Object Detection on Embedded System

Author

MUHAMMAD AMMAR HASSAN

NUST201463100MSMME62114F

Supervisor

DR. SYED OMER GILANI

DEPARTMENT OF ROBOTICS AND INTELLIGENT MACHINE ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD, PAKISTAN

AUGUST, 2018

Object Detection on Embedded System

Author

MUHAMMAD AMMAR HASSAN

Registration Number

NUST201463100MSMME62114F

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics and Intelligent Machine Engineering

Thesis Supervisor:

DR. SYED OMER GILANI

Thesis Supervisor's Signature: ____

DEPARTMENT OF ROBOTICS AND INTELLIGENT MACHINE ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD, PAKISTAN

AUGUST, 2018

Thesis Acceptance Certificate

It is certified that the final copy of MS Thesis written by NS Muhammad Ammar Hassan (Registration No. NUST201463100MSMME62114F), of SMME (School of Mechanical & Manufacturing Engineering) has been vetted by undersigned, found complete in all respects as per NUST statutes / regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfilment for award of MS/MPhil Degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: ______________________________

Name of Supervisor: Dr. Syed Omer Gilani

Date: __________________________________

Signature (HOD): _______________________

Date: __________________________________

Signature (Principal): ___________________

Date: __________________________________

Declaration

I certify that this research work titled "*Object Detection on Embedded System*" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

MUHAMMAD AMMAR HASSAN

NUST201463100MSMME62114F

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student
MUHAMMAD AMMAR HASSAN
NUST201463100MSMME62114F

Signature of Supervisor
Dr. Syed Omer Gilani

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed, I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express special thanks to my supervisor Dr. Syed Omer Gilani for his help throughout my thesis and also for Dr. Syed Irtaza Ali Shah and Computer Vision course which he has taught me. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to pay special thanks to my wife for his tremendous support and cooperation. Each time I got stuck in something, she came up with the solution. Without her help I wouldn’t have been able to complete my thesis. I appreciate her patience and guidance throughout the whole thesis.

I would also like to thank Dr. M. Jawad Khan, Dr. Mohsin and Dr. Emad uddin for being on my thesis guidance and evaluation committee and express my special thanks to Dr. Muhammad Naveed for his help. I am also thankful to M. Javed Iqbal and Umair Bin Ahmed for their support and cooperation.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional parents and my wife whose tremendous support and cooperation led me to this wonderful accomplishment.

Abstract

Human detection in thermal infrared images postures a difficult challenge because infrared images have a lot of noise and they have very low resolution. We use Local Adaptive Steering Kernel (LASK) method to extract features from infrared images on the basis of image geometry concisely. We perform training less detection of pedestrians in low resolution infrared images by creating a rigid detector template. We detect humans by cross correlation in frequency domain. The cross correlation is between features of detector template and target image features. This speeds up the detection as compared to sliding window approach. The aim of this work was to implement method that can detect humans in thermal infrared images on computer and as well as on embedded system. Our implementation is performed on benchmark thermal infrared images of dataset "*OSU Thermal 01 Pedestrian database"*. We implement this method on dell 6th generation, core i5, 2.4 GHz with 8GB Ram on MATLAB and on single board computer ODROID Xu4. It has "*Samsung Exynos5422 Cortex™-A15 2Ghz and Cortex™-A7 Octa core"* alongwith 2GB Ram of "*LPDDR3*". The execution of this method is computationally cost effective than sliding window approach. There are two main tasks of this thesis, first to employ an algorithm which can extract features from images and then form a feature vector and second is to design a detector based on features extractor which can classify objects from non-objects. The main input of this thesis is the first part where we have optimized an algorithm to be used on an embedded system.

Key Words: *Locally Adaptive Steering Kernel, Thermal Infrared Images, Features, OSU Thermal Pedestrian database, ODROID Xu4*

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

HOG	Histogram of Oriented Gradients
SIFT	Scale Invariant Feature Transform
ACF	Aggregated Channel Features
SURF	Speeded Up Robust Features
ATV	All-Terrain Vehicle
SUV	Support Utility Vehicle
HRI	Human Robot Interaction
LoG	Laplacian of Gaussian
DoG	Difference of Gaussian
DPM	Deformable Parts Model
SVM	Support Vector Machine
RVM	Relevance Vector Machine
EM	Expectation Maximization
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
LASK	Locally Adaptive Steering Kernels
GPU	Graphical Processing Unit
PCA	Principal Component analysis

CHAPTER 1: INTRODUCTION

Computers nowadays, are seen everywhere in our daily lives. Computers can perform mathematical expressions much faster than humans, perform same task again and again with accuracy, handle large data sets efficiently. There are some tasks that humans perform in daily life on regular basis without even consciously dedicating to that task such as seeing a scene they can recognize objects of different classes ranging from mountains, trees, land, sea, animals, humans etc. to man-made structures like buildings, cars, bicycles, ships, airplane and what not. We can detect humans in almost any pose and the appearance may vary to great extent. We can recognize people by seeing their face, observing their walking gait, listening to their voice without even knowing that we are doing this all together in real time. While doing so we don't really strive to recognize this never ending list of objects that are part of our daily life. Apart from seeing, there are also such domains where we don't focus on them while doing them, including speech recognition and voice recognition tasks, for example if we know someone we can easily know the presence of them just by listening their voice, similar is the case with human visual system. The low level tasks in which computers have surpassed the humans like solving complex mathematical problems, handling large data, it seems natural to make computers perform high level tasks such as computer vision and voice recognition. The main focus of researchers in the field of computer vision" and "image processing" is to make such systems that can see, detect different objects present in a specific scene which means basically understand a scene on the basis of different contents of that scene or in other words extract useful information from image. Computers are currently far behind humans in this domain, thus one goal in "computer vision" and "machine intelligence" is to give computers the ability to see, analyze data visually in images and videos. Firstly, the task is to make computers distinguish among different classes of objects like car is a class, human is another class, inside each class a lot of variations exist, for example car can be sub-classified into sedan, hunchback, ATV, SUV, etc. Such an ability would have many real-world applications.

Computer vision can be stated as a sub field of a wider field of study named as Image Processing. Image processing means performing an operation on an image in order to obtain some useful information from the image. Improvement in the quality of image may include a large number of processes like:

- To restore the image,
- To sharpen the Image,
- processing of videos
- computer vision i.e. vision through robotic machines,
- Remotely sense the images,
- use image in medical field,
- transmission of data,
- differentiate among different colors,
- recognition of designs,
- to process images of microscopic level.

Computer vision is a process to observe surroundings. This is an attempt to make computers see like the humans can see and judge. Obviously this is currently not achievable or in near future, but the work on this has been started and there are significant improvement in this work. Some algorithms if properly trained and applied can perform well in well-defined settings. Object detection is one of the very important field of computer vision. There are different methods of object detection which also include human detection and this Human detection is a topic of commercial value. It has many applications where we need the machines or robots to locate humans e.g. Human robot interaction (HRI), surveillance, autonomous vehicles, pedestrian detection.

In present scenario when the world is facing bomb blasts threats, the demand of pedestrian detection systems has increased manifold. There is also a demand of improvement in available detection methods.

Although a lot of work has been done on object detection and human detection especially pedestrian detection during the last decade [1-5], yet a lot of room to do more work in this field is available for the community.

Earlier works mainly located pedestrians through photographs or videos. These images were taken with visible range sensors. Most of the surveillance systems using visible sensors work in the presence of light, hence dark places especially in the night can be a source of threat. Use of infrared cameras and devices present a solution to this problem. The use of infrared systems is not new but earlier it was mainly utilized for defense purposes or medical instruments probably because this system was not affordable by small organizations. However, with the passage of time, this technology has become affordable and the prices are reducing further, so small organizations are also interested in utilizing this technology. As security has become one of the main concerns of every organization and country in present circumstances, the use of infrared devices is increasing in surveillance systems due to its advantages over conventional visible range sensors. The IR (infrared) devices are improving with the passage of time and their cost is also decreasing but still these devices are not affordable by all customers. The research work on IR systems is in progress and techniques are under way to use IR data especially for automotive applications [6,7]. The efforts are also being made to reduce its cost further.

If we compare IR image with visible sensors (VS) images, we see that VS image is clearer in illuminated areas or day light. For example, clothes texture can be seen in VS image and this is a good feature but in the darkness, VS system does not work. On the other hand, the IR image is not so clear but it can work even in darkness in the night. Further the pedestrians far from the camera can also be detected through IR systems. The spectral sensitivity of thermal IR image sensors fall between 7 to 14 microns. Many factors affect the working performance of these sensors. The surface of the object is one of the factors. Similarly, the material of the object also plays an important role in forming image. These and some other factors can result in misrepresentation of images such as halo effect [8]. Noisy nature of the IR system also creates problems in making clear cut images of the objects. Hence a robust noise handling component is necessary for object detection. Similarly, in order to make the system reasonably effective there must be some component to cover the weak image signals.

CNN system is although fast enough, yet it may take sometimes a lot of time for training [10]. Similarly boosting [9] is also a good detection method but the same delayed training process sometimes creates the necessity of a faster system.

Histogram of Oriented Gradients (HOG) method although gives solution of pose detection, yet runtime costs make it unaffordable. Further in this method detection of parts is not possible if the object is too far away. Some studies have depicted that a rigid detector can give very good results if wisely designed. So, we have mainly utilized this technology.

1.1 Object Detection

Detection is the ability of the system to distinguish an object from the background while Recognition can be defined as the ability to classify the object i.e. animal, human, vehicles etc.

Object detection is to find occurrences of objects from a specific class in an image. The purpose of object detection is to detect all objects of a class such as people, vehicles or faces in an image.

1.1.1 The problems in human detection in Thermal Infrared Images

Due to noisy nature of image, the Human / Pedestrian detection is difficult in thermal infrared pictures. The other problem is low resolution of such images.

Sensor noise is much higher in infrared images, and features are very less varying in feature space as compared to colored natural Images.

Textures are visible in natural color images while these are restricted in infrared images. Hence in learning process, it is very complicated to separate between background and humans in feature space.

Existence of large amounts of variations in images and videos counts to the foremost challenges faced in design of visual object detector. There are many factors that may have certain effects on this like:

• Image formation process loses the depth information of the object with respect to the capturing lens which results in loss of information about size, shape etc. of object. The shape of object actually varies respective to the pose and orientation of camera relatively to the object being captured, that is why it is difficult to class an image as object or non-object. Another issue needs to be addressed is about the number of variations in scale of an object in which it can appear. A good object detector must cater for such variations due to viewpoint and change in scale of object and should be invariant to such changes.

• The second challenge faced in object detection is the variation in natural occurrence of objects i-e within a class an object may vary in shape or appearance, for example there is a class of objects car, now within this class lot of variants exist, a car can be of many types as mentioned above in introduction. For instance, in human class the objects vary in pose, varying ways in how they cloth and many more factors which makes it quite a challenging task to design a detector capable of classifying objects because the detector should be robust enough to be invariant to such changes in pose, clothing and appearances.

• Thirdly, the images and videos taken in natural environments seem to be a challenge for object detector because of the cluttering background which also varies from image to image. The detector often confuses the features of background clutter with the foreground object that needs to be classified hence leading in false positives or sometimes no detections, so this must be handled with proper precautions should the detector be able to separate the background from the foreground.

• Here comes another challenge that aforementioned two challenges conflict each other such that if a more class specific detector is designed it will perform very well and will give very less false detections on images with cluttered background but on the other hand will miss many instances of that object due to varying poses and appearances. To address this problem if the detector is made more generic than the number of false positives will increase due to cluttering environment, so these two conflicting challenges should be handled simultaneously.

• Detecting objects in partially occluded environments is also a difficult and challenging task because the objects are not fully visible and some parts of objects are hidden with each other. This is a major problem specifically in human detection as humans may appear in groups of many people and may be occluded intra-person or with other non-class objects.

