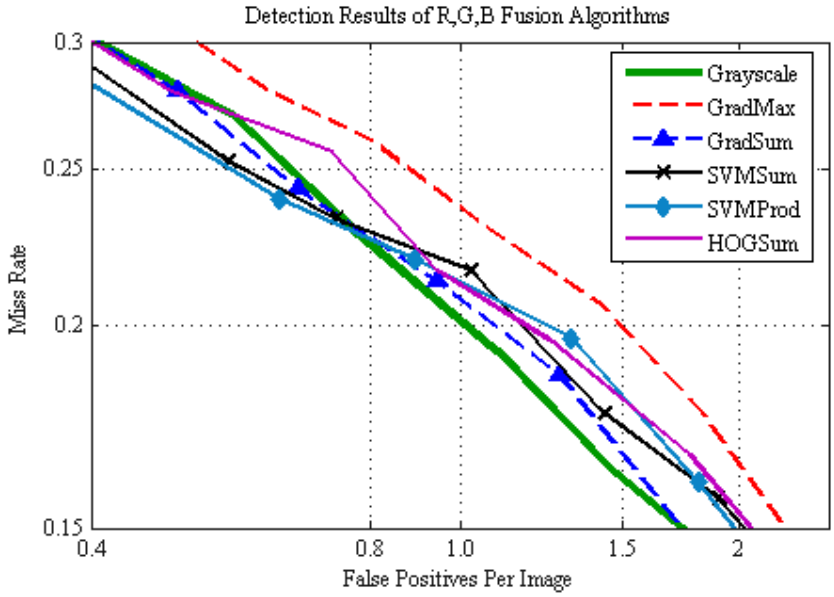

Figure 5.19. Comparison of R,G,B fusion algorithms

5.4. Score Fusion of Different HOG Based Detectors

Fusion operations produce better results when the sources differ from each other. Sometimes weak results may produce a good final result when they are combined properly. In this approach, we have designed different HOG based detectors by adjusting parameters such as block size, HOG bin number and gradient operator. A given test image is analyzed with these detectors and SVM scores of these detectors are combined using Sum or Product fusion rule. In Figure 5.20, the result of fusing three different HOG detectors is shown. At 1 FPPI reference point, the performance of the fusion is 2% better than the best of these detectors.

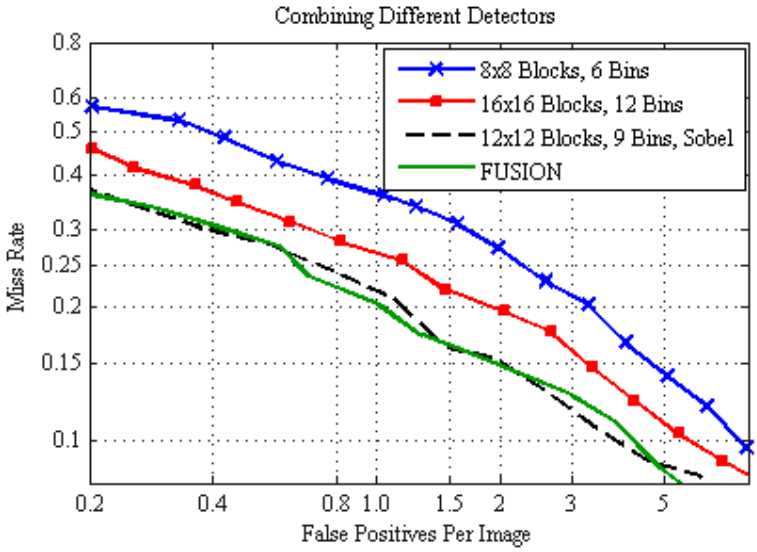

Figure 5.20. Performance of the fusion of different HOG based detectors

6. PEOPLE DETECTION USING GABOR ENERGIES

A Gabor filter applied to images is a bandpass filter with local support to reveal the spatial frequency distribution and achieves an optimal resolution in both spatial and frequency domains. Gabor filters have been successfully applied in various computer vision applications, such as texture segmentation and recognition, face recognition [23], scene recognition, and vehicle detection [24]. The basic issue of Gabor analysis is how to select the parameters of the filters so that they respond mainly to a target object, such as a vehicle or a pedestrian. Accurate detection only occurs if the parameters defining Gabor filters are well selected.

The 2D Gabor filter $\psi_{f,\theta}(x, y)$ can be represented as a complex sinusoidal signal modulated by a Gaussian kernel function as follows:

$$\psi_{f,\theta}(x, y) = \exp\left[-\frac{1}{2}\left\{\frac{x_{\theta_n}^2}{\sigma_x^2} + \frac{y_{\theta_n}^2}{\sigma_y^2}\right\}\right]\exp(2\pi f x_{\theta_n})$$
$$x_{\theta_n} = x\cos\theta_n + y\sin\theta_n$$
$$y_{\theta_n} = -x\sin\theta_n + y\cos\theta_n$$

where σ_x and σ_y are the standard deviations of the Gaussian envelope along the x- and y-dimensions, f is the central frequency of the sinusoidal plane wave and θ_n is the orientation. The rotation of the x-y plane by an angle θ_n will result in a Gabor filter at the orientation θ_n. The angle θ_n is defined by:

$$\theta_n = \frac{\pi}{p}(n-1) \text{ for } n = 1, 2, ..., p \text{ where } p \text{ denotes the number of orientations.}$$

Design of Gabor filters is performed by tuning the filter with a specific band of spatial frequency and orientation by appropriately selecting the filter parameters; the spread of the filter σ_x, σ_y radial frequency f, and the orientation of the filter θ_n. The

variation of θ_n changes the sensitivity to edge and texture orientations. The variation of σ changes "scale" and the variation of *f* changes the sensitivity at high/low frequencies.

The Gabor representation of an image is computed by convolving the image with the Gabor filters. Let *f(x,y)* be the intensity at the coordinate *(x,y)* in a gray scale image, its convolution with a Gabor filter $\psi_{f,\theta}(x, y)$ is defined as:

$$g_{f,\theta}(x, y) = f(x, y) * \psi_{f,\theta}(x, y)$$

The response to each Gabor kernel representation is a complex function with a real part $\Re\{g_{f,\theta}(x, y)\}$ and an imaginary part $\Im\{g_{f,\theta}(x, y)\}$. The magnitude response $\|g_{f,\theta}(x, y)\|$ is expressed as:

$$\|g_{f,\theta}(x, y)\| = \sqrt{\Re^2\{g_{f,\theta}(x, y)\} + \Im^2\{g_{f,\theta}(x, y)\}}$$

In this work, we have used the magnitude response in each image block as its feature set. We have selected a circular support, hence identical values for σ_x and σ_y. In the design of Gabor filters, instead of defining *f*, we preferred using the inverse quantity λ which can be expressed in pixel unit. It is also preferable to use the parameter $\gamma = \frac{\lambda}{\sigma}$ instead of using λ directly so that a change in σ corresponds to a true scale change in the Gabor function. In other words, large support sizes will have low frequencies and small support sizes will be investigated with high frequencies. In Figure 6.1, we illustrate the effect of the variation of parameters on the shape of the Gabor function and in Figure 6.2, Gabor filter responses on a pedestrian image are provided.

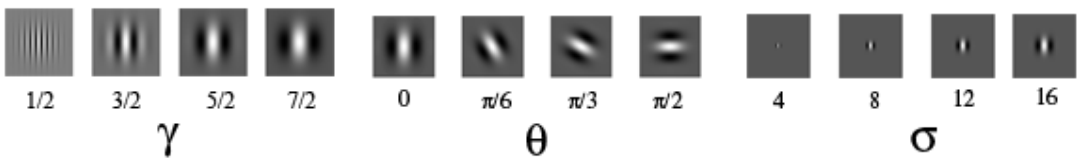

Figure 6.1. Effect of parameters [γ, θ, σ] on Gabor functions

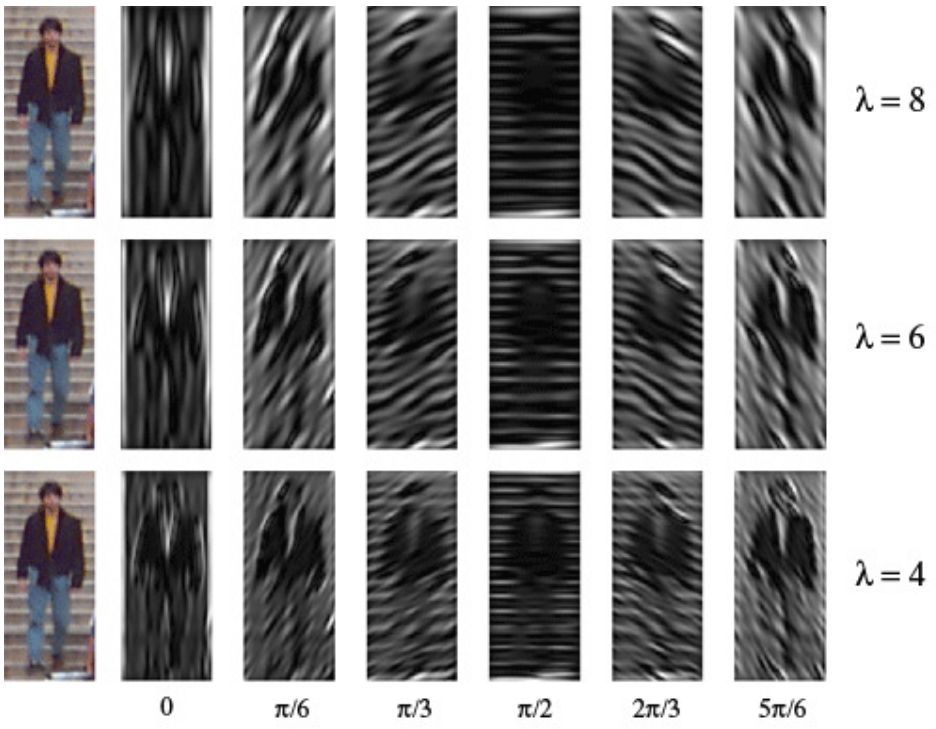

Figure 6.2. Gabor filter responses on a pedestrian image

(for λ = [8, 6, 4], σ = 2.4, θ = [0, π/6, π/3, π/2, 2π/3, 5π/6])

