Automatic Analysis of Crowd Dynamics Using Computer Vision and Machine Learning Approaches

Muhammad Saqib

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney, NSW - 2007, Australia

A Thesis submitted in fulfilment
of the requirements of the degree of
Doctor of Philosophy

March 2019

ABSTRACT

As the population of the world increases, urbanization generates crowding situations which pose challenges to public safety, security and to the management of crowd. Manual analysis of crowded situations is a tedious job and usually prone to errors. Computer scientists have been trying to bring about the cognitive video understanding abilities of human brains onto video systems using computer vision techniques. However, crowd analysis is a complex phenomenon in computer vision because of the intrinsic difficulties such as perspective distortion, occlusion, shadowing effect and variations in scale, pose and illumination. These difficulties degrade the performance of a computer vision system. This research proposes solutions based on classical machine learning approaches as well as state-of-the-art deep learning approaches to automate the process of visual surveillance. This thesis will investigate three research areas of visual analysis of crowds: crowd counting and density estimation in low-to-medium density crowds, crowd motion analysis, and describing crowd motion with useful motion information.

To this end, we investigated the texture features for crowd counting and density estimation. We outline an SVM-based framework for the evaluation of different texture features and also studied the performance of the various combination of features for crowd counting and density estimation. Currently, object detection using Deep Convolutional Neural Networks have shown tremendous improvement over hand-crafted feature extraction techniques. Current state-of-the-art object detectors were used for head detection task. Furthermore, Faster R-CNN was used for counting by detection, and the performance is improved with the proposed Motion Guided Filter by using spatio-temporal information.

In this study, we also propose a novel technique of crowd analysis, the aim of which is to detect different dominant motion patterns in real-time videos. It is very important for a computer vision system to capture movement as accurately as possible. For such a system, a motion field is generated by computing the dense optical flow. The performance of our approach is evaluated on different real-time videos. Furthermore, a framework is developed that extract motion information from the crowd scene by generating trajectories using particle advection approach. The trajectories obtained are clustered using unsupervised hierarchical clustering algorithm into flows. The motion information extracted includes, the speed, density and mean direction of dominant flows/global flows.

This study will provide real-time and practical solutions for the autonomous vision system

used in the surveillance. The analysis of information before the occurrence of any disaster or disruptive event will result in the indication of dangerous and disruptive behaviour which seeks urgent attention.

CERTIFICATE OF ORIGINAL AUTHORSHIP

I Muhammad Saqib declare that this thesis, submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science, Faculty of Engineering and Information Technology at the University of Technology Sydney, Australia, is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Signed:

Production Note:
Signature removed
prior to publication.

Muhammad Saqib
March 15, 2019

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to my supervisors Prof. Michael Blumenstein (Principal supervisor), Dr Nabin Sharma (Associate Supervisor) and Dr Sultan Daud Khan (External Supervisor) for their fruitful guidance and support throughout my PhD candidature. Their encouragement, interactive regular meetings have made my experience very productive. I am especially thankful to Prof. Michael for his support during the stressful time when I was diagnosed with Bechet's Syndrome and also for my transfer of PhD studies from Griffith University to the University of Technology Sydney.

I am thankful to Griffith University for giving me the opportunity to pursue my PhD studies by granting me scholarships to cover my tuition fee and living expenses. I am thankful to the School of ICT, IIIS and GGRS of Griffith University for their support during the first year of my PhD.

I am thankful to UTS for providing me with the scholarship for the remaining period of my PhD. I cannot forget to mention the state-of-the-art facilities for doing my experiments. I express my gratitude to all staff at the School of Computer Science, CAI, GRS and FEIT staffs of UTS for their assistance. I am lucky to find great friends. I am grateful to Dr Gul Zameen Khan, Dr Afzaal Qamar, Chandranath Adak, Dr Abhijit Das, Di Wu, Fahimeh Alaei, Forough Rezaei Boroujeni, Dr Ranju Mandal, Nazar Waheed, Imran Makhdoom, Wali Rashid, Dr Rupam, Muhammad Fazal, Dr Sajjad, and Dr Asma Alkalbani for their support and great company.

Finally, I would not have been able to focus on my PhD studies without the constant support, sincere prayers and unconditional love of my wife, parents and siblings.

LIST OF PUBLICATIONS

List of conference Publications

1. Saqib, Muhammad, Sultan Daud Khan, and Michael Blumenstein. "Texture-based feature mining for crowd density estimation: A study." In 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ), pp. 1-6. IEEE, 2016. **Research Question 1**

2. Saqib, Muhammad, Sultan Daud Khan, and Michael Blumenstein. "Detecting dominant motion patterns in crowds of pedestrians." In Eighth International Conference on Graphic and Image Processing (ICGIP 2016), vol. 10225, p. 102251L. International Society for Optics and Photonics, 2017. **Research Question 2**

3. Saqib, Muhammad, Sultan Daud Khan, Nabin Sharma, and Michael Blumenstein. "A study on detecting drones using deep convolutional neural networks." In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1-5. IEEE, 2017.

4. Coluccia, Angelo, Marian Ghenescu, Tomas Piatrik, Geert De Cubber, Arne Schumann, Lars Sommer, Johannes Klatte et al. "Drone-vs-bird detection challenge at ieee avss2017." In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1-6. IEEE, 2017.

5. Saqib, Muhammad, Sultan Daud Khan, Nabin Sharma, and Michael Blumenstein. "Extracting descriptive motion information from crowd scenes." In 2017 International Conference on Image and Vision Computing New Zealand (IVCNZ), pp. 1-6. IEEE, 2017. **Research Question 2 and 3**

6. Saqib, Muhammad, Sultan Daud Khan, Nabin Sharma, and Michael Blumenstein. "Person head detection in multiple scales using deep convolutional neural networks." In 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1-7. IEEE, 2018.**Research Question 1**

7. Das, Abhijit, Abira Sengupta, Muhammad Saqib, Umapada Pal, and Michael Blumenstein. "More Realistic and Efficient Face-Based Mobile Authentication using CNNs." In 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. IEEE, 2018.

8. Saqib, Muhammad, Sultan Daud Khan, Nabin Sharma, Paul Scully-Power, Paul Butcher, Andrew Colefax, and Michael Blumenstein. "Real-Time Drone Surveillance and Population Estimation of Marine Animals from Aerial Imagery." In 2018 International Conference on Image and Vision Computing New Zealand (IVCNZ), pp. 1-6. IEEE, 2018.

List of Journal

1. Saqib, Muhammad., Daud Khan, Sultan., Sharma, Nabin. and Blumenstein, M. (2019). "Crowd Counting in Low-Resolution Crowded Scenes Using Region-Based Deep Convolutional Neural Networks." Accepted in IEEE Access.

CONTENTS

LIST OF FIGURES

List of Tables

List of Acronyms

BPR	Bayesian Process Regression
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
FTLE	Finite Time Lyapunov Exponent
GLCM	Gray-level Co-occurrence Matrix
GMM	Gaussian Mixture Model
GOCM	Gradient Orientation Co-occurrence Matrix
GPR	Gaussian Process Regression
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
LBP	Local Binary Pattern
LBPCM	Local Binary Pattern Co-occurrence Matrix
LCS	Lagrangian Coherent Structure
LOI	Line-of-Interest
LSTM	Long Short-Term Memory
MGF	Motion Guided Filter
MRF	Markov Random Field
NMS	Non-Maxima Suppression
R-CNN	Region-based Convolutional Neural Network
RELU	Rectified Linear Unit
ROI	Region-of-Interest
RPN	Region Proposal Network
RR	Ridge Regression
SIFT	Scale Invariant Feature Transform
SS	Selective Search
SSD	Single Shot Detector
SURF	Speeded Up Robust Feature
SVM	Support Vector Machine

VGG16	Visual Geometry Group 16
VGGM	Visual Geometry Group Medium
YOLO	You Only Look Once
ZF	Zeiler and Fergus

"The member of a crowd gains a sentiment of invincible power which allows him to yield to instincts which, had he been alone, he would have kept under restraint."

- Gustave Le Bon, 1899

1 Introduction

A crowd constitutes a large number of people gathered together in public places. In public places, CCTV cameras are installed to monitor the activity of the crowd. Traditionally, human operators are required to continuously monitor the sheer number of CCTV cameras installed, to detect any unusual and abnormal aspect within the crowd. It is very difficult for a human operator to continuously monitor and analyze a huge amount of visual surveillance data coming from different viewpoints. Therefore, a large staff is required to monitor a large crowd, which results in very high costs. On the other hand, if very few people are employed for the monitoring task, they may miss very important and critical situations. Many researchers are turning toward computer vision for such a system, which will automate the whole process of crowd monitoring, thus reducing the need for human intervention very significantly. An important application of visual analysis of crowd is counting and density estimation of people within the crowd. The term counting is used in detection based approaches for the exact number of people in the crowd. While density is the estimation of the crowd used in a regression-based analysis of high-density crowd. Crowd counting and density can be used in the detection of potentially dangerous situations such as congestion. Additionally, civil engineer and architect use counting in the design of public space. Motion analysis in a crowd is a pre-processing step in the overall understanding of crowd behavior over some time in many frames of video. As a first step, motion information using optical flow is computed. The motion information is then used to find trajectories, dominant motion pattern, and overall behavior of the crowd.

Recent advancement in deep learning has shown outstanding results in detection using Convolutional Neural Networks (CNN). In a typical CNN based approach, there is local connectivity of a region in the input image to the output image as compared to the traditional feedforward

neural network. In a feedforward neural network, every input layer is fully connected with the output layer. Deep CNN is a compositional model, in which features are extracted ranging from low-level along the pipeline of CNN towards the final layers. The lower layers represent low-level features such as edges, and the subsequent layers represent abstract features such as shapes, etc. CNN features are expressive, robust and generic than hand-crafted features.

The main contribution of the thesis will be to automate the process of crowd counting and density estimation in real-time using computer vision and machine learning. The thesis will study the crowd using traditional machine learning and deep learning based approaches. Moreover, motion information will be used to study the behavior of the crowd.

1.1 Background and motivation

Crowds of the pedestrian can be considered as a complex phenomenon because of constant interaction among the people within the crowd, complex behavior of the individuals and inter-object occlusions make it a difficult and challenging task. World population has been doubled in the years 1959 (3 billion) to 1999 (6 billion) [1] . The current growth of global population increased at 1.13 percent per year. According to latest predictions by *United Nation Predictions*[2], the estimated population will reach 10 billion in 2056. As the population of the world is increasing at a rapid rate and more people are moving towards the major cities in pursuit of better life and facilities. The world has experienced unprecedented urban growth; big cities are more densely populated than ever before. It is expected that 70 percent of the world population will be urban by 2050. The number of cities having over one million population had grown from 14 to 83. Also, the phenomena of a crowd like activities such as sports, festivals, social gatherings, political and religious gatherings, concerts and public transport terminals tend to attract a large number of people competing for limited space and resources; this can lead to serious safety and security concerns for the participant and their organizers. Therefore understanding the crowd is a critical and challenging task.

There has been multidisciplinary research in the analysis and understanding of crowd behavior. Urban planner, civil engineers, architects, study crowd behavior in normal and extraordinary situations *e.g*. evacuation help them in planning and designing buildings and making policies which can cope with such extraordinary situations like riots, fire, stampede. Similarly, organizer and manager of big events do crowd analysis and study the behavior of the crowd such as ceremonies, races, carnivals, concerts, parties *etc*. to do better management in case of any abnormal events. Some of the examples of the crowd are shown in Fig. 1.1. Therefore, understanding the dynamics of a large number of people is very important in designing and management of such big events. Similarly, biologist studied the behavior of animals in herds, packs and bird flocks, fish schools and their habitat. A social psychologist studied different aspects of the behavior of an individual in the crowd as well as the overall behavior of the crowd in relation to crowd size. According to

[1] http://www.worldometers.info/world-population
[2] http://www.prb.org/Publications/Lesson-Plans/HumanPopulation/Urbanization.aspx

the Gustave Le Bon, a French psychologist, wrote in his famous book, The Crowd: A Study of the Popular Mind, about the infectious behavior of the crowd psychology in his contagious theory. According to his contagion theory, which argues that crowds cause people to act in a certain way. The theory suggests that crowds exert a sort of hypnotic influence on their members. The hypnotic influence combined with the anonymity of belonging to a large group of people, even just for that moment, results in irrational, emotionally charged behavior. Or, as the name implies, the frenzy of the crowd is somehow contagious, like a disease, and the contagion feeds upon itself, growing with time. In the end, the crowd has assumed a life of its own, stirring up emotions and driving people toward irrational, even violent action.

The focus of the thesis will be about the visual analysis of the crowd for the public safety and security. Due to the sheer number of cameras installed, a large number of staff is required to monitor manually and report any abnormal activity in the crowd. Unfortunately, manual surveillance requires constant observation, adequate training, and a great deal of attention which is not possible for a human observer to focus all the time and therefore may be prone to error due to human negligence. It is very often that videos of an incident are investigated after the occurrence of the event. The aim is to make real-time alerts for any abnormal activity within the crowd and gives the authority a chance to prevent injuries and fatalities at the earliest. During the time of panic in large size crowded area, the study of the crowd helps the organizers of event important information for mass evacuation. Therefore, crowd detection and estimation has been the area of interest of most of the computer scientist and researchers. Although, the main purpose of surveillance is to provide safety and security. However, it can be incorporated into business intelligence such as big shopping malls. Such intelligence can provide valuable information about the number of customers and their interests in particular products. It can also provide information about the peak hours sales of different products, which can help the owner of the business to restructure shopping mall according to the customer's interests. Monitoring and surveillance is a very important area of study, which has applications in the safety and security of the public.

The human eye is the most intelligent visual device which passes low-level information to the brain for processing. Brain extracts high level semantic and contextual information about the scene. Computer vision scientists are trying to bring about the same intelligence and perceptual capabilities as the human eye. Till now, researchers in the computer vision community have succeeded in the detection of different kind of objects in a variety of scenes. But the crowd phenomenon is much more complex than simple detection and tracking tasks. And therefore, simple extension of detection and tracking of human in the simple scene cannot apply to complex scenes of crowds. Because in the case of crowd analysis in real-world surveillance setting, scientist faces a difficult challenge to deal with severe occlusion, due to an excessive number of objects and their interaction with each other in the crowd.

The general implementation pipeline for the high-density crowd may include the following steps. The first important step in feature-based crowd analysis is to calculate the crowd flow, which can easily be computed by well established and optimized techniques of optical flow techniques. Generally, flow vectors having identical orientation are combined, then followed by segmentation.

Another mostly used feature-based tracking is based on SIFT. In general two steps are followed in a SIFT feature-based tracking approach to segment the flow of the crowd. In the first step, SIFT-based features are computed and their correspondence is tracked at some interval of time. SIFT features seem to be more accurate in complex scenarios as compared to corner based features. As a result, local tracks are obtained called tracklets. These tracklets provide good information about the local motion in a very short duration of time. Combining local tracklets to form global tracks over time is a clustering problem. In the second step, to get the global flow information, these local tracklets are combined to form global tracks using some clustering algorithms.

There are some commercial products in the market for surveillance and monitoring. However, they are expensive, not reliable and robust. The performance is limited to one scene and needs to be retrained for a new scene. Therefore in video surveillance applications, there is a dire need to automatically detect the crowd density, dominant flows, crowd size, and abnormal events and provide intelligence to the analyst beforehand. Despite all the security measures, crowd disasters still occur frequently. The summary of crowd disasters of the last ten years is shown in Table 1.1.

1.2 Significance of visual analysis of crowd

Crowd analysis has many real-world critical applications. The applications are given below.

1. Crowd Management The study of crowd analysis help managers and event organizers to develop evacuation procedure in emergencies. e.g., fire, riot, stampede
2. Public Space Design Crowd analysis provides useful information to civil engineers and architects in the design of public spaces.
3. Visual Surveillance It can be used in to detect anomalous event to ensure public safety and security
4. Intelligent Environments The intelligence can be used in a shopping mall to find the pattern of customers and their interest in a particular object.

1.3 Challenges

The assumption that the mass of the crowd behaves more or less the same is usually incorrect. Because the person behavior is more complex and depends on many different factors such as the layout of the scene, density of people, goals of the individuals and coherence among the individuals in the crowd. Most of the time the motion and behavior of the individual are tied to his role in the crowd; whether the individual is working as police, a runner, a protestor or any other possible role. The traditional pipeline of object detection and tracking are core technologies in computer vision which are successfully applied to low-density crowds. The main steps of these methods are to localize the moving objects, track them for some duration and analyze them for

Table 1.1: Crowd disasters

Year	Place	Deaths
2015	Mina Crowd Crush, Mecca Saudi Arabia	2277
2014	China Shanghai New Years Eve crush	34
2013	Madhya Pradesh Crowd crush, India	115
2012	Stadium Bamako, Mali	36
2011	Pilgrimage Kerala, India	119
2010	Love parade, Germany	23
2009	Phnom Penh Crowd Crush, Cambodia	347
2008	Houphout-Boigny stampede, Cte d'Ivoire	19
2007	Jodhpur temple,Rajasthan India	249
2006	PhilSports Stadium, Manila Philippines	73

scene semantics. However, they find limited use in the case of high-density crowds because of the inherent challenges and complexities. Some important challenges need to be addressed for the understanding of crowd behavior. I have discussed some of the challenges given below.

1. As the density of people in the crowd increases, there is less number of pixels per person, which makes the detection process much harder.

2. Human behavior is very complex because of high variability in situation and settings, and there is no existing mathematical model to model such complexities.

3. Due to the perspective problem or far-shortening effect, geometric correction of camera calibration is required to get a reasonable estimate of crowd density.

4. The individuals in the crowds exhibits different behaviors and usually goal-directed. Which makes it very challenging to figure out an appropriate level of granularity to model crowd dynamics [2].

5. Complex interaction between objects, physical scene characteristics, varying illumination, and shadowing effect, pose severe occlusions both at the spatial and temporal level, result in loss of information about the considered object.

6. Another challenge is to detect any specific behavior is to train the system in that particular behavior.

These challenges can be seen in the images of the crowd shown in the Fig. 1.1.

(a) Sports (b) Egypt uprising

(c) Concert (d) Marathon

Figure 1.1: Illustration of challenges in the crowd

1.4 Aims and objectives

The objectives of this research are to develop robust algorithms for visual analysis of crowds. In this study, we will be using core technologies such as machine learning and computer vision to make intelligent solutions for surveillance and monitoring of crowds. Substantial efforts have been made in feature engineering for different computer vision applications. However, in this study, different features and their combination will be evaluated for crowd density estimations, crowd counting, crowd flow analysis, and behavior of the crowd. The aims of the stakeholder for crowd management are listed below.

1. The computer vision system should automatically give the count and density estimation of the crowd.
2. The system should be able to generalize to unseen situations.
3. The system should be able to do motion analysis.
4. The system should be able to provide useful motion information about crowd flows.

To achieve the aims of the stakeholder, the following objectives have been proposed.

1. Development of real-time solutions for crowd counting and density estimation using computer vision and Convolutional Neural Network.

2. To evaluate proposed features for cross scene evaluation, and to see how they generalize to the unseen situations.

3. To identify crowd behavior in terms of local motion estimates (motion patterns) and dominant motion patterns.

4. To study the behavior of crowd using motion analysis.

1.5 Research questions

The above motivation and objectives lead to the following research questions

1. How to find the size of the crowd both in terms of counting and density of the crowd by evaluating different features and finding the most distinguishing features for crowd density estimation both for an outdoor and indoor scene?

2. How to make use of crowd flows to determine dominant motion patterns in the crowded videos?

3. How to extract descriptive motion information from the crowd scene?

1.6 Evaluation of crowd analysis methodology

There are several publicly available benchmark datasets for the evaluation of crowd analysis algorithms. The brief description of these datasets is given below.

1. PETS 2009 Dataset [1]
   This dataset of crowded video scene released as a challenge before the eleventh international workshop on performance and evaluation for activities such as crowd counting and density estimation, tracking under varying lighting conditions. This dataset contains calibration data, training data, person counting, and estimation data, people tracking and flow analysis data.

2. UCF Crowd Dataset [3]
   This dataset contains images from BBC motion gallery and Getty Images. The dataset provides different lighting and field of view conditions for evaluation of crowd analysis algorithms.

3. UCSD Anomaly Detection Dataset
   Chan [24] introduced this dataset for evaluation of anomalous behavior in a crowd.

4. Mall
   The Mall dataset was introduced by chen [36] for the evaluation of indoor crowd scene with different environmental and crowd configurations.

5. UCF Crowd Behavior Dataset
   This dataset contains images from BBC motion gallery, YouTube, PETS2009 for evaluation of crowd behavior recognition.

1.7 Contributions

The proposed work has led to the following contributions to this thesis. The following are the contributions.

- A new hierarchical clustering algorithm has been proposed which can detect dominant motion patterns in crowd videos.
- A new method for extracting motion information presented. The proposed method can extract useful information such as speed, density, as well as the direction of dominant motions from crowd flows.
- Most of the existing approaches for crowd counting is based on regression for crowd counting. We have evaluated state-of-the-art object detectors for head detection as well as for person detection. A Motion Guided Filter has been proposed to exploit the spatio-temporal information for the refinement of detections. i,e. False detections are suppressed, and missed detections are recovered.
- A study of various texture features and their fusion have been evaluated for crowd counting and density estimation

1.8 Stakeholders

The thesis will assist the crowd management and law enforcement agencies in the management of crowd of pedestrians.

1.9 Outline of the thesis

Based on the proposed research questions, the proposed methodologies are divided across several chapters. The thesis is divided into seven chapters and is briefly described as follow:

- Chapter 1 provides an introduction to the research area, aims, and motivation of the research, its significance, major challenges, brief description of the datasets and evaluation criteria, and outline the contributions to the research.

- Chapter 2 provides a critical analysis of the existing state-of-the-art work published in the visual analysis of the crowd using computer vision and machine learning approaches. The crowd dynamics covers different aspects of crowd analytics which include crowd counting, crowd motion analysis, crowd behavior analysis, and crowd anomaly detection.

- Chapter 3 describe the proposed framework for crowd counting using texture feature. Texture feature is an important feature descriptor for many image analysis applications. The objectives of this research are to determine distinctive texture features for crowd density estimation and counting. A two-stage classification and regression-based framework have been proposed for performance evaluation of all the texture features for crowd density estimation and counting. According to the framework, input images are divided into blocks and blocks into cells of different sizes, having varying crowd density levels. Due to perspective distortion, people appearing close to the camera contribute more to the feature vector than people far away. Therefore, features extracted are normalized using a perspective normalization map of the scene. At the first stage, image blocks are classified using multi-class SVM into different density level. At the second stage, Gaussian Process Regression is used to regress low-level features to count. Various texture features and their possible combinations are evaluated on a publicly available dataset.

  The focus of the second part of the chapter is that of using state-of-the-art object detectors for the task of head detection. The choice of head detection for detecting the number of people is a viable option because it is visible most of the time even when other body parts are fully occluded.

- This chapter examines the use of Region-Based Deep CNN for person detection. The detection framework is combined with the proposed Motion Guided Filter to refine the detection results.

- Chapter 5 presents the motion analysis of crowd videos to detect dominant motion patterns using hierarchal clustering.