1.2 Applications of Human Detection

The main focus of our research is finding individuals in images. Person detection has evoked the attention of researchers in the recent years though it is a tedious and challenging task. For instance, take example of a normal digital camera owner, most of the times he takes pictures during parties, family get togethers, vacations etc. and could easily reach a picture count of 15000+ during 3 to 4 years, it can be a wearisome task to locate the pictures of a specific person, or to find all the pictures of some person or pictures taken at a specific location. Thankfully, intelligent people tagging software can be designed now which will intelligently tag images with the name of person, so that when required user can easily search and find the relevant content. Nowadays, there is excessive ongoing research on autonomous cars, it is the future, such cars also need to have robust visual system so that they can see and drive accordingly, an important part of such visual system should be a pedestrian detector which will alert the driver and the car regarding any passing by pedestrian and try to minimize any danger of run-over or collision.. Human detection is slightly different from ordinary object detection because humans are occluded both inter and intra-person and illuminations and poses vary from picture to picture which makes it quite a challenging task. Human detection techniques can be divided into two categories depending upon their approach, i-e, parts-based [50-52] detection and sub window-based [53-54] detection. In parts-based detection technique a specific location of specific parts constitutes a human body, while in features-based detection a feature extraction algorithm is used, this feature set is then compared with feature sets of training data.

Common applications of human detection include surveillance, search and rescue and customer analytics (for example, people counters).

1.3 Objective

This thesis aims at the problem of visual object detection in images. Specifically, it focuses on design of visual object detection from image processing point of view such that in a given image they can look for desired objects and localize them. There are two main tasks of this thesis, first to employ an algorithm which can extract features from images and then form a feature vector and second is to design a detector based on features extractor which can classify objects from non-objects. The main input of this thesis is the first part where we have optimized an algorithm to be used on an embedded system. To resolve the above mentioned challenges/limitations, the aim of this work was to implement method that can detect humans in thermal infrared images on computer and as well as on embedded system.

1.4 Research Methodology

Human detection in thermal infrared images postures a difficult challenge because infrared images have very low resolution. We have used Local Adaptive Steering Kernel (LASK) method to extract features from infrared images on the basis of image geometry concisely. We perform training less detection of pedestrians in low resolution infrared images by creating a rigid detector template. We detect humans by cross correlation in frequency domain. The cross correlation is between features of detector template and target image features. This speeds up the detection as compared to sliding window approach.

Our implementation was carried out on benchmark thermal infrared images of dataset "*OSU Thermal 01 Pedestrian database*". We implemented this method on dell 6th generation, core i5, 2.4 GHz with 8GB Ram on MATLAB and on single board computers ODROID Xu4. It has "*Samsung Exynos5422 Cortex™-A15 2Ghz and Cortex™-A7 Octa core*" alongwith 2GB Ram of "*LPDDR3*". The execution of this method is computationally cost effective than sliding window approach in addition to having a much higher speed.

1.5 Thesis Structure

This chapter introduces the idea of Human Detection and its application and defines the problem of this research – How to detect pedestrians in thermal infrared Images. In the following chapter, we discuss some notable contribution to Human detection methods. The proposed methodology is presented in detail in chapter 3. The implementation on Embedded systems and Laptop is discussed in chapter 4. The Dataset on which human detection is performed and their results are presented in chapter 5 along with the analysis of results achieved. Chapter 6 then concludes this document and presents potential research directions on this problem.

CHAPTER 2: LITERATURE REVIEW

If we study the history of object detection, we would have to go beyond two decades and if we imagine its future, the object detection methods may take more than two decades for further improvements especially in human detection methods.

In the beginning i.e. 1980s, object detection was being dealt under the umbrella of visual recognition. The work on visual recognition was done under two major disciplines. In one discipline the idea was to make image prior to analyzation. These models were named as compositional models [14-15]. In the second method constraints having two or more dimensions were used [13]. Both these disciplines were basically geometry based and did not work when big datasets required algorithms. However the methods served a lot for developing recent models of object detection.

Viola and Jones in early 21st century [16] did a reckonable work. Present day object detection methods have been derived from their work. Machine learning was also initially used by them.

Constellation [17] is a larger frame of models. It includes Pictorial Structure method [18-19] which is basically a geometrical method. Present deformable part model is a modern shape of that method. An object detection method basing on parts is also known as Implicit Shape model [20-21]. These methods normally use rigid templates. With the passage of time datasets sizes increased. Consequently, a need arose for matching existing methods with machine learning algorithms. The work done on this new technique was named as Kernel method [22]. A large number of good detectors were prepared by applying this technique. Dalal & Triggs [23] are well known for designing template features for best detectors of that time. It involves training process which gives better results that earlier methods. This detection method can manage more variations of object. Even pose can be detected by this method [24]. Mixtures of different parts are used in this method to detect different types of objects. However, feature learning was not covered in this method which was a big drawback of this method.

Feature learning lead to another method of detection. This method is known as boosting method [25-26]. The name of Piotr dollar is remembered for this work [27-30]. This type of detector can easily detect people walking or standing [31-32].

Convolutional net working [33] brought very useful progress in detection methods. Learning illustrations have been adopted by most of the people working in this field. Region based proposals are common in this method [34-38].

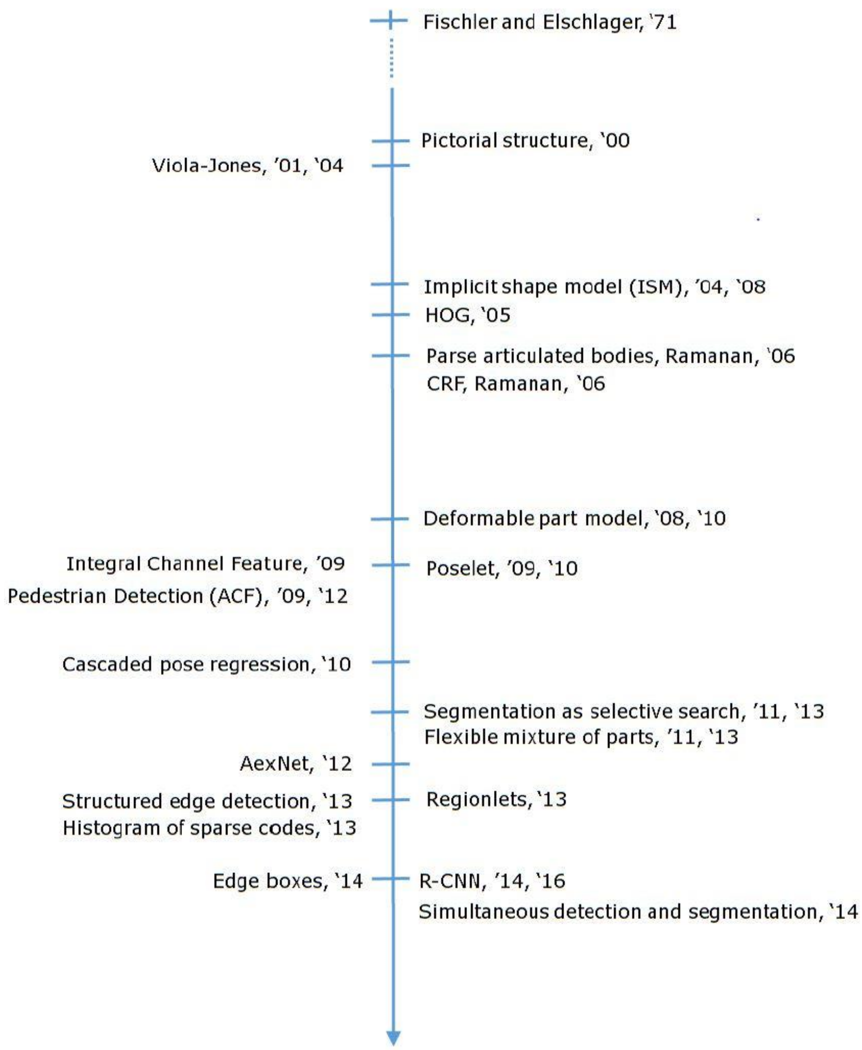

Figure 2.1: History

Despite above mentioned progress in the field of object detection, a lot is still to be done. Training needs further improvement. It needs to be made more efficient. Orientations also create problems which need to be resolved. Run time also needs attention of this community. Part complexities is still a question which needs to be solved. There are two categories in which the previous work on object detection can classified.

- The image representations or in other words feature vectors that are used
- The design of framework which uses these representations.

2.1 Image Features

Image feature set should be robust enough to extract the most important and relevant information from the image, which should be invariant of any illumination changes, colour changes, scale or appearance invariant. One way of achieving this is in spite of using original image gradients and intensities as features one should use local advanced image features. Such features can based on point features [39-40] Laplacian of Gaussian (LoG) [41] or Difference of Gaussian (DoG) [42] intensities [43-44] gradients [45-46] texture, colour, or a combination of many of them [47]. The many approaches are further divided into two main categories: (i) dense representations which use intensities and gradients of images, (ii) point based, parts-based sparse features.

2.1.1 Sparse Local Representations

The use of key point detectors and parts-based detectors results in formation of pertinent local image regions upon which sparse representations are based.

Point Detectors

There have been pervasive work on object detection using local point detectors, major contributions are reviewed in this section [48-55] and [42]. Above mentioned approaches use sparse local point features extracted from local points or keypoints or area of interest. The supposition here is that the image regions that are more stable and reliable are selected by keypoints so it can be said that more the image regions are informative more the success rate will be, so a great emphasis should be given to selection of keypoint features. Forstner and Harris [56,57,39] Scale Harris-Invariant Laplace [39], Difference of Gaussians [50] are some of the commonly used keypoint based detectors. They are usually designed to fire up on desired objects or blobs, so they might not perform well when specified to work with particular within class object classes, however, local scale and dominant scale information can be provided by using DoGs and H-Laplace keypoints. There is another advantage that such point features use sparse image regions which means that they will be compact as compared to pixels used as features so this an advantage in using such keypoint features. There are many approaches about using such keypoint features, i-e they can be used individually each at a time or combination of more than one features, like Kadir et al's [43] region detector was the basis for the entropy used by Fergus [52], similarly, the interest point operator of Forstner et al [57] is used by Agarwal [55] and Weber [53], and Harris [39], H-Laplace [40] and Laplacian of Gaussian [40] are used by Dorko [51] .

The other famous approaches that use areas around the keypoints as feature representations and descriptions are *SIFT* (Scale Invariant Feature Transformation) [58-59] and features using *shape contexts* [60-61]. Above mentioned representations use image gradients by computing local histograms over image gradients surrounding the keypoint regions and local histograms computed over edges, SIFT particularly uses a framework in which feature are extracted by the voting performed by keypoint detector, voting is done depending upon the scale and orientation of feature vectors and then the weights are computed as to which features are more dominant than the other, in this way SIFT selects feature representations. The scale information is required to define the appropriate smoothing of scale which are then used to compute image gradients, as for SIFT rectangular grids of histograms are computed however, log polar grids are used in shape-context features. The shape-context [61] firstly used 2 dimensional spatial histograms which was further extended by [62] which uses 3 dimensional histogram of oriented gradients, the both approaches are somehow similar to the one used in SIFT.

Parts-Based Detectors

Another type of detectors used in object and human detection systems are parts-based detectors, as the name suggests, they basically detect within object class objects, as the example of bicycle is mentioned in chapter 1, each part is detected individually and then a set of such parts is classified a bicycle. For instance, take example of human body parts detector, which employs each limb as an approximation of cylinders, to achieve these parallel detectors are used which are applied on different image segments, some are based on body geometry and applied on corresponding regions of images by using articulation limitations, graphical models of human body parts. Recently, 3-Dimensional limb detectors [63] are also being used to detect different body parts in parallel with each other. [63-66] and [45] all use parts-based modeling and detection for classification of human body. The parts used are mainly divided as “upper leg”, “lower leg”, “upper arm”, “lower arm”, “torso”, “head” and “neck”. There is one difficulty in using this technique is the assumptions made are too generic or may seem quite simple to one because the representation of legs or arms by straight line may not produce desired results due to the huge variations in poses.