6.1. People Detection based on Dense Gabor Energy Representation

We have applied dense Gabor feature representation approach similar to HOG based detection approach explained in section 5. First, the Gabor filter response of the detection window is calculated at various frequencies and angles. Then, these Gabor filter responses of the detection window are block sampled in that the total Gabor energy of blocks is computed and used as a descriptor. The blocks are of fixed size and overlap. For example, if we use 48x112 size detection window, 12x12 size cells with 6 pixels overlap, the total number of blocks within the detection window becomes 119 as explained in Section 3.1. If we calculate Gabor energies for 2 different frequencies and 6 different angles, we obtain 2x6=12 Gabor bands. Hence we obtain a 12-dimensional feature vector for each block. Finally, the dimension of the feature vector per detection window becomes 119x12=1428.

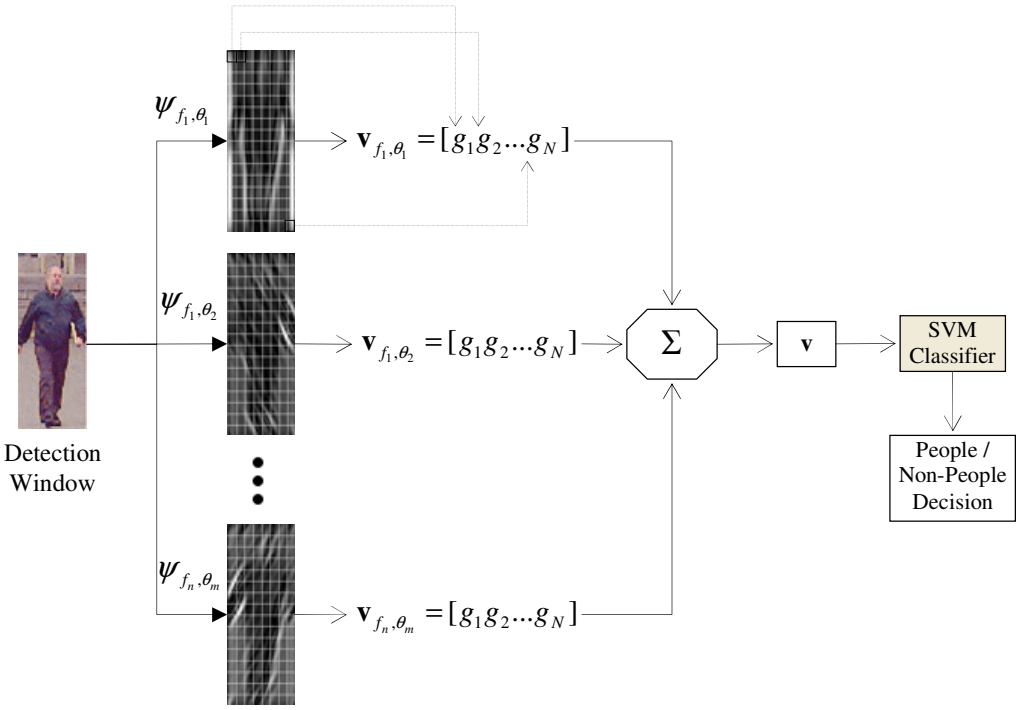

Figure 6.3. People detection based on dense Gabor energy representation

6.2. Selection of the Highest and the Lowest Energy Blocks

In this approach, for each f and θ, average Gabor response is calculated over all the positive training images. To facilitate feature selection, we first calculate the average response image per band; blocks are ranked according to their Gabor energies. Some of the highest and lowest block indexes are selected as blocks of interest. Instead of dense Gabor energy representation approach presented in Section 6.1, feature extraction is performed only on these selected blocks and these per-block features are concatenated to form the feature vector of the detection window. The rationale for choosing low energy blocks side by side with the high energy blocks is to establish a contrast between people regions and non-people regions. In Figure 6.4, average response image for a Gabor band and selected blocks are shown.

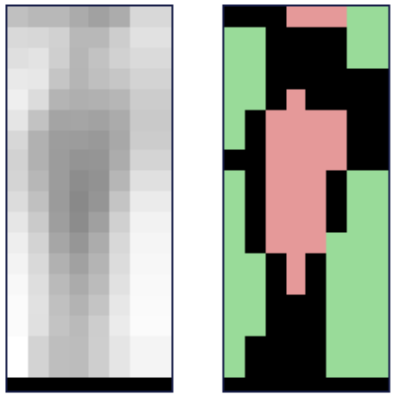

Figure 6.4. (a) Average Gabor filter response image for λ=8 and θ=30° (b) Selected low and high energy blocks (red and green) and ignored blocks (black)

In Figure 6.5, we can see that instead using all blocks, if we use only the highest and lowest energy blocks, we can still get similar detection performance.

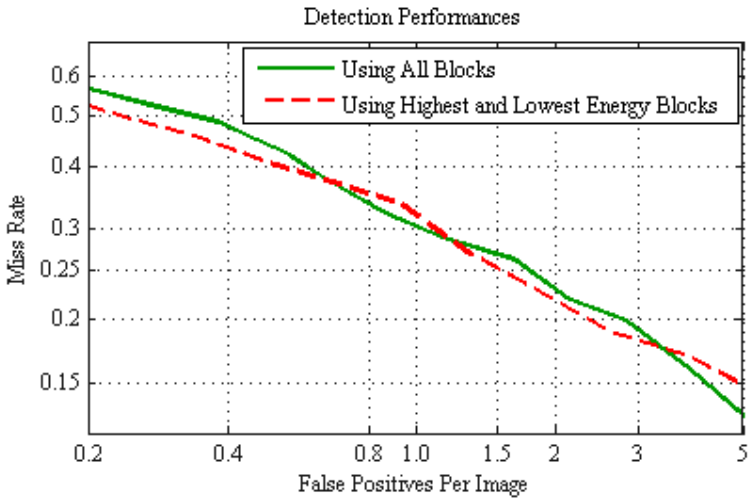

Figure 6.5. Performance of using only the highest and lowest energy blocks

6.3. Detection Results

In this section, detection performance results obtained using Gabor energy features, the effect of different parameter values on the detection performance and results of fusion

of HOG and Gabor based detectors will be presented. Gabor energies are calculated at two different frequencies and various orientation angle resolutions are experimented.

6.3.1 Effect of Wavelength

We have experimented Gabor based people detection using different wavelengths pairs. For each block, Gabor energies are calculated at two different wavelengths and the extracted Gabor energy features are concatenated. In Figure 6.6, people detection performances for different wavelength combinations are presented.

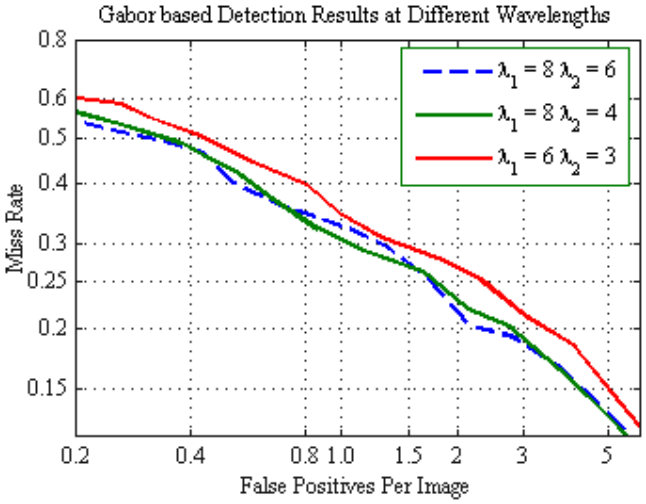

Figure 6.6. Gabor based detection results at various wavelength combinations

6.3.2 Effect of Gabor Angles

We have experimented with Gabor based people detection by calculating Gabor energies at different orientations. The resolution of these angles also affects the detection performance. In Figure 6.7, the effects of Gabor orientation angles on the performance are presented. We can see from this figure that using 4 angles with 45° resolution produces the worst performance. Using 9 angles with 20° resolution produces better results at very low and high FPPI regions, but around 1 FFPI region, it is worse than using 6 angles with 30° resolution.