- Chapter 6 describes the proposed work for extracting useful motion information such as density, speed, and direction of different flows in the crowd.

- Chapter7 Summarizes the thesis and outline the future research work in the visual analysis of the crowd.

"Research is to see what everybody else has seen, and to think what nobody else has thought."

- Albert Szent-Gyorgyi

2

Background and Related work

This chapter provides a brief review of current state-of-the-art approaches and their limitations. The discussion about limitation and shortcomings will pave the way for research direction and will lay the foundation for the work discussed in subsequent chapters.

2.1 Introduction

Crowd analysis is an active field of research in computer vision. In practice, computer vision is used in most of the visual surveillance systems. In a typical surveillance system, manual analysis is usually prone to error due to the short attention span of a human. However, many researchers are focusing on intelligent solutions for automatic analysis of visual streams of crowded and cluttered scenes. From a computer vision perspective, the aim of analysis is to find the count and density estimation of a crowd, motion analysis, congestion, and ultimately the behavior of the crowd over a period to detect anomalous situations.

The problems presented by crowd analysis are more complicated than a typical computer vision system's problem. Here we will discuss some of the common problems faced by typical computer systems as well as by the system used for crowd analysis as illustrated in Fig 2.2.

1. ***Intra-class variations:*** In crowd analysis, the focus is to detect human as accurately as possible. However, due to intra-class variations in shape, size, texture, and color make it difficult for a computer vision system to analyze. Sometimes, there is a need to consider different pose and context around the object for accurate analysis.

2. ***Background clutter:*** Sometimes the object of interest closely resembles the background where it is almost impossible to see objects. Moreover, in such situations background is dominated in the image.

3. ***Illumination changes:*** The change in illumination depending on the time of the day and season make the object to appear differently.

4. ***Occlusions:*** This is the most common problem in crowd analysis where one object is occluded by another object either fully or partially.

5. ***Shape variations:*** Due to the articulated nature human body, different variation in pose may cause a problem for computer vision in recognition.

6. ***Viewpoint variations & Perspective distortion:*** Due to viewpoint variation, the object may appear differently from different viewpoints. The problem particular to crowd analysis is that of perspective distortion. In perspective distortion, the object close to the camera may contribute more to the feature vector than the object far away from the camera. Most of the time perspective normalization is required as a pre-processing step before the analysis of crowd.

This literature review aims to do a critical analysis of current state-of-the-art techniques. In particular, our survey will be closely related to crowd counting and density estimation, crowd motion analysis, crowd flow segmentation, crowd behavior analysis, the study of dominant motion patterns, and high-level scene semantics understanding of the crowd. Computer vision researchers have carried out successful research for object detection, recognition, image segmentation and tracking in very low-density crowds. The number of people in each density levels of crowd varies. There are no agreed-upon criteria of different density levels in the crowd. There remain blur boundaries between different density level of the crowd, often depend on specific applications. In a low-density crowd, there is clear visibility of individual and even their parts. Therefore detection, recognition, and even tracking are relatively easy in low-density crowds. On the other hand, detection and tracking algorithms are usually inapplicable in high-density crowds.

Classification of crowd analysis is based on the specific problem domain. The crowd can be broadly classified into two categories.

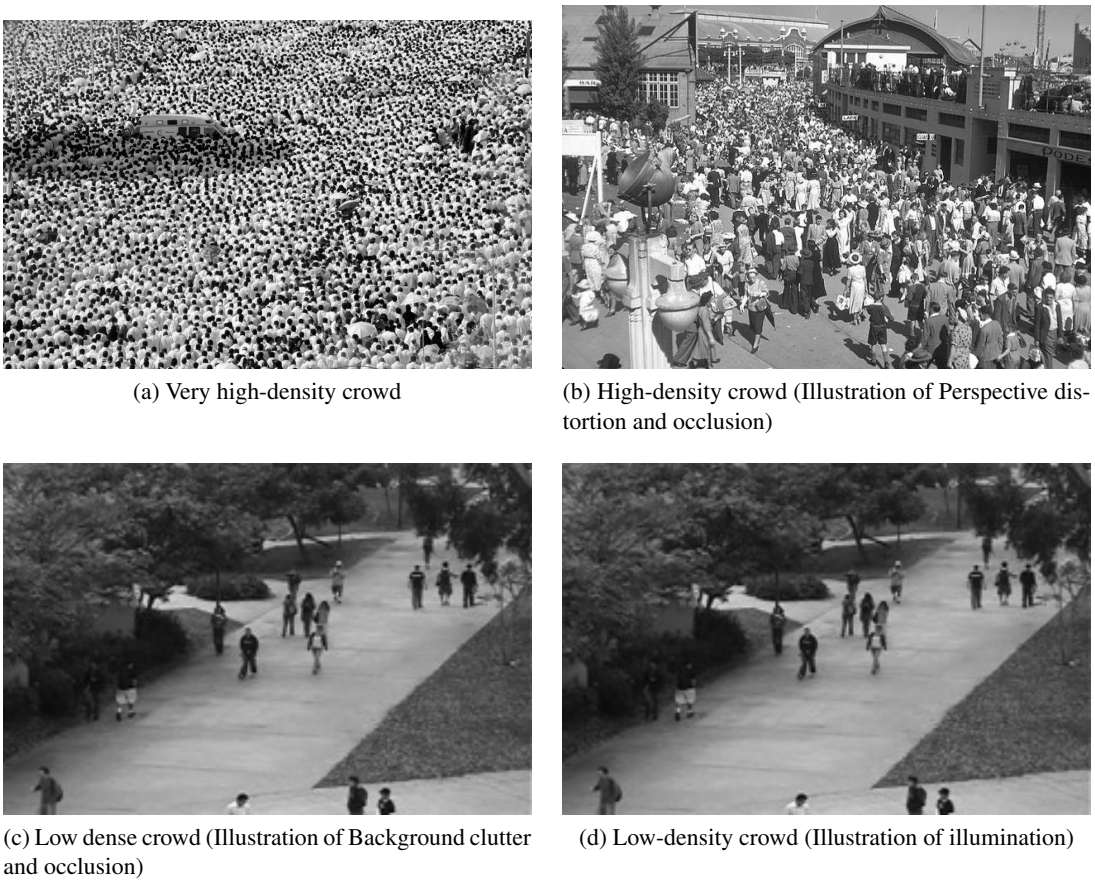
1. Structured crowds

2. Unstructured crowds

High-density crowds can be broadly classified into structured and unstructured crowds [145] as illustrated in Fig 2.1. In the structured crowd, people move coherently in common directions. In such type of crowds, the motion of the people is predictable, goal oriented, and most of the time the goal is common. The direction of movement of such a crowd is fixed and does not change with time. *e.g*. pedestrian in a marathon race. In contrast, people in unstructured crowd move freely in one direction or another direction, and their motion is unpredictable *e.g*. road crossings, railway stations, expos, airports.

(a) Structured crowd (b) Unstructured crowd

Figure 2.1: Classification of crowds based on motion patterns



(a) Very high-density crowd

(b) High-density crowd (Illustration of Perspective distortion and occlusion)

(c) Low dense crowd (Illustration of Background clutter and occlusion)

(d) Low-density crowd (Illustration of illumination)

Figure 2.2: Illustrations of various problems in publically available datasets

2.2 Crowd counting and density estimation

Moreover, estimating crowd density or estimating the number of people attending the event can substantially reduce the cost by deploying an exact number of security personnel required for public safety and security.

Various methods for estimating the crowd count are proposed in the literature. Generally, we

can classify these methods into two major categories, 1) *Regression-based methods*, 2) *Detection-based methods*. Regression-based methods employ machine learning techniques like Support Vector Regressor [188], Gaussian Process Regression(GPR) [25], linear regression [45], K-nearest neighbor [210], and neural network [121] are employed to estimate the crowd count by performing regression between the image features and crowd size. Regression-based crowd counting algorithms can be further categorised into two groups: *Holistic* and *Local*.

2.2.1 Holistic approaches

In *holistic approaches*, image features like size, shape, edges, keypoints, and texture are extracted from the entire image and regression is then applied to estimate the size of the crowd. In [25], edge and texture features are extracted from the whole image, and the correspondence between the number of people and features is learned through Gaussian Process Regression. [112] extracts shape, color, size, texture features from the image and the self-organizing neural network is employed for crowd density estimation. The neural network is employed by [69] to learn the correspondence between the foreground pixels extracted from the whole image and number of people. A method is proposed by [199] that transforms an image into multiple scales using wavelet transform and then the first and second order features are extracted as a density character vector. A support vector machine classifier is trained that classify density character into different density levels. The term density of crowd is used whenever an exact number of count is not required, The scene may be defined in term of density levels such as low, very low, medium, high, very high. The boundaries between these different density levels are a blur and are often defined in the context of the application being considered. Marana *et al*. [118, 119, 120] uses subset of second order statistical features based on Gray Level Co-occurrence Matrix(GLCM) for crowd density estimation originally proposed by Haralick *et al*. [63]. The underlying assumption for the use of texture features was that low-density crowd exhibit coarse texture and high-density crowd exhibit fine textures and the background of the scene are relatively smooth texture as compared to foreground texture. Similarly, Manfredi *et al*. [116] used Gradient Orientation Co-occurrence Matrix(GOCM) and Wang *et al*. [189] used Local binary Pattern Co-occurrence(LBPCM) instead of using gray-levels for crowd density estimation. Other features used are foreground pixels [168, 28],gradient features [116], minkowski fractal dimension [119],fourier features [86], Gabor features [79], Translational Invariant Orthonormal Chebcheve Moments [139]. Number of studies [58, 109, 128, 206] also addressed the crowd density estimation using LBP [131]. In [88], edge and blob size histogram features are extracted, and the neural network is trained to find the relationship between the number of people and extracted features. [83] proposed crowd flow segmentation as the first step and then applying counting framework to count the number of people in each flow segment.

Previously perspective normalization was used to normalize for perspective in the single image. However, Ryan *et al*. [152] proposed a system, in which human is approximated by 3-D cylinder model. The density map is constructed by projecting this cylinder into the scene. The value of the density map at each location is based on the size of the object, centered at that lo-

cation. According to weight or density map, distant objects get higher weights as they are far away in the scene. The density map was used for scene invariance across multi-camera. In their proposed approach before feature extraction, first background is modeled, and foreground objects are segmented in the form of blobs before feature extraction. The full feature vector for each blob contains features such as size, shape, edge, and key-points. An adopted Gaussian process regression is trained by annotating each blob with the number of people contained in and tested with the data from different data sets. The holistic estimate for a frame is the total number of people in each blob in the frame. To avoid counting multiple times in multi-camera environments, density-map is modified in an overlap region in such a way that the multiple occurrences of the object get less weight. Loy *et al*. [104] presented a semi-supervised uniform framework. The framework makes use of sparsely labeled data and unlabeled data to learn a kernelized ridge regression model. The model was used to get more informative and representative samples for annotations by exploiting the geometric structure of the unlabeled data. Spectral clustering was used to annotate the most informative frames reducing the burden of annotating all the frames. Finally, knowledge of one scene was used in the annotation of another scene by feature representation transfer techniques.

2.2.2 Local approaches for crowd counting

In local approaches, image features are extracted from the local patches of image and regression is applied locally to each patch of an image and estimate the number of people in each patch. In this case, crowd count is the direct sum of these local estimates. The size and shape features are extracted from the local patches of the image [85], and linear (cylinder model) is employed to estimate crowd count. [94] proposed pixel based density function for counting problem. The density function is a mapping between the feature vector associated with every single pixel value and its ground density value. The ground-truth density value of the pixel is approximated by fitting the normalized Gaussian kernel to the pixel dotted annotation. A multi-output regression model is proposed by [37] for crowd counting by extracting size, shape, edge and texture features from the local regions of the image. Their proposed regression model is able to estimate people count in spatially localized regions. [43] extracts SURF features from the local region of the image and support vector regression are employed to learn the correspondence between the features and the count of people. [76] proposed a counting framework based on multi-source multi-scale approach, which used multiple features extracted from the local regions of an image. These features are taken from different sources like HOG, Local binary pattern (LBP) and Fourier analysis. These features are computed at different scales for accurate and reliable counting. This work is extended by [12] which added more features like wavelets, SIFT, and GLCM. [114] proposed a local histogram-of-orientation gradient, in contrast to the standard histogram-of-orientation-gradient, used to describe the parts of the person independently and therefore helps in extraction of features even in partial occlusions.

A great deal of work in [24, 31, 150, 9] for crowd counting and density estimation has focused on local features such as edges, blobs extracted from the foreground. Typically, regression tech-

niques such as Ridge Regression (RR) [36], Bayesian Poison Regression (BPR) [31], Gaussian Process Regression (GPR) [24] are used to learn the model between the local features and count. However, a significant amount of information is lost in the calculation of such features. The accuracy and performance of such features heavily rely on the segmentation of the foreground. The foreground segmentation is a challenging problem especially because of varying lighting conditions and shadowing effect [176]. [153] considered texture features that are directly related to the crowd density and counting. The more texture means high crowd density which is not always true because of the incorrect foreground segmentation. Foreground segmentation of crowd only caters for moving crowd, but it performs poorly in the case of a static or very slow moving crowd. Furthermore, these features cannot be used for contextual crowd scene understanding. [56] proposed the simplified version of the previous work by estimating the object density using regression random forest improving the training accuracy. Similarly, [10] proposed an interactive and iterative density estimation technique. In this technique, the user annotates the object with the dot for object and line segment for its diameter to estimate the density. The Low-level features extracted are mapped to the density value. The learned mapping can be visualized intuitively by the user for error. The error indicates the need for further annotations to refine results in the next iteration. As in extremely dense crowd images, a single feature or detection does not work properly. Idrees *et al*. proposed a counting framework based on multi-source multi-scale approach, which uses multiple features from HOG based head detections, Fourier analysis and different sources taken at different scales for accurate and reliable counting [77]. Fourier analysis is applied along with head detections, and interest points are applied in a local neighborhood on multiple scales. Markov Random Field is used to impose smoothness among the counts from different patches and also among different scales. Crowd counting algorithms, in particular under the umbrella of local approaches, can either be based on 1) line-of-interest (LOI), 2) Region-Of-Interest (ROI) or a combination of both. There are some approaches which are based on head detection for counting and density estimation.

2.2.2.1 Line-of-Interest & Region-of-Interest based approaches

Barandiaran *et al*. proposed a solution for crowd counting in the pre-defined observed area by single overhead camera [13]. The observed area is drawn with the vertical equidistant lines such that the lines orthogonal to the direction of motion and the distance between them is greater than half of a person's width. When somebody crosses the line, the line counter is incremented by calculating non-zeros interval on the line using frame difference method. The algorithm makes uses of optical flow by Lucas and Kanade to find the direction the of motion. To count the multiple persons either crossing line simultaneously or one after the other without any separation between them. In the first case, the author suggested the use of overall non-zero interval and person's width to count. Also, when the person stops at the line, the algorithm may count it many times. While in this case, time is calculated independently for each interval based on the magnitude of optical flow, the height of the camera, and person's width. The assumption that the motion will only be orthogonal to equidistant lines drawn in the observed area, equidistant lines may not hold

in the real-world situation due to perspective distortions. The similar approach followed by [16] spatio-temporal maps for counting the number of people crossing the line. The author makes use of spatio-temporal maps which consist of slices of pixels of a person crossing the line with the assumption that color varies as the person crosses the line. Also, a spatial map of orientation and velocities of pixels using optical flow was used to find the direction of the blob and helps in the merging of blocks having a similar orientation. Finally, the author used linear regression for counting the number of persons in each blob.

The current state-of-the-art Line-of-Interest (LOI) methods are based on extracting and counting the blobs from the temporal slice of the video. In these blob centric approaches, counting is only done when the blob crosses the Line-of-Interest. In some cases, large blob extraction causes poor instantaneous count estimates. Ma and Chan proposed a local histogram-of-orientation-gradient which in contrast to standard histogram-of-orientation-gradient, can describe the parts of the person independently and therefore helps in extraction of features even in partially occluded people. The line counting algorithm based on integer programming is used to estimate the instantaneous count [113]. Usually, spatio-temporal normalization is required to cancel out the effect of the camera perspective. Flow mosaicking which caters for perspective normalization uses the variable width of the line, according to the speed of person crossing the line. The speed of the person is estimated using optical flow techniques. In his line counting framework, the crowd is first segmented into different directions using a mixture of dynamic textures [28]. The temporal slice image is obtained by sampling the line having a fixed width. An ROI is obtained by using a sliding window with a step size of a single pixel. The people crossing the Line-of-Interest will appear wider than the people crossing slowly. The author uses a temporal weight map to compensate for spatio-temporal perspective. A faster-moving person has high weights than a slower moving person. Furthermore, a spatial map is structured for perspective to normalized the effect of the angled camera. The regression is the most common technique to directly map the low-level feature input space to the number of people or density of the crowd. The author uses Bayesian poison regression with discrete outputs as compared to Gaussian process regression which models the real-valued output.

2.2.2.2 Deep Learning-based approaches

The performance of regression-based methods are improved further by employing convolution neural networks (CNN) [19, 155, 82, 214, 122, 133, 182]. In these methods, density maps are generated from the image patches, where count for each patch is obtained by performing the integration over the density map. Zang *et al.* [209] proposed a CNN model which can generate both crowd count and density maps using switchable alternative learning for counting and density map. The training and testing require a perspective map for perspective normalization which might not be available in practice. A Multi-column Convolutional Neural Network (MCNN) is proposed in [214], which utilizes three columns with a filter size of a different receptive field is used to compensate for perspective distortion. MCNN is trained to estimate crowd density at only three different scales in extremely crowded still images. Zhang *et al.* and Boominathan *et al.* [214][19]

proposed multicolumn CNN approaches, in which different columns with different filter sizes are used to capture multiple scales variation along with perspective. The final prediction obtained from columns is averaged to get a density map. Finally, the integral of the density map gives crowd count. Similarly, Switch-CNN [155] proposed switching architecture which intelligently switches appropriate regressor for particular crowd patch based on variation in density within the single image. However, these approaches are highly scene-specific and may perform poorly on cross scene analysis. [182] used shallow CNN architecture in the framework of ensemble learning where new models are added to the ensemble to fix the error from the previous model. This ensemble was used to estimate the density map. The density map is then spatially integrated to count. However, the research did not clearly explain the stopping criteria for adding new models to the ensemble. Therefore, ultimately adding new models to the ensemble might end up in overfitting. [82] used contextual information such as perspective weights, camera tilt angle and camera height to estimate crowd density. The contextual information is an auxiliary input to the Filter Manifold Network (FMN) to produce filter weights for the convolutional layer according to the scene context. Thus convolutional layer adapts itself to the context of the scene in contrast to the fine-tuning and transfer learning. The current datasets do not provide contextual information; therefore comparison with current datasets is not possible. The convolution is made adaptive to only parameters related to perspective. There might be more complex parameters related to the scene which are ignored.

Shen *et al.* [161] proposed Adversarial Cross-Scale Consistency Pursuit (ACSCP) approach using adversarial loss instead of traditional Euclidean loss to mitigate the blurry effect due to l_2 regularisation in the generation of density maps from crowd patches. Moreover, a new regulariser is proposed to enforce the scale consistency such that the number of crowd count in the large patch is coherent with the sum of crowd counts in the corresponding smaller non-overlapping patches. Similarly, Cao *et al.* [22] proposed Scale Aggregation Network (SANet) for accurate and efficient high-resolution of density maps using new training loss called local pattern loss. Sindagi and Patel [165] proposed Contextual Pyramid CNN (CP-CNN) in which local and global contextual information is incorporated with Density Map Estimator(DME) to generate high-quality density maps. Finally, all the maps are fused to estimate the crowd count and density. Crowd counting is formulated as a semantic scene model [74]. The three key factor including pedestrian, head, and their context as a composite body-part semantic structure information for two types of scene semantic models. These models are turned into different sub-tasks to train deep CNN for counting and scene semantic analysis. Xiong *et al.* [201] proposed Convolutional LSTM(convLSTM) to exploit temporal correlation along with spatial dependencies to boost the count accuracy in a complex scene. However, most of the datasets of the high-density crowd are still images of the crowd and therefore do not carry temporal information. Regression-based methods work well in high-density situations since they can capture generalized density information but suffer from the following limitations. 1) The performance of these methods degrades when applied to low-density situations due to overestimating the count. 2) These methods can not localize pedestrian in the scene and thus provide no information about the distribution of pedestrians in the environment

which sometimes very crucial for the crowd managers and security personnel.

2.2.3 Detection-Based approaches

On the other hand, *detection based methods* [44, 54, 93, 180, 59, 215], train object detectors to localize the position of each person, where crowd count is the number of detections in the scene. Detection-based methods can be further divided into two categories, 1) hand-crafted feature based models[180, 44, 47, 48, 54, 186, 212] and 2) deep features models [107, 187, 134, 135, 95, 211, 75, 117, 72, 213, 173]. In the first category, hand-crafted features like edges [27, 29], texture [199, 156, 27], and shape [27] are extracted from image to train SVM or boosting classifiers. After training, learned weights of the classifier are considered as a template for the entire human body. Subburaman *et al*. proposed a head detection based counting system, in which first interest points based on gradient information are extracted [169], which gives the approximate location of candidate location for the head detector, thus reducing the search space. Interest points are then masked using foreground masks obtained from background subtraction. A sub-window is used on interest points to detect head or non-head using an AdaBoost classifier. Finally, individual detections are finally merged to get the final count. These hand-crafted features have a low representation of the human body and performance of classifier degrades when applied to complex, crowded scenes. To model complex poses of pedestrians, DPM [54, 220, 100] learn a mixture of local templates for each body part. Although DPM is robust to complex poses, but feature representation and classifier cannot be jointly optimized to improve performance.

Recent advancement in deep learning has shown outstanding results in detection using CNN. In a typical CNN based approach, there is local connectivity of a region in the input image to the output image as compared to the traditional feedforward neural network. In a feedforward neural network, every input layer is fully connected with the output layer. Deep CNN is a compositional model, in which features are extracted ranging from low-level along the pipeline of CNN towards the final layers. The lower layers represent low-level features such as edges, and the subsequent layers represent abstract features such as shapes, etc. We have used Faster R-CNN for the detection of pedestrians in the low-to-medium density crowd videos. In Faster-RCNN [143], a small network namely Region Proposal Network (RPN) is used on top of the feature map to extract object candidates or region proposals in contrast to other approaches like Selective Search [174], CPMC [23], MCG [8], Edge boxes [221], etc. The advantage of RPN make Faster R-CNN an end-to-end pipeline for the detection and also does not add to the computation of the network. To detect objects at multiple scales, region proposals at various scale and aspect ratios are extracted using anchor boxes. The center of the anchor box is having a different aspect ratio and size and coincide with the center of the sliding window.