2.1.2 Dense Representations

In this approach the features are extracted densely whether over a detection window or an image is used, then the features are gathered into a large feature vector which is very descriptive by using this feature vector the classification is done about an image as positive or negative. High order differential operators, gradients and intensities are the conventional basis of such representations.

Regions and Fragments Based on Image Intensity

Eigenfaces approach coined by [67] and [68] in which large feature vectors are formed by combining separate feature sets of same size face pictures and afterwards, re-organizing them in such a way that the features become invariant to usual variations in images by using PCA (Principal Component Analysis) [69] have proposed a method in which face detection is carried out on intensity images by applying histogram equalization which simply corrects the intensities of images and then a neural network classifier is applied for further classification [70] as positive or negative detection. Fragments are produced arbitrarily, with the most important ones being chosen eagerly by boosting the common data between the part and the class. Pairwise measurable conditions between fragments is likewise used to make non-linear tree based models. The creators show moved forward discovery execution with respect to wavelet based descriptors (portrayed underneath), yet constrain their assessments to generally rigid article classes, for example, countenances and autos [71] and [44].

Edge and Gradient Based Detectors

Edge and image gradient based detectors are also popular among other pedestrian detection approaches, Gavrila and Philomin’s work [72] proposes a method of pedestrian detection which uses edges as features and for classification they use chamfer distance and apply on learned exemplars. Gavrila et al [73] have used this to practically implement a human detection system. Ronfard et al [45] and Mikolajczyk [46] have used image gradients as features for their human detection system. An ariculated detection system is built by using SVM (Support Vector Machine) classifiers to detect body parts which is comprised of 1st and 2nd order Gaussian filters by Ronfard et al [45] similar work is done by Felzenszwalb et al [74] and Ioffe et al [75]. Mikolajczyk [46]. They have designed features that are specifically optimized for human parts detection.

Wavelet Based Detectors

Dense encoding of image regions is used in many detectors that are designed on the basis of Haar-wavelets, some examples of such detectors are the famous Viola and Jones [76], Papageorgiou et al. [77] and Mohan et al. [78]. The absolute value of Haar-wavelet coefficients is used particularly in the work of Papageogiou et al.[77] in which they first compute local descriptors from different orientations and varying scales then the spatial analysis in which pixels are used to map images in a rich way so that the description can be portrayed from patterns. They used three kind of wavelets i-e horizontal, vertical and diagonal, each have its own importance in this technique. Translation Invariance can be achieved by deploying over-complete basis by overlapping. There are two kind of SVMs that can be used to classify objects, (i) one used by Papgeorgiou et. [77] in which a SVM based kernel is used in such a way that the decision is based on the learned kernel distances from training examples. (ii) linear SVM, both classifiers perform comparable to each other, there is not any significant difference in performance of these two approaches. Linear SVM is basically the weighted sums of rectified wavelets such that the final decision criteria is a weighted sum of training examples' kernel distances. According to Viola and Jones [76] and Viola et al. [79] a generalization of Haar wavelets can be used to design a progressive rejection based classification chain. The chain works as rejection stages, each progressing towards depth, with the depth the complexity of the classifiers starts to increase, the basic idea that at first naive classification is done on images, if the classifiers at initial stages reject the image than there is no need to apply complex classifiers. AdaBoost is used as a greedy feature selector, such that the all the dimensions of rectangle are tested and weights are reassigned. Viola and Jones [76] used this method to design a real-time face detection system and then implemented a pedestrian detection system in Viola et al. [79].

2.2 Classification Methods

Many classification methods exist, they are mainly divided into two major categories depending upon the underlying approach, (i) Discriminative and (ii) Generative.

2.2.1 Discriminative Approaches

Such machine learning classification techniques due to their ease of use, automatic selection of features of relevancy and performance have become quite popular in machine learning community. Common examples of discriminative classification techniques are Support Vector Machines (SVMs) and boosting [80-84].

Support Vector Machine (SVM) Classifiers

Now it's been a while the SVMs are being used for classification in the field of machine learning. The principle of SVM is that it separates the classes with a hyperplane by maximizing the distance between object class and non-object classes, regardless of the version i-e either input features space based or kernelized version. Papagrorgiou [77,85] and Mohan [78] have used SVM in parts-based detection system. The later used a two stage SVM classifier in such a way that the first stage uses Haar wavelets to form part detectors and the second stage combines them to gain the final object detector capable of detecting humans. Ronfard [45], Dorko [51] use SVMs as an intermediate stage classifier, in a way that Relevance Vector Machine (RVM) and 15 SVMs are used to make base limb detectors on first and second order image gradients, however similar to Ioffe and Forsyth's [75] the final classification is done by programming dynamically

over the part detections. Dorko [51] have used SVM based classifiers as intermediate limb detectors for object detection over interest points.

The final two types of classifiers are tested (i) likelihood ratios of detecting parts 2.1 (ii) mutual information among detected parts and object classes.

$$\frac{\boldsymbol{S}(\boldsymbol{part}|\boldsymbol{object}=\mathbf{1})}{\boldsymbol{S}(\boldsymbol{part}|\boldsymbol{object}=\mathbf{0})} \qquad \textbf{(2.1)}$$

Cascaded AdaBoost

A collection of weak classifiers is combined to get a strong classifier, in the field of computer vision it is used by matching the pattern that is most relevant to be rejected at every level of cascade, known as cascades of pattern rejectors. The training time of AdaBoost is slower however, their performance in run time comparatively better as compared to SVMs depending upon the selection of feature encoder. AdaBoost is used to train weak classifiers for face detection and pedestrian detection in Viola and Jones [76] and Viola et al. [76], in doing so spatial and temporal difference of rectangle features were used. AdaBoost is also used to train weak interest point based classifiers in [49,65,66] have proposed a model in which the parts are defined as functions of wavelet coefficient specific groups with a coordinate frame used to represent them respectively. Using this technique, the relationship (geometrical) between parts is captured and hence, by combining naive Bayes independent combinations of probable proportions 2.1 the final classifier is build. In the work of [46] 2.2 are used as week classifiers by relaxing the assumption that the pairs of descriptors be used as weak classifiers should be independent. [86] have used oriented histograms by using cascade of rejectors.

$$\frac{\boldsymbol{S}(\boldsymbol{part}|\boldsymbol{object}=\mathbf{1})}{\boldsymbol{S}(\boldsymbol{part}|\boldsymbol{object}=\mathbf{0})} \qquad \textbf{(2.2)}$$

2.2.2 Bayesian Graphical Models

[44] and [71] propose a method in which image fragments are selected so that the mutual information can be maximized among the object class label and fragment so that a useful and descriptive information can be extracted. In their approach Naive Bayes classification is used which proves that no significant improvement is achieved by applying Bayesian networks on the original fragment repetitively. [53] uses generative Bayesian models that are learned using EM also use similarity ratios 2.1 for classification among object class and non-object class. [52] also used similarity ratios but that also includes the scale, appearance and other conditional probabilities such as position and scale of features.

2.3 Fusion of Multiple Detections

A typical binary classifier slides a detection window upon different scales of an image of interest which in return gives detections more than one of that object and also overlapped, which is basically a single detection so a good practice is to merge the. According to [69] the number of specific detections under a specific region or neighborhood is greater than a specific user defined threshold then the centroid of all the detections is reckoned which is then treated as the location of detection. Such centroids are calculated in 3 Dimensional space and the detection score is given by the number of detections while scale value represents the region for detected object. The bounding boxes having lower score values are removed hence, still existing detections conclude the final result. A simpler method was proposed by [87] which incorporates disjoint subsets, each such subset consists of detections which are partitioned such that each partition thus represents a final detection. The detections with overlapping bounding regions belong to same partition and for final detection average of all partitions is calculated.

CHAPTER 3: DETECTION METHODOLOGY

3.1 Locally Adaptive Steering Kernels (LASK)

The proposed methodology is presented in detail in this chapter. Properties of LASK are

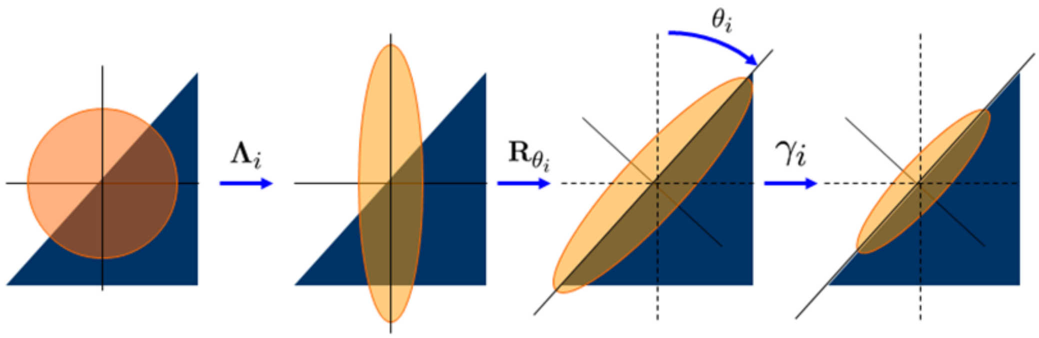

Figure 3.1: Locally Adaptive Steering Kernels

- Kernel adapted to locally dominant structure
- The steering matrices scale, elongate, and rotate the kernel footprints locally.
- Local dominant orientation estimation

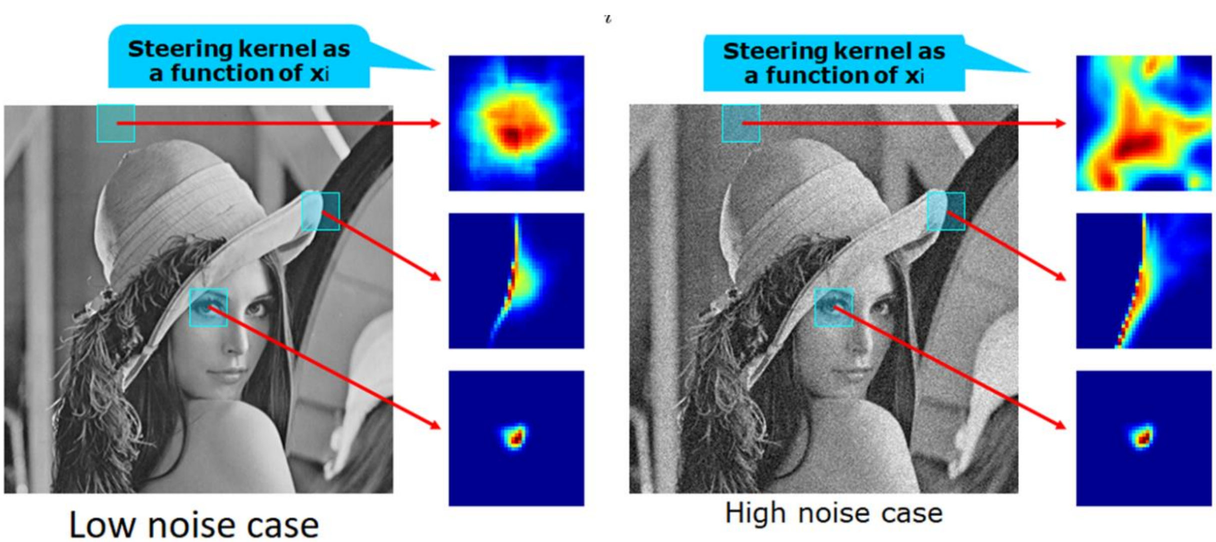

Figure 3.2: Stability of Steering Kernels

Steering approach is stable even in the presence of significant noise.

Here ST stand for steering matrix (gradient matrix). The rectangular window (w) centered at i^{th} pixel and $I(p_i)$ is the intensity value at i^{th} pixel of image (I).

In a single pixel ST is based on gradients. A single pixel however makes the ST unstable. Therefore, ST is averaged over a rectangular window.

$$\boldsymbol{ST_w = \sum_{i\in w} \begin{bmatrix} \frac{\partial I(p_i)^2}{\partial x} & \frac{\partial I(p_i)}{\partial x} . \frac{\partial I(p_i)}{\partial y} \\ \frac{\partial I(p_i)}{\partial x} . \frac{\partial I(p_i)}{\partial y} & \frac{\partial I(p_i)^2}{\partial y} \end{bmatrix}} \qquad \textbf{(3.1)}$$

After decomposition of steering matrix:

$$\boldsymbol{ST_w = \lambda_1 U_1 U_1' + \lambda_2 U_2 U_2'} \qquad \textbf{(3.2)}$$

Here U_1 and U_2 are the eigenvectors representing principal directions and λ_1 and λ_2 are the eigen values.