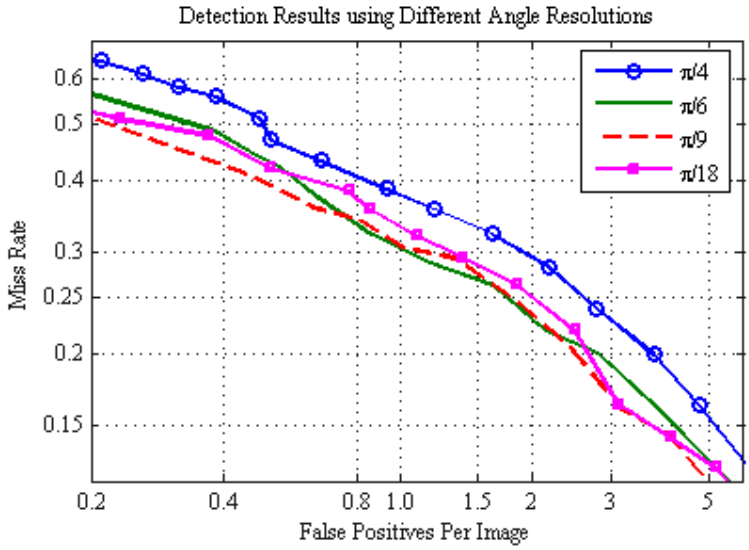

Figure 6.7. Gabor based detection results using different angle resolutions

(The 0°-180° range is evenly divided using the given angle resolutions)

6.3.3 Fusion of the HOG Based and Gabor Based Detectors

We have observed that when we combine the results of detectors based on different features, the overall detection performance becomes better. Fusion operation has been performed as described in Figure 6.8. Sum and Product fusion rules are applied on the SVM decision values and SVM posterior probabilities, respectively.

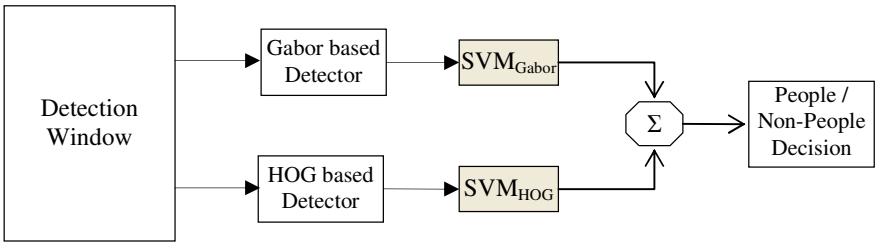

Figure 6.8. Fusion of Gabor and HOG based detectors

In Figure 6.9, performances of the HOG based detector, the Gabor based detector and the fusion of these detectors are shown. The configuration of the HOG based detector is as follows: 12x12 block size, 6 pixel block overlap, 9 HOG bins, L1 block

normalization. The configuration of Gabor based detector: 12x12 block size, 6 pixel block overlap, L1 block normalization, λ_1=8, λ_2=4, σ = 2.4, θ = [0, π/6, π/3, π/2, 2π/3, 5π/6]. For both detectors detection window size is taken as 112x48 pixels and [-1 0 1] operator is used for gradient calculations. As it can be seen from the figure, Sum fusion of HOG and Gabor based detectors improves the detection performance by 5-6% at 1 FPPI reference point.

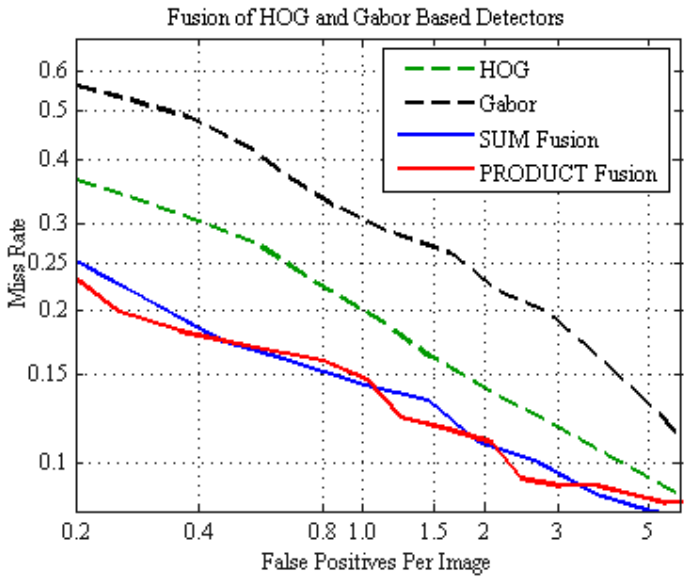

Figure 6.9. Fusion results of Gabor and HOG based detectors

In Figure 6.10, results of fusing HOG and Gabor based detectors at 1 FPPI is shown for two test images.

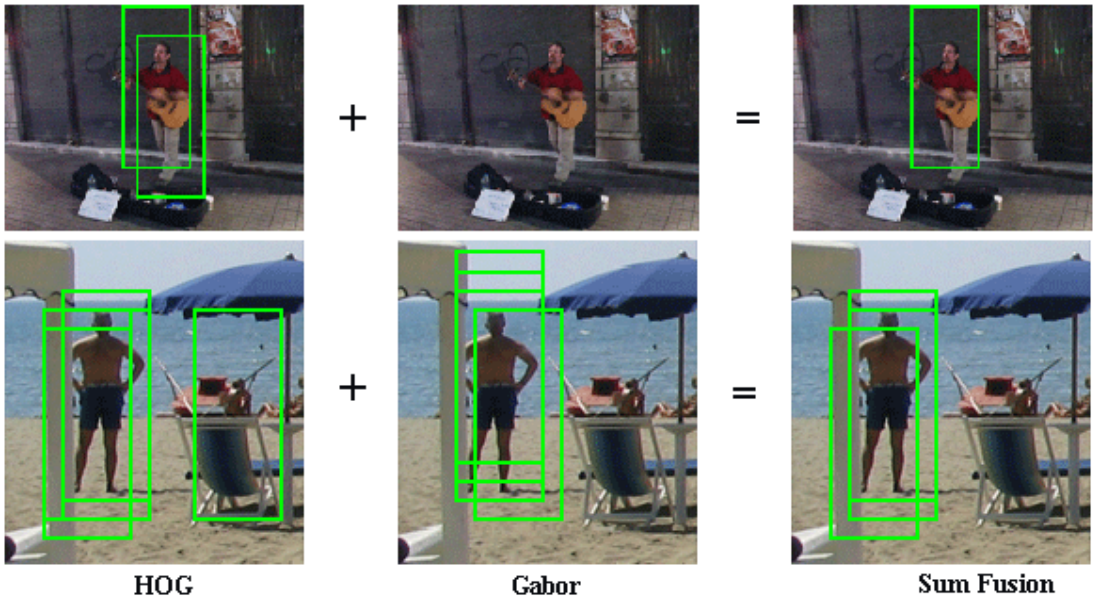

Figure 6.10. Result of applying sum fusion at 1 FPPI on two test images

When we analyze the detection results of the HOG and Gabor based algorithms at 1 FPPI we see that 63.0% of the people are detected by both algorithm, 14.0% of the people are not detected by either, 6.3% of the people are detected only by the Gabor-based algorithm and 16.7% of the people are detected only by the HOG-based algorithm.

6.3.4 Detection Results by the Gabor-based Detector

In Figure 6.11, some sample false positive detection results of the Gabor-based detector are presented. From the figure, we can see that generally vertical structures and head-like objects are detected as people.

Figure 6.11. Various false alarms by the Gabor based detector at 1 FPPI

In Figure 6.12, some people which cannot be detected by the Gabor based detector at 1 FPPI are shown. From the figure, we see that similar to the results of HOG based detector, viewpoint and scale changes, poor illumination, body articulations and partial occlusion are also problems of the Gabor based detector.

Figure 6.12. Various missed detections by the Gabor based detector at 1 FPPI

In Figure 6.13, some detections performed by the Gabor based detector at 1 FPPI are shown.