Table 2.1: Summary of crowd counting and density estimation using traditional machine learning approaches

Authors	Features							PN	ML model	Application	
	Segment	Texture	Edges	Shapes	Motion	Keypoints	Gradients			Density	Counting
Local approaches											
Barandiaran *et al*. [13]	✓	–	–	–	✓	✓	–	–	Linear Regression	–	✓
Benabbas *et al*. [16]	✓	–	–	–	✓	–	–	–	Linear Regression	–	✓
Ma and Chan [113]	✓	✓	✓	–	–	–	–	✓	Bayesian Poison Regression	–	✓
Cong *et al*. [42]	✓	–	✓	–	–	✓	–	✓	Quadratic Regression	–	✓
Li *et al*. [96]	✓	–	–	–	✓	✓	–	–	Adaboost Classification	–	✓
Subburaman *et al*. [169]	✓	–	✓	–	–	–	✓	✓	Adaboost Classification	–	✓
Ryan *et al*. [152]	✓	✓	✓	✓	✓		✓	–	Adaboost Classification	–	✓
Xu *et al*. [203]	✓	–	✓	✓	✓	✓	–	–	KNN, NN, Gaussian Process Regression	–	✓
Van Oosterhout *et al*. [178]	✓	–	–	✓	–	–	–	–	Kalman Filter	–	✓
Holistic approaches											
Marana *et al*. [118, 119]		✓	✓	✓			–	–	NN, Bayesian Classifier	✓	
Loy *et al*. [104]	✓	–	✓	✓	✓	✓	–	–	Adaboost Classification	–	✓

Chan *et al*. [25]	✓	✓					–	✓	Kernel Ridge Regression, Semi-supervised Regression	–	✓
Chan and Vasconcelos [32, 30]	✓	✓	✓	✓	✓	✓	–	✓	Gaussian Process Regression, Bayesian Process Regression	✓	✓
Kong *et al*. [89]	✓	–	✓				–	–	NN, Linear Regression	–	✓
Ryan *et al*. [151]	✓	–	✓				–	✓	NN	–	✓
Wang *et al*. [190]		–	✓				✓		NN	✓	
Ma [110, 111]		✓					✓	✓	Bag-of-Word,NN	✓	
Wu *et al*. [198]		–	✓					✓	SVM	✓	
Xiaohua [200]		✓						✓	SVM	✓	
Rahmalan [139]		✓					✓		SOM	✓	

The aforementioned approaches rely on the static background with the high camera angle. The total number foreground pixels, edges or textures are less likely to be good features for density estimation unless the effects of perspective are not considered.

Features extracted may either be normalized before or after for perspective distortion.

2.3 Crowd Motion Analysis

There are numerous approaches followed by researchers to understand the behavior of the crowd in real-time videos. Crowded video can be characterized by detecting objects in the crowd and then understanding the behavior of those objects. However, there is another approach based on the collection of local estimates of motion patterns and subsequently understanding the crowd holistically.

2.3.1 Motion representation of crowd

The collective behavior exhibit distinct motion patterns which are important in understanding the crowd behavior and crowd scene classification. Numerous studies [3, 125, 38] have been developed to detect, segment motion patterns by studying motion exhibited by individuals in the crowd. Moreover, how it contributes to the overall motion of crowd which ultimately help in the study of behavior analysis. But before looking at the crowd holistically, there is a need to look at the crowd motion representations at a more granular level. A considerable amount of literature has been published on the representation of the crowd. The following are some of the representation of the crowd.

2.3.1.1 Optical flow

Optical flow is a movement of each pixel from one frame to another with the assumption that brightness almost remains constant [14, 68, 106]. The flow vectors obtained may contain redundant information. To remove noise from flow vectors researcher used unsupervised [6, 7] or supervised [70, 71] dimensionality reduction techniques. Optical flow is widely used in crowd motion analysis. The current optical flow techniques are robust to camera motion as well as object motions. Optical flow only gives instantaneous motion information in the scene using two consecutive frames and therefore not capable of capturing long-range trajectories of the people in the crowd.

2.3.1.2 Particle flow

According to Lagrangian framework [160] people in the crowd act like a particle in the fluid. The particle flow is computed by moving grid of particle with optical flow through numerical integration. Impressive results have been demonstrated in crowd segmentation [3], abnormal crowd

behavior detection [125, 193]. However, particle flow doesn't consider spatial changes and also time delay is significant.

2.3.1.3 Streaklines flow

Streaklines are used in flow visualization and analysis based on Lagrangian framework [67, 179]. The Streaklines are defined as traces left as line upon injection of colored material in the flow. In computer vision, streaklines are obtained by repeatedly initializing the grid of particles at each frame. These particles are traced using optical flow. The streaklines flow is capable of capturing crowd motion better and faster than particle flow.

2.3.1.4 Spatio-Temporal features

Spatio-temporal features are computed locally. In extremely dense crowd, sometime optical flow may not give enough information because high variability of pedestrian movements. In these local approaches, the image is divided into cells of different sizes [207] or cubiods [90]. The Spatio-temporal gradients are used to capture steady-state 3-D motion characteristics of the scene. Kratz and Nishino [90, 91] demonstrated the use of spatio-temporal gradients in the detection of abnormal activities.

2.3.1.5 Tracklets

In dense crowds, keeping track of trajectories followed by the pedestrian in the crowd is important for the crowd behavior. Therefore the trajectories obtained are fragmented and might not be useful in the semantic understanding of the scene. The tracklet is a fragment of trajectory obtains by a tracker with a short period of time. In numerous studies [216, 218, 40, 184], tracklets are a computed from dense feature points, and certain model is applied to find coherency between tracklets for a long-range understanding of crowds.

Motion pattern is defined as a set of flow vectors that are a part of the same physical processes or a motion pattern [70]. To learn dominant motion patterns in the crowd or traffic flow. Min Hu *et al*. [70] used instantaneous motion flow-field of the video instead of long-term trajectories. The instantaneous flow field is a set of independent flow vectors and is readily available regardless of the density of the objects in the scene. The instantaneous motion flow-field is global and can be easily computed by optical flow techniques. As the density of crowd increases, detection of the object and keeping track of their trajectories for a longer period of time is not reliable. Therefore, the approaches based on object detections are applicable only in a low-density crowd. While the instantaneous motion flow-field is not really affected by occlusion due to its global nature. The motion field is obtained by optical flow techniques, and all the motion flow vectors are combined to form a global motion flow field. The motion flow can either be sparse which uses the interest points only or it can be dense which consider all the pixels in the computation of flow-field. The flow field obtained contain thousands of flow vectors. The flow vectors belonging to the background

are removed by thresholding the velocity magnitude. The author made use of sink seeking process for motion patterns instead of long trajectories. The sinks are endpoints of the path and can be learned from start and end points of trajectories. The flow vector having a similar movement pattern also has a similar path shown by their respective sinks. After clustering the similar sinks, a super track is extracted with maximum arc length representing the dominant motion pattern. Similarly, in another study, Min Hu *et al*. [71] used Gaussian ART [191] to reduce the number of flow vectors in the motion flow-field and still able to maintain the geometric structure of the motion flow-field. The author computed spatial and directional distance for the flow vectors to find the close neighbors in a directed graph. Furthermore, agglomerative clustering was used to cluster the flow vectors in a similar direction and proximity.

Cheriyadat and Radke [39] used low-level features such as Shi-Thomasi-Kanade detector [162] and Rosten-Drummond [147] detectors in the initial frame and keep detecting the new features in every fifth frame. New features detections in close proximity spatially are discarded. The feature points tracks which are closed spatially having similar orientation are clustered together using distance metric called Longest Common Sub-Sequence(LCSS). However, these approaches assume homogenous motion in each area of the scene.

Ali and Shah [4] proposed a flow segmentation framework assuming the moving crowd as an aperiodical system manifested by two dependent flow-field. The flow segments are the segments arise from spatio-temporal interactions between each other as well as from external forces. To locate these flow segments, the authors made use of the Lagrangian particle dynamic system. The flow-field can represent the physical properties of the scene. The Lagrangian particle dynamic system finds Lagrangian Coherent Structures (LCS) which divide the flow into different regions. The notion of Lagrangian Coherent Structures is synonymous to edges in the images. The Lyapunov exponent is to locate the coherent structures in the crowd. Mehran *et al*. [126] used social force model to model the complex interaction of individual in the crowd originally proposed by Helbing [66]. There are two kinds of forces involved in social force modeling of a crowd, i.e., interaction force and personal desire forces. These forces are mapped to their corresponding image frames resulting in force flow, which is used to model the normal behavior of the crowd using the bag-of-words model.

2.3.2 Crowd behavior classification

Numerous studies [5, 124, 127] model crowd based on hydrodynamics model with the assumption that people in the crowd behave like a particle in the fluid. Researchers use three different scales to model people behavior in the crowd. At a microscopic scale, every individual is tracked in the crowd of low-density. Sometimes, it might not be feasible to track every individual whereas, on another side, some application may require how the group of people interact with the crowd, their study of normal and abnormal behavior employ a mesoscopic point of view. A macroscopic point of view is used for segmenting global motion patterns in the crowd. The author used macroscopic scale point of view to segment the crowd flow using a technique called ”particle advection.” The

particle advection is performed by overlaying the scene with the grid of particles. The approach was then able to use the common mathematical framework of Lagrangian dynamics. But particles in the liquid are affected only by external forces exerted on them while people in the crowd may have their internal desire for a destination which cannot be modeled by the hydrodynamic model. [166] used linear dynamical system Jacobian matrix defined by optical flow to classify crowd behavior into five categories.

- Lane formation
- Arches
- Fountainheads
- Bottleneck
- Blocking

The proposed pipeline does not need detection or tracking to classify and is able to identify multiple behaviors in one scene. Similarly, [196] used radial information along with tangential trajectories to describe motion information with Curl and Divergence of motion trajectories (CDT). The CDT classify the behavior into five categories.

2.3.3 Crowd anomaly detection

Anomaly detection is a very important and key research aspect in visual analysis of crowd. Anomalous events are those events which violate our assumption about the scene is subjective and scene-specific and might have different interpretations. For example the presence of moving vehicle or bicycle in pedestrian space, running person in relatively slow-walking pedestrians. There has been a lot of research done [194, 4, 126, 115, 183] and still, there are open problems which need further research exploration. Crowd anomaly detection problem can be view as a supervised or unsupervised learning problem. In the case of supervise-learning, normal behavior is learned from data. If the unseen behavior deviates from the learned normal behavior is considered to be anomalous. While in the case of unsupervised learning normal and abnormal behavior is learned with the assumption that most of the data is normal.

Various approaches have been proposed in the literature for abnormal event detection. These approaches can be broadly classified into local and global approaches.

2.3.3.1 Local approaches

The local abnormality detection approaches use 2-D patches or 3-D cuboid motion features to model and detect an anomaly in a crowd scene. These modeling techniques include flow-field model, social force model, Hidden Markov model, and Bag-of-words model. Kratz and Nishino [90] used HMM to model the spatial and temporal relationship between local motion

patterns. The approach demonstrated good results for detecting unusual events. However, the model learned is not generalized and needs to be retrained for unseen crowded scenes. Kim and Grauman [87] proposed space-time Markov Random Field to model abnormal events. The nodes in MRF refers to optical flow motion pattern of the local region in the grid. The model is learned with a mixture of probabilistic principal component analysis for normal behavior. The model is used to detect abnormality at each node in MRF and is updated incrementally to incorporate new observations. A dynamic texture is spatio-temporal generative model for video [49]. It represents video as an observation from the linear dynamical system. The model, however, is not able to model multiple regions, each of which belongs to the semantically different visual process. Chan and Vasconcelos [28] use a mixture of dynamic texture for modeling, clustering and motion segmentation of video. In a mixture of dynamic texture where a collection of video sequences are modeled as samples from a set of underlying dynamic textures. Li *et al*. [98] used a joint detector which accounts for both appearance and dynamics for anomaly detection. Roshtkhari and Levine [146] uses online unsupervised learning method for detection of anomalies. In his work, he describes anomaly a spatio-temporal composition video or set of videos with a low probability of occurrence concerning previous observations. The author used the probabilistic framework to capture spatio-temporal video volumes. There are thousands of video volumes in a video of short duration. To remove the redundant information spatio-temporal are clustered using the BOV approach. Huam and Tjhoase used sparse subspace clustering algorithm to detect abnormal event detection by observing the difference between local space and normal space. The difference between local space and normal space is large for abnormal crowds. Ali *et al*. [4] used a Lagrangian Coherent Structure in Finite Time Lyapunov Exponent field for detection of unstable crowd flow. The LCS divide the flow into the region of qualitatively different dynamics and also locate their boundaries. Instability is detected if the number of flow segments and correspondences changes over time. Similarly, Mehran *et al*. uses particle advection to move particle as an individual with the average optical flow. Their interaction forces are estimated using Social Force Model [126]. The interaction forces are mapped into an image plane to obtain Force flow for each pixel. The Bag-of-words approach is used to differentiate normal from abnormal events. Raghavendra *et al*. [138] proposed particle swarm optimization(PSO) to perform particle advection and optimize force for each particle obtained from the social force model. In normal behavior, the population of particles drifts toward the area of main image motion by minimizing the interaction force in PSO fitness function.

2.3.3.2 Global approaches

In global anomaly detection approaches, anomalies are detected based on holistic features such as motion information. There are numerous studies [124, 167, 15] applied for global anomalies detection. Xiaong *et al*. [202] proposed a model to detect two abnormal activities: pedestrian gathering and running. The model uses kinetic energy and potential energy. Firstly, potential energy is used to compute crowd density and Crowd Distribution Index (CDI). Finally, both are used to detect pedestrian gathering. Based on CDI, improved kinetic is used to detect pedestrian running.

Ren *et al*. [142] proposed Behavior Entropy model to detect and localize abnormal behavior. Several types of abnormal behavior such as crowd dispersion, individual conversing the walking scene are detected by the computing the scene behavior entropy (SBE) and localized by computing their pixel behavior entropy (BE). Yang *et al*. [205] incorporated local density and velocity information in the pressure model to compute a Histogram of Oriented Pressure(HOP) for anomalies detection. Chen and Huang [34] use graph model to model the human crowd. Each isolated region is modeled as q vertex in the graph. Thus event detection is seen as a topological variation of the consecutive graph over the period of time. The features such as triangle deformation, eigenvalue base subgraph, and moments are used to detect the anomaly in human crowds. Wu *et al*. [195] uses a Bayesian framework to model escape and non-escape detection in crowds.

2.4 Summary

In this chapter, a comprehensive literature review of the different aspects of crowd analysis was presented. This literature review shows that most of the work has been done in the area of counting and density estimation using classical machine learning approaches for feature extraction. As it is evident from literature, not much work was done by combining various features and evaluating their effectiveness for crowd counting and density estimation. Therefore, we have combined various texture features such as LBP, GLCM, and GOCM. Most recent approaches are using deep learning for counting. However, most of them generate perspective normalized density maps and then apply regression to generate count from a density map. We have used state-of-the-art object detection for counting and then proposed Motion Guided Filter for recovering misdetections and suppressing false detections.

Moreover, to understand crowd behavior, we proposed hierarchical clustering for detecting dominant motion patterns as well as extracting useful motion information from a crowd scene.

"Why does my brain insist on counting the steps every time I walk up a flight of stairs? I just can't help myself. There's something about my mind that always wants to keep counting."

- Rachel Nichols

3

Crowd Counting and Density Estimation : Part I

In the first half of this chapter, the focus of this chapter is to explorer various feature extraction techniques using classical machine learning approaches. As it is evident from the literature review, there is no single feature to perfectly capture the complexity of the crowd. Therefore, various texture features and their combination has been evaluated for counting and density estimation.

In the second half of the chapter, the chapter focuses on counting and density estimation using deep learning techniques particularly deep convolutional neural network in the crowd of low-to-medium density videos for head detection task. For evaluation, we have used state-of-the-art object detection approaches based on Region-based Convolutional Networks as well as Single Shot Detectors. Major parts of this chapter have been published in the paper titled *Texture-based feature mining for crowd density estimation: A study*, Saqib *et al*. [156] and in the paper titled *Person Head Detection in Multiple Scales Using Deep Convolutional Neural Networks*, Saqib *et al*. [159].

3.1 Introduction

Crowd density estimation can be done using different techniques; one way is to model the background and subtract it from every frame to get foreground. Thus crowd density level is roughly in direct proportion to the pixel occupancy of foreground pixels relative to background pixels in the entire image. However, due to high occlusion in a dense crowd, such techniques may lead to incorrect results for crowd density estimation. Moreover, the efficiency of such a system is also dependent on how well the background is modeled. Besides occlusion, background modeling is

also prone to illumination and shadowing effect. Another approach to crowd estimation is using texture analysis. Researchers studied different texture features for crowd density estimation based on the fact that different crowd density levels exhibit different texture patterns.

3.2 Background and related work

Preliminary work on crowd density estimation was undertaken by Marana *et al*. [118, 119, 120]. According to Marana *et al*., low-density crowds exhibit coarse texture while high-density crowds exhibit very fine texture. Haralick *et al*. [63] proposed a large set of statistical features, which are based on a Gray Level Co-occurrence Matrix (GLCM). Marana *et al*. [120] used only a subset of these features (Contrast, Homogeneity, Energy, and Entropy) and evaluated its performance using a self-organizing map for crowd density estimation. Ma performed local texture analysis, such as GLCM and LBP, on a grid of multi-resolution cells to implicitly normalize the effect of perspective distortion [110, 111]. Similarly, Wu *et al*. [198] applied multi-resolution density cells for texture analysis. The resolution of the cell is dependent upon the location in the image to compensate for perspective distortion. Marana *et al*. [119] also proposed Minkowski fractal dimension features for crowd density. These features are extracted from binary images of a crowd to get different dilated images, each for different size of the morphological structure element. Different sizes of the structure element used are then mapped to the number of pixels by linear regression. Thus, the slope obtained from linear regression is used to calculate Minkowski fractal dimension features. Minkowski fractal value increases as the density of the crowd increases. Minkowski fractal may provide unreliable results when the background scene contains strong gradients with varying illumination. Xiaohua [200] used the 2D discrete wavelet transform (DWT) to extract texture based on the fact that high-density crowds exhibit high-frequency texture patterns while low-density crowds exhibit low-frequency patterns. A tree of Support Vector Machine (SVM) classifier was used to classify the crowd into four different density levels. Translation Invariant Orthonormal Chebyshev Moments (TIOCM) features used by Rahmalan [139] for crowd density estimation are also based on the fact that different moments of order exhibit different behavior for different densities of a crowd. Local Binary Pattern is a local feature descriptor, which has been used effectively for texture classification [132].

Numerous studies [58, 109, 128, 206] used the LBP histogram-based approach for crowd density estimation. While Zhe *et al*. [189] proposed LBPCM for crowd density estimation by applying GLCM on the LBP image instead of applying on an original image. Furthermore, the performance is evaluated on patches for crowd density estimation using SVM. Fradi *et al*. [57, 58] used a combination of principal component analysis (PCA) and linear discriminant analysis (LDA) to project high-dimensional LBP feature space to low-dimensional discriminative feature space, subsequently used for crowd density classification using multi-class SVM. In another study, a 24-dimensional Gabor filter [79] was applied to the LBP image to get the mean and standard deviation of the output of filter images as a feature vector to SVM. Furthermore, GLCM-based features are extracted and given to the firefly classifier for crowd density classification. Man-

fredi *et al*. [116] used a combination of two cameras in a master-slave configuration, where the master camera acquired crowded zones and the slave camera calculated Gradient Orientation Co-occurrence Matrix(GOCM)-based features for the patch. Then, a patch-based classifier along with motion and temporal information is used to select static crowded patches. Spectral texture-based Fourier features are used for crowd density estimation, using a multi-class Adaboost algorithm instead of a self-organizing map [86].

Together, these studies provide important insights into the application of texture for crowd density estimation. However, previous studies do not fully explore one distinctive feature for crowd density estimation. Therefore, in this study, we have comprehensively reviewed different texture features and all their possible combinations. Many such systems proposed in the literature either use texture to determine the crowd density level holistically at the scene level or in small patches. We have carried out extensive experimentation both at scene level and also at the block level to evaluate the features. We proposed the application of Fourier descriptor on the LBP image and also on the gradient image, as two new feature combinations.

3.3 Proposed methodology

We outline an SVM-based framework for the evaluation of different texture features as shown in Figure 3.1. According to the framework, images are divided into blocks of different crowd density levels. For feature extraction, blocks are further divided into overlapping cells. Features extracted from cells are concatenated to get a single feature vector for each block. Blocks are manually annotated to establish the ground truth according to the crowd density level as shown in Table 3.2. This kind of framework has two advantages. (1) Overlapping cells within the block can code more textures. (2) Division of frames into blocks can help in localization of potentially crowded areas and inherently reducing the effect of perspective. A multi-class SVM with RBF kernel is trained by blocks of different density levels. The hyper-parameters value of gamma and c were evaluated from tuning set in the logarithmic scale of $2^{-2}...2^{8}$. The best values of $c = 2^{2}$ and $gamma = 2^{5}$ were selected during the training which later on used for testing. Block accuracy is reported by testing trained model on unseen test blocks.

3.3.1 Perspective normalization

Objects or persons close to the camera contribute more to the feature vector than persons away from the camera. The effect is called perspective distortion, or far-shortening effect. To get a better estimate of the density and count, the feature extracted must be normalized. Typically, a density map or perspective map is constructed which assigns weights to the pixels of the object according to the depth of the object in the scene. Pixels belonging to the person near the camera are multiplied by lower weight than the person away in the far-field view. Chan *et al*. [26] uses two reference lines in the image plane, which corresponds to the two horizontal lines of equal distance in the real world. These two lines, AB and CD, are taken at both extremes of the scene

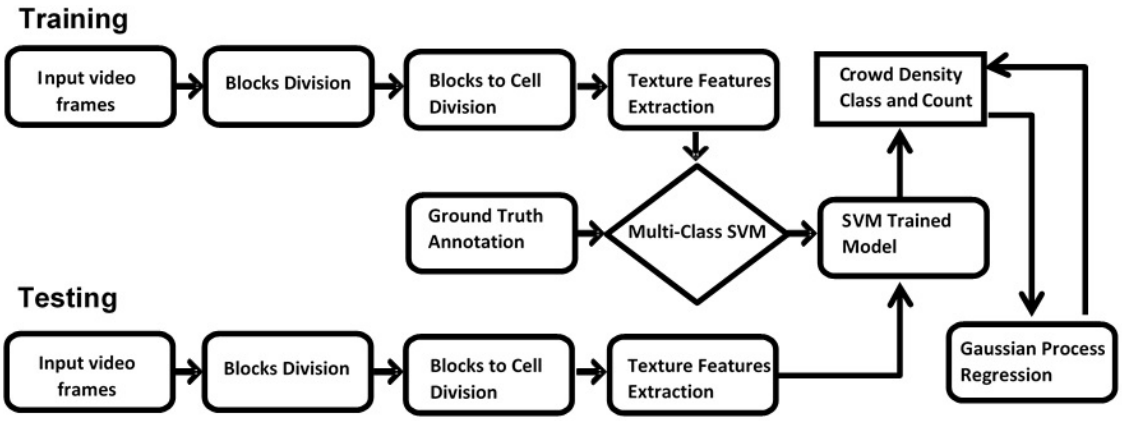


Figure 3.1: Block diagram of proposed crowd density estimation system

in the j^{th} dimension of the scene. The height of the object is manually annotated by the user at both of the lines. Then, a density map is approximated by interpolation for objects in between these two reference lines and heights. We have used detection based perspective map in second stage regression to normalize the effect automatically as opposed to the scene geometry (parallel lines), manual annotation or calibration from user [26] [108]. Not every scene contain parallel lines which can be used in calibration. Currently available crowd analysis datasets used for crowd density estimation contains multiple instances of pedestrians which can be automatically detected.