To avoid numerical instabilities, we used Riemannian matrix by keeping the eigenvectors unchanged. In order to minimize the presence of noise in image, we can change parameter value (α) to speed up and limit the information of local gradient. The smoothing function for steering matrix is followed as:

$$\boldsymbol{ST_w = (ab + \epsilon)^{\alpha} \left(\frac{a+1}{b+1} U_1' U_1 + \frac{b+1}{a+1} U_2' U_2 \right)}, \qquad \textbf{(3.3)}$$

Where ϵ is set at 10^{-1} and a and b are the singular eigen values as $a = \sqrt{\lambda_1}$ and as $b = \sqrt{\lambda_2}$

The LASK is stated as the below mentioned function between a pixel p_i and its neighboring pixels p_j .

where $j = 1,2,\ldots,n^2$ and n is the size of window.

$$\boldsymbol{L = \frac{\exp(-\Delta p_{ij} ST_{w_j} \Delta p_{ij}')}{\sum_j \exp(-\Delta p_{ij} ST_{w_j} \Delta p_{ij}')}, j = 1, 2, \ldots n^2} \qquad \textbf{(3.4)}$$

Where $\Delta p_{ij} = (p_i - p_j)$ is the difference between pixel p_i and p_j.

3.2 Methodology of Person Detection

Overall methodology to detect humans is shown in the figure below. Procedure is divided into four major steps. Detailed procedures are explained in the next sessions one by one.

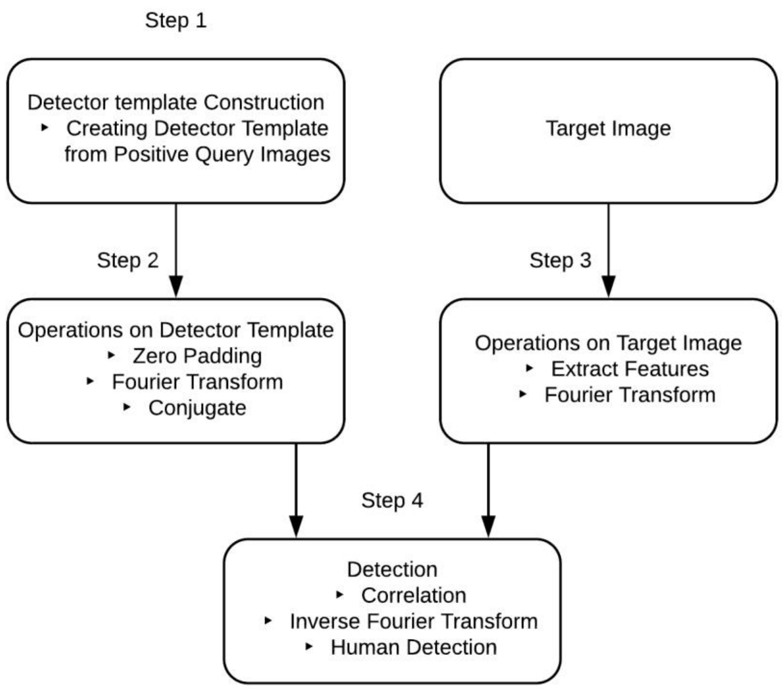

Figure 3.3: Methodology

In first step, detector template is constructed by taking only positive images. After that zero padding is performed on detector template with (FFT) “Fast Fourier transform” and then its conjugate.

First two steps were performed and results saved. For the run time detection, operations are performed on target image and then correlated with results of target image and saved conjugate of detector template. Human detection is executed by converting frequency domain to time domain by executing “IFFT”.

3.2.1 Detector Template Construction

Overall construction is shown in the figure 3.4. First take the positive images of humans of size 36x28. After applying LASK, we get the nine features of query image. To reduce the feature space, “Principal component Analysis” (PCA) is performed by taking 70 % of energy of features.

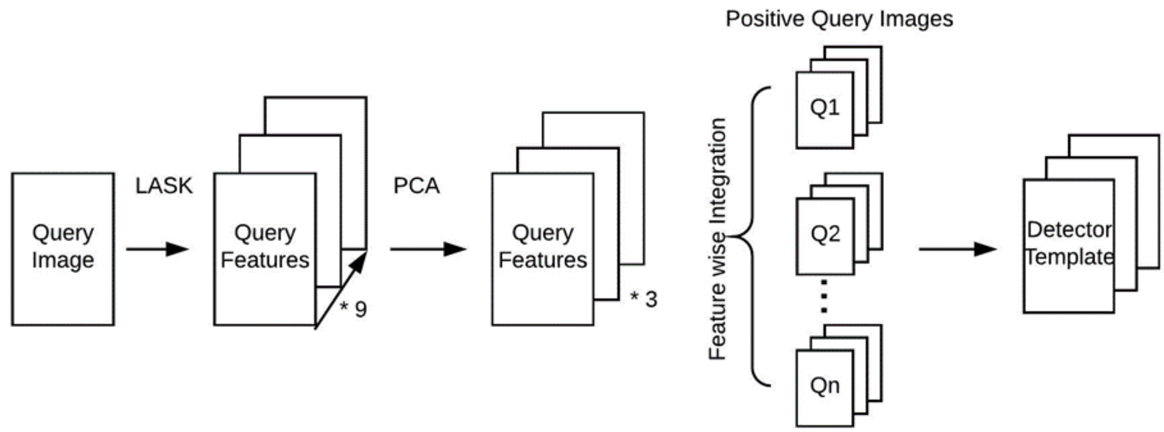

Figure 3.4: Detector Template Construction

Top three features of query images are taken. To make the detector template, feature wise integration is performed to make a detector template with top three features which are common.

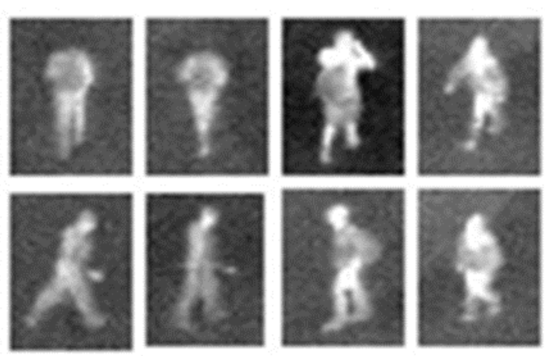

Figure 3.5: Positive Query Images

Humans in different poses are shown in figure 3.5. These positive images of dimensions 36x28 pixels are extracted from dataset. Detector size of 36x28 pixels is made with top three features.

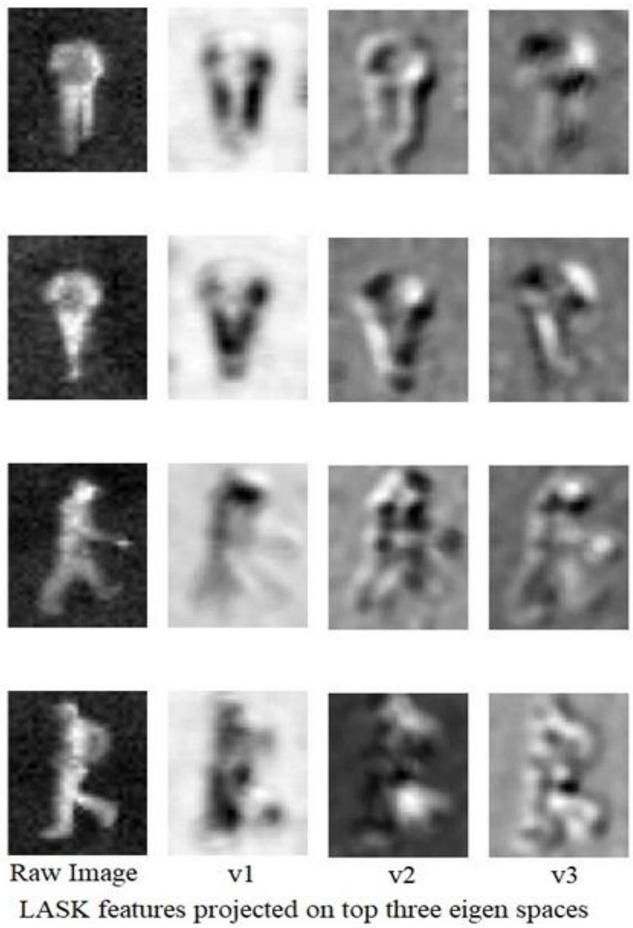

Figure 3.6: LASK top 3 features

Figure 3.6 shows the feature space of positive images. After applying PCA, three top features are extracted. These features were obtained by projecting three eigen vectors on LASK features. Above figure is three times bigger than the original image.

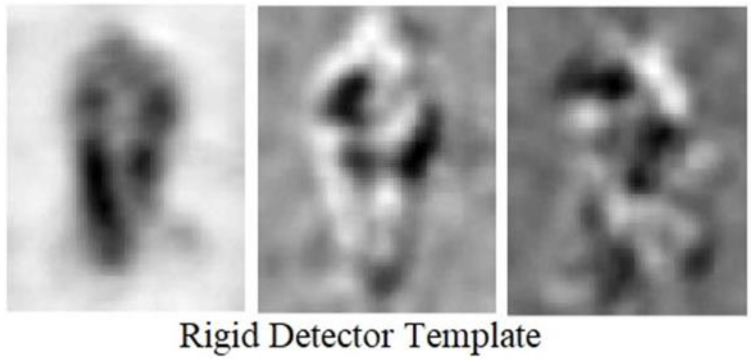

Figure 3.7: Template with Top 03 Features

This above figure 3.7 is final detector template. These are the first three features from the left side. This final template is searched within target image to detect humans. To detect humans, we use cross correlation in frequency domain because sliding window approach is computationally expensive than cross correlation.

3.2.2 Operations on Detector Template

After construction of detector template following operations are performed as shown in figure 3.8. First apply FFT after zero padding. We get three features in frequency domain. After that conjugate these features and save the resulting features to perform run time detecting operations.

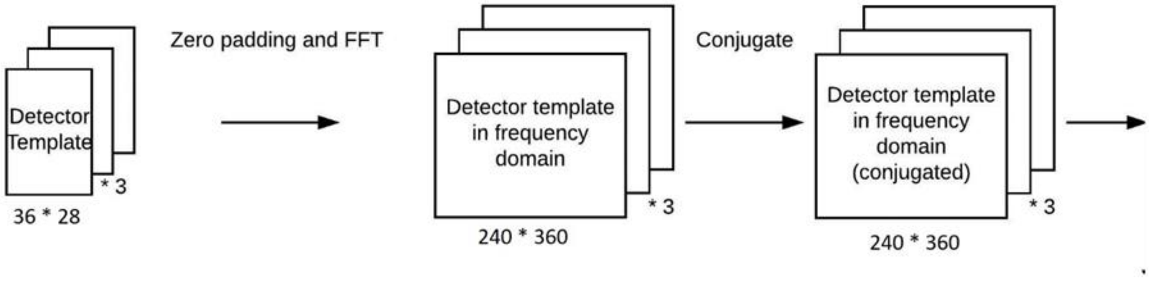

Figure 3.8: Operations on Detector Template

Figure 3.9: Conjugate Detector template in time domain

Figure 3.9 shows features of detector template conjugated. These figures show in time domain converted from Fourier domain by taking IFFT. By taking the conjugate, images are inverted. Features of detector template in Conjugated form in Fourier domain are saved. These features of detector template are correlated with target image features which is described in next session.

3.2.3 Operation on Target Image

The first step of run time calculations is to extract features of target image by using LASK. After applying PCA, top three features are extracted as shown in figure 3.11. These features are then converted in Fourier domain by taking FFT as illustrated in figure 3.10.