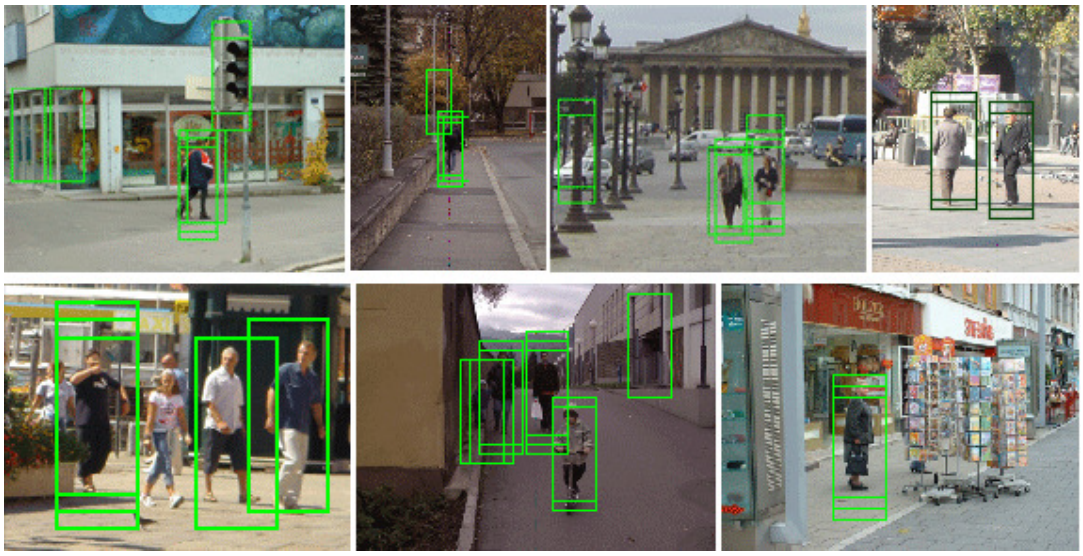

Figure 6.13. Sample detections by the Gabor based detector at 1 FPPI

7. CASCADING WITH SKIN COLOR

Skin color has proven to be a useful and robust cue for face detection, localization and tracking. Image content filtering, content-aware video compression and image color balancing applications can also benefit from automatic detection of skin in images. However, skin color cannot be used for pedestrian detection as a standalone tool since human skin may not be available in many cases, but it can be a useful cue if the pedestrian is facing the camera. In addition, skin color is very sensitive to illumination and in case of outdoor scenes; there may be some background structures with colors very close to skin color.

In order to increase the reliability of our people detection system, we have integrated skin color detection as a cascading detector to other more powerful detection algorithms. During this integration, only existence of skin color is used as a cascading tool since absence of skin color does not infer the absence of people. Skin color cascade operation is not applied on all detection windows in an image. We have checked the SVM detection scores of the detection windows in an image, if the SVM score of the detection window is just below the SVM decision threshold by a small amount, then skin color cascading operation is performed. If we detect skin color in the detection window, then we increase detection score so that it becomes a positive detection.

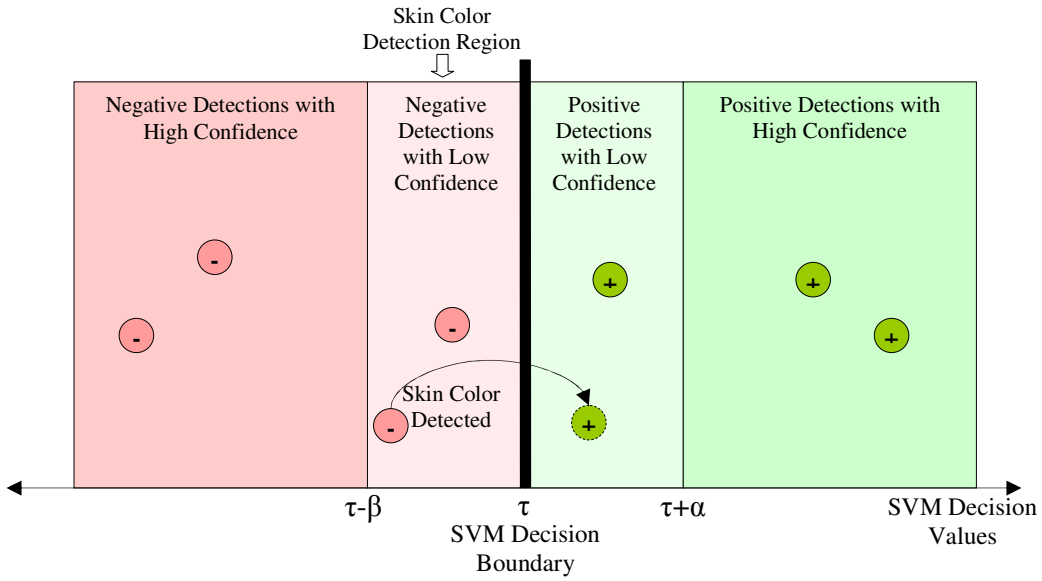

Figure 7.1. Description of skin color cascade operation

(Skin color detection is performed only for the detection windows having score in the range [τ-β, τ])

7.1. Skin Color Detection Algorithm

There are many skin color detection algorithms proposed in the literature. We have tried the explicitly defined RGB skin region approach presented in [25] and the algorithm presented in [26], which are based on human perception of colors using YUV and YIQ color spaces. We have observed that the performance of the latter was superior; therefore we have selected to use it in our people detection system.

The chromaticity information is encoded in the *U* and *V* components. Hue and saturation are obtained by the following transformation:

$$Ch = \sqrt{|U|^2 + |V|^2}$$
$$\theta = \tan^{-1}\left(|V| / |U|\right)$$

where θ represents hue, which is defined as the angle of vector in YUV color space. *Ch* represents saturation, which is defined as the mode of *U* and *V*. Proper hue thresholds are

obtained according to the observation that the hues of most people's skin vary in the range from 100° to 150°.

Like YUV color space, YIQ is the primary color system adopted by NTSC for color TV broadcasting. In YIQ color space *I* is the red-orange axis and *Q* is roughly orthogonal to *I*. The less *I* value means the less blue-green and the more yellow. It has been shown that most people's skin varies in the range from 20 to 90 in the term of *I*. The combination of YUV and YIQ color space has been found to be more robust than each other. As shown in Figure 7.2, if a pixel satisfies $I \in [20,90] \cap \theta \in [100,150]$, it is possible be relevant to skin color.

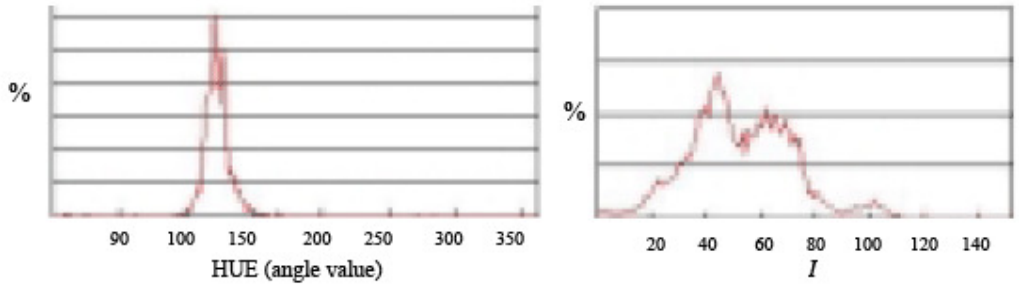

Figure 7.2. Distribution of skin color in YUV and YIQ color spaces [26]

It can also be seen from the Figure 7.2 that probability of being a skin pixel increases if we narrow the defined *Hue* and *I* ranges. Therefore, we have segmented these ranges into 5 regions and assigned separate scores for each color range segment and then calculated the sum of these scores as the final skin color score of a pixel. Let N_i be the number of skin pixels in range R_i, then overall skin score *S* of a pixel is calculated as follows:

$$S = \sum_{i=1}^{n} \alpha_i N_i$$

$$\text{where } \sum_{i=1}^{n} \alpha_i = 1 , \quad \alpha_1 < \alpha_2 < ... < \alpha_n \text{ for } R_1 \supset R_2 \supset ... \supset R_n$$

In Figure 7.3 the results of this range segmentation and scoring scheme is shown for three sample images. In the third sample image, we can see that some background structures are also considered as skin pixels.