Firstly, HOG [44] is used to detect multiple instances of pedestrians which is sufficient to know the variations in sizes of the pedestrians across the scene. The width $w(j)$ is the width of pedestrian's bounding box along the j^{th} dimension of the scene and is linearly interpolated between minimum width w_{min} and maximum width w_{max} of the pedestrian's bounding boxes. The scaling weight factor for object width is $S_w(j) = w_{max}/w(j)$. Similarly, height $h(j)$ is the height of the pedestrian's bounding box along the j^{th} dimension of the scene and is interpolated between minimum height h_{min} and maximum height h_{max} of the pedestrian's bounding boxes. The scaling weight factor object height is $S_h(j) = h_{max}/h(j)$. Similarly, full scale ratio, which accounts both for vertical and horizontal perspective, is given by (3.1)

$$S(i,j) = (h_{max}.w_{max}) \,/\, (h(j).w(j)) \tag{3.1}$$

3.3.2 Feature extraction

Texture can be defined as a repeating pattern of local variations in image intensity which is too fine to be distinguished as a separate object at the observed resolution. Texture is an important descriptor and has many applications in different machine-learning tasks, like classification, shape determination, object orientation, etc. The aim of this study is to extract distinguishing texture features and evaluate their performance in the crowd density estimation. First, different texture features are extracted from images having crowds of varying density, and then images are annotated manually according to the density of the crowd.

The following features are extracted and evaluated for crowd density estimation.

3.3.2.1 Gray-Level Co-occurrence Matrix

GLCM is widely used second-order statistical-based texture features, based on the gray-level distribution of the pixels and the neighboring spatial relationship between the pixels. Originally, many texture features were proposed by Haralick *et al.* [63], Only four of them were used mainly for crowd density estimation by Marana *et al.* [120]. The following are the image texture features mainly used for crowd density estimation: energy, entropy, contrast, and homogeneity. The Gray-Level Co-Occurrence Matrix (GLCM) $P[i,j]$ is calculated by counting all the pairs of pixels having gray level i and j separated by distance d and at an orientation $\theta \in \{0°, 45°, 90°, 135°\}$. The following features are extracted from GLCM $P[i,j]$ given by the (3.2), (3.3), (3.4), (3.5).

$$Entropy = -\sum_{i}\sum_{j} P[i,j] \log P[i,j] \tag{3.2}$$

$$Energy = -\sum_{i}\sum_{j} P^2[i,j] \tag{3.3}$$

$$Contrast = -\sum_{i}\sum_{j} (i-j)^2 P[i,j] \tag{3.4}$$

$$Homogeneity = -\sum_{i}\sum_{j} P[i,j]/(1+|i-j|) \tag{3.5}$$

3.3.2.2 Local Binary Pattern

Local Binary Pattern (LBP) is an excellent representation model for image texture classification [132]. In feature extraction, $LBP_{Q,R}$ is calculated for each pixel with an angular quantization of $Q = 8$ and spatial resolution of $R = 1$. The center pixel value is compared with its 8 neighbor values; if the center pixel value is greater than any of its neighbor values, this results in 0 or otherwise 1. Thus, 1's and 0's obtained are traversed clockwise to get the binary string, which is a unique representation of a spatial structure. The binary string is often converted to decimal for calculation of histograms. Finally, a normalized histogram is calculated over the image, for the frequency of each decimal number.

3.3.2.3 Local Binary Pattern Co-occurrence Matrix

Normally, texture features are extracted from the co-occurrence matrix of gray-levels. However, by applying LBP on the image, a histogram or LBP image is obtained. In LBPCM, instead of directly calculating the Gray-Level Co-occurrence Matrix from gray-level values, a co-occurrence matrix is computed from the LBP image, thus called a Local Binary Co-occurrence Matrix (LBPCM) [189]. Then, similar texture features are extracted as that of a GLCM, as shown in (3.2), (3.3), (3.4), (3.5) respectively.

3.3.2.4 Gradient Orientation Co-occurrence Matrix

GOCM [116] as opposed to GLCM, is co-occurrence matrix of different orientations based on gradient map. Gradient map is a two dimensional matrix incorporating both magnitude and direction. Gradient map G is calculated by applying Prewitt filter. Given an image block B_i, where $I(p)$ is the gray-level intensity of the pixel $p(x, y)$. Gradient map G is calculated by (3.6), (3.7).

$$Gmag(p) = \sqrt{(\nabla I_x(p))^2 + (\nabla I_y(p))^2} \tag{3.6}$$

$$Gdir(p) = \arctan(\nabla I_y(p)/\nabla I_x(p)) \tag{3.7}$$

Where $Gmag(p)$ and $Gdir(p)$ are magnitude and direction respectively. GOCM is a $9X9$ matrix extracted from the $Gdir(p)$ values by binning it to nine different bins of equal spacing. At first step in $Gmag$, each pixel's magnitude value is compared to the magnitude values of the other pixels in the $3x3$ window of neighborhood. In order to look for most similar pixel magnitude values, the following magnitude function is used.

$$V(x, y) = \arg\min ||Gmag(x_i, y_i) - Gmag(x, y)|| \tag{3.8}$$

where $Gmag(x_i, y_i)$ are pixel magnitude values in the neighborhood of $Gmag(x, y)$. In the next step $Gdir$ of pixel $p(x, y)$ and the $Gdir$ of the most similar pixel are added together with $Gmag$ value of the $p(x, y)$. The aim is to strengthen the information carried by strong gradient as compared to weak gradient. The complete algorithm adopted from [116] is given below.

Algorithm 1 Gradient Orientation Co-occurrence Matrix

Input: B_t,$Gmag(x, y)$,$Gdir(x, y)$
Output: A $9X9$ **GOCM matrix** $P(i, j)$

Initialization :
GOCM matrix $P(i, j) = 0$
Loop process :
for Every reference pixel $P_r(x_r, y_r)$ in B_t **do**
    Neighborhood of reference pixel given by $N_r = p_i \in B_t | \; ||p_i - p_r|| < D$
    Searching for the most similar pixel in the neighborhood N_r of $P_r(x_r, y_r)$ using the magnitude function $p_b = V(x, y) = \arg\min ||Gmag(x_i, y_i) - Gmag(x, y)||$
    Orientation of reference pixel and orientation of most similar pixel is given by $O_r = G_{dir}((x_r, y_r))$ and $O_b = G_{dir}((V(x, y)))$
    $P(O_r, O_b) = P(O_r, O_b) + Gmag(x_r, y_r)$
end for
return $P(i, j)$

The following features are extracted from the GOCM matrix and concatenated to form a single feature vector of $\mathbb{R}^{24}$.

1. Sum: sum of all the elements of GOCM matrix $S = \sum \sum_{i,j} P(i, j)$

2. $Hom = -\sum\sum_{i,j} P(i,j)/1 + |i-j|$
3. Max maximum element in P
4. Energy $E = \sum\sum_{i,j} P(i,j)^2$
5. Entropy $E_t = -\sum\sum_{i,j}(P(i,j). * \log(P(i,j)))$;
6. Conditional mean $con\_meani = \sum_j jP(i,j)$
7. Correlation
8. Diag vector is the vector of principal diagonal values of P.

3.3.2.5 Gabor filters

Gabor filters are used effectively in many applications, such as face recognition, texture classification, and image and document analysis. The function of the Gabor filter is very similar to that of mammalian visual cortical cells. Gabor filter has the ability to use different orientation and different scale to represent texture for classification. In this study, it has been used to classify different density levels of crowds based on Gabor features. The first step in Gabor features extraction is to make a filter bank in which filters are taken at the scale of 5 and orientation of 8. In the spatial domain, Gabor transform is a short-time Fourier transform with Gaussian window and is given by the following formula.

$$Gf(x,y) = \frac{f^2}{\pi\gamma\eta}\exp\left(\frac{-\left(x'^2+\eta^2y'^2\right)}{2\sigma^2}\right)\exp\left(j2\pi fx'+\phi\right) \tag{3.9}$$

where
$x' = x\cos\theta + y\sin\theta$
$x' = -x\sin\theta + y\cos\theta$
By convolving each filter in the filter bank with the image to get the response image. Let $G(i,j)$ be the response matrix. The following features are extracted from response matrices.

1. $Local\_Energy = \sum_i\sum_j |G(i,j)|^2$
2. $Mean\_Amplitude = \sum_i\sum_j |G(i,j)|$

Features extracted are concatenated together to get a feature vector of $\mathbb{R}^{n\times 80}$for n images.

3.3.2.6 Fourier features

Fourier transform is the universal mathematical formulation for understanding the structures in images. Fourier transform of an image $f(x,y)$ of size MxN is given by (3.10).

$$F(u,v) = 1/(MN)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)} \tag{3.10}$$

where $(M/2, N/2)$ is the center of the frequency spectrum, called the frequency rectangle, which shows the average gray-level values in the image. By careful observation, the width of the frequency rectangle is related to the density of crowds. The width of the frequency rectangle increases as the density in the crowd increases, as shown in figure 3.2. Another important aspect, one can infer from the spectrum is that low-density crowds have a coarse and uneven spectrum, while high-density crowds have more sharper and finer pattern as we move away from frequency rectangle. It is more convenient to extract features and do analysis in a polar coordinates system. Let $F(r, \theta)$ be spectrum in a polar coordinate system where r and θ are the radius and angle, respectively. Fourier features are extracted for the original image, gradient image, and LBP-image using equation (3.11) and equation (3.12)

$$F_R(r_i, r_{i+1}) = \sum_{r=r_i}^{r_{i+1}} F(r) \tag{3.11}$$

$$F_\theta(\theta_j, \theta_{j+1}) = \sum_{\theta=\theta_j}^{\theta_{j+1}} F(\theta) \tag{3.12}$$

where R and θ are typically taken $N/2$ and π respectively. The $F_R(r_i, r_{i+1})$ $F_\theta(\theta_j, \theta_{j+1})$ are quantized into 16 and 8 levels results in feature vector of $\mathbb{R}^{24}$.

3.3.3 Gaussian Process Regression

At the second stage of the framework, Gaussian Process Regression(GPR) [26] is used to regress low-level features in each block to the number of people as shown in Figure 3.1. The GPR is fully specified by covariance function being used, which model the relationship between input features and output count.

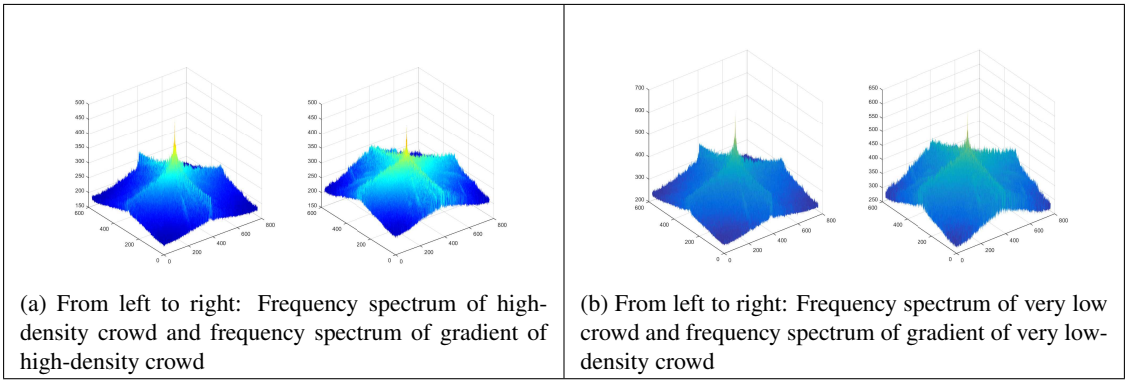
$$k(x_p, x_q) = h_1 \exp^{\frac{-|x_p - x_q|^2}{h_2}} + h_3(x_p^T x_q + 1) + h_4\sigma(p, q) \tag{3.13}$$

Equation (3.13) uses four hyper-parameters $h1, h2, h3$ and $h4$ learned from training blocks during training phase. While x_p and x_q could be samples from training and testing datasets.

3.4 Experimental results and analysis

3.4.1 Preparation of dataset

In this work, we have evaluated the performance of the co-occurrence-based and spectral-based texture feature descriptors and their possible combinations on benchmark $PETS2009$ dataset. The video dataset contains frames having a resolution of $768X576$ from different viewpoints.The video sequences $12-54$, $12-43$, $13-47$, $13-59$ and $14-06$ were used for evaluation of features. The dataset is divided into four crowd density levels according to the number of persons as shown



(a) From left to right: Frequency spectrum of high-density crowd and frequency spectrum of gradient of high-density crowd

(b) From left to right: Frequency spectrum of very low crowd and frequency spectrum of gradient of very low-density crowd

Figure 3.2: Difference in Fourier spectrum of high-density and very low-density crowds

(a) Very Low Density Crowd

(b) Low-Density Crowd

(c) Medium-Density Crowd

(d) High-Density Crowd

Figure 3.3: Crowd density levels

in Table 3.2 and Figure 3.3. As there is a temporal correlation between consecutive frames training samples are taken at sparse intervals to avoid overfitting. For performance analysis, each frame of video is divided into set of blocks $B = \{b1, b2, ..., bn\}$. The size of each block is considered as hyper-parameter, features extracted are evaluated at different blocks sizes and the accuracy is shown in Table 3.5. Frames are divided into blocks of size $128X128$, $256X256$, $512X512$. Block sizes lower than $128X128$ are not explored because the crowd density is not discernible at a lower resolution. Block-level ground truth is established by manually annotating the blocks according to the different density levels into very low, low, medium and high. In order to code more texture, each block is further divided into overlapping cells $bi = \{c1, c2, ...cn\}$ of $64X64$. Features extracted from each cell of the block are concatenated together to get one feature vector for each block of the image.

3.4.2 Performance analysis

Before classification, all frames are converted to gray-level images; features are extracted from the images and are fed to multi-class SVM for training and testing. As Table 3.5 shows, there is a significant decrease in accuracy for all the features based on a co-occurrence matrix *e.g.* GLCM, GOCM, LBPCM. In co-occurrence features GLCM, GOCM performed relatively well as compared to LBPCM. The reason behind the poor performance of LBPCM is that it is calculated from

Table 3.1: PETS 2009: S1 Person count and density estimation

Sequence	Test Set	Train Set	Crowd Size
13-57	110	110	5 to 34
13-59	120	120	3 to 24
14-06	100	100	0 to 42

Table 3.2: Reference crowd density levels

Density Level	No of Persons
High	>21
Medium	17 to 21
Low	9 to 16
Very Low	<8

LBP-image instead of the original image, thus not able to properly code texture. Interestingly, spectral features such as Fourier, Gabor, and our proposed combination of Fourier on gradient image work well both at frame level and block-level. The accuracy increases for almost all the features when the block size decreases from $521X512$ down to $128X128$. There is no significant change in accuracy for block size $128X128$ as compared to $256X256$. Therefore, block size $256X256$ is taken as an optimal block size for the experimentation. Our other proposed combination of Fourier does not work well, because their calculation is based on LBP-image instead of an original image. Among the spectral features, Gabor features turned out be the most distinctive and discriminative features as shown in Table 3.5 and its confusion matrix 3.4. Though the training time for Gabor features is marginally high, it has shown good overall performance both at the block level and frame level.

In the second stage of regression, texture features alone does not perform well because of the poor correlation between input feature space and output count. Therefore, other low-level features such as size, shape, edges taken together with texture performed well with a Mean Absolute Error (MAE) of 1.6.

Table 3.3: Frame level accuracy

Extracted Features	Features dimension	Accuracy
GLCM	nX16	86.6%
GOCM	nX24	86.6%
LBP	nX439684	85.71%
LBPCM	nX16	76.19%
Gabor	nX80	95.23%
Fourier	nX26	90.38%
Gradient-Fourier	nX26	91%
LBP_Img_Fourier	nX26	80%

Table 3.4: Confusion Matrix of Gabor Filter

	Very Low	Low	Medium	High
Very Low	**89.4%**	10.5%	0%	0%
Low	0%	**100%**	0%	0%
Medium	0%	5.2%	**94.7%**	0%
High	0%	0%	5.5%	**94.4%**

Table 3.5: Block level accuracy

Extracted Features	Frame level Accuracy	Block Size Based Accuracy		
		128X128	256X256	512X512
GLCM	76.6%	78%	78%	86.6%
GOCM	76.6%	78%	80.95%	86.6%
LBPCM	75.71%	75%	76.19%	86.6%
LBP	66.19%	76%	85.71%	76.6%
Gabor	85.23%	90%	**95**%	91%
Fourier	80.38%	88%	90%	82%
Gradient-Fourier	81%	85%	92%	80%
LBP_Img_Fourier	70%	78%	80%	75%

3.5 Head detection using deep convolutional neural network

Person detection is an important problem in computer vision with many real-world applications. The detection of a person is still a challenging task due to variations in pose, occlusions and lighting conditions. The purpose of this study is to detect human heads in natural scenes acquired

Table 3.6: Gaussian Process Regression results (Mean Absolute Error)

Extracted Features	**MAE**
Size	2.06
Edges	2.75
Shape	3
Texture	4.2
Size+Edges+Shape+Texture	1.6

from a publicly available dataset of Hollywood movies. In this work, we have used state-of-the-art object detectors based on deep convolutional neural networks. These object detectors include region-based convolutional neural networks using region proposals for detections. Also, object detectors that detect objects in the single-shot by looking at the image only once for detections. We have used transfer learning for fine-tuning the network already trained on a massive amount of data. During the fine-tuning process, the models having high mean Average Precision (mAP) are used for evaluation of the test dataset.

3.6 Introduction

Object detection is a significant and critical part of many practical applications of computer vision ranging from face detection and surveillance to many other industrial applications. In classification, a classifier is trained to categorize and label the content of the image globally. While in detection, detector not only categorizes and label the object but also localize the bounding box of the object. The object detector faces challenges similar to that of classification. These challenges include viewpoint variation, scale changes, illumination changes, occlusion, background clutter, and deformation. Person detection is a popular research topic due to its applications in safety and security. Although technology has reached maturity in face detection and recognition, person detection still poses many challenges due to the articulated nature of the human body. However, in this study, the focus is on head detection in the images taken from video. The choice of head detection for detecting the number of people is because it is visible most of the time even when other body parts are fully occluded. The feature extracted for head detection are not discriminative enough to be used alone for person detection. Therefore contextual information in conjunction with the features extracted for the head is used for detection [181]. The contextual information provides additional cues and useful information for detection and recognition. The use of contextual information is in contrast to the traditional way of extracting features for an object inside the bounding box.

Previously, machine learning approaches applied to computer vision tasks used hand-crafted features. The features are less robust and discriminative as compared to the learned features. Recently, learned feature using deep learning techniques had been successfully applied for ma-

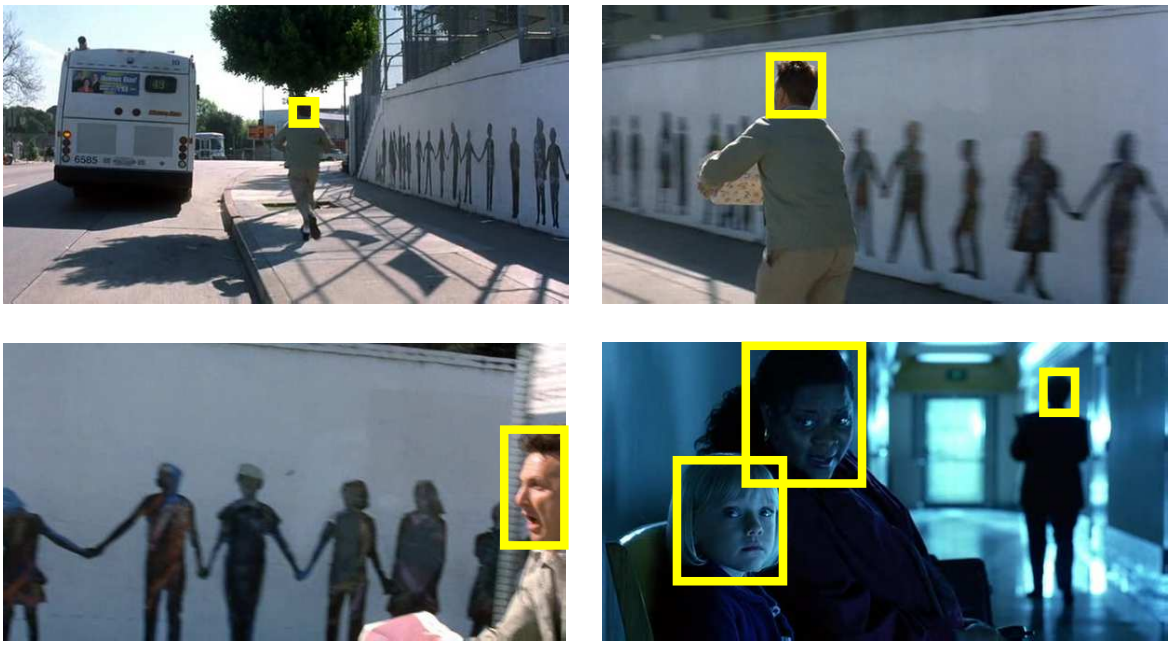


Figure 3.4: Sample images from HollywoodHeads(HH) dataset with ground-truth annotations. (Best viewed in colour)

jor computer vision tasks such as segmentation [102], classification [65], detection and recognition [143]. The deep learning techniques have been winning the classification and detection challenges such as ImageNet [46] for so many consecutive years surpassing the performance of hand-crafted features by a wider margin.

The structure of typical deep Convolutional Neural Network (CNN) consists of several layers followed by classification layer. However, object detector based on deep CNN differs in such a way that it not only classify but also localize the bounding box containing the object. Each layer in the deep CNN extract features at different abstraction level. The lower layers extract features representing more fine details in the image like edges and texture. More complex and abstract features are extracted such as shape, as we go across the pipeline of the neural network towards the final layers. By truncating the last layer, most of the previous layers can represent input images in abstract space of network parameters or weights. Deep CNNs are data-hungry architectures, to work properly for particular task requires a massive amount of data. So in most of the cases, more real data is useful if available. However, data augmentation is used to fulfill the requirement for deep CNN. If the last classification layer is removed, the network weight or parameter are pre-trained weights, and the models are called pre-trained models. These models can be generalized to many other similar tasks. This is widely known fact that these pre-trained models work best for unknown datasets. The pre-trained network parameter can be used for fine-tuning the network for a new task. Thus the resultant network converges faster and performs well instead of training from scratch. Most of the pre-trained models are available are trained on a hugely popular, most extensive collection of images such as ImageNet [46]. In this study, we have extensively carried out experimentation with state-of-the-art object detectors based on deep learning to detect head in

the videos as shown in the Fig 3.4.