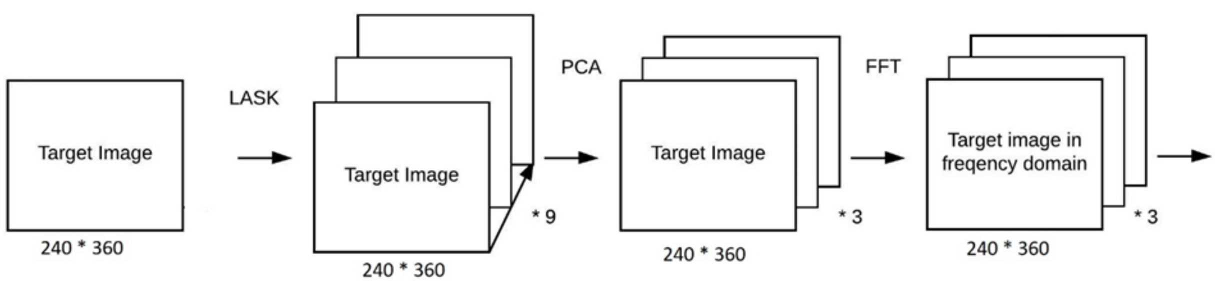

Figure 3.10: Operations on Target Image

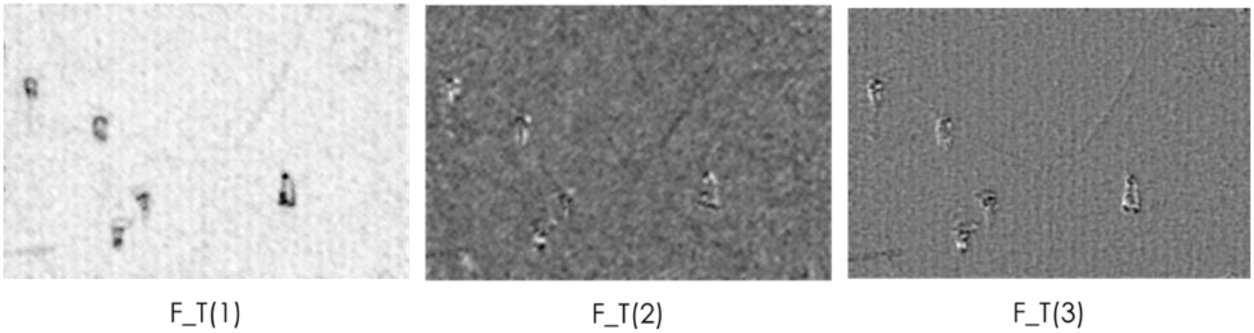

Figure 3.11: Top 03 Features of Target Images

3.2.4 Detection

Overall detection procedure is illustrated in figure 3.12. Cross correlation is done in frequency domain which is basically pointwise multiplication. Correlation is performed between detector template in conjugated form and target features in frequency domain. After correlation, we get three features. These three features are then summed pointwise to get 2D Fourier image.

$$\frac{\boldsymbol{\mathcal{F}^{-1}}\ \{\ \sum_{\boldsymbol{d}} \boldsymbol{\mathcal{F}}\{\boldsymbol{F_T}(:,:,\boldsymbol{d})\ \}.*\ \boldsymbol{\mathcal{F}}^{\dagger}\ \{\boldsymbol{F_Q}(:,:,\boldsymbol{d})\}\}}{||\boldsymbol{F_T}||^{\boldsymbol{2}} * ||\boldsymbol{F_Q}||^{\boldsymbol{2}}} \qquad \textbf{(3.5)}$$

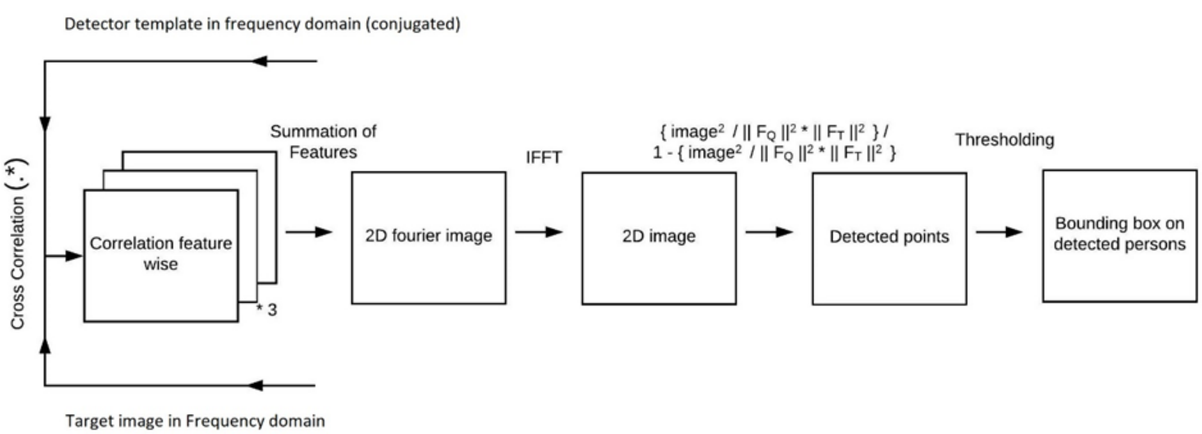

Figure 3.12: Detection Procedure

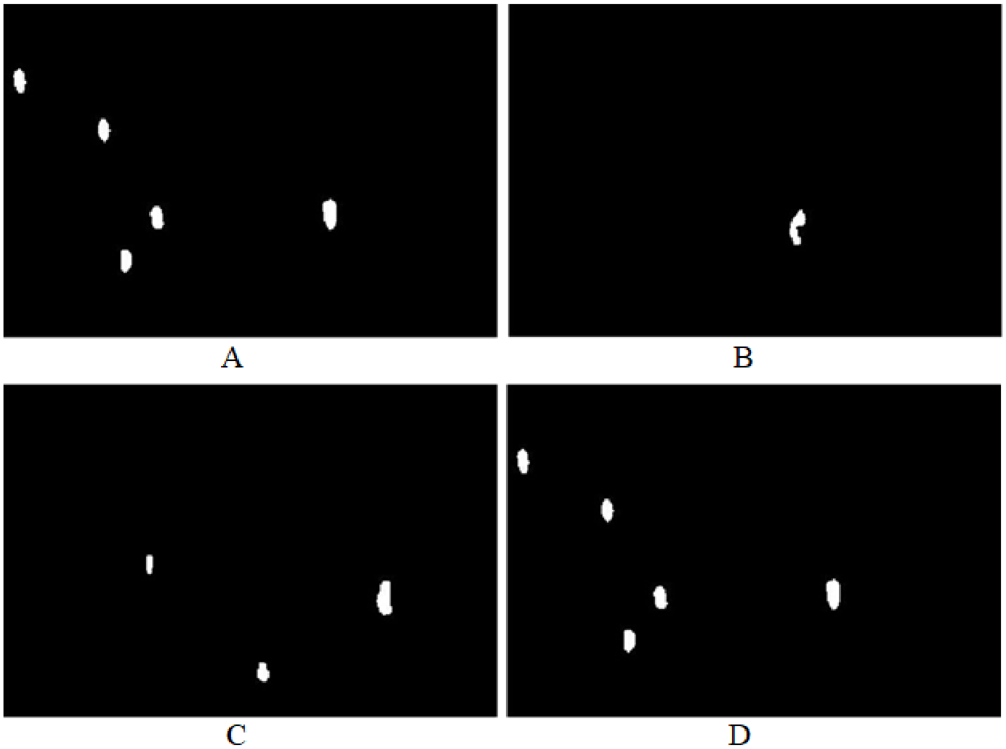

Figure 3.13: Detected Points on 04 different Target Images in Time domain

After applying IFFT on this resultant image, we get detected points in time domain as shown in figure 3.13.

These detected points are then extracted in coordinate form. Bounding rectangular box of size 36* 24 pixels is drawn on extracted coordinates on original image figure 3.14.

These detected white points are maximum likelihood that matches the detector template features in target image. The image we get after IFFT is normalized by product of $||F_T||$ and $||F_Q||$ to suppress the false alarms in equation 3.5.

- F_Q ----- features of query image.
- F_T -----features of query image.
- d ------ No. of features
- $\mathcal{F}$ -----FFT.
- $\mathcal{F}^{\dagger}$--- Conjugated
- $\mathcal{F}^{-1}$--- IFFT

- $\|F_T\|$----frobenius norm.

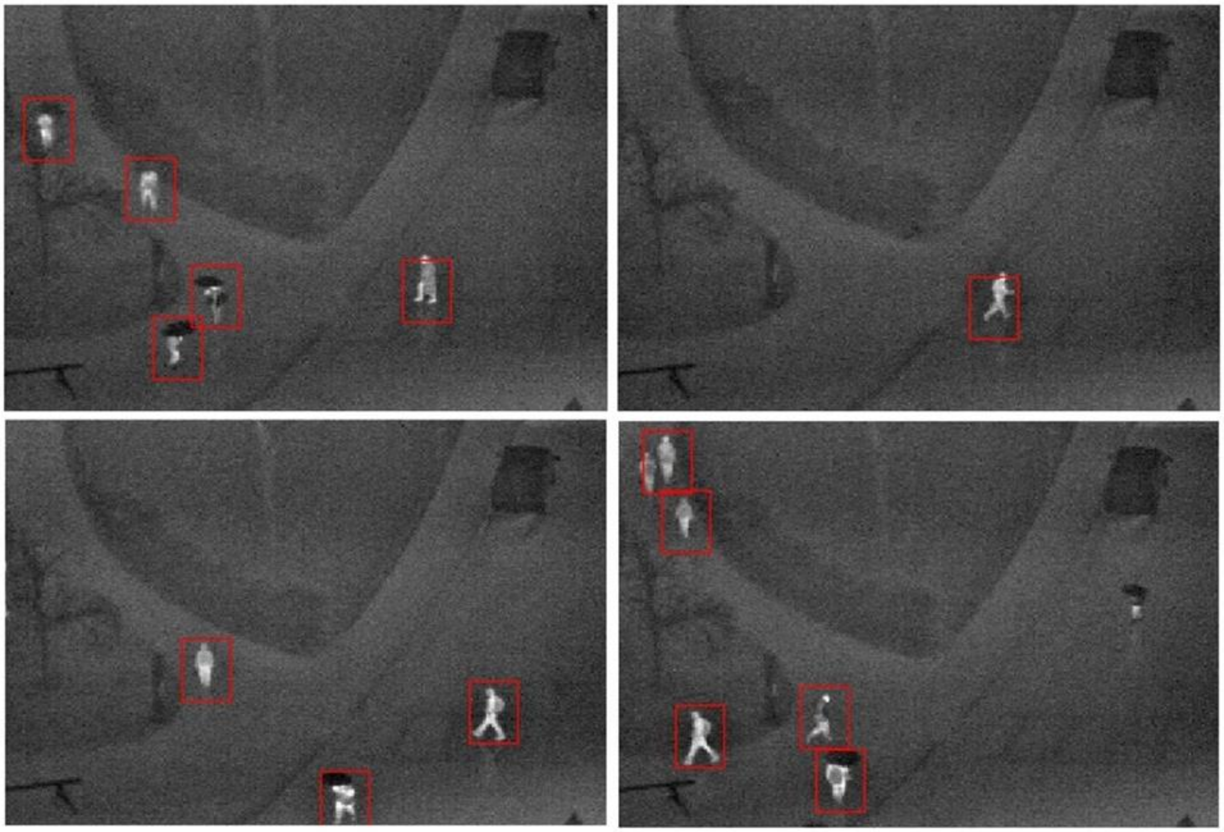

Figure 3.14: Results on 04 different Target Images

CHAPTER 4: IMPLEMENTATION

The implementation phase is divided into two phases, first the detection algorithm was implemented in MATLAB on Laptop and then the same algorithm was implemented in C++ on ODROID xu4. Then the results were generated and compared which is discussed in chapter 5. The implementation details on embedded systems and computer are described later in this chapter. The following sections provide hardware details of our framework.