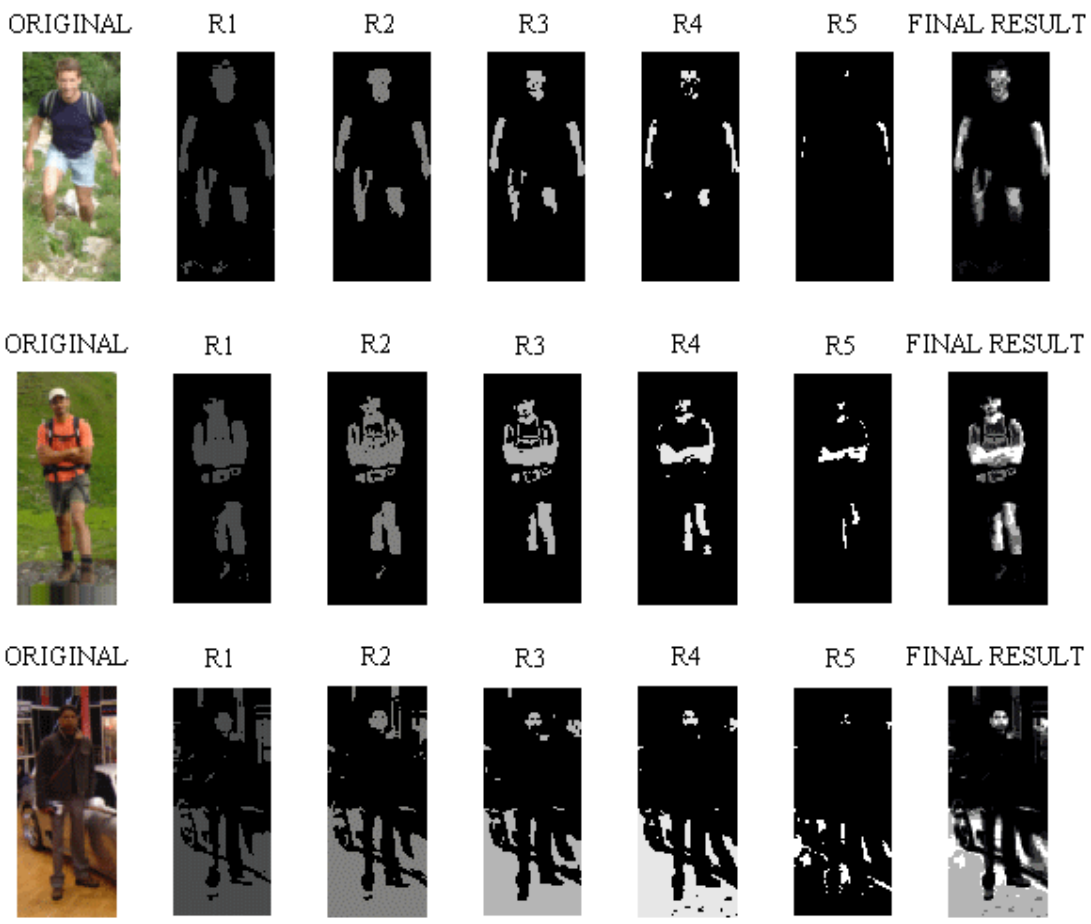

Figure 7.3. Sample skin pixel detection results

$$R1: Hue \in [100,150], I \in [20,90]$$
$$R2: Hue \in [105,145], I \in [25,85]$$
$$R3: Hue \in [110,140], I \in [30,80]$$
$$R4: Hue \in [115,135], I \in [35,75]$$
$$R5: Hue \in [120,130], I \in [40,70]$$

7.2. Using Skin Color Information in People Detection

First we have constructed a new positive training set where all humans contain some amount of skin pixels since our original training database also contains humans with no skin pixels. We have designed a skin color based people detector and used the results of

this detector as a cascade for previously developed people detection algorithms like HOG based detectors to improve them.

First, we run a people detector such as HOG based detector on the image. Then we look for skin pixels within the negative detections with low confidence SVM scores. If the calculated skin pixel score exceeds the threshold then the negative detection is changed to a positive detection.

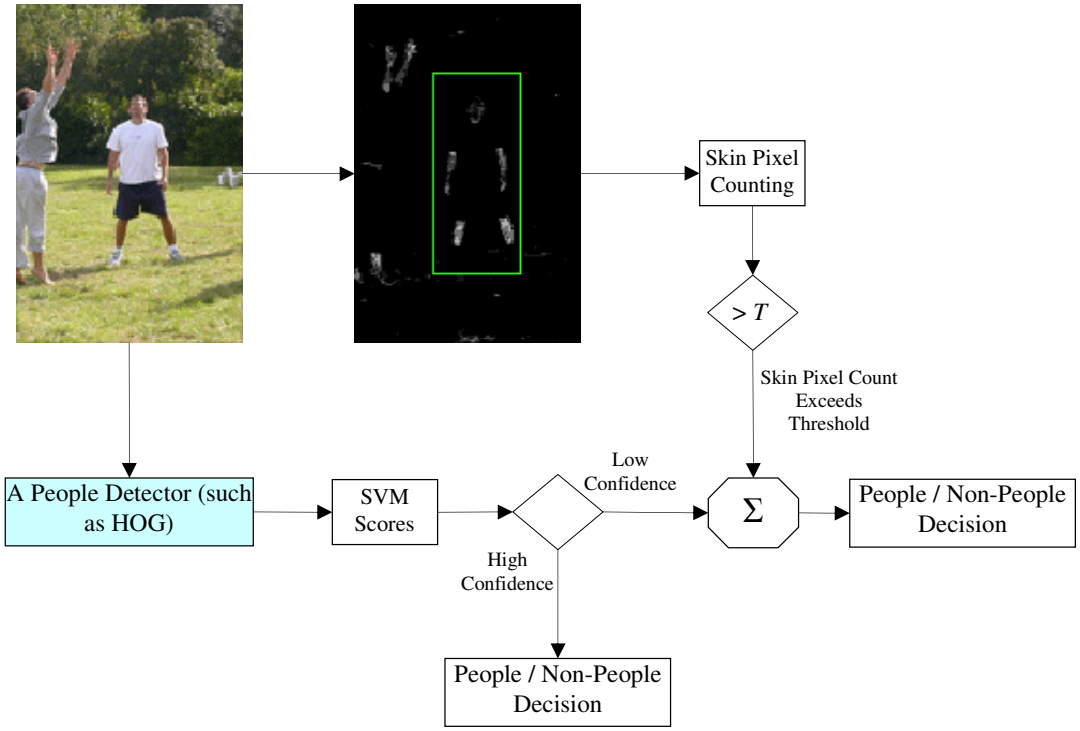

Figure 7.4. Cascading skin color based detector with other detectors

In Figure 7.5, performance of skin color cascading is shown, where skin color cascading has been applied on the HOG based detector. We see that the improvement made by skin color cascading is limited, only 2% performance increase is obtained at 1 FPPI.

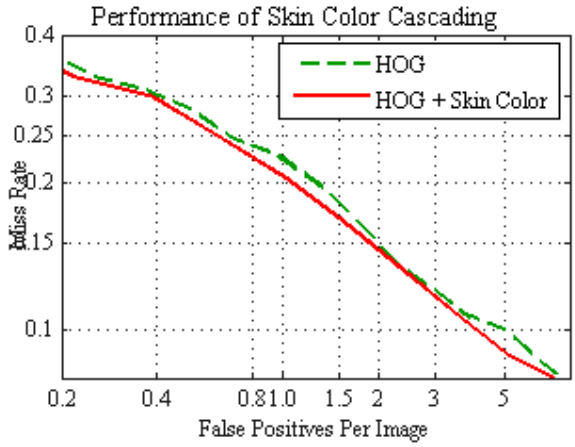

Figure 7.5. Performance effect of skin color cascading on the HOG based detector

In Figure 7.6, some example results of applying skin color cascade algorithm to the HOG based detector at 1 FPPI are shown. We see that although skin color cascading may have positive effects on the performance, it may also introduce false positives due to skin-like colors at the background.

Figure 7.6. Examples of positive and negative effects of skin color cascading

8. ANALYSIS OF PROJECTION PROFILES

Projection profiles have been used for various purposes in the literature such as character and line segmentation in OCR systems and human posture identification [27]. We have also studied projection profiles as a feature to use for people detection. In Figure 8.2, average vertical and horizontal projection profiles of people in our training image database are shown. From this figure, we see that on the overage, people have a common projection profile, especially in the vertical projection profile.

Figure 8.1. Vertical projection profiles of some true and false detections

However, as it can be seen from the Figure 8.1, because of the factors like background clutter, viewpoint changes and body articulations, most of the time it is not possible to differentiate pedestrians from other objects by using only projection profiles. Some obvious cues extracted from projection profiles can be utilized to reject false positive detection. For example, as shown in Figure 8.1, we do not expect flat and low magnitude profiles. The right-most false positive detection in Figure 8.1 has a flat and low magnitude profile at the upper part. Hence this information can be utilized to reject this false positive.