3.7 Background and related work

The efficient object detector needs to detect an object of different scale and aspect ratio anywhere in the image. Traditionally, the concept of the image pyramid is used to represent the image at a different scale and then sliding window approach is applied to search object of various scale anywhere in the image. The features are extracted from sliding window position on the image and passed on to the classifier for detection. Much of the previous research extract hand-crafted feature such as LBP [132], HOG [44], and SIFT [103] for describing the objects. The complete pipeline of object detection includes the preparation of data samples into positive and negative followed by training the classifier with extracted features along with their corresponding bounding boxes. Most of the time, trained classifier is applied to negative samples, if the classifier classifies them as an object, then these false positive samples are included as hard negative, and the classifier is retrained. Due to sliding window approach, classifier detects multiple bounding boxes of an object. Therefore, a non-maximum suppression is applied to remove redundant and overlapping bounding boxes [44]. However, all of these approaches are computationally expensive, and furthermore, the training process is complex, not end-to-end and often done in stages. The complex training process makes these approaches not suitable for real-time applications.

The current resurgence of neural networks especially CNN has shown remarkable results in image classification challenges surpassing human-level performance on certain tasks [149]. The success of the CNN can be attributed to the availability of huge amount of data as well as the processing power in the form of GPUs to process the data. The features extracted from CNN are powerful, robust and expressive as compared to its traditional counterpart like DPM [53], HOG [44], SIFT [103]. A series of convolution layers followed by non-linear activation units constitutes the convolutional neural network. In CNN, regions in one layer called receptive field are connected to another layer. A series of filters are applied to each layer to derive the output. The weights or parameters of the filter are learned during the training process. In contrast to the feedforward neural network in which every layer is fully connected to other layers. The features extracted from CNN architecture are locally invariant and compositional, in which higher more complex features like objects and shapes are constructed from low-level features such as edges. Most of the popular architectures are based on CNN e.g. AlexNet [92], ZF [208], GoogLeNet [172], VGG16 [164] and ResNet [65].

In classification, sliding windows applied on image require much computation for scanning each possible location in the image. To avoid scanning all possible location, Selective Search (SS) [174] computes region proposals of different sizes and start merging from pixel level into objects. The merging of pixels is based on similarity in low-level features such as texture and color. The SS is a class agnostic method which detects multiple foreground objects without knowing the class of the object. The computation of SS is fast and act as a pre-processing step for the more highly specialized classifier to find out the exact class within the bounding box. The initial

implementation of the selective search was not optimized for GPU based implementation and thus being a major bottleneck in the object detection framework. Edgeboxes [221] relatively takes less time to compute region proposal as compared to Selective Search.

A specialized classifier called Region-based Convolutional Neural Network (R-CNN) [61] is used to classify an object into their specific classes. The R-CNN expects fixed size images, i.e., $227X227$ as R-CNN uses Alex Net [92] as its backbone CNN architecture. Therefore, the region proposals ($\sim$2k) are warped before passing it on to the R-CNN which uses SVM to classify the object and linear regression to find out more exact bounding box boundaries for the object. Despite the fact that R-CNN works very well, however, the computation is very slow. For each region proposal, the image has to be passed on through the network. Moreover, the region proposals are computed in an offline manner. Thus end-to-end training of R-CNN is very complex. In the next iteration of the R-CNN called Fast R-CNN [60], there is a change in network architecture; region proposals are extracted from convolutional feature map using Region of Interest Pooling (RoI) rather than directly from images. The fact that most of the region proposals are overlapped, thus convolutional feature map is computed once and shared across all computation of the region proposals. This change in architecture drastically reduces the overall time for detection. Furthermore, the Fast R-CNN architecture unified the network for extracting the region proposals followed by its classification and regression to its tighter boundaries. The SVM classifier is replaced by softmax layer in Fast R-CNN [60]. The region proposals from convolutional feature map project to its corresponding spatial counterpart. In Fast R-CNN [60], Spatial Pyramid Pooling (SPP) make sure Fully Connected layer (FC) get fixed-size feature vector. The major shortcoming of Fast R-CNN is the time taken for region proposals at the test time. Also, the training pipeline of the architecture is complex. In yet another iteration, Faster R-CNN [143], a fully convolutional neural network is used as a Region Proposal Network (RPN) at the cost of very low computation instead of using SS for region proposal from convolutional feature map. The use of RPN as region proposal made Faster R-CNN a streamlined end-to-end trainable network for object detection.

All the previously discussed methods consider detection as a classification problem. The detection is carried out in stages. The first stage computed proposals and classified in the second stage into object categories. However, there are some methods, e.g., YOLO [140], SSD MultiBox [101], which consider detection as a regression problem and glance at an image once for detections. Therefore called single shot detectors. The YOLO [140] divides the image into a grid of $13x13$ resolution. Each cell in the grid predicts the bounding box of the object along with its confidence score. The bounding boxes with low confidence score are removed by setting a threshold. The YOLO [140] performance is fast but not as accurate as its other counterparts. The reason behind its low accuracy is that it only predict one type of class in a grid. The Single Shot Detector MultiBox [101] compute a convolutional feature map by looking only once at the image and predicts bounding boxes at multiple of these feature maps for the objects of various scales.

Previous studies used different models to capture various aspects of head detection in contextual reasoning [181]. The Global model predicts the scale and coarsely localize the object using full image, while Local CNN captures the object appearance. Moreover, the Pairwise model

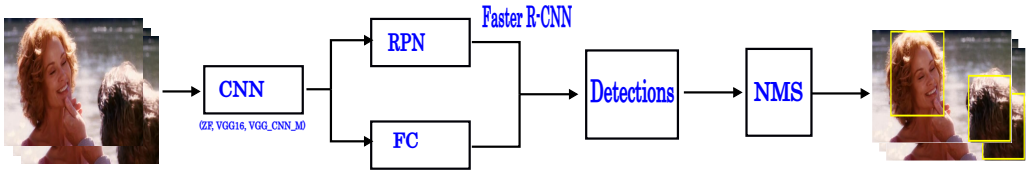


Figure 3.5: Faster R-CNN detection framework

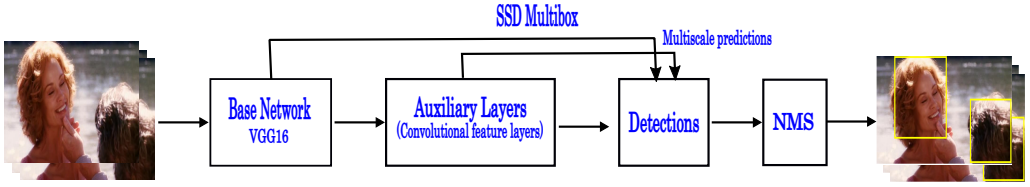


Figure 3.6: Single Shot Detector MultiBox

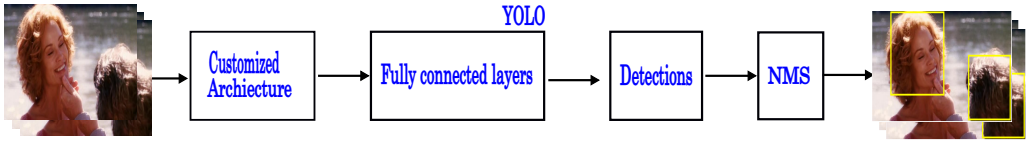


Figure 3.7: YOLO framework

captures the relationship between the objects. All the above models are jointly optimized for contextual head detection. However, such models do not provide a real-time unified architecture for head detection. Similarly, the performance of methods used for face detection [123] on head detection task is very low. The choice of particular object detection depends upon the trade-off between speed and accuracy. The single shot detectors are fast and suitable for the embedded real-time application. However, region proposals technique such as Faster R-CNN [143] is more accurate and intended for applications where accuracy is more important than speed. Furthermore, the size of an object is another consideration for the selection of detector to be used for detection. Both types of detector perform poorly on the very small scale objects.

3.8 Proposed methodology

Transfer learning is an approach widely used in deep learning applications in which pre-trained CNN is used as a feature extractor for a given dataset. The features are typically extracted from the last layer in the form of activations and cached to the disk. Then a standard machine learning classifier such as SVM is used for training and testing on the features. There is another type of transfer learning, more powerful than just feature extraction approach is called fine-tuning. In fine-tuning, the last fully-connected layers are replaced with the new set of fully-connected layers suitable for a given task. Typically the added fully-connected start learning from the previous convolutional layers while keeping the convolutional layers frozen so that their weights cannot be modified. However, in this study, all the layers are included in the fine-tuning process. The

process involves the state-of-the-art neural network architecture already trained on a large collection of images such as ImageNet [92] and PascalVOC [52]. These network architecture includes VGG16 [164], ZF [208], VGG_CNN_1024 [33] which contains rich and discriminative filters. A very small learning rate is used during retraining of the architecture making sure that already learned *CONV* filters do not deviate dramatically.

For Faster R-CNN [143] and SSD MultiBox [101] detector, experiments have been carried out in Caffe [80] framework. The first approach proposes regions in the bounding boxes and then applying the high-quality classifier to classify into object class categories. These approaches are called Region-based convolutional Neural network. In Faster R-CNN, the convolutional layers are shared by both the detection network and by small Region Proposal Network (RPN) as shown in the Fig. 3.5. The RPN is a small network computes region proposal from the last shared convolutional layers. The RPN consist of few CNN layer which does not add to the overall computation. The RPN can be trained in end-to-end fashion for object detection proposals. The concept of anchor boxes is introduced to cater for the object of different scale and aspect ratio. In anchor boxes, a single image and single filter size are used whereas for each location of feature map several anchor boxes of different scale and aspect ratio are used. In RPN, the sliding window on the last shared convolutional feature map encodes the output into $256 - d$ feature vector and $512 - d$ feature vector in the case of ZF [208] and VGG16 [164] respectively followed by non-linear activation unit of ReLu [143]. The extracted features are fed into regression for finding bounding box coordinates and classification layer for classifying the bounding box region into one of the class.

The Single Shot Detector detects the object by a single pass of the image through deep Convolutional Neural Network without explicitly proposing the more probable regions. SSD MultiBox [101] detector is one of the techniques which uses the same powerful CNN network architecture VGG16 [164] generally used for classification. However, the last classification layers are truncated, and some auxiliary layers are added for object detection as shown in Fig. 3.6. The network architecture without additional layers is called base network. For all the experiments, VGG16 [164] has been used as a base network. The additional layers are organized in such a way that the CNN feature map size decrease down in size until the extraction of final feature vector. These layers produce a fixed-size collection of bounding boxes by applying a convolutional filter on these feature maps. As a result, there are some redundant boxes which are suppressed by Non-Maxima Suppression (NMS) for boxes having Intersection over Union (IoU) less than 0.7 with the ground-truth bounding boxes. The usage of all feature map of additional layers serves an opportunity to predict the detections at various scales.

The usage of several feature maps is in contrast to other state-of-the-art object detector YOLO [140] where single feature map is used as shown in Fig. 3.7. The feature map is divided into a grid of cells. The set of default bounding boxes are associated with each feature map cell position. The position of default box is fixed relative to the cell. These default boxes act like anchor boxes in Faster R-CNN [143] to capture the object of different aspect ratio. At each cell location, position offset is predicted along with the class score of an object present in the box. The offset indicates the coordinates how much far away is the predicted bounding box from one

of the default bounding boxes.

3.9 Experimental results and discussion

We trained our models with Nvidia Quadro $P6000$ GPU with the common implementation settings of weight decay, momentum, learning rate and batch size of 0.0005, 0.9, 0.0001 and 64 respectively for all the detection frameworks.

3.9.1 Datasets

We have carried out several experiments to evaluate the performance of various detectors on publicly available datasets. In this section, we discuss the two datasets that have been used for experimentation namely HollywoodHeads (HH)) [181] and Casablanca datasets [144].

3.9.1.1 HollywoodHeads (HH) [181]

The HH dataset contains $369,846$ human heads annotated in $224,740$ frames taken from 21 Hollywood movies [181]. In the annotations, keyframes are annotated with the bounding box for the head. The remaining frames are linearly interpolated to head position and verified manually. However, some of the interpolated frames are not correctly annotated. Upon visualization, some of the XML annotation file corresponding to the frames are found to be empty. Therefore, the empty frames were not included in the fine-tuning process. The dataset is divided into the splits of training, testing and validation sets of 216719, 1302, and 6719 frames. The training, testing and validation sets are taken from 15, 3, and 3 movies respectively.

3.9.1.2 Casablanca [144]

The Casablanca dataset is taken from the movie Casablanca . The dataset contains 1466 frames in which head is annotated with bounding boxes [144]. However, the annotations only consider the frontal face part as a human head. The annotations were corrected for scale and aspect ratio like HH. This dataset is mainly used for evaluating the models fined-tuned on HH dataset.

3.9.2 Analysis of the results

In Faster R-CNN we have followed the same implementation setting as that of original paper [143]. However, few changes were made while fine-tuning the network from ImageNet trained models. The images are rescaled to 600 pixels on the shorter-side of dimension. The anchor boxes of 3 different scale and aspect ratio used to capture various scales in the process of region proposals. During training, the cross-boundary anchors are causing the network to converge therefore ignored, while in testing phase the cross-boundary anchors are clipped to the image boundary. A

Non-Maxima Suppression (NMS) approach is applied to reduce overlapping proposals based on their IoU intersection with ground-truth bounding boxes. To evaluate the detection performance, we have adapted the average precision (AP) a standard metric calculated from the area under the Precision-Recall (PR) curve [51]. The mean Average Precision (mAP) is the mean over classes for the set of detections. In true positive detections, there is more than 70% overlap calculated as Intersection over Union (IoU) between the detected class and their corresponding ground-truth bounding box as shown in (3.14) [51]. The remaining detections are considered as false negative. Similarly, the detections having no overlap with any corresponding ground-truth box are false negative or background.

$$IoU = area\left(B_{pred} \cap B_{gt}\right) / area\left(B_{pred} \cup B_{gt}\right) \tag{3.14}$$

Where B_{pred} and B_{gt} denotes predicted bounding box and ground truth bounding box respectively. In the first set of experiments using Faster R-CNN [143], we have used three architectures ZF [208], VGG16 [164], and VGG_CNN_1024 [33]. In the process of fine-tuning, the networks were trained for $100k$ iterations. The snapshot of fine-tuned models is saved at an interval of $10k$ iteration. The performance analysis of each of these saved models was tested on test dataset as shown in the Fig. 3.8, 3.9, and 3.10 for Faster R-CNN [143], SSD MultiBox [101], and YOLO [140] respectively. Faster R-CNN [143] with VGG16 [164] and VGG_CNN_1024 [33] converge at $70k$ iteration while ZF [208] converge at $100k$ as reported in Table 3.7. These models having high mAP's are considered best models later on used for evaluation on Casablanca dataset. Furthermore, in experiments with Single Shot Detectors (SSD), YOLO [140] has been fine-tuned for HH dataset according to the original data splits [181]. The performance of YOLO is worst amongst all the detectors with mAP of 0.63 which is lower by 13.34% than the current baseline of mAP 0.72 as reported in the Table 3.7. The SSD MultiBox [101] has also been used for experiments with VGG16 [164] as its base network architecture. The performance of SSD MultiBox is comparable with the performance of Faster R-CNN [143] as reported in Table 3.7. SSD MultiBox with $300X300$ input archives 0.788 mAP outperforming the current best approach by 8.4% as reported in Table 3.9. Similarly, Faster R-CNN archives 0.791 mAP on HH dataset outperforming the baseline approach by 8.8% as shown in Table 3.9. The best-fine-tuned models for HH dataset are used to evaluate on Casablanca dataset without fine-tuning on Casablanca. The results are reported in Table 3.8 shows the superior performance and generalization of Faster R-CNN for the unknown dataset. In Fig. 3.11 some detection samples are taken from a test set of HH and Casablanca datasets for each of the network architecture used in the experiments.

3.10 Summary

In the first half, we have evaluated different texture features for crowd density estimation. We also proposed two new feature combinations for crowd density. It is shown by experiments that proposed combination features gives significant improvement over the other methods. In the second half, state-of-the-art object detectors have been used for people detection. The detectors were

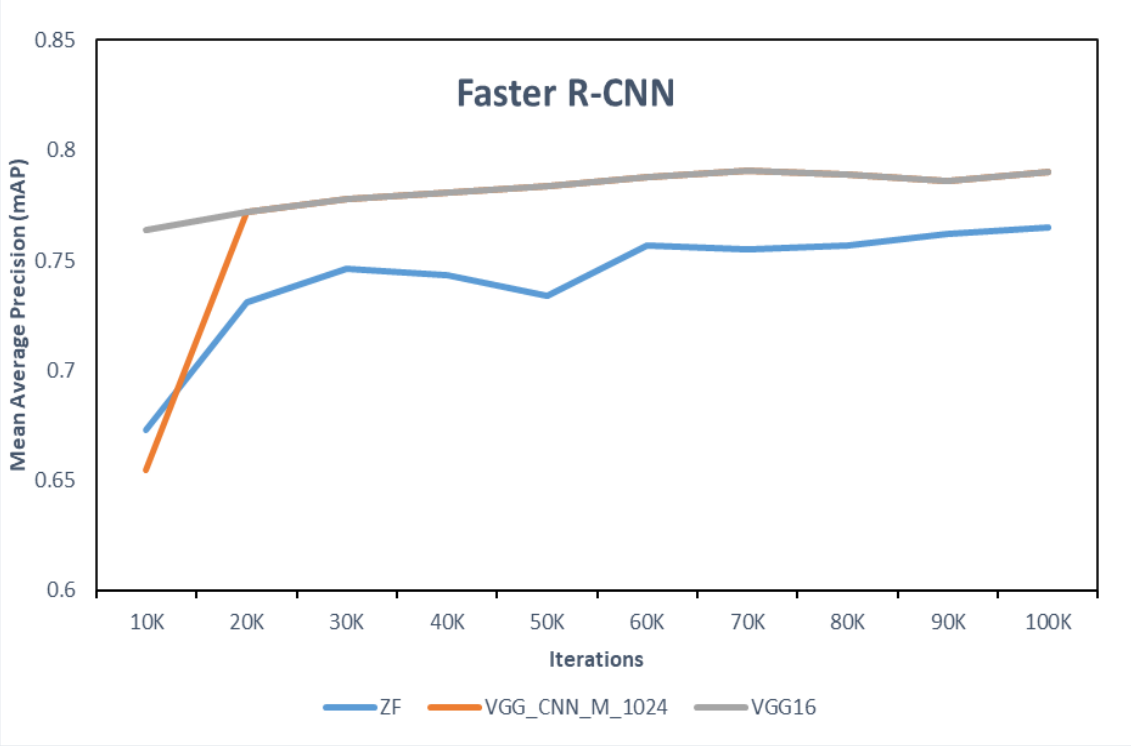


Figure 3.8: Performance of network architectures at every $10k$ iteration

Table 3.7: Testing error on HollywoodHeads Dataset

Detection frameworks	**CNN architectures**	**Iteration**	**mAP**
Faster R-CNN [143]	ZF [208]	$100k$	0.765
	VGG16 [164]	$70k$	0.791
	VGG_CNN_M_1024 [33]	$70k$	0.791
YOLO v2 [140]	13-layered architecture	$46k$	0.631
SSD MultiBox [101]	VGG16 [164]	$40k$	0.788

Table 3.8: Selected best models fine-tuned on HollywoodHeads(HH) dataset performance on Casablanca dataset

Detection frameworks	**CNN architectures**	**mAP**
Faster R-CNN [143]	ZF [208]	0.486
	VGG16 [164]	0.556
	VGG_CNN_M_1024 [33]	0.513
YOLO v2 [140]	13-layered architecture	0.518
SSD MultiBox [101]	VGG16 [164]	0.52

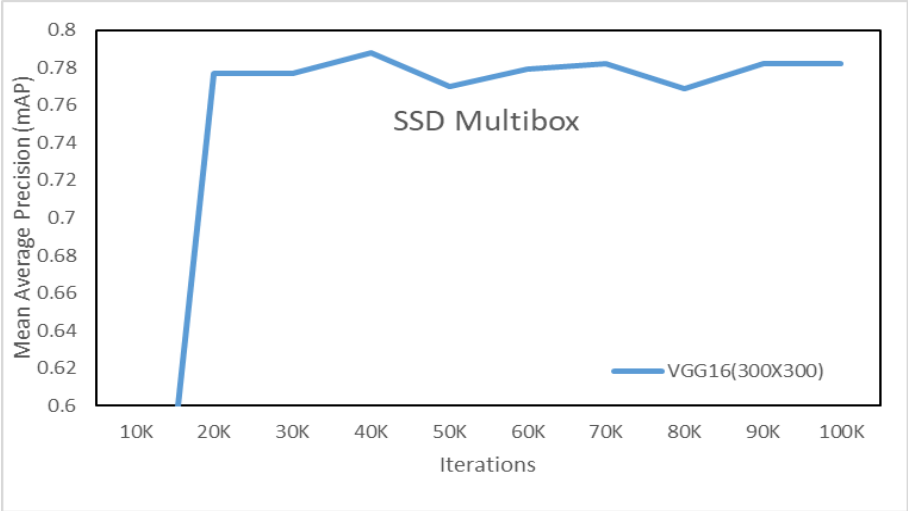


Figure 3.9: Performance of network architecture at every $10k$ iteration

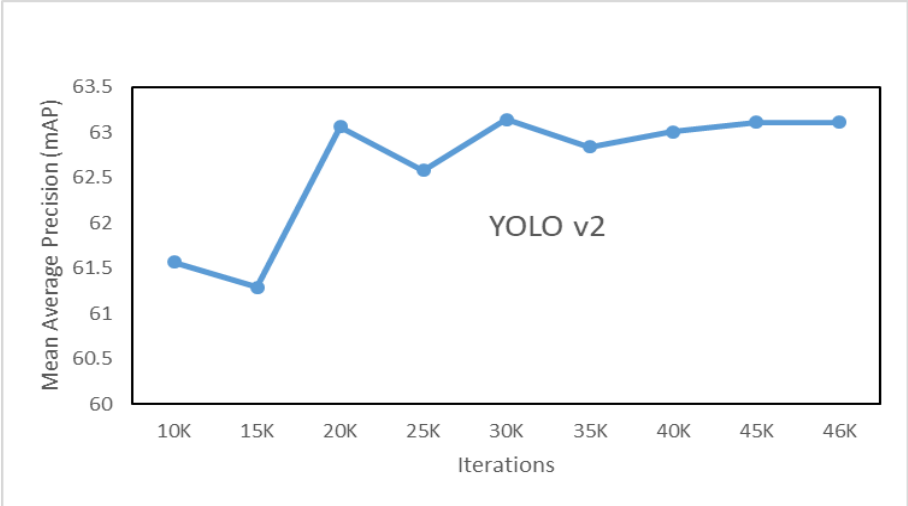


Figure 3.10: Performance of network architecture at every $10k$ iteration

trained and evaluated on contextual head detection task. It is demonstrated through experiments that Faster R-CNN [143] with VGG16 [164] works very well for head detection as compared to the single shot detectors. However, all of the object detectors perform poorly for the small-scale and high-dense head detections. Moreover, it is observed that region-based CNN are more suitable for situations where high accuracy is required. While single shot detectors are fast although less accurate, thus more suitable for real-time and mobile applications.