4.1 Embedded System Details

The embedded system used for this project is ODROID Xu4 which is basically credit card sized computer.

4.2 ODROID Xu4

The processor of ODROID Xu4 is “ARM® big. LITTLE™” technology. It is “Heterogeneous Multi-Processing”.

Figure 4.1: ODROID Xu4

4.2.1 Layout of ODROID Xu4

The detailed block diagram of ODROID Xu4 in figure 4.2.

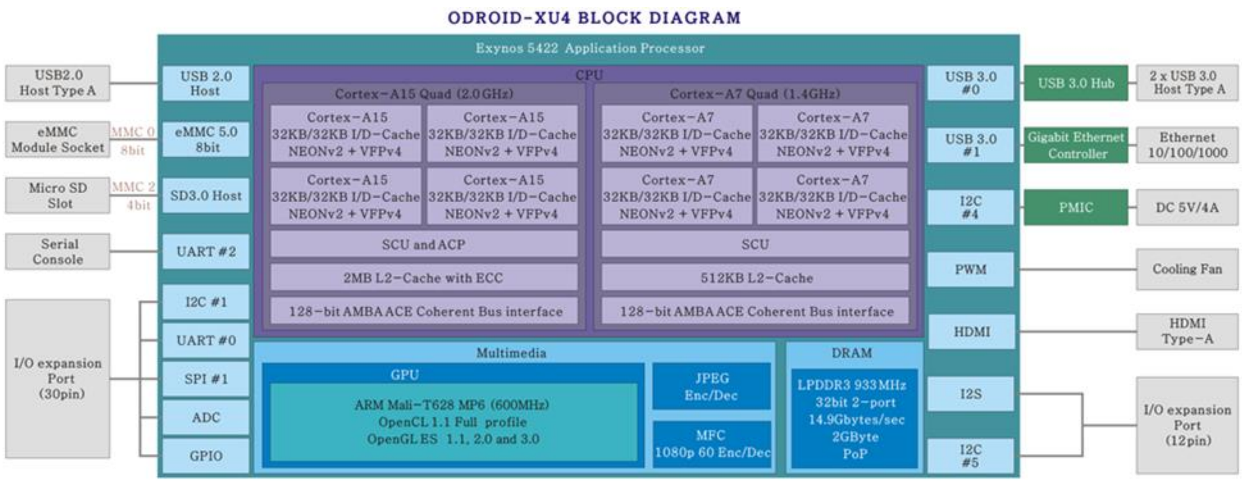

Figure 4.2: Block Diagram

4.2.2 Board Details

Board details are shown in figure 4.3. Every controllers, slots, ports, buttons, connectors, pins

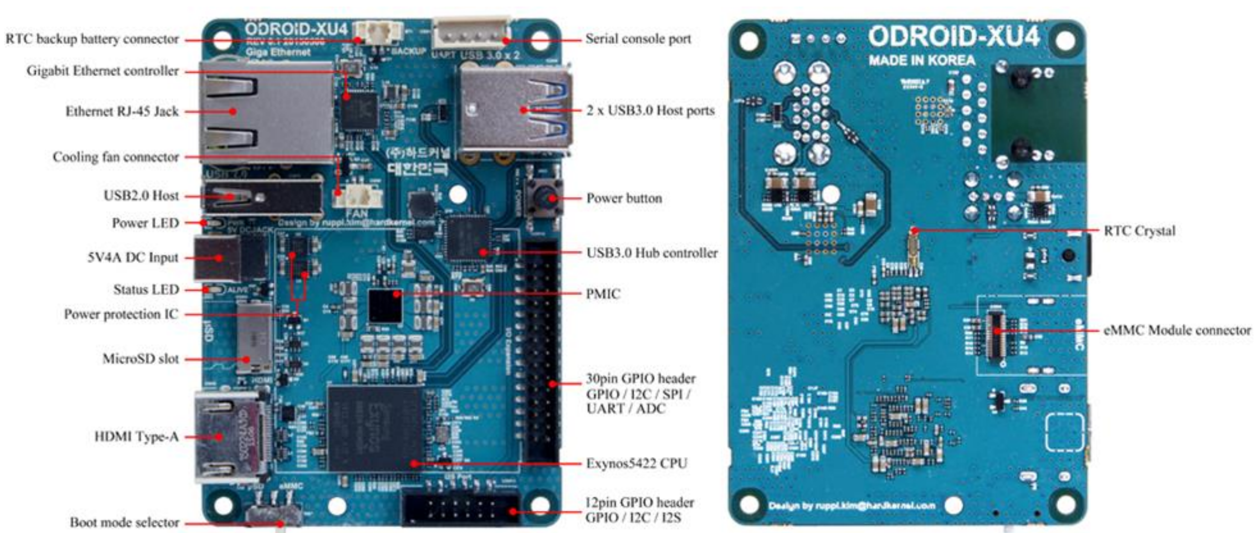

Figure 4.3: Board Details

and more are shown in figure 4.3.

“General Purpose (GPIO)”

It has Input as well as Output ports which can be utilized for General Purpose (GPIO) as shown in figures 4.4 and 4.5 respectively. Detailed information about GPIO are also shown in figures.

It has “30-pin GPIO port” which is operated as “IRQ”, “SPI” and “ADC”. It has also “12-pin GPIO port”. Electronics and robotics utilize this port as “GPIO/I2S/I2C”. Buttons and LEDs can be interfaced with the help of GPIO pins through small Linux controller. Interfacing

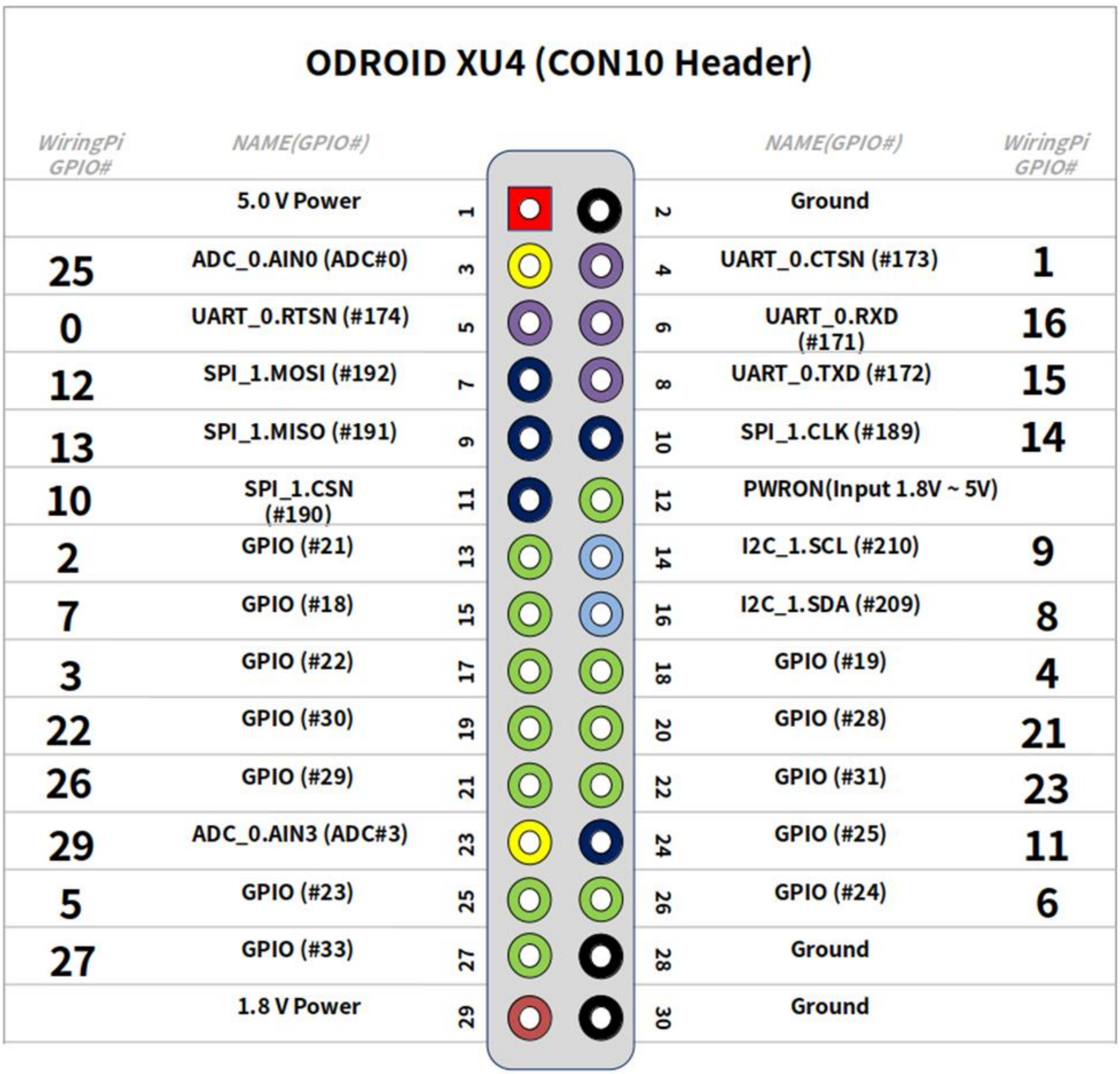

Figure 4.4: CON10 Header

with the pins is handled through WiringPi (a useful library) for developers of Python or C and C++. The ADC inputs have been restricted to 1.8Volt and all of the "GPIO ports are 1.8Volt". The XU4 has a Level Shifter Shield and in case sensor or peripheral needs higher voltage, it helps in level shifting of the GPIO ports from 3.3V or 5V.

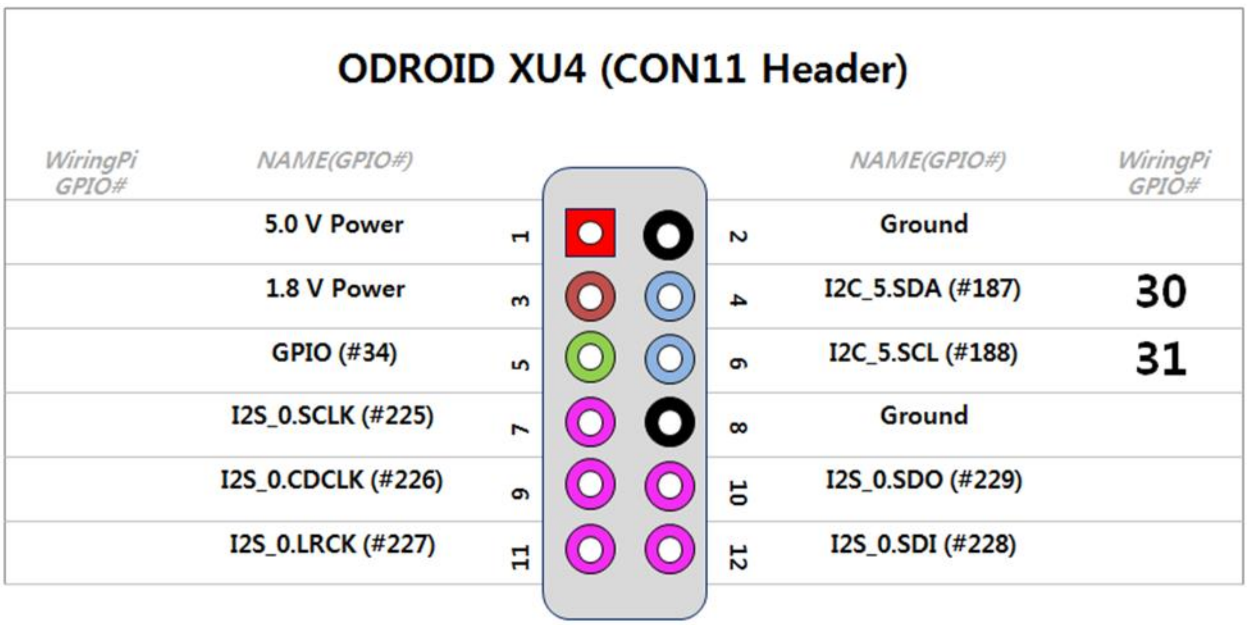

Figure 4.5: CON11 Header

"Power Supply"

Its board needs DC power source which should be 5V/4A. It has a power connector (miniature barrel jack). A DC plug can be connected to it. However, the connecting plug should have an inner and outer diameter of 2.1mm & 5.5mm respectively. Outer cylinder has been kept negative and the inner core has been kept as positive "attached to the black and red wires respectively in the cable".