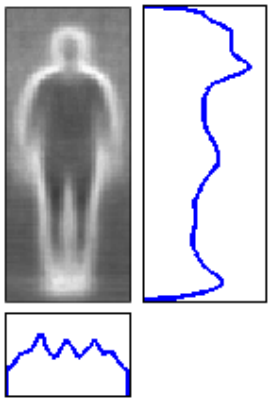

Figure 8.2. Average vertical and horizontal projection profiles of positive training images

We have also analyzed energy (variance), kurtosis and skewness properties of vertical projection profiles of positive and negative images. Histograms of these properties for the training set are provided in Figure 8.3. It is obvious from these histogram plots that it is very difficult to differentiate pedestrians from other objects by using only projection profiles information

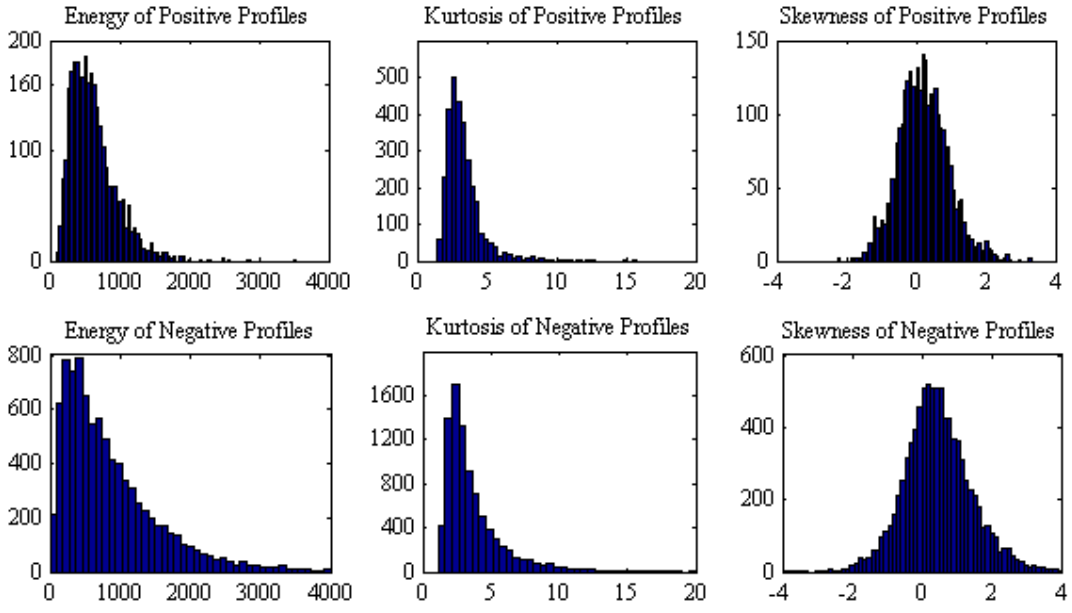

Figure 8.3. Energy, kurtosis and skewness histograms of vertical projection profiles of positive and negative training images

9. PEOPLE DETECTION BASED ON BLOCK GRADIENT ORIENTATION VECTORS

In this approach, we use gradient orientation of pixels similar to the HOG based methodology, but we do not construct an orientation histogram. Instead, gradient orientation vectors of pixels within a block are combined (vector summed) to form a single orientation vector, which represents the overall orientation of the block. An illustration of block orientation calculation is given in Figure 9.1. In this figure, a block contains 4 pixels (*p1*, *p2*, *p3* and *p4*) with gradient orientation vectors shown. These vectors can be expressed in terms of x- and y- gradient components (g_x, g_y) as follows: $\mathbf{g}_1$= (0, 2), $\mathbf{g}_2$= (1, 2), $\mathbf{g}_3$= (2, 1) and $\mathbf{g}_4$= (1, -1). The overall orientation of the block $\mathbf{g}_b=(g_{bx}, g_{by})$ is calculated as follows:

$$g_{bx} = \sum_{i=1}^{n} g_{x_i} = 4 \text{ and } g_{by} = \sum_{i=1}^{n} g_{y_i} = 4$$

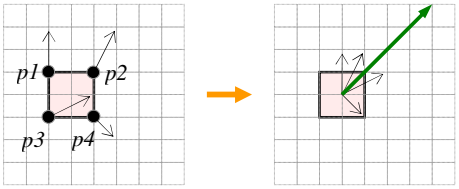

Figure 9.1. Example of block orientation calculation

9.1. Integral Orientation Image

Combined orientation of a block is easily calculated by just summing x- and y-gradient components of each pixel separately as described in the previous section. This rectangular calculation scheme allows us to construct an integral orientation image, which is a similar idea introduced in [28] for image intensities. Let $g_x(x,y)$ and $g_y(x,y)$ be the horizontal and vertical gradients respectively at pixel location (x,y), then integral orientation images are calculated as follows:

$$gg_x(x,y) = \sum_{x' \le x, y' \le y} g_x(x', y') \text{ and } gg_y(x,y) = \sum_{x' \le x, y' \le y} g_y(x', y') \text{ and}$$

where $\boldsymbol{gg_x}$ is the horizontal integral orientation image and $\boldsymbol{gg_y}$ is the vertical integral orientation image. For example, in Figure 9.2, point 1 contains the sum of orientations in rectangle A, point 2 contains A+B, point 3 contains A+C and point 4 contains A+B+C+D. The orientation of block D can be easily calculated as 4-3-2+1.

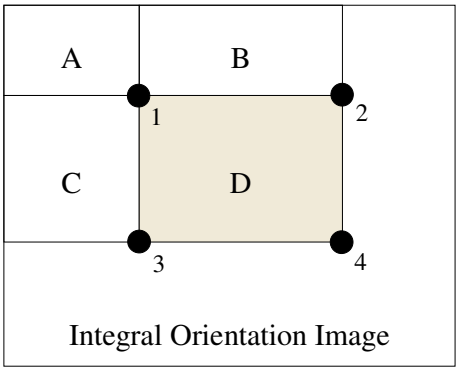

Figure 9.2. Calculation of block orientation using integral orientation image

9.2. People Detection based on Dense Representation of Block Orientation Features

We scan the detection window using fixed size block in an overlapping manner and for each block we calculate the combined orientation vector. The feature vector of the detection window is constructed by using the norm and angles of these block orientation vectors or just by using the *x*- and *y*-components of the orientation vector.

9.3. Searching for the Most Relevant Blocks

The goal of this approach is to find the blocks in a search window, which best differentiates humans from non human objects. For this purpose, we have defined 23 blocks with different sizes such as 8x8, 4x8, 16x8 etc. We have constructed a training image set containing 1000 human and 1000 non-human images and a test set with again 1000 human and 1000 non-human images. As a feature to be used for detection, we only use the combined gradient orientation vector of one block at one position. The SVM

classifier is trained using only the orientation vector of that block at the given position. Then, the differentiating capability of the block size-position pair is measured by performing detection experiment on the test set.

We have studied 11630 different combinations of block size-position pairs. The detection scores of these pairs are sorted in descending order. This way we have the opportunity of using the most relevant block features in a search window instead of using fixed size blocks at all positions. In Figure 9.3, 400 highest-score and lowest-score blocks are shown on a pedestrian image. We can see from this figure that the most relevant people/non-people features are located on boundaries of the people body and narrow (tall) rectangular blocks contain more relevant information than wide (flat) rectangular blocks. We can also see from the figure that the least relevant people/non-people information is located at the bottom boundaries of the search window and the chest area of the human body, where there may exist different textures depending on the clothing of the human.

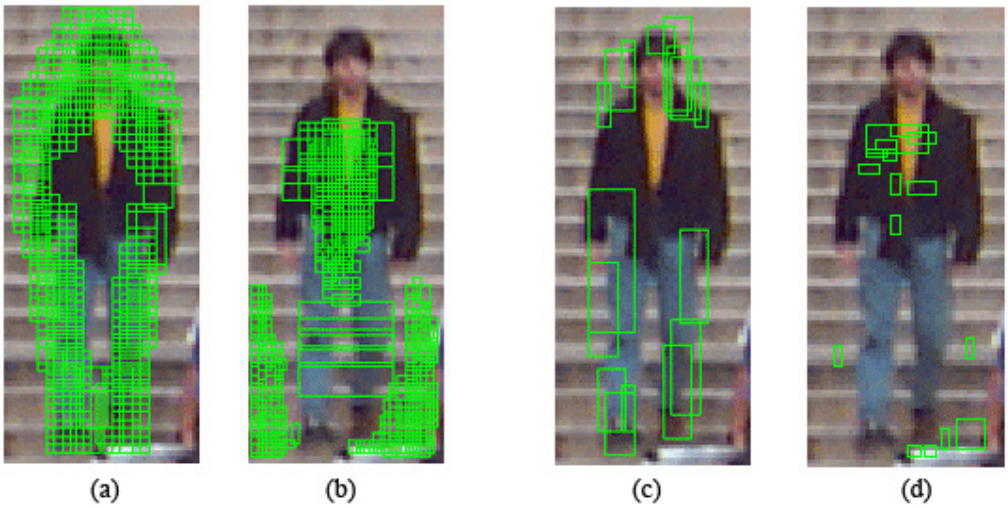

Figure 9.3. (a) 400 Highest score blocks (b) 400 lowest score blocks (c) 16 highest score blocks (d) 16 lowest score blocks

9.4. Detection Results

During the search for the most relevant blocks to differentiate between people and non-people images, it was interesting to see that using the orientation vector of only one

block, we were able to get 70% detection success on our training images. However, when we combine the orientations of many blocks and feed them into SVM classifier, the overall performance on our test images was not so good. Therefore, as a future work, Adaboost algorithm can be used to find out the best blocks and then classification can be performed using Adaboost or SVM algorithms.