Table 3.9: Comparative analysis with other approaches

Detection frameworks	**CNN architectures**	**mAP**
DPM Face [123]		0.374
R-CNN [61]		0.671
Local model [181]		0.718
Local+Global+Pairwise model [181]		0.727
Faster R-CNN [143]	ZF [208]	0.765
	VGG16 [164]	0.791
	VGG_CNN_M_1024 [33]	0.791
YOLO v2 [140]	13-layered architecture	0.631
SSD MultiBox [101]	VGG16 [164]	0.788

Figure 3.11: Qualitative Results: The first column shows the sample frames from test sequences overlaid with the ground-truth annotations. The second, third and fourth column shows the detection results using models fine-tuned on SSD MultiBox [101], YOLO [140] and Faster R-CNN [143] respectively. The rows from $1-4$ and $5-7$ show the results from Hollywood-Heads [181] and Casablanca [144] respectively. (Best viewed in colour)

"A person who never made a mistake never tried anything new"

- Albert Einstein

4

Crowd Counting and Density Estimation : Part II

Crowd counting and density estimation is an important and challenging problem in the visual analysis of the crowd. Most of the existing approaches use regression on density maps for the crowd count from a single image. However, these methods cannot localize individual pedestrian and therefore cannot estimate the actual distribution of pedestrians in the environment. On the other hand, detection-based methods detect and localize pedestrians in the scene, but the performance of these methods degrade when applied in high-density situations. To overcome the limitations of pedestrian detectors, we proposed a Motion Guided Filter (MGF) that exploits spatial and temporal information between consecutive frames of the video to recover missed detections. Our framework is based on Deep Convolution Neural Network (DCNN) for crowd counting in the low-to-medium density videos. We employ various state-of-the-art network architectures namely Visual Geometry Group (VGG16), Zeiler and Fergus (ZF) and VGGM in the framework of a region-based DCNN for detecting pedestrians. After pedestrian detection, the proposed motion guided filter is employed. We evaluate the performance of our approach on three publicly available datasets. The experimental results demonstrate the effectiveness of our approach which significantly improves the performance of state-of-the-art detectors.

Major parts of this chapter have been accepted for publication in the paper titled *Crowd Counting in Low-Resolution Crowded Scenes Using Region-Based Deep Convolutional Neural Networks*, Saqib *et al*. (accepted in IEEE Access).

4.1 Introduction and motivation

The crowd scene understanding is an important and challenging problem in computer vision. The phenomenon of crowd is commonly observed in sports, festivals, social, political and religious gatherings which tends to attract and gather a huge number of people in a constrained environment. Such mass gatherings pose serious challenges to crowd safety and raise security concerns for the participant as well as organisers. Therefore, crowd analysis is one of most important and challenging task in video surveillance due to complex behaviour of pedestrians. Crowd analysis can be used for detecting critical crowd levels, detecting and counting of people and also for detecting anomalies in crowded scenes. Moreover, it can be used for tracking individuals or group of people in crowds. Among these applications, estimating the number of people from a single image becomes extremely important for crowd control and crowd safety. In public gatherings, it is important to know the number of people attending the event which can provide useful piece of information for future event planning and public space design.

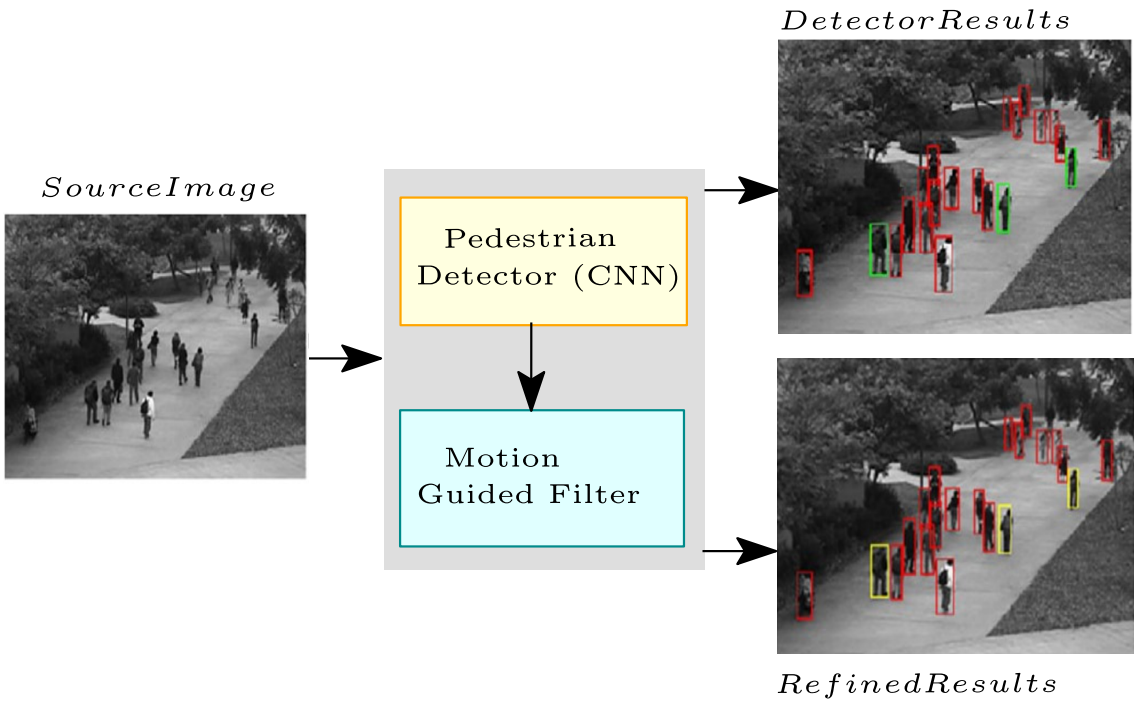


Figure 4.1: The performance of a detector is improved by employing our approach as depicted in the Fig 5.1a. The left image is the input sample frame. The upper right image shows the pedestrians in red bounding box are detected by the detector while the pedestrians in green bounding boxes are the missed detection. The lower right image shows that the missed pedestrians are recovered by our proposed approach and highlighted in yellow color. (Best viewed in colour)

Moreover, estimating crowd density or estimating the number of people attending the event can substantially reduce the cost by deploying an exact number of security personnel required for public safety and security. Various methods for estimating the crowd count are proposed in literature. Generally, we can classify these methods into two major categories, 1) *Regression-based methods*, 2) *Detection-based methods*.

4.1.1 Regression-based methods

These employ machine learning techniques like Support Vector Regressor [188], Gaussian Process Regression(GPR) [25], linear regression [45], K-Nearest Neighbor [210], and neural network [121] are employed to estimate the crowd count by performing regression between the image features and crowd size. Regression-based crowd counting algorithms can be further categorised into two groups: *Holistic* and *Local*.

In *holistic approaches*, image features like size, shape, edges, keypoints, and texture are extracted from the entire image and regression is then applied to estimate the size of crowd. In [25], edge and texture features are extracted from the whole image, and the correspondence between the number of people and features is learned through Gaussian Process Regression. [112] extracts shape, color, size, texture features from the image and self-organizing neural network is employed for crowd density estimation. The neural network is employed by [69] to learn the correspondence between the foreground pixels extracted from the whole image and number of people. A method is proposed by [199] that transforms an image into multiple scales using wavelet transform and then the first and second order features are extracted a as density character vector. A Support Vector Machine classifier is trained that classify density character into different density levels. In [88], edge and blob size histogram features are extracted and neural network is trained to find the relationship between the number of people and extracted features. [83] proposed crowd flow segmentation as the first step and then applying counting framework to count the number of people in each flow segment.

In *local approaches*, image features are extracted from the local patches of image and regression is applied locally to each patch of image and estimate the number of people in each patch. In this case, crowd count is the direct sum of these local estimates. The size and shape features are extracted from the local patches of the image [85] and linear (cylinder model) is employed to estimate crowd count. [94] proposed pixel based density function for counting problem. The density function is a mapping between the feature vector associated with every single pixel value and its ground-truth density value. The ground-truth density value of the pixel is approximated by fitting the normalized Gaussian kernel to the pixel dotted annotation. The multi-output regression model is proposed by [37] for crowd counting by extracting size, shape, edge and texture features from the local regions of the image. Their proposed regression model is able to estimate people count in spatially localized regions. [43] extracts SURF features from the local region of the image and support vector regression are employed to learn the correspondence between the features and the count of people. [76] proposed a counting framework based on multi-source multi-scale approach, which used multiple features extracted from the local regions of an image. These features are taken from different sources like HOG, Local Binary Pattern (LBP) and Fourier analysis. These features are computed at different scales for accurate and reliable counting. This work is extended by [12] which added more features like wavelets, SIFT, and GLCM. [114] proposed a local Histogram-of-Orientation Gradient, in contrast to the standard Histogram-of-Orientation-Gradient, used to describe the parts of the person independently and therefore helps in extraction of features even in

partial occlusions.

A great deal of work in [24, 31, 150, 9] for crowd counting and density estimation has focused on local features such as edges and blobs extracted from the foreground. Typically, regression techniques such as Ridge Regression (RR) [36], Bayesian Poison Regression (BPR) [31], Gaussian Process Regression (GPR) [24] are used to learn the model between the local features and count. However, a significant amount of information is lost in the calculation of such features. The accuracy and performance of such features heavily rely on the segmentation of foreground. The foreground segmentation is a challenging problem especially because of varying lighting conditions and shadowing effect [176]. [153] considered texture features that are directly related to the crowd density and counting. The more texture means high crowd density which is not always true because of the incorrect foreground segmentation. Foreground segmentation of crowd only caters for moving crowd, but it performs poorly in the case of a static or very slow moving crowd. Furthermore, these features cannot be used for contextual crowd scene understanding. [56] proposed the simplified version of the previous work by estimating the object density using regression random forest improving the training accuracy. Similarly, [10] proposed an interactive and iterative density estimation technique. In this technique, user annotates the object with the dot for object and line segment for its diameter to estimate the density. The low-level features extracted are mapped to the density value. The learned mapping can be visualized intuitively by the user for error. The error indicates the need for further annotations to refine results in the next iteration.

The performance of regression-based methods are improved further by employing Convolution Neural Networks (CNN) [19, 155, 82, 214, 122, 133, 182]. In these methods, density maps are generated from the image patches, where count for each patch is obtained by performing the integration over the density map. Zang *et al.* [209] proposed a CNN model which can generate both crowd count and density maps using switchable alternative learning for counting and density map. The training and testing require a perspective map for perspective normalisation which might not be available in practice. A Multi-column Convolutional Neural Network (MCNN) is proposed in [214], which utilizes three columns with filter size of a different receptive field is used to compensate for perspective distortion. MCNN is trained to estimate crowd density at only three different scales in extremely crowded still images. Zhang *et al.* and Boominathan *et al.* [214][19] proposed multicolumn CNN approaches, in which different columns with different filter sizes are used to capture multiple scales variation along with perspective. The final prediction obtained from columns is averaged to get a density map. Finally, the integral of density map gives crowd count. Similarly, Switch-CNN [155] proposed switching architecture which intelligently switches appropriate regressor for particular crowd patch based on variation in density within the single image. However, these approaches are highly scene specific and may perform poorly on cross scene analysis. [182] used shallow CNN architecture in the framework of ensemble learning where new models are added to the ensemble to fix the error from the previous model. These ensembles were used to estimate the density map. The density map is then spatially integrated to count. However, the research did not clearly explain the stopping criteria for adding new models to the ensemble. Therefore, ultimately adding new models to the ensemble might end up in

overfitting. [82] used contextual information such as perspective weights, camera tilt angle and camera height to estimate crowd density. The contextual information is an auxiliary input to the Filter Manifold Network (FMN) to produce filter weights for the convolutional layer according to the scene context. Thus convolutional layer adapts itself to the context of the scene in contrast to the fine-tuning and transfer learning. The current datasets do not provide contextual information, therefore, comparison with current datasets is not possible. The convolution is made adaptive to only parameters related to perspective. There might be more complex parameters related to the scene which are ignored.

Most recently, Shen *et al.* [161] proposed Adversarial Cross-Scale Consistency Pursuit (ACSCP) approach using adversarial loss instead of traditional Euclidean loss to mitigate the blurry effect due to l_2 regularisation in the generation of density maps from crowd patches. Moreover, a new regulariser is proposed to enforce the scale consistency such that the number of crowd count in the large patch is coherent with the sum of crowd counts in the corresponding smaller non-overlapping patches. Similarly, Cao *et al.* [22] proposed Scale Aggregation Network (SANet) for accurate and efficient high-resolution of density maps using new training loss called local pattern loss. Sindagi and Patel [165] proposed Contextual Pyramid CNN (CP-CNN) in which local and global contextual information is incorporated with Density Map Estimator (DME) to generate high-quality density maps. Finally, all the maps are fused to estimate the crowd count and density. Crowd counting is formulated as a semantic scene model [74]. The pedestrian, head, and their context are three key factors, that are considered as a composite body-part semantic structure for two types of scene semantic models. These models are turned into different sub-tasks to train deep CNN for counting and scene semantic analysis. Xiong *et al.* [201] proposed Convolutional LSTM (convLSTM) to exploit temporal correlation along with spatial dependencies to boost the count accuracy in a complex scene. However, most of the datasets of the high-density crowd are still images of the crowd and therefore do not carry temporal information.

Regression-based methods work well in high-density situations since they can capture generalized density information but suffer from following limitations. 1) The performance of these methods degrades when applied to low-density situations due to overestimating the count. 2) These methods cannot localize pedestrian in the scene and thus provide no information about the distribution of pedestrians in the environment which is sometimes very crucial for the crowd managers and security personnel.

4.1.2 Detection-based methods

On the other hand, *detection-based methods* [44, 54, 93, 180, 59, 215], train object detectors to localize the position of each person, where crowd count is the number of detections in the scene. Detection based methods can be further divided into two categories, 1) hand-crafted feature based models[180, 44, 47, 48, 54, 186, 212] and 2) deep features models [107, 187, 134, 135, 95, 211, 75, 117, 72, 213, 173]. In the first category, hand-crafted features like edges [27, 29], texture [199, 156, 27], and shape [27] are extracted from image to train SVM or boosting classifiers.

After training, learned weights of the classifier are considered as a template for the entire human body. These hand-crafted features have low representation of human body and performance of classifier degrades when applied to complex crowded scenes. In order to model complex poses of pedestrians, DPM [54, 220, 100] learn mixture of local templates for each body part. Although DPM is robust to complex poses but feature representation and classifier cannot be jointly optimized to improve performance.

Recent advancement in deep learning has shown outstanding results in detection using CNN. In a typical CNN based approach, there is local connectivity of a region in the input image to the output image as compared to the traditional feedforward neural network. In a feedforward neural network, every input layer is fully connected with the output layer. Deep CNN is a compositional model, in which features are extracted ranging from low-level along the pipeline of CNN towards the final layers. The lower layers represent low-level features such as edges, and the subsequent layers represent abstract features such as shapes, etc. We have used Faster R-CNN for the detection of pedestrians in the low-to-medium density crowd videos. In Faster-RCNN [143], a small network namely Region Proposal Network (RPN) is used on top of the feature map to extract object candidates or region proposals in contrast to other approaches like Selective Search [174], CPMC [23], MCG [8], Edge boxes [221], etc. The advantage of RPN make Faster R-CNN an end-to-end pipeline for the detection and also does not add to the computation of the network. To detect objects at multiple scales, region proposals at various scale and aspect ratios are extracted using anchor boxes. The center of the anchor box is having a different aspect ratio and size and coincide with the center of the sliding window.

The models mentioned above achieved a considerable improvement in object detection in particular and pedestrian detection in general when applied to static images. However, the performance of a detector on videos is limited due to the following reasons; 1) Most CNNs are trained on fixed training sets and cannot incorporate the un-constrained video environments which may have different illuminations, backgrounds, and viewpoints. Also, it is very expensive to label and retrain the detector for every new video. 2) CNNs are designed to learn features from raw image pixels and cannot able to leverage the information existed in different video frames of the video such as consistent brightness and color pattern of the object. In this study, we propose an approach to estimate crowd count by improving the detection performance of a detector when applied to videos. We first apply state-of-the-art object detection techniques to detect people in low-density crowds and then the performance of a detector is improved by leveraging the spatio-temporal information between the frames of video as shown in Fig. 5.1a. In general, our method takes predicted detection of a detector as input and generates refined detections as output with higher average precision.

4.2 Proposed methodology

We proposed a framework for crowd counting using state-of-the-art deep CNN as shown in Fig. 4.2. According to the framework, input frames are given to the Region-based CNN. We have

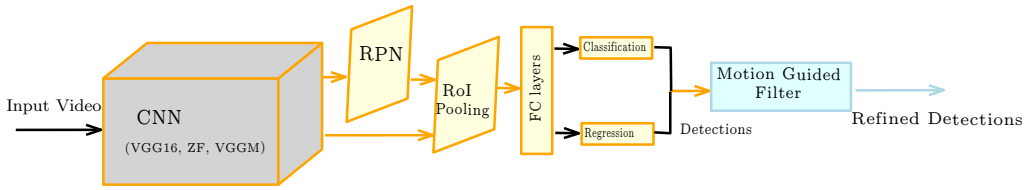


Figure 4.2: Proposed framework

used Faster R-CNN [143] with Caffe [80] for the detection of pedestrians. The datasets that we have used contain few numbers of frames which are not sufficient enough for training deep CNN. Therefore, we have used transfer learning from ImageNet [46] to fine-tune our models. These fine-tuned models are used in the testing phase to test on unseen frames. We have used various network architectures such as ZF [208], VGG16 [164], and VGGM [164] to train the system and evaluate the performance on the test dataset. ZF is a 8 layered architecture containing 5 convolutional layers and 3 fully-connected layers. Similarly, VGG16 is a 16 layered architecture that has 13 convolutional layers and 3 fully connected layers.

In Fast R-CNN [60] the order of the extracting region of proposals and running the CNN is exchanged as compared to RCNN [62] architecture. In this architecture whole image is passed once through the CNN and the regions are now extracted from convolutional feature map using ROI pooling. This change in architecture reduces the computation time by sharing the computation of convolutional feature map between region proposals. The region proposal is projected to the corresponding spatial part of convolutional feature volume. Finally, the fully connected layer expects the fixed-size feature vector, and therefore the projected region is divided into a grid and Spatial Pyramid Pooling (SPP) is performed to get fixed-size vector. SPP deals with the variable window size of pooling operation and thus end-to-end training of the network is very hard. The generation of the region proposals is the bottleneck at the test time. In the above-mentioned approaches, CNN was used only for regression and classification. The idea was further extended to use CNN also for region proposals. The latest offspring from the RCNN family, the Faster R-CNN [143] proposed the idea of a small CNN network called Region Proposal Network (RPN), build on top of the convolutional feature map. RPN is two-layered network which does not add to the computation of overall network. A sliding window is placed over a feature map in reference to the original image. The notion of anchor box is used to capture object at multiple scales. The center of the anchor box having a different aspect ratio and size coincide with the center of the sliding window. RPN generates region proposals of different sizes and aspect ratios at various spatial locations. Finally, regression provides finer localization with reference to the sliding window position.The complete architecture is shown in Fig. 4.2.

4.2.1 Motion Guided Filter

Convolution neural networks like SSD [101], YOLO [141], and Squeezedet [192] showed a significant improvement in domain of real-time object detection using a single image. However, the

performance of these networks can be improved further by leveraging the temporal information available in real-time videos. Leveraging temporal information in object detection is not a trivial problem. Usually, end-to-end learning is a sophisticated way of solving computer vision problems, but in the case of videos, this approach cannot be applied. Feeding multiple frames to the CNN is not possible due to the limitation of memory. Therefore, as a solution, we propose a Motion Guided Filter that recovers the missed detection in the frames by using the flow estimation. It is observed that pedestrian detected in the first frame travels a few pixels in the next frame. For estimating the displacement, we utilize an energy function which is based on three assumptions: brightness constancy assumption, gradient constancy, and spatio-temporal smoothness constraint.

4.2.1.1 Brightness constancy

For estimating the displacement, it is assumed that the gray value of a pixel does not change [68]

$$\Omega(x, y, t) = \Omega(x + u, y + v, t + 1) \tag{4.1}$$

$\Omega : \lambda \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ denotes bounding box sequence, and $\mathbf{w} := (u, v, 1)$ is the displacement vector between an image at time t and another image at time $t + 1$. Here it is to be noted that bounding box Ω is a 4-D vector. For estimating the displacement, we use only the spatial coordinates of pixels while we assume the size of the bounding box (width and height) is the same. Therefore we omit the size of the bounding box in the equations.

4.2.1.2 Gradient constancy

It is also assumed that the gray value of the pixel does not change instantaneously. However, this assumption is weak since a slight change in the environment or change in illumination may change gray values of the image, therefore, in this case, we allow some small variations and determine the displacement vector by a criterion that is invariant to gray value changes. We, therefore, use the gradient of gray value instead of considering the gray values directly. We then assumed that gradient of gray value does not vary due to the displacement [177] and is given by

$$\nabla\Omega(x, y, t) = \nabla\Omega(x + u, y + v, t + 1) \tag{4.2}$$

where ∇ is the gradient. Equation (4.2) deals with translatory motion while (4.1) is best suited for complicated motions.

4.2.1.3 Spatio-temporal smoothness

Up till now, the model estimates the displacement of one pixel from one frame to another without taking into account neighboring pixels. Therefore, the model runs into problems as soon as the gradient disappears somewhere. Furthermore, we also expect some outliers in the estimates.

Therefore it is very useful to use smoothness assumption. This constraint can be either applied only in the spatial domain if we want to compute flow between two images or to the spatio-temporal domain, if the displacement in the whole sequence of images is needed. Here, since we are recovering the bounding box of the next frame from the current frame, therefore we use only spatial smoothness constraint.

With this discussion, we now derive an energy function that will penalize deviations from these aforementioned assumptions. Let $\mathbf{x} := (x, y, t)$ is the pixel of frame at t and $\mathbf{w} := (u, v, 1)$ is its displacement vector. Then deviations from the grey value constancy and gradient constancy are measured by the following energy function

$$E_d = \int_{\lambda} (\mid \Omega(\mathbf{x} + \mathbf{w}) - \Omega(\mathbf{x}) \mid^2 + \gamma \mid \nabla\Omega(\mathbf{x} + \mathbf{w}) - \nabla\Omega(\mathbf{x}) \mid^2) d\mathbf{x} \tag{4.3}$$

where γ is a balancing parameter between brightness and gradient constancies.