“Ethernet and WiFi”

It has 3 USB ports and any one of them can be used for the WiFi adapter having USB cable. The transmission rate of the ethernet port ranges upto1 GB/second. A standard ethernet cable should be used.

“MicroSD Card”

It has a microSD card connector. The SD card must be properly mounted. The metal strips of the microSD card should be aligned with the pins of the SD card and slowly pushed in until it settles down in its place appropriately.

“eMMC Module”

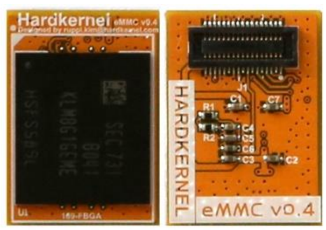

Figure 4.6: eMMC Module

It has an eMMC connector (figure 4.6) on the XU4 board and there is a rectangle of white colour on the PCB for guidance for aligning it with eMMC module. The eMMC module has female portion which needs to be connected with male portion of the board. It should be slowly pushed in until it settles down in its place appropriately. There would appear a writing on the card if inserted properly.

“LED status”

Includes several “LED lights” that indicate the status of the ODROID-XU4 is shown with the help of LED lights as shown in the following table:

Table 4-1: LED status

Red LED	Power is on
Solid blue LED	Bootloader is functioning
Slowing blinking blue LED	The kernel is functioning
Quickly blinking blue LED	The kernel is working extensively

“USB host ports”

It has three USB host ports as under:

- Host ports 2.0: one
- Host ports 3.0: two

Additional ports: In case more than three ports are needed, add an external self powered USB hub. The above ports can be utilized for a number of devices like mouse, keyboard, adapters etc.

“HDMI port”

The connector it contains is HDMI (standard A type).

“Ethernet RJ-45 jack”

For LAN connection, the available port is RJ45 standard Ethernet. It has a flashing system showing connectivity speed as under:

- 100 Mb per second: green flashing light
- 1000 Mb per second: yellow flashing light

The port is meant for Mb per second speed ranging from 10,100,1000.

“Serial console port”

In order to reach the Linux console you will require to connect it with a PC. After logging in you can carry out root maintenance and also check the boot process. A 1.8 volt interface is required and suitable USB-UART module kit should be used like Hardkernel’s.

- “Molex 5268-04a” (2.5mm pitch): to be mounted on the PCB, and its mate
- “Molex 50-37-5043”: Wire-to-Board Crimp Housing.

“RTC (Real Time Clock) backup battery connector”

All circuits relating to real time clock are available on this system. Just connect a battery (preferably Lithium coin CR2032 or equal). If you want to add a RTC functions for logging or keeping time when offline, just connect all of the RTC circuits are included on the ODROID-XU4 by default. It connects with a “Molex 53398-0271” “1.25mm pitch Header”, Surface Mount, Vertical type “Mate with Molex 51021-0200”.

“Gigabit Ethernet”

Ethernet transceiver is available on it. It is Realtek RTL8211F. It is integrated 10/100/1000M.

It conforms with

- “10Base-T”
- “100Base-TX”
- “1000Base-T” IEEE 802.3 standards.

“USB MTT hub controller”

The Hub controller is “Genesys GL3521”. It has following specifications:

- “2-port”

- “low-power”
- “configurable USB 3.0” SuperSpeed.

“USB VBUS controller”

Power supply Protection IC is: NCP380. It is from OnSemi.

“Boot media selector”

The boot media is selected by eMMC/SD card switch. It is on the side of the board.

“Power supply circuit”

For power supply of:

- DRAM
- IO
- CPU

Following are used:

“Discrete DC-DC converters LDOs”.

“Power protector IC”

The power protector IC is:

“NCP372” It controls “over-voltage”, “over-current”, “reverse-voltage”. It is from “OnSemi”.

“Heat Sink and fan”

It is well known fact that Electronic components produce heat while functioning and different machineries produce different quantities of heat. Some components heat up during operation and need to be cooled. Composite mechanisms such as the processor may heat upto 95°C. At such degree, the processor will regulate itself to control further increase in heat. Mostly people prefer to install a heat sink so that the heat does not touch high degrees. People normally use heat sink of Hardkernel’s. Heat transfer from parts to the surrounding air depends on the existing surface area to transmit heat to the surrounding air. The heat sink has a larger area than processor and is therefore able to disperse more heat into the surrounding air. There is also a fan which is controlled through software to change the amount of cooling keeping in view the heat produced. It makes heat sink more cool by drawing air.

“USB3.0 eMMC Module Writer”

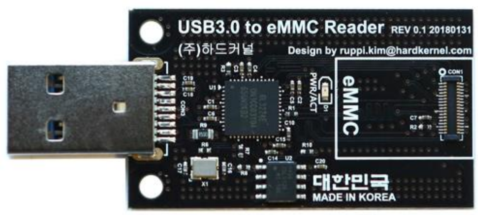

Figure 4.7: eMMC Module Writer

It is much quicker way to show your OS image into the eMMC Module (figure 4.7). No need of a separate USB card reader. Via this writer board your eMMC module can be directly attached with your computer. Dimensions are 60x26x4.5 mm. Specification and advantages are shown in the table 4.3.

Table 4-2: Specification of Module

1. data interface: Native eMMC 8bit wide	2. Interface: USB 3.0
3. Working mode: HS200	4. Windows / Mac / Linux Compatible
5. PC usage: with Etcher or Win32DiskImager software	6. ODROID Orange, Red and Blue eMMC modules are compatible
7. Rated Power: 5V/500mA (including eMMC module)	8. You can access the eMMC hidden boot blocks

“eMMC Module Reader Board for OS upgrade”

Figure 4.8: eMMC Reader Board

Upgradation of OS is only possible in the eMMC in this way. It is extremely suggested item shown in figure 4.8.

How this board is used.

- Attach the eMMC Module with this Reader Board.

- Slot the board to a high quality USB Multi-reader to experience a much faster speed.

- Attach the USB Multi-reader with PC.

4.2.3 Implementations on ODROID Xu4

The first step in implementation of Human detection on hardware is to setup the ODROID Xu4 so that we can program it to perform our desired task.

The following are the steps of booting the ODROID Xu4 for the first time.

- Insert newly configured eMMC Card. (Pre-installed Ubuntu 18.04)
- Connect display (HDMI or analogue display)
- Plug-in input peripheral devices.
- Connect to a network using ethernet cable.
- To power up use a 5V charger.

Implementation is done in C++ Programming Language. “Opencv” Library (publicly available) downloaded and configured in Code Blocks.

CHAPTER 5: RESULTS

5.1 Dataset

For testing the algorithm and detecting humans in Images, benchmark “OSU Thermal Pedestrian Database” [88] is used which is publicly available. This dataset is used for testing the computer vision methods and algorithms. Images of different sequence of the dataset are shown in the figure 5.1.

Figure 5.1: Dataset

Sensor Details

Images are taken from “Raytheon 300D thermal sensor” of “core 75mm lens”. Camera is mounted on the top floor. Building has eight (8) floors. Gain and focus is manually controlled to control the data.

The information of the dataset is in the tables 5.1.

Table 5-1: Information about Dataset

No. of Sequences	10
Total No. of Target Images	284
Total Humans	984
Image Format	8-bit “grayscale”
Image Size	240 * 360 pixels
Sampling rate	Less than 30 Hz, non uniform

Images are taken in different conditions which are given in the table 5.2. In the table only minimum and maximum conditions are shown. Images are taken at different conditions which are varying from minimum to maximum.

Table 5-2: Atmosphere Conditions

Conditions	**Minimum**	**Maximum**
Temperature	37F	68F
UV index	1 Minimal	7 High
Dew Point	28ºF	47ºF
Humidity	38%	82%
Visibility	3.0 miles	10 miles
Wind	0 mph	29 mph
Timing	Day	Night
Whether	Partly Cloudy, Mostly Cloudy	Light Rain, Haze

5.2 Experimentation

Experiments are performed on two hardware's laptop and ODROID Xu4. We have a ground truth file.txt along with dataset [88]. On the basis of ground truth and results of our implemented algorithm on both hardware's, evaluation metrices are computed.

5.3 Evaluation Metrices

After the implementation of human detection on hardware different evaluation analysis are done on results to actually analyze the efficiency and accuracy of our system. Evaluation Metrices which are used to evaluate the detection and algorithm are

- F_P (False Positives) - "Things which are not human but have been detected as human"
- F_N (False Negatives)- "Humans in the image which are not detected"
- T_P (True Positives)- "Humans that are detected as humans"
- P – "Total number of Images"

It is a statistical analysis criteria in binary classification problems, and is used to basically measure the accuracy of a test.

$$\boldsymbol{Accuracy = \frac{T_P}{P} * 100} \qquad \textbf{(5.1)}$$

$$\boldsymbol{Precision = \frac{T_P}{T_P + F_P} * 100} \qquad \textbf{(5.2)}$$

$$\boldsymbol{Recall = \frac{T_P}{T_P + F_N} * 100} \qquad \textbf{(5.3)}$$

$$\boldsymbol{Miss\ Rate = \frac{F_N}{T_P + F_N} * 100} \qquad \textbf{(5.4)}$$

5.4 Results

This chapter focuses on experimentation results of this thesis work. Sample detection results in which humans are detected are also shown figure 5.2. It can be seen that our detector is robust enough to detect humans. Two different types of hardware architectures were used to test and analyze the detection algorithm. The results are represented both in tabular and graphical form.

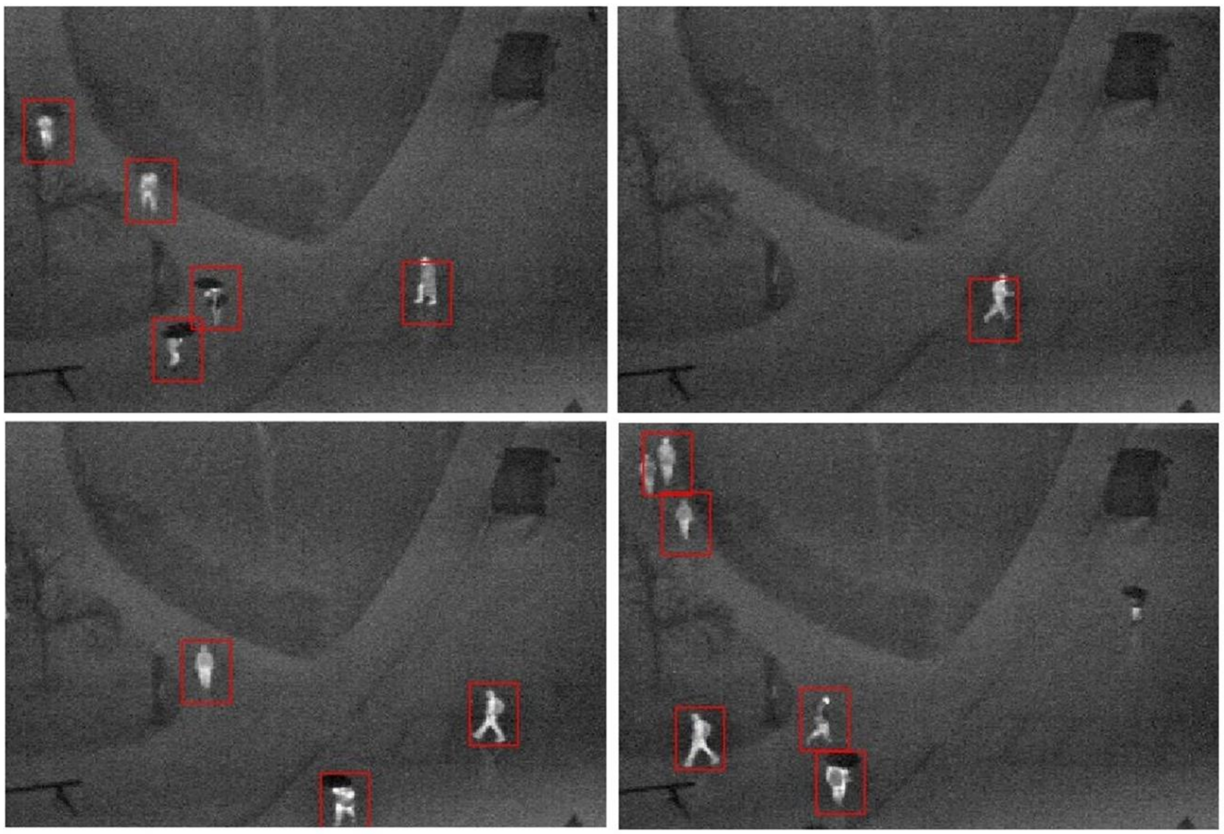

Figure 5.2: Results

Detection time on Laptop using MATLAB implementation and detection on ODROID Xu4 using C++ implementation are shown in table 5-3. On both devices GPU is not used. On MATLAB average 7.5 images per second are processed and on ODROID Xu4 average 03 images are processed per second. This detecting speed on ARM processor based low cost single board computers (embedded systems), have not yet been achieved with such accuracy and precision.