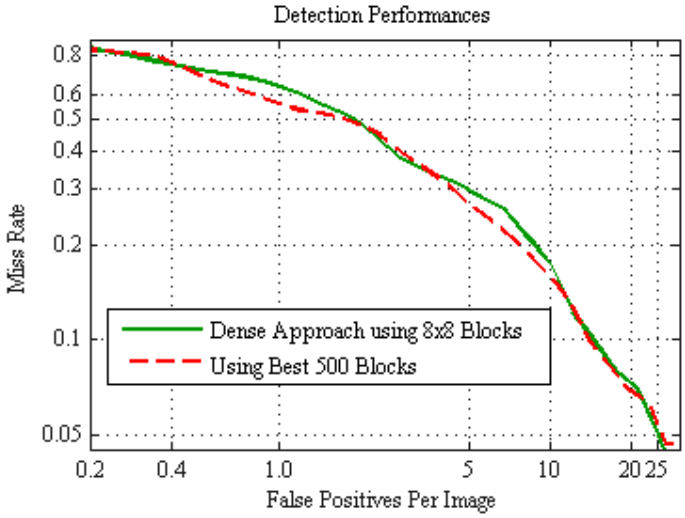

Figure 9.4. Performances of block orientation based detectors

Although this algorithm may not be used as a standalone people detector, it can be used as a preprocessing detector. From the Figure 9.4, we can see that at 25 FPPI point, the miss rate is 5%. At this point, the number of total detections found by the algorithm is 5694 and the total number of detection windows within all test images is 42026. So, if we accept the risk of losing 5% of people, we can reduce the number of windows to be searched to 5694, instead of looking at all 42026 windows; this means 86.5% reduction in the number of search windows and it may help to speed up the detection process.

10. DETECTION BASED ON CLUSTER DISTANCES

In this approach, first we have clustered people images in our training set. Clustering is performed on the HOG features of the positive training images by using the K-Means algorithm. Then during training phase, for each image in the training set, Euclidian distance to cluster centroids is calculated. The cluster centroids of the positive training images are recorded in order to use them later in the detection phase. Cluster distances are used to form a feature vector and are fed into the SVM classifier for detection. Since the number of features are very few, linear SVM does not produce good results; therefore SVM with RBF kernel is preferred.

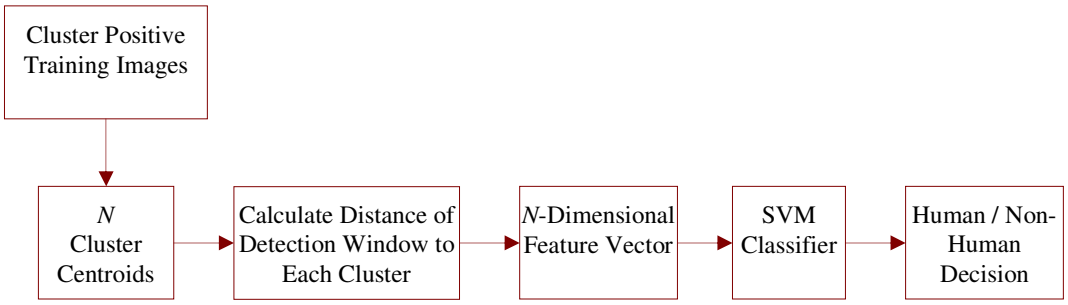

Figure 10.1. Flow of detection algorithm based on cluster distances

In Figure 10.2, we show detection performances of the cluster distance based detector at various cluster numbers. We can see from the figure that as we increase the number of clusters, the performance of the detector improves. However, if the dimension of the image feature vector used in clustering is high, the detection time gets longer when we increase the number of clusters since cluster distance calculations take more time during detection.

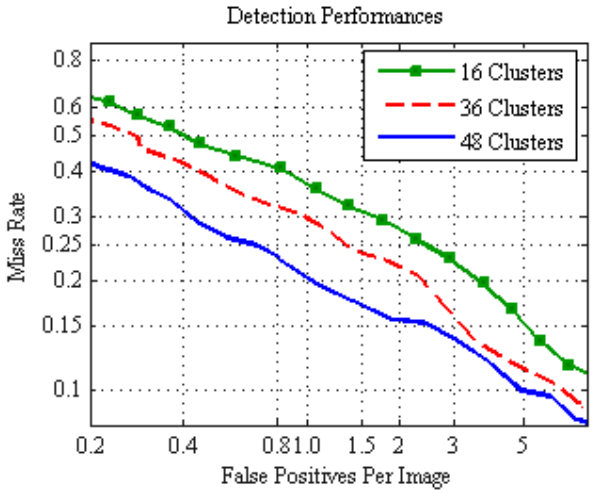

Figure 10.2. Detection performances of the cluster distance based detector

In Figure 10.3, the effect of increasing the number of clusters at 1 FPPI reference point is shown. We can see that as we increase the number of clusters, false positives are removed and true positive detections get better.

Figure 10.3. Effect of number of clusters on two test images at 1 FPPI

11. SUMMARY AND CONCLUSION

We have performed people detection in still images using various features and algorithms. An image database is constructed by utilizing MIT [20] and INRIA [21] data sets. Our image database contains positive (people) and negative (non-people) training images and test images in order to measure performances of the detectors.

In order to detect people in an image, the image is scanned by a search window in an overlapping manner. At each position of the search window, image features are extracted and these features are fed into a linear soft margin SVM classifier, which is trained by using the same type of features on the positive and negative training images. Dense feature representation of the search window is constructed by dividing the window into fixed size overlapping blocks and extracting local features for each block. These local block features are concatenated to form the overall feature vector of the detection window. We have observed that for each detection scale, the selection of appropriate values for HOG parameters like block size and histogram bin number is crucial for the performance of the detector. At the scale of our training images, we have obtained the best performance by using 12x12 size blocks, 9 histogram bins and L1 block normalization scheme.

We have observed that normalization of image features is a critical issue in order to get higher detection rates. L1, L2, LSUV and LSUR normalization methods are applied on the extracted image features. Normalization is either applied locally for each block separately in the search window or globally for the whole search window. We have seen that local normalization scheme performs better than global normalization. Among the normalization methods, L1 and L2 normalization algorithms produce better results than others and LSUV algorithm produces very poor results.

As far as image feature types are concerned we have studied Histogram of Oriented Gradients, Gabor energies, block gradient orientations, skin color, projection profiles and cluster distances. Among these features, locally normalized HOG features produce the best detection results, 80% recall rate at 1 FPPI. Gabor energies approach produces 70% recall

rate at 1 FPPI. The fusion of HOG and Gabor based detector results in the best detector with an 86% recall rate at 1 FPPI. Skin color could not be used a standalone feature for detection but when cascaded with other detectors it helps to improve detection performance by 1-3%. Projection profiles could not be used as features for pedestrian detection. Although human body has a common projection profile, especially the vertical projection profile, because of the factors like background clutter, viewpoint changes and body articulations, most of the time it is not possible to differentiate pedestrians from other objects by using only projection profiles information. The performance of the detector which is obtained by combining the HOG and Gabor based detectors is shown in Figure 11.1

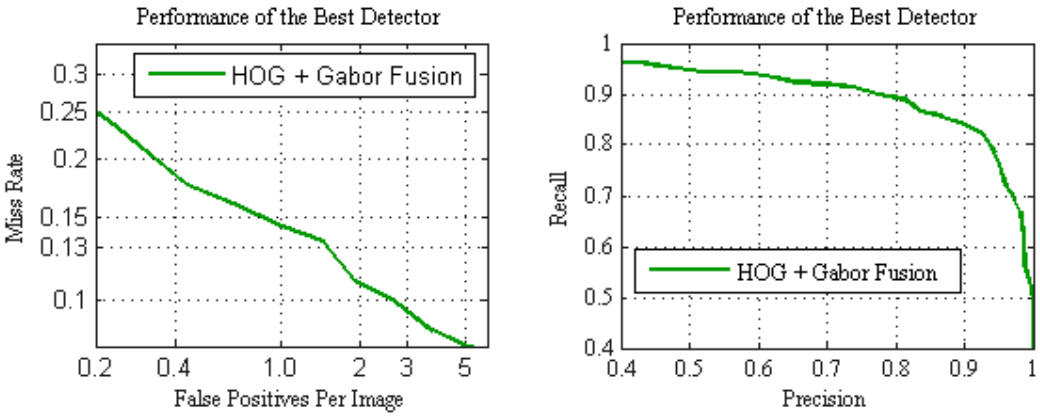

Figure 11.1. Performance of the best detector: Fusion of HOG and Gabor based detectors

We have also introduced block gradient orientation as an image feature which can be used for detection. Although we have not obtained high detection rates using this feature, it merits to be studied further, since it is can be calculated very fast using the integral orientation image approach and might have a role as a preprocessing tool.

In order to obtain higher and more reliable detection performances, we have applied various fusion techniques at the data, feature and decision levels. The best fusion results are obtained by fusing the SVM scores of different detectors by applying Sum and Product fusion rules. The best performances of different people detectors developed in this project are given in Table 11.1 for three different FPPI points.