Finally, we write the smoothness term which penalizes the total variations in the flow field [148] and can be expressed as

$$E_s = \int_{\lambda} (\mid \nabla u \mid^2 + \mid \nabla v \mid^2) d\mathbf{x} \tag{4.4}$$

The total energy function is the weighted sum of the above two equations and is given by

$$E(u, v) = E_d + \alpha E_s \tag{4.5}$$

with some regularization parameter the value of $\alpha > 0$. The goal is to find displacement vector (u, v) that minimizes the energy function given by (4.5)

For every pixel $\mathbf{x_t} \in \Omega_t$ in a frame at t, we compute its corresponding pixel in a frame at $t+1$ by using the following equation

$$\mathbf{x_{t+1}} = \mathbf{x_t} + \mathbf{w} \tag{4.6}$$

4.2.1.4 Detection refinement

After detecting pedestrians in each frame of the analyzed video sequence, we then leverage temporal information across the multiple frames to further refine the detection results. For this purpose, we use (4.6) for low-level tracking to establish temporal correspondence across multiple frames. Exploiting temporal information can suppress false alarms generated due to the noise and other random distortions. We integrate temporal information across the frames to re-score detections and suppress the false positives. Let $\Omega_t = \{\omega_1, \omega_2, \ldots, \omega_n\}$ represents a set of n bounding boxes (or detections) in a frame at t. We then represent $D = \{\Omega_1, \Omega_1, \ldots, \Omega_N\}$ as a container of all sets of bounding boxes for a video sequence containing N number of frames. In order to refine Ω_t

for the current frame at t, we employ a matching hypothesis based on overlap area between the current bounding box $\omega_i \in \Omega_t$ and $\omega_j \in \Omega_{t+1}$ in the subsequent frame. We propose a refinement Algorithm 3 which takes Ω_t as input and gives the corresponding refined Ω_R as an output. Given a set of bounding boxes Ω_t in the current frame, we define a temporal window of the size W. For each bounding box $\omega_i \in \Omega_t$ in a frame at t, we first predict its location in the next frame at $t+1$ by computing displacement vector $\mathbf{w}$ as in (4.6). We then compute Δ between ω_i and set of bounding boxes Ω_{t+1} in the next frame at $t+1$ and select the best match (maximum value of Δ). We compute Δ between two detections ω_i and ω_j as Intersection over Union and formulated as $\frac{\omega_i \cap \omega_j}{\omega_i \cup \omega_j}$. Final confidence score σ is computed for each ω_i by accumulating confidence score over temporal window W, as in line 8 of the Algorithm 3. We then delete the bounding box for which confidence score σ is less than ϵ. We set the value of ϵ = 0.5 in all our experiments. We refine the container D in the same way. Let $R = \left\{\Omega_1{}', \Omega_2{}', \ldots, \Omega_N{}'\right\}$ is a container of refined sets of bounding boxes for a video sequence containing N frames. The bounding boxes obtained after this step are refined and trusted detections.

In some cases, a given set of bounding boxes Ω_t may not contain a detection for a particular person due to occlusion, or missed detection, etc. In order to address this issue, we integrate temporal information by reliably tracking pedestrian through time and use it to find the missed detection. Our tracking approach operates in two modes: 1) *detection mode*, 2) *Low-level tracking mode*. We initialize a tracker for each detection in a frame at t. Whenever the tracker finds and matches a detection in the next frame at $t+1$, it follows the detection mode. This mode enables tracking more robust to variations in scale, appearances and pose. If for some reasons, tracker cannot find detection in the next frame, the tracker relies on low-level tracking. In low-level tracking mode, the tracker estimates the displacement vector and predicts the next location by using (4.5) and (4.6). It is to be noted that we are not interested in long-range tracking, instead our goals is to use low-level tracking to fill in the gap by recovering the missed detection. Let $\{\mathbf{x_t}, \mathbf{s_t}\}$ be the position and size of a pedestrian being tracked. Let $\ddot{x}_t$ and $\ddot{s}_t$ are the observations of x_t and s_t, with Gaussian noises of co-variance R_x and R_s. For each track in a frame at t, we have predictions $\left\{\hat{x}_{t|t-1}, \hat{s}_{t|t-1}\right\}$ and we search for pedestrian detection around position $\hat{x}_{t|t-1}$ and size $\hat{s}_{t|t-1}$. For any pedestrian detection P will be assigned to a track if $\parallel \hat{x}_{t|t-1} - x_f \parallel < \alpha$ and $\parallel \hat{s}_{t|t-1} - s_f \parallel < \alpha$, where x_f is the position and s_f is the size of P. We set α = 0.3 in our experiments. We adopt a greedy strategy of data association and score each track by N_d, where N_d represents the number of detections it has matched. During the detection mode, we maintain a pedestrian template $P_{template} = I(x_t, s_t)$ at location x_t and of size s_t. We use normalized correlation to search for best match in the image. In case, a tracker can not find and match a detection, then tracker switch to low-level tracking mode and continue tracking. In this case we update the template linearly as in (4.7)

$$P_{template} = (1 - \beta_{update} P_{template} + \beta_{update} I(x_t, s_t)) \tag{4.7}$$

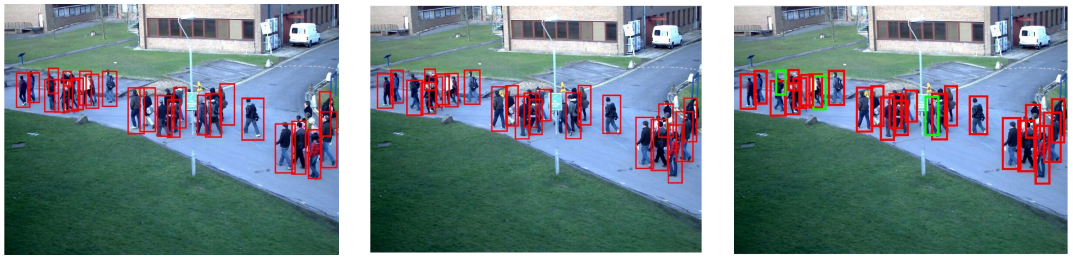
Where β_{update} is set to 0.1 in our experiments. For every track, we keep track of N_d and N_{llt},

Algorithm 2 : Refinement of Detection Results

Input: Sets of Bounding Boxes Ω_t
Output: Refined Bounding Boxes Ω_R

function REFINEMENT(Ω_t)
  $T = \{\Omega_{t+1}, \Omega_{t+2}, \ldots, \Omega_{t+W}\}$
  Initialize evidence accumulator σ to zero
  for all bounding box ω_i in Ω_t **do**
    for all Ω_j in T **do**
      Compute displacement vector **w** using (4.5)
      Predict next location $\omega_i^{'}$ as ω_i + **w**
      $\sigma = \sigma + \arg\max_{j \in T} \Delta(\omega_i^{'}, \Omega_j)$
      Update ω_i as $\omega_i \longleftarrow \omega_i^{'}$
    end for
    if $\frac{1}{W}\sum_1^W \sigma > \varepsilon$ **then**
      Insert ω_i in tail of Ω_R
    end if
  end for

  return Ω_R
end function



(a) (b) (c)

Figure 4.3: (a) Detections in the first frame (b) Detections in the second frame (c) Recovered detection missed in second frame

where N_d is the number of step that a track follows the detection mode and N_{llt} is the number steps in which track follows low-level tracking mode. We terminate a track if N_{llt} / N_d ¿ 1.5.

4.3 Experiments

In this section, we discuss the qualitative and quantitative analysis of the results obtained from the experiments. We evaluate our approach using three publicly available datasets, PETS2009 [55], UCSD dataset [31] and Mall dataset [37]. These datasets include indoor and outdoor scenes with varying densities. Traditionally, regression-based methods are evaluated on these datasets. Therefore the available annotations are only suitable for regression-based analysis and not for

Table 4.1: Crowd datasets

Dataset	Resolution	Color	Location	Test frames	Train frames	Crowd Size
PETS2009 [55]	$768X576$	RGB	Outdoor	800	1200	8 to 26
UCSD [31]	$238X158$	Grayscale	Outdoor	800	1200	11 to 50
Mall [36]	$640X480$	RGB	Indoor	800	1200	15 to 60

detection base methods. Typically, there is a dot annotation for every person in the scene. These annotations also include perspective map used for the normalization of perspective distortion. Such dot annotations are not suitable for training a CNN model for pedestrian detection. Therefore, for the first time, we annotated each pedestrian with a bounding box that covers the whole body of the pedestrian. The complete details of the datasets are given in the Table. 4.1. The sample images along with overlaid ground-truth annotations from three datasets are shown in the Fig. 4.7 (d) (e) (f).

After annotating all video sequences, we then trained different models, i.e, *ZF* [208], *VGGM* [164] and *VGG16* [164] on Nvidia Quadro P6000 GPU with a learning rate of 0.0001 and batch size of 64. The RPN batch size is kept constant at 128 for region based proposal networks.

We then evaluate and compare the performance of our method with other reference methods. For the sake of a comprehensive evaluation, we divide the experiment setup into two phases. In the first phase, we evaluate and compare the detection/localization performance while in the second phase, we evaluate and compare the crowd counting performance.

4.3.1 Localization performance

In this section, we evaluate and compare the localization performance of different models. The purpose of evaluating localization performance is to measure how well the model localized the pedestrian in the given scene. Precise localization of pedestrians is very crucial for the crowd managers and security personnel to effectively respond to the anomalous situations.

The localization accuracy by which a model can predict the bounding box of a pedestrian is typically judged by Intersection over Union (IoU) between predicted and ground-truth. In most of the cases, IoU is used with fixed threshold value 0.5 for deciding whether a bounding box is successfully detected. However, with the fixed threshold value, one cannot overview the range of performance with varying the thresholds. Therefore, we use mean Average Precision (mAP) as an evaluation metric that averaged the performance over a wide range of IoU thresholds.

We evaluate the localization performance of these models in two ways, i.e., pre-trained and fine-tuning. In the pre-training phase, these networks are trained from scratch by using *ImageNet* dataset and then the learned models are directly used for detecting pedestrians during the testing phase. In fine-tuning case, we fine-tuned these pre-trained models by using the images from

PETS2009, UCSD and Mall datasets.

We analyzed the performance of each network architecture at a different iteration during the fine-tuning phase. During training, the snapshot of trained models are saved at the interval of $10k$ as shown in the Fig. 4.4 for Mall dataset [37]. All the network architectures were able to converge after $20k$ iterations. The best-trained model obtained at iteration $90k$ of $VGG16$ having mAP of $.701$ was used for evaluation on the testing sequence of Mall dataset [37]. Similarly, trained model based on ZF architecture for USCD [31] with high mAP of 0.783 is obtained at $80k$ iteration as shown in Fig. 4.6. The reason for high mAP can be attributed to the low-resolution of the dataset as well as the smaller filter size used in ZF architecture. Thus ZF shows stable performance throughout all the iterations. Furthermore, a $VGG16$ model with high mAP of 0.692 is obtained for PETS2009 dataset [55] as shown in Fig. 4.5.

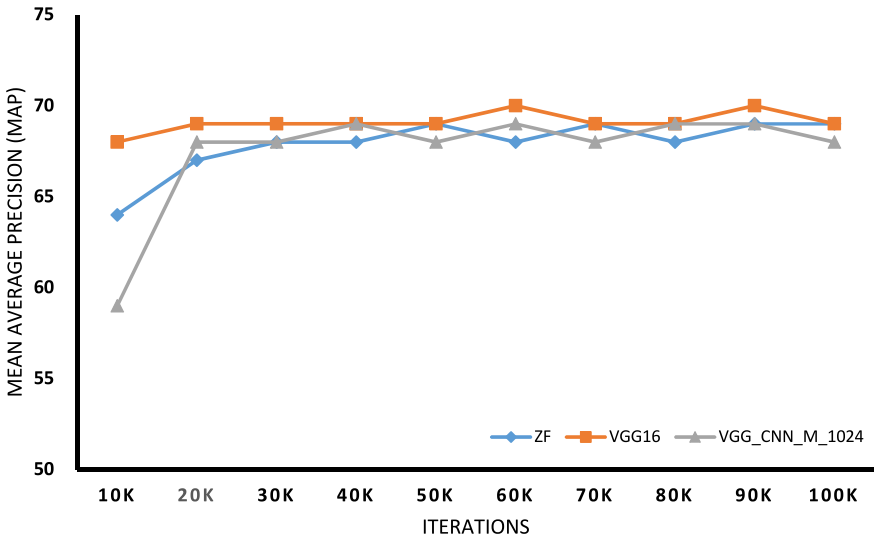


Figure 4.4: Performance at different iteration for for Mall dataset [37]

After fine tuning the models, we then employ spatio-temporal filtering approach discussed in the section, which further refines the detection by exploiting spatial and temporal information between the consecutive frames.

Table. 4.2 shows the performance of these models obtained during pre-training, fine-tuning and after employing a Motion Guided Filter. It is obvious from the table that all the base models show poor performance during the pre-trained phase. The reason for poor performance is the models were trained on ImageNet dataset and not on pedestrian datasets. However, these models are generic enough to be fine-tuned for pedestrian detection. We have used 50% samples of pedestrian datasets during fine-tuning phase. The models fairly learn the representation of pedestrians and as a result, mAP is increased up to 72% on average for all the datasets. Among the fine-tuned CNN models, VGG16 outperforms other methods. Even after fine-tuning, there is still room for improvement since the information like consistent brightness and color pattern of pedestrian existed

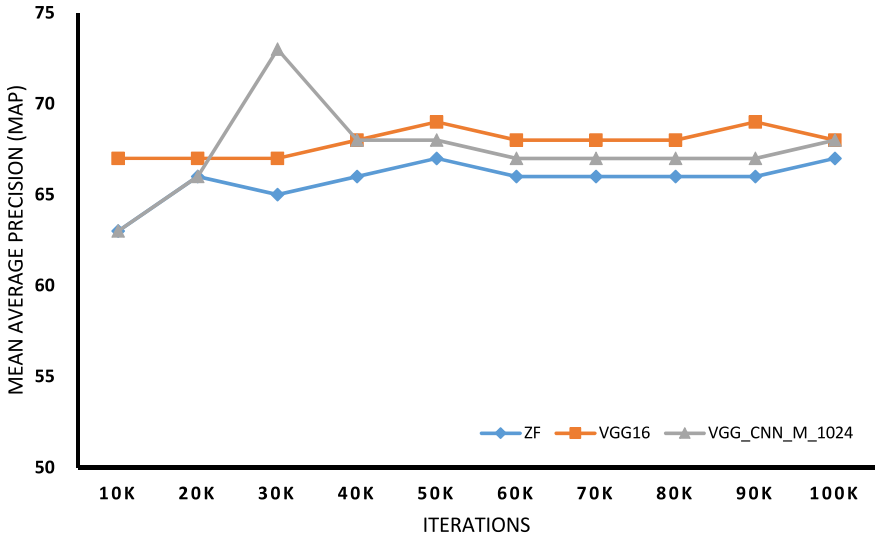


Figure 4.5: Performance at different iteration for PETS dataset [55]

Table 4.2: Localization performance at different steps

Methods	Pre-Training			Fine Tuning			Proposed		
Models	VGG16	VGGM	ZF	VGG16	VGGM	ZF	VGG16 + MGF	VGGM + MGF	ZF + MGF
UCSD	0.45	0.25	0.40	0.67	0.65	0.59	0.73	0.71	0.63
PETS2009	0.40	0.15	0.20	0.75	0.73	0.69	0.82	0.78	0.75
Mall	0.41	0.25	0.30	0.68	0.63	0.57	0.71	0.65	0.58

between the subsequent frames are not exploited. For example, the detector detects pedestrians in one frame while detector completely missed that detection in subsequent frames. Therefore, by exploiting the spatio-temporal relationship and brightness consistency constraint, our proposed Motion Guided Filter is able to recover the detection which are missed due to occlusions. As a result, our proposed methodology is able to improve the mAP for all the models.

4.3.2 Counting performance

In this section, we evaluate the performance of different crowd counting methods. In addition to CNN based methods, we used four different regression-based models, i.e Gaussian Process Regression [25], linear regression [45], K-Nearest Neighbor [210] (K=4) and neural network [121] with sigmoid activation function for crowd counting. These regression models are trained on local features, i.e., size, shape, edge and keypoints. The features are extracted from the local regions of image by first dividing the image into patches. We then apply a regression technique to each patch

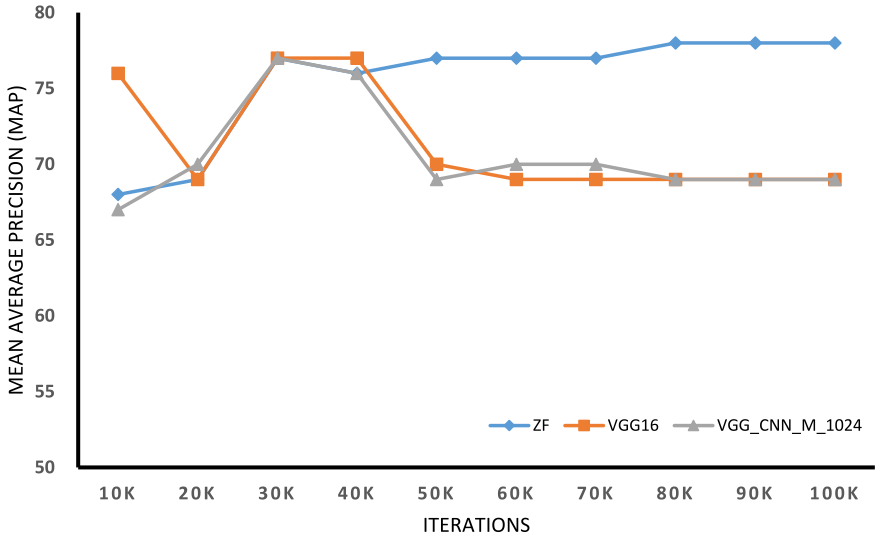


Figure 4.6: Performance at different iteration for for UCSD dataset [31]

Table 4.3: Evaluation of crowd counting methods

Methods	**Models**	**UCSD**		**PETS 2009**		**Mall**	
		MAE	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
Regression (hand-crafted features)	GPR [25]	1.46	6.23	1.78	16.97	2.58	8.86
	Linear [45]	1.56	6.48	1.77	17.75	2.58	9.65
	KNN [210]	2.72	9.63	3.00	18.69	2.89	9.23
	NN [121]	8.13	33.08	4.11	30.42	26.06	163.41
CNN	VGG16 [164]	2.89	9.25	2.67	18.53	3.52	10.25
	VGGM [164]	3.92	10.47	2.63	17.56	4.85	13.65
	ZF [208]	3.55	11.26	2.64	16.28	3.65	11.46
CNN + MGF (proposed)	VGG16+MGF	1.27	5.62	1.21	5.43	1.89	7.29
	VGGM+MGF	1.38	6.29	1.28	5.66	2.32	8.35
	ZF+MGF	1.41	6.35	1.36	6.48	2.56	8.78

of an image. The local features are extracted in the following ways

Size refers to the area of foreground object. The area of object is measured as the count of foreground pixels. In order to compensate for perspective distortions, we assign weight $W(x, y)$ to each foreground pixel as in [25] based on the relative size of reference object in the scene. The weighted area A of blob B is computed as follows

Table 4.4: Comparative analysis with other techniques on UCSD [31] dataset

Method	**MAE Test**
Density + MESA [94]	1.7
Crowd CNN Model with global regression [209]	1.6
COUNT forest [137]	1.6
CNN Model with no boosting [182]	1.63
Boosted CNN (1 boost) [182]	1.35
Boosted CNN (2 boost) [182]	1.29
Boosted CNN (3 boost) [182]	1.28
Fine-tuned (2 boost) [182]	2.01
Twice as deep [182]	1.82
Thrice as deep [182]	2.42
Ensemble of 2 CNNs [182]	1.55
Ensemble of 3 CNNs [182]	1.53
Zhang2015 [209]	1.60
MCNN [214]	1.07
Hydra-CNN	1.65
Switching CNN [155]	1.62
ConvLSTM [201]	1.30
BSAD [74]	1.0
ACSCP [161]	1.04
SANet [22]	1.02
Proposed Method (VGG16 + MGF)	**1.27**

$$A = \sum_{(x,y)\in B} W(x,y)$$

Shape is computed by measuring the orientation of perimeter pixels. Perimeter pixels contain important and useful information about the shape of the object. For computing shape feature, we generate a histogram of orientations with four bins. Each bin corresponds to the orientations of pixels. The four bins correspond to four shape features and denoted by S(h), where h $\in [1, 4]$.

Edge is computed by taking the histogram of edge pixels of the foreground object. We divide edge orientation histogram into six bins over the range of $[0, 180°]$. In this case, for perspective

Table 4.5: Comparative analysis with other techniques on Mall [36] dataset

Method	**MAE Test**
CA-RR [35]	3.43
COUNT forsest [137]	2.50
CNN Model with no boosting [182]	9.54
Boosted CNN (1 boost) [182]	2.43
Boosted CNN (2 boost) [182]	2.08
Boosted CNN (3 boost) [182]	2.13
Fine-tuned (2 boost) [182]	2.01
Twice as deep [182]	10.41
Thrice as deep [182]	15.37
Ensemble of 2 CNNs [182]	6.52
Ensemble of 3 CNNs [182]	6.57
ConvLSTM-nt [201]	2.53
ConvLSTM [201]	2.24
Bidirectional LSTM [201]	2.10
Proposed Method (VGG16 + MGF)	**1.89**

Table 4.6: Comparative analysis with other techniques on PETS [55] dataset

Method	**MAE Test**
Shape+Edges+Keypoints [154]	1.77
Proposed Method(VGG16 + MGF)	**1.21**

normalization, each edge pixel assigns a weighted vote of $\sqrt{W(x,y)}$ to a corresponding histogram bin h as follows.

$$E(h) = \sum_{(x,y)\in K} \begin{cases} \sqrt{W(x,y)}, & \text{if } \theta_h \leq \theta_{x,y} \leq \theta_{h+1} \\ 0 & \text{otherwise} \end{cases}$$

where K is set of edge pixels of a blob of the foreground object and $\theta_{x,y}$ is the orientation of edge pixel. Upper and lower bound of bin h is represented by θ_h and θ_{h+1} respectively.

Interest points refers to keypoints in the scene and provide useful information about the human crowding. We extract two types of features, i.e. FAST and SURF from the blob and denoted as P_F and P_S respectively and computed as follows.

$$P_F = \sum_{(x,y)\in M} \sqrt{W(x,y)}$$

$$P_S = \sum_{(x,y)\in N} \sqrt{W(x,y)}$$

where M represents $FAST$ features extracted from the blob, and N represents $SURF$ features extracted from foreground blobs. In this case, we also assign weights W to each interest point to compensate for perspective distortions.

We train four different regression models using these local features and the results of regression and CNN based methods are reported in Table. 4.3 in terms of Mean Absolute Error (MAE) and Mean Square Error (MSE). MAE and MSE are mostly used evaluation measures for counting and formulated as

$$MAE = \frac{1}{|T|}\sum_{t\in T}(\mu_t - G_t)^2 \tag{4.8}$$

$$MSE = \frac{1}{|T|}\sum_{t\in T}|\mu_t - G_t| \tag{4.9}$$

where T is the total number of testing frames. While μ_t and G_t are the predicted and ground-truth count of pedestrian respectively in a frame at t. From the Table. 4.3, it is obvious that regression-based methods perform well than CNN based detection methods. It is attributed to the fact that in high-density situations, regression models perform well in approximating the count by leveraging the rich context in crowded patches while CNN based detection models are unable to localize and detect pedestrians due to the small size of the head, occlusion, and perceptive distortions. We observed from the experiments that detection based methods provide a reliable estimation in sparse crowds where the pedestrians are fully visible. Based on our experiments and as obvious from the table we find that detection and regression-based counting methods show different performances depending on the densities of the crowd. The regression-based methods provide reliable estimates when applied to congested scenes. However, these methods cannot provide localization information for persons in the scene and tend to overestimate the count when applied in low-density situations. The detection based methods, on the other hand, can localize each person precisely in low dense situations. However, the performance of detection based methods improves in all situations after employing our proposed Motion Guided Filter.