Table 5-3: Detection time

Evaluation Metrices	General purpose Laptop (Intel Core i5) MATLAB	ODROID Xu4
Average Time in seconds	0.13472	0.33964
Miss rate	10.36%	10.36%
No. of images per second	7.5	03

The values of evaluation metrices which are used to evaluate the detection and algorithm are shown in the table 5-4. Detected results are compared with the given ground truth data.

Table 5-4: Evaluation Metrices

Evaluation Metrices	Values
True Positives	882
False Positives	92
False Negatives	102
Accuracy (%)	89.6
Precision (%)	90.5
Recall (%)	89.5
Miss rate (%)	10.36

According to the ground truth information, 984 persons in the given dataset should have been detected. The implemented algorithm detected 882 persons out of 984. Hence 102 persons were not detected.

The algorithm detected 92 persons which were actually not human. So, by using this information, the accuracy of the implemented algorithm is 89.6%, precision is 90.5%, recall is 89.5% and miss rate is 10.36%. (Equations from 5.1 – 5.4). These metrices values show that our implemented algorithm is good in detecting person for this specific dataset.

CHAPTER 6: CONCLUSION

Our Results are very encouraging as the objectives of the research have been achieved. Detection time on Laptop using MATLAB and detection on ODROID Xu4 using C++ have been calculated (Table 5-3). On both devices GPU is not used. On MATLAB average 7.5 images per second are processed and on ODROID Xu4 average 03 images are processed per second. This detecting speed on ARM processor based low cost single board computers (embedded systems), have not yet been achieved with such accuracy and precision. The accuracy of the implemented algorithm is 89.6%, precision is 90.5%, recall is 89.5%.

The research has also proved to be very significant from commercial point of view. It is obvious that with the use of better embedded systems, the results will be better. We can foresee availability of cheaper and better embedded systems in the market in near future, in view of the fast-growing technology of mobile embedded systems. Hence the surveillance systems using this technique shall become more economical as there shall be no requirement of wire net work for cameras and only screens will be required instead of computers. This would obviously reduce the cost of surveillance system. Further its maintenance cost shall also be much lesser.

6.1 Key Contribution

This thesis describes a complete framework for detecting humans in images. We have proposed a method which can detect humans in images using embedded hardware i-e ODROID Xu4. It is very easy to use and can be used in many applications. The main emphasis of this research is implementation of Human Detection on hardware on thermal-infrared Images which has not been done before on such a small credit card size computer.

6.2 Future Work

- We can improve the detection time by using multithread programing.
- We can improve the detection time by resizing the images.

- We can implement this detection method on Ubuntu (desktop computer) with high specifications. This will increase the detection time.
- This algorithm can be optimized on different bench mark thermal infrared images.

REFERENCES

[1] X. He and P. Niyogi, “Locality preserving projections,” in *Advances in Neural Information Processing Systems*, 2004, pp. 153–160.

[2] S. Agarwal, A. Awan, and D. Roth, “Learning to detect objects in images via a sparse, part-based representation,” *IEEE Transactions on Pattern Analysis and Machine Intelli- gence*, vol. 26, no. 11, pp. 1475–1490, 2004.

[3] B.Wu and R. Nevatia, “Simultaneous object detection and segmentation by boosting local shape feature based classifier,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[4] A. Kapoor and J. Winn, “Located hidden random fields: Learning discriminative parts for object detection,” in *European Conference on Computer Vision*. Springer, 2006, pp. 302–315.

[5] J. Mutch and D. J. Lowe, “Multiclass object recognition with sparse, localized features,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp.11–18.

[6] H. J. Seo and P. Milanfar, “Training-free, generic object detection using locally adaptive regression kernels,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1688–1704, 2010.

[7] H. A. Rowleya, S. Baluja, and T. Kanade, “Neural network based face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[8] E. Shechtman and M. Irani, “Matching local self-similarities across images and videos,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[9] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[11] P.Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008,pp.1-8.

[12] P.Felzenszwalb, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” in *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627-1645, 2010.

[13] W. E. L. Grimson, D. P. Huttenlocher,et al., *Object recognition by computer: the role of geometric constraints.* The MIT Press, 1990.

[14] A. Yali and D. Geman, “A computational model for visual selection,” *Neural computation*, vol. 11, no. 7, pp. 1691–1715, 1999.

[15] E. Bienenstock, S. Geman, and D. Potter, “Compositionality, mdl priors, and object recognition,” *Advances in neural information processing systems*, pp. 838–844, 1997.

[16] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2001, pp. I–511.

[17] M.W. M. C. Burl and P. Perona, “A probabilistic approach to object recognition using local photometry and glocal geometry,” in *European Conference on Computer Vision (ECCV)*. Springer,1998,pp.628-641.

[18] P. Felzenszwalb and D. P. Huttenlocher, “Efficient matching of pictorial structures,” in *IEEE conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2000, pp.66–73.

[19] P. Felzenszwalb and D. Huttenlocher, “Pictorial structures for object recognition,” *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[20] B. Leibe, A. Leonardis, and B. Schiele, “Combined object categorization and segmentation with an implicit shape model,” in *European Conference on Computer Vision (ECCV)*. Springer, 2004.

[21] B. Leibe, A. Leonardis, and B. Schiele, “Robust object detection with interleaved categorization and segmentation,” *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, 2008.

[22] C. H. Lampert, *Kernel methods in computer vision*. Now Publishers Inc, 2009.

[23] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 886–893.

[24] Y. Yang and D. Ramanan, “Articulated human detection with flexible mixtures of arts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013.

[25] P. Felzenszwalb, R. Girshick, and D. McAllester, “Cascade object detection with deformablepart models,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2241-2248.

[26] L. Bourdev and J. Brandt, “Robust object detection via soft cascade,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2005, pp. 236–243.

[27] P. Dollár, Z. Tu, P. Perona, and S. Belongie, “Integral channel features,” 2009.

[28] P. Dollár, P.Welinder, and P. Perona, “Cascaded pose regression,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1078–1085.

[29] P. Dollár and L. C. Zitnick, “Structured forests for fast edge detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1841–1848.

[30] L. C. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 391–405.

[31] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[32] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: A benchmark,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 304–311.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[34] C. Szegedy, A. Toshev, and D. Erhan, “Deep neural networks for object detection,” in *Advances in Neural Information Processing Systems*, 2013, pp. 2553–2561.

[35] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, “Scalable object detection using deep neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2147–2154.

[36] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv preprint arXiv:1312.6229*, 2013.

[37] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[39] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[40] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European conference on computer vision*, pages 128–142. Springer, 2002.

[41] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[42] David G Lowe. Local feature view clustering for 3d object recognition. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–682. IEEE, 2001.

[43] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[44] Shimon Ullman, Erez Sali, and Michel Vidal-Naquet. A fragment-based approach to object representation and classification. In *Visual Form 2001*, pages 85–100. Springer, 2001.

[45] Rémi Ronfard, Cordelia Schmid, and Bill Triggs. Learning to parse pictures of people. In *European Conference on Computer Vision*, pages 700–714. Springer, 2002.

[46] Krystian Mikolajczyk, Cordelia Schmid, and Andrew Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision*, pages 69–82. Springer, 2004.

[47] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549, 2004.

[48] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 878–885. IEEE, 2005.

[49] Andreas Opelt, Michael Fussenegger, Axel Pinz, and Peter Auer. Weak hypotheses and boosting for generic object detection and recognition. In *European conference on computer vision*, pages 71–84. Springer, 2004.

[50] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[51] Gyuri Dorkó and Cordelia Schmid. Selection of scale-invariant parts for object class recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 634–639. IEEE, 2003.

[52] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE, 2003.

[53] Markus Weber, Max Welling, and Pietro Perona. Unsupervised learning of models for recognition. In *European conference on computer vision*, pages 18–32. Springer, 2000.

[54] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.

[55] Shivani Agarwal and Dan Roth. Learning a sparse representation for object detection. In *European conference on computer vision*, pages 113–127. Springer, 2002.

[56] Wolfgang Forstner and Alfred Pertl. Photogrammetric standard methods and digital image matching techniques for high precision surface measurements, 1986.

[57] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, pages 281– 305, 1987.

[58] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416– 423. IEEE, 2001.

[59] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[60] Greg Mori, Serge Belongie, and Jitendra Malik. Shape contexts enable efficient retrieval of similar shapes. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–723. IEEE, 2001.

[61] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[62] Greg Mori and Jitendra Malik. Recognizing objects in adversarial clutter: Breaking a visual captcha. In *Computer Vision and Pattern Recognition, 2003. Proceedings.2003 IEEE Computer Society Conference on*, volume 1, pages I–134. IEEE, 2003.

[63] Leonid Sigal, Michael Isard, Benjamin H Sigelman, and Michael J Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *Advances in neural information processing systems*, page None, 2003.

[64] Deva Ramanan and David A Forsyth. Finding and tracking people from the bottom up. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–467. IEEE, 2003.

[65] Henry Schneiderman and Takeo Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, 2004.

[66] Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 746–751. IEEE, 2000.

[67] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3):519–524, 1987.

[68] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991

[69] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1): 23–38, 1998.

[70] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[71] Michel Vidal-Naquet and Shimon Ullman. Object recognition with informative features and linear classification. In *ICCV*, volume 3, page 281, 2003.

[72] Dariu M Gavrila and Vasanth Philomin. Real-time object detection for âA.smartâAI vehicles. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 87–93. IEEE, 1999.

[73] Dariu M Gavrila, Jan Giebel, and Stefan Munder. Vision-based pedestrian detection: The protector system. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 13–18. IEEE, 2004.

[74] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient matching of pictorial structures. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 66–73. IEEE, 2000.

[75] Sergey Ioffe and David A. Forsyth. Probabilistic methods for finding people. *International Journal of Computer Vision*, 43(1):45–68, 2001.

[76] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[77] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[78] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example-based object detection in images by components. *IEEE transactions on pattern analysis and machine intelligence*, 23(4):349–361, 2001.

[79] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153– 161, 2005.

[80] Vapnik N Vladimir and V Vapnik. The nature of statistical learning theory, 1995.

[81] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[82] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[83] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

[84] Robert E Schapire. The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification*, pages 149–171. Springer, 2003.

[85] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Computer vision, 1998. sixth international conference on*, pages 555–562. IEEE, 1998.

[86] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1491–1498. IEEE, 2006.

[87] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[88] J. W. Davis and M. A. Keck, "A two stage template approach to person detection in thermal imagery." In *Proc. Workshop on Applications of Computer Vision.* IEEE, 2005. [Online available]: http://vcipl-okstate.org/pbvs/bench/