Table 11.1 Performances of the detectors at 3 different FPPI points

Algorithm	Recall Rates		
	0.4 FPPI	1 FPPI	3 FPPI
HOG	69.3%	79.7%	88.3%
Gabor	52.3%	69.3%	80.3%
HOG + Gabor	80.3%	85.7%	91.0%
HOG + Skin	69.8%	81.6%	88.7%
Block Orientations	25.7%	43.7%	62.3%
Cluster Distances	68.3%	79.3%	86.0%

11.1. Future Work

We have performed detections using only one scale; performing detection on several scales can improve the detection performance. Our test image database contains people which are shorter than people in our training dataset. Therefore, most detectors have missed these short people. For different scales, it would be better to search for the best detector parameters, instead of using the detectors with the same parameters.

The torso part of the human body has a more consistent appearance than the other parts of the body since its appearance is not much affected by the movement. Therefore, people detector can be improved by cascading with a torso detection algorithm. For example, after that the image is scanned with the detector, low-confidence windows can be further processed by a torso detector. Template matching can be applied for torso detection. Moreover, we have performed skin color detection over the whole detection window. However, most of the time, we can find skin color on people's faces. Therefore, skin color detection can be restricted to the upper part of the detection window.

In order to further improve the detectors, both miss rate and false positive rate should be decreased. We have observed that most of the false positives are caused by vertical human like objects. Hence, further studies can be performed to eliminate these false detections. As far as the missed detections are concerned partial occlusion is the most challenging problem. In order to tackle partial occlusion problem parts-based approaches should be studied.

We have only worked on grayscale and R,G,B color spaces, other color spaces can also be experimented. Due to complex background structures in our test images, we could not utilize projection profiles. However, if the detector is used with a static camera, then it would be possible to apply background subtraction algorithm to remove noise caused by background clutter. In that case, projection profiles can also be used for various purposes like posture identification in addition to people detection task.

Block orientation approach mentioned in Chapter 9 is a very fast algorithm but its detection performance with a linear SVM classifier was poor. Using Adaboost approach, the performance of this feature can be improved. Also we have observed that at 25 FPPI point, this algorithm provides 95% recall rate, so it can be used as a preprocessing detector in order to reduce the number of search windows. As we have mentioned in section 9.4, it is possible to obtain 86.5% reduction in the number of search windows.

Multiple camera usage is another active research area in people detection in order to develop more robust people detectors. This approach can be studied especially to solve the occlusion problem by processing evidence obtained from several cameras simultaneously.

In addition, the recently introduced Compressive Sensing (CS) theory can also be studied for people detection task. This theory proposes a new sampling technique and a different approach to the reconstruction and classification issues. Some studies have shown that CS based classification algorithms do not require heavy pre-processing and are generally more successful than traditional ones.

We have also observed that the same person in an image can be detected by several detection windows, on the just left, right, up and down of the original position. In order to obtain a clearer detection picture, these detections can be combined to show as few detections as possible.

Since we have implemented our detection algorithms mostly using the MATLAB toolbox, we could not measure our detection speed benchmark values, which are very important for real-time usage. Therefore, algorithms can be implemented with a suitable language like C/C++ in order to analyze the real-time performance.

APPENDIX A: FUSION TECHNIQUES

Fusion rules make use of the output values of the base classifiers. The outputs of these classifiers can be class labels, distances or class posterior probabilities. In this project, we have used various fusion techniques in order to combine detection results of different detection experiments. At each position of the detection window over the image, SVM produces a distance score and also a probability of human existence within that detection window. The following fusion rules are applied on these SVM results.

Let C_{ij} be some numerical outcome of classifier j for class i, $\mathbf{x}$ represents the object and $Q_i(\mathbf{x})$ is the combining classifier.

- **Sum Rule**

$$Q_i(\mathbf{x}) \sim \sum_j C_{ij}(\mathbf{x})$$

- **Product Rule**

$$Q_i(\mathbf{x}) \sim \prod_j C_{ij}(\mathbf{x})$$

- **Max Rule**

$$Q_i(\mathbf{x}) \sim \max_j \{C_{ij}(\mathbf{x})\}$$

- **Min Rule**

$$Q_i(\mathbf{x}) \sim \min_j \{C_{ij}(\mathbf{x})\}$$

- **Median Rule**

$$Q_i(\mathbf{x}) \sim \text{median}_j \{C_{ij}(\mathbf{x})\}$$

APPENDIX B: COLOR SPACE TRANSFORMATIONS

B.1. RGB to YUV Transformation

Historically, YUV color space was developed to provide compatibility between color and black/white analog television systems. YUV color image information transmitted in the TV signal allows proper reproducing an image contents at the both types of TV receivers, at the color TV sets as well as at the black/white TV sets. The *Y* component determines the brightness of the color (referred to as luminance), while the *U* and *V* components determine the color itself (the chroma). The RGB values are transformed into YUV values using the following formulation:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

B.2. RGB to YIQ Transformation

The conversion from RGB to YIQ may be accomplished using the following transformation:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & -0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

REFERENCES

1. Mikel Rodriguez and Mubarak Shah, “Detecting and Segmenting Humans in Crowded Scenes”, *ACM Multimedia*, 2007

2. P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: A Benchmark", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009

3. Enzweiler, M., Gavrila, D.M., “Monocular Pedestrian Detection: Survey and Experiments”, *PAMI(31)*, 2009

4. Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon, “Hierarchical Part-template Matching for Human Detection and Segmentation”, *In ICCV*, 2007

5. D. Gavrila and V. Philomin, “Real-time Object Detection for Smart Vehicles”, *ICCV*, 1999

6. N. Dalal, B. Triggs, I. Rhone-Alps, and F. Montbonnot, “Histograms of Oriented Gradients for Human Detection”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005

7. C. Papageorgiou and T. Poggio, “A Trainable Pedestrian Detection System”, *International Journal of Computer Vision (IJCV)*, 2000

8. Qiang Zhu, Shai Avidan, Mei-chen Yeh and Kwang-ting Cheng, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients", *CVPR*, 2006

9. F. Suard, A. Rakotomamonjy, and A. Bensrhair, "Pedestrian Detection using Infrared images and Histograms of Oriented Gradients", *Intelligent Vehicles Symposium*, 2006

10. B.Leibe, E. Seemann, and B. Schiele. “Pedestrian Detection in Crowded Scenes”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005

11. A. Mohan, C. Papageorgiou, and T. Poggio, “Example-based Object Detection in Images by Components”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001

12. Bo Wu and Ram Nevatia, “Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors”, *IEEE International Conference on Computer Vision (ICCV)*, 2005

13. Bin Hu, Shengjin Wang, Xiaoqing Ding, "Multi Features Combination for Pedestrian Detection", *Journal of Multimedia*, Vol 5, No 1 (2010), 79-84, Feb 2010

14. Mikolajczyk, K. and Schmid, C. and Zisserman, A. “Human Detection Based on a Probabilistic Assembly of Robust Part Detectors”, *The European Conference on Computer Vision (ECCV)*, volume 3021/2004, 2005

15. P. Viola, M. Jones, D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance", *IEEE International Conference on Computer Vision (ICCV)*, 2003

16. Gavrila, D.M., Munder, S., “Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle”, *IJCV(73),* 2007

17. Payam Sabzmeydani and Greg Mori, “Detecting Pedestrians by Learning Shapelet Features”, *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2007

18. S. Agarwal and D. Roth, “Learning a Sparse Representation for Object Detection”, *In Proceedings of the 7th European Conference on Computer Vision*, Copenhagen, Denmark, 2002

19. Chih-Chung Chang, and Chih-Jen Lin, “A Library for Support Vector Machines”, Department of Computer Science, National Taiwan University, http://www.csie.ntu.edu.tw/~cjlin

20. MIT Pedestrian Dataset, http://cbcl.mit.edu/software-datasets/PedestrianData.html

21. INRIA Person Dataset, http://pascal.inrialpes.fr/data/human

22. Josef Kittler, Mohamad Hatef, Robert P.W. Duin, Jiri Matas, "On Combining Classifiers", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, Mar. 1998

23. Al-Amin Bhuiyan, and Chang Hong Liu, “On Face Recognition using Gabor Filters”, *World Academy of Science, Engineering and Technology*, 2007

24. Hong Cheng, Nanning Zheng, Chong Sun, “Boosted Gabor Features Applied to Vehicle Detection”, *18th International Conference on Pattern Recognition*, 2006

25. Peer, P., Kovac, J., and Solina, F., “Human Skin Color Clustering for Face Detection”, *In submitted to EUROCON – International Conference on Computer as a Tool*, 2003

26. Lijuan Duan, Guoqin Cui, Wen Gao and Hongming Zhang, “Adult Image Detection Method Based on Skin Color Model and Support Vector Machine”, *The 5th Asian Conference on Computer Vision*, 2002

27. Foroughi, H. Aski, B.S. Pourreza, H. , “Intelligent Video Surveillance for Monitoring Fall Detection of Elderly in Home Environments”, *ICCIT*, 2008

28. P. Viola and M. Jones, “Rapid Object Detection Using a Boosted Cascade of Simple Features”, *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001