The average time for processing each frame for detection was 0.044 seconds, 0.130 seconds and 0.048 seconds, for ZF, VGG16 and VGG M, respectively. On average the frame was processed in 0.130 sec/frame.

We also compare our method with other crowd counting methods using UCSD dataset, and the results are reported in Table. 4.4. We use MAE metric as an evaluation measure. From the

table, it is obvious that our proposed method outperforms most of the state-of-the-art methods. Our approach out-performs most of the state-of-the-art methods in UCSD data set. However, our approach shows lower performance in comparison to few state-of-the-art methods. The low performance attributes to the following reasons: (1) UCSD data set consists of extremely low-resolution videos. Each video frame has to be re-sized and padded before input to the network. As a result, frame lost most spatial information and become too coarse to describe pedestrians. (2) Faster R-CNN lacks the ability to detect small objects lies in various scales. In the same way, we compare our method with other methods using Mall and PETS datasets, and the results are reported in Table. 4.5 and 4.6 respectively. In this case, we use VGG16 as best model for comparison with other methods. As obvious from tables, our proposed method produced superior results as compared to the state-of-the-art methods.

4.4 Summary

In this work, we proposed a real-time detection framework for counting of a crowd in a low-to-medium density crowd videos — the framework use state-of-the-art detector Faster-RCNN to detect pedestrians in crowd video. We used Motion Guided Filter to recover misdetections and therefore improve mean Average precision of the overall detections. The improvement in the accuracy of detection also leads to the improvement in the counting and density estimation of a crowd. The proposed approach can be easily incorporated in the real-time monitoring and surveillance applications and as well as high-level scene understanding of crowd.

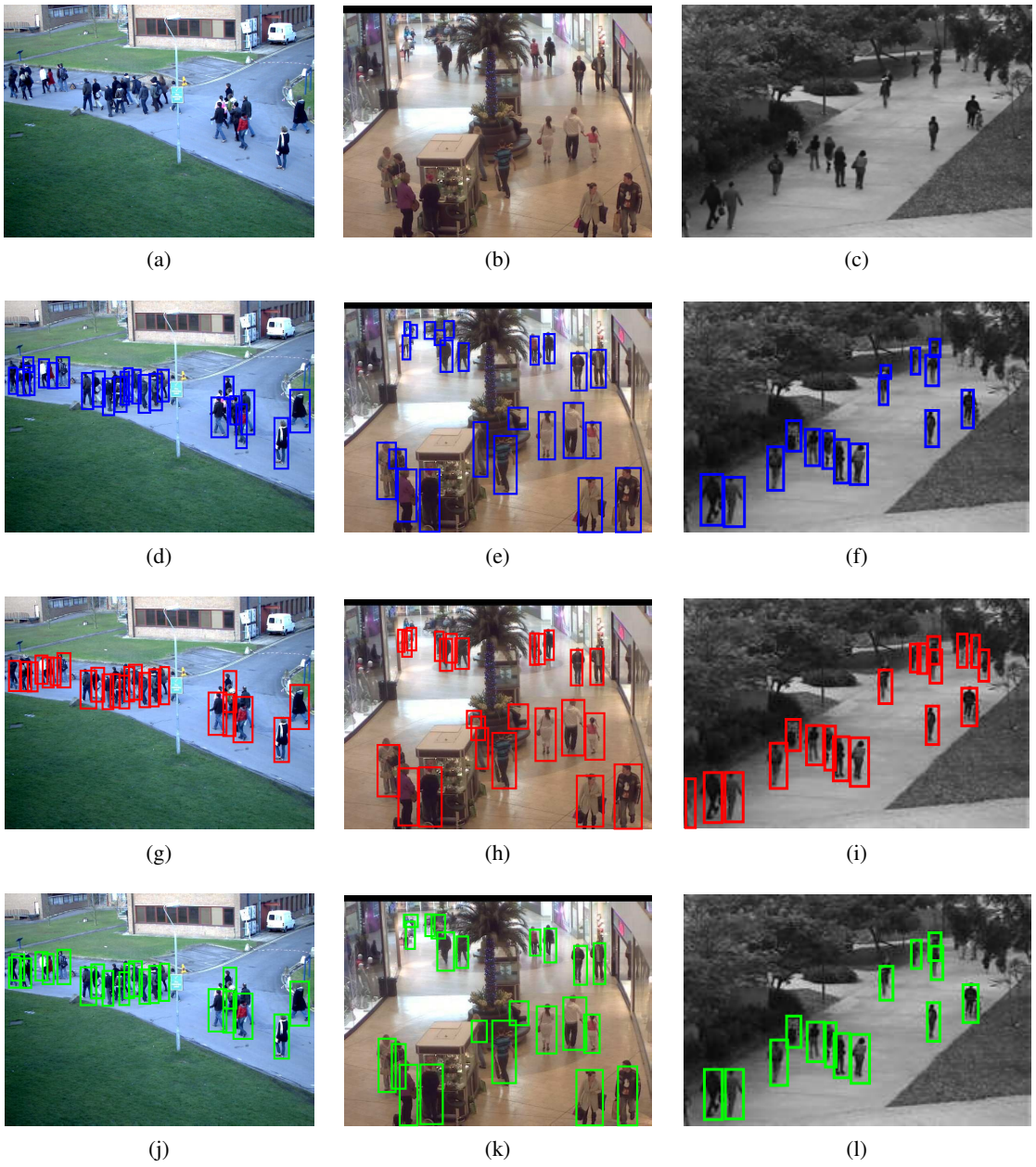


Figure 4.7: Qualitative Results: Row:1 (a-c)Sample images from datasets (a) PETS2009 (b) Mall dataset (c) UCSD dataset Row:2 (d-f) Ground-truth annotations overlaid on samples images Row3: (g-i) detection results Row4: (j-l) Refine results after spatio-temporal smoothing (Best viewed in colour)

"The only way to discover the limits of the possible is to go beyond them into the impossible"

- Arthur C. Clarke

5

Detection of Dominant Motion Patterns in Crowd

This chapter deals with the proposed methodology that exploits motion information from videos to better understand crowd scenes. This chapter will focus on the detection of dominant motion patterns in the crowd.

Major parts of this chapter have been published in the paper titled *"Detecting dominant motion patterns in crowds of pedestrians"*, Saqib *et al*. [157].

5.1 Introduction and motivation

In this study, we propose a novel technique of crowd analysis, the aim of which is to detect different dominant motion patterns in real-time videos. A motion field is generated by computing the dense optical flow. The motion field is then divided into blocks. For each block, we adopt a *Intra-clustering* algorithm for detecting different flows within the block. Later on, we employ *Inter-clustering* for clustering the flow vectors among different blocks. We evaluate the performance of our approach on different real-time videos. The experimental results show that our proposed method is capable of detecting distinct motion patterns in crowded videos. Moreover, our algorithm outperforms state-of-the-art methods.

Public safety and security are becoming a crucial problem in most urban areas. Large events such as marathons, religious festivals, sports, and political gatherings attract thousands of people in a constrained area. Despite safety measures, public disasters still occur frequently. One issue that presents itself in such disasters is the presence of large and conflicting motion patterns, i.e., crowds composed of groups of people motivated by a common goal. It is because of this common interest

that people in crowds compete with each other for common resources. In some cases, conflicting groups come face-to-face with each other in a constrained environment leading to stampedes.

Crowd management is becoming a daunting job because of the complex dynamics of the crowds. To manage high-density crowds, several surveillance cameras are installed at different locations that can cover the whole scene. The whole scene can be overseen by an analyst whose job is to detect some specific activities. Such a manual analysis of the crowd is a tedious job and usually prone to errors. Therefore, we need an automatic technique that can reliably detect specific activities. Developing such techniques and methods has been the goal of many researchers in the area of computer science. Usually in automatic video surveillance, detection, and tracking of pedestrians forms part of pre-processing steps. But in crowded situations, detection, and tracking of pedestrians becomes an even more challenging task because of severe occlusions, clutter and a representation of a small number of pixels per individual. Therefore, to understand crowd dynamics in high-density situations, researchers have focused on global features (optical flow) instead of local ones (head, etc.). In this study, we generate motion fields through extraction of global features by computing the dense optical flow using the Brox and Sand method [20]. A motion field is divided into blocks. Subsequently, we extract dominant motion patterns by employing *Intra-clustering* and *Inter-clustering* algorithms.

5.2 Background and related work

A crowd can form as a result of individual motions or interacting motions of objects or people. Crowds can either be classified as structured or unstructured based on the motion patterns followed by individuals comprising the crowd [145]. In structured crowds, an individual motion is in a certain order, and the orientation of the crowd remains the same for most of the time *e.g.* crossing the road, entry to a train, using the elevator. While in an unstructured crowd, objects or people exhibit motion either in one direction or in another direction randomly, *e.g.* a chaos situation. Another classification is based on the density of individuals in a crowd. Researchers have studied individual tracking or even partial tracking in a crowd of low density. While in high-density crowds, it is more feasible to study the overall behavior of the crowd. Ozturk *et al.* [136], divide the entire scene into small regions to determine the flow vectors in each region. Flow vectors having similar orientations are clustered followed by spatial clustering. As a result, major groups are identified in those small regions. Dominant flow is determined from average location, orientation, and the number of flow vectors in each group in the region. Global dominant flows are determined from the small regions connecting local dominant flow vectors. Wu *et al.* [197], use a region-growing algorithm to segment the crowd flow along with a shape optimization model to update the contours of the domain region adaptively. A KLT tracker is used by [39] to extract tracklets from dense crowds. The Longest Common Subsequence (LCS) algorithm is then used to cluster similar point track segments, which results in dominant motion in a dense crowd. Normalized optical flow features along with different similarity measures are fed to a clustering algorithm [50]. The two clustering algorithms used are (1) Self-tuning spectral clustering (2) Nonlinear dimension re-

duction isomap followed by k-means clustering. A min-cut/max segmentation algorithm is used to segment the crowd flow [175]. First, the foreground is extracted from video using a Gaussian Mixture Model (GMM). The frame is divided into fixed-size blocks, and the correlation of crowd flow among the blocks is computed in a spatio-temporal domain. To remove noise and to have a more consistent representation of crowd flow, a grid of the particle is initialized on the foreground and tracked using Lucas Kanade optical flow with a pyramid. A histogram-based adaptive optical flow segmentation algorithm is used to segment the crowd into different flows based on their direction [163]. Ihaddadene *et al*. [78] used a motion intensity heat map to determine any abnormal activity in the crowd. The first set of points are tracked using optical flow, and then a variation is estimated to determine abnormal activity. Khan *et al*. [84] constructs a more robust and optimal foreground mask from the logical product of the optical flow-based mask and the Gaussian background mask. Segmentation of crowd flow is determined using K-mean clustering followed by a blob absorption algorithm to absorb small blobs, which do not contribute to the dominant flow. Sparse and dense optical flows are used on interest points to compute the global motion field. Min Hu *et al*. [71] used an undirected graph to construct directed graphs to measure the similarity and proximity of flow vectors. The original flow field contains thousands of flow vectors, which are reduced to hundreds of flow vectors by removing noise and redundant information. Then, hierarchical agglomerative clustering was used to cluster the flow vectors in the same direction to determine dominant motion patterns. Wei li [99] proposed a crowd flow segmentation approach based on histogram analysis of optical flow angles. The histogram curve is used to determine segmentation points and subsequently segmentation of crowd flows.

5.3 Proposed methodology

In this section, we discuss the proposed methodology for extracting motion patterns from the given sequence of frames. Given a sequence of frames, the main steps involved in extracting the motion patterns are (1) optical flow field computation followed by construction of the motion field; (2) dividing the motion field into similar non-overlapping blocks and (3) finding clusters in each block by employing our proposed circular clustering algorithm. The final clusters of each block represent motion patterns in the given scene.

5.3.1 Optical Flow and Motion Field estimation

In this step, we compute the optical flow between two adjacent frames of the given sequence followed by the construction of the motion field. Let $f_{(x,y,t)}$ and $f_{(x,y,t+1)}$ be the two consecutive frames of the given sequence, we compute the dense optical flow by employing the Brox and Sand (BS from now on) method followed by a Median filter and Gaussian filter in order to remove noise. Usually, the objects in crowded situations move in a wide area covering the whole scene. Therefore, to capture whole motion information, we compute dense optical flow, that enables us to compute a change in every pixel, in contrast to the sparse optical methods that compute change

(a) sample frame (b) optical flow (c) direction map

Figure 5.1: 5.1a: Sample frame 5.1b: Corresponding Optical flow 5.1c: Direction map

only in interest points, e.g., corners or edges, etc. After computing the optical flow, the next step is to construct a motion field. The motion flow field is the set of independent flow vectors, and each flow vector is associated with its spatial location. The motion flow field contains the instantaneous and temporal information of the scene and can be used for learning motion patterns of the given video. Consider a feature point i in F_t, its flow vector Z_i includes its location $X_i = (x_i, y_i)$ and its velocity vector $V_i = (v_{x_i}, v_{y_i})$, i. e. $Z_i = (X_i, V_i)$ where θ_i is the angle or direction of V_i, where $0° \leq \theta \leq 360°$ and Ω_i is the magnitude. Then, $\{Z_1, Z_2, \ldots, Z_k\}$ is the motion flow field of all the foreground points of an image. The motion flow field $\{Z_1, Z_2, \ldots, Z_n\}$ is an n x 4 matrix where each row represents a flow vector i and columns represent its spatial location X_i and the velocity vector V_i. While n represents the total number of flow vectors (foreground points). Each flow vector represents motion in specific directions (represented by different colors) as shown in Figure 5.1c.

The direction map in Figure 5.1c does not show any dominant motion pattern in the scene and hence does not provide any meaningful information about the motion patterns. Therefore, we need to devise a method that can automatically analyze the similarity between the flow vectors and cluster them into multiple groups. In the following section, we proposed a *circular clustering* approach that automatically can detect dominant motion patterns in the scene.

5.3.2 Circular clustering

The direction map in Figure 5.1c could not characterize dominant motion patterns, and therefore we cannot extract information about motion patterns. To characterize dominant motion patterns in the scene, we need a clustering algorithm that groups the similar flow vectors belonging to a motion pattern in the scene. Before applying our clustering algorithm on the motion flow field, we first divide the motion flow field into N x M blocks. Let $\{b_1, b_2, \ldots, b_n\} \in B$ is the list of blocks, where each block b_i is of size i x j. Our proposed algorithm involves two steps, (i) *Intra-clustering* and (ii) *Inter-clustering*. At the *Intra-clustering* level, we cluster the flow vectors within the block. While in *Inter-clustering*, we cluster flow vectors among the blocks. In the following, we summarize our *Intra-clustering* algorithm.

1. For every block $b_i \in B$, select a set of points whose magnitude Ω_k is greater than a threshold η. Let P be the set of points (or flow vectors) whose magnitude is greater than Ω_k.

2. Initially we assume each point of the block as a separate cluster, and we assign a distinct identifier to each point of the block.

3. Compute the centre of the cluster by computing the circular mean μ of each cluster as in [17]. Let $C = \{c_1, c_2, \ldots, c_n\}$ be a set containing a list of clusters for all n number of blocks, where $c_i = \{\mu_1, \mu_2, \ldots, \mu_k\}$, is a vector of k number of clusters detected in block b_i.

4. Compute a distance matrix D, where each element of the matrix is the distance between circular means μ_i and μ_j and is given by $D_{i,j} = \mu_i - \mu_j$.

5. Compute $d_{min} = \arg\min D$ for all $i \neq j$.

6. If d_{min} is less than δ, then cluster the flow vectors whose distance between the means is d_{min} and continue to step 3 through step 6. Otherwise, go to step 7.

7. Repeat the same steps (1-6) until all the blocks in set B are processed.

After *Intra-clustering*, the flow vectors belonging to each block b_i are clustered into k distinct groups. Now, in the next step, we cluster two distinct blocks b_i and b_j by employing our *Inter-clustering* algorithm 3. The algorithm iterates until all the blocks are processed. The final cluster G represents the dominant motion patterns in the scene.

5.4 Experimental results and discussion

In this section, we discuss the qualitative and quantitative results obtained from the experiments conducted. The experiments are carried out on a PC with a 2.6 GHz core i5 processor and with 8.0 GB memory. We used a publicly available data set from the University of Central Florida [4]. The data set covers so-called structured and unstructured crowds with different density situations. The first row of Fig 5.2 shows a frame from the video in which the people are moving in opposite directions in a square. As is obvious from the figure, the clustering of flow vectors near the boundaries of the opposite flows becomes very hard. By employing our proposed clustering algorithm, the flow vectors are correctly distinguished, and the corresponding motion patterns are correctly detected. The second row of the figure shows a relatively simple scenario, where the people are running in a marathon in a U-shape. Our proposed algorithm detects three different motion patterns. A very complex scenario is shown in the third row of the figure, where the people are moving in different directions making it very challenging for the clustering algorithm. Our method detects four different motion patterns which also correspond to the ground truth data. From further experiments, we also observe that small clusters are also detected near the boundaries of two conflicting flows. This is mainly due to the following reasons; (1) if the object moves slowly, the inside and outside flow vectors of the object are not the same and hence are classified as two different flows;

Algorithm 3 Inter-Clustering of Blocks

Input: Set of Local Clusters C
Output: Global Cluster G

function INTERCLUST(C)
    initialize array G as empty
    G ←insert the first list of cluster c_1 from C.
    for all List of cluster c in C **do**
        G = MergeBlocks(G,c)
    end for
end function

Input: Cluster c_1**,** c_2
Output: Updated Cluster G

function MERGEBLOCKS(G, c)
    initialise cluster identifier "id" to 1.
    for i = 1 to sizeof(G) **do**
        for j = 1 to sizeof(c) **do**
            if $\|G[i] - c[j]\| \leq \tau$ **then**
                $\mu_m \leftarrow$ compute the mean of c[j] and G[i]
                G[id] = μ_m
                Delete c[j]
            end if
        end for
        increment id by 1.
    end for
    Return G
end function

Table 5.1: Performance evaluation using different parameter settings

Video sequence	Threshold values	Block Size $20X20$	Block Size $30X30$	Block Size $40X40$	Accuracy Actual number of motion patterns
People	$(\eta = 0.16, \delta = 1.3, \tau = 1.2)$	4	4	6	5
Marathon	$(\eta = 0.07, \delta = 1.3, \tau = 1.3)$	3	3	4	3
Road cross	$(\eta = 0.04, \delta = 1.5, \tau = 1.1)$	2	3	4	2
Hajj	$(\eta = 0.06, \delta = 1.4, \tau = 1.3)$	2	2	3	2
Escalator	$(\eta = 0.05, \delta = 1.2, \tau = 1.4)$	2	2	3	2

(2) the optical flow near the boundaries of two conflicting flows is ambiguous. We get rid of these small clusters by adopting a blob absorption approach as proposed in [84].

The performance of our algorithms depends on different threshold values given in Table 5.1. Whereas η, δ and τ are *intra-clustering*, *inter-clustering* and distance between the means of the clusters respectively. The detection of motion patterns is evaluated at different block sizes. However, there is no significant improvement shown in the accuracy of motion pattern detections for

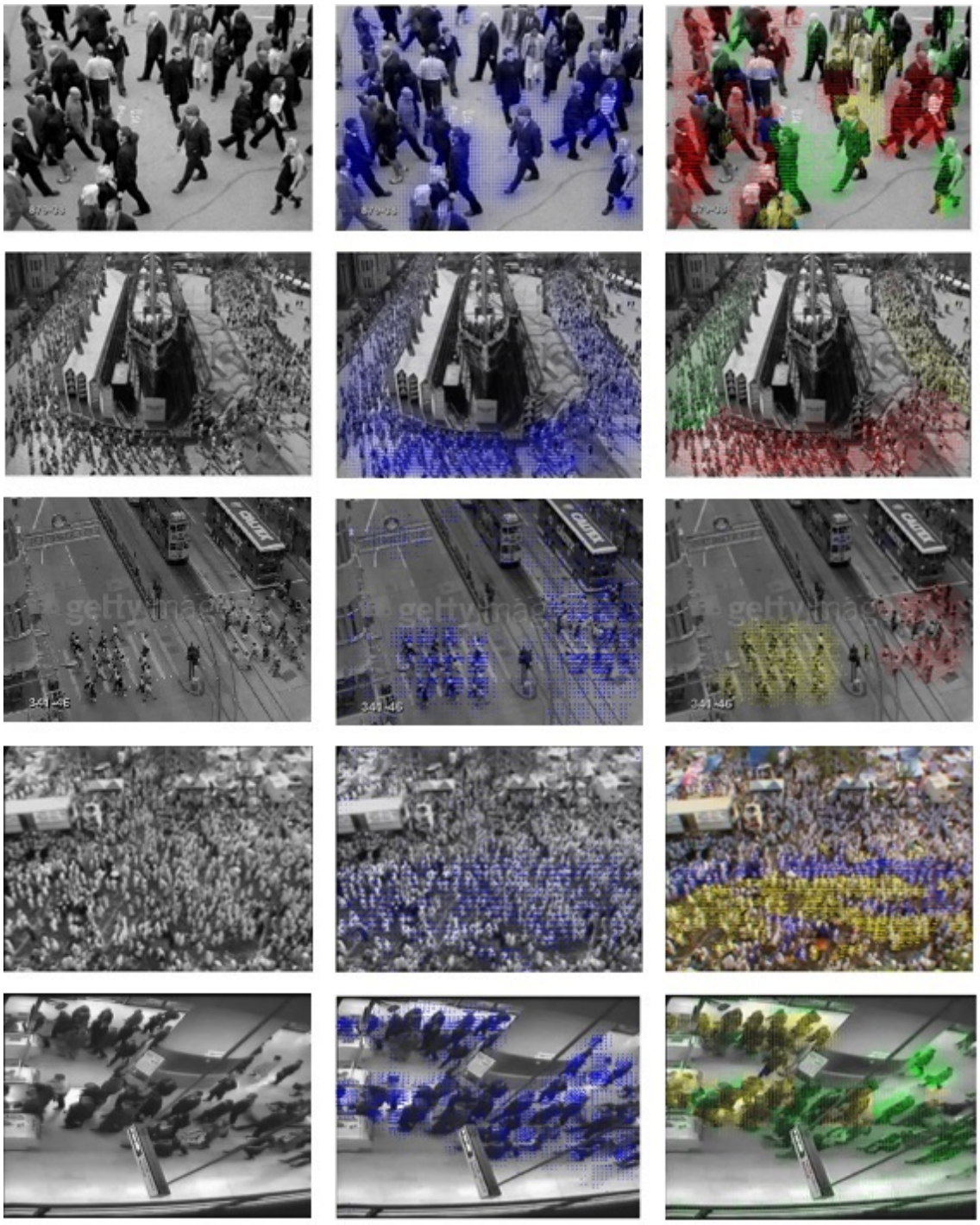


Figure 5.2: The first column shows the input frames. The second column shows the motion flow field of the corresponding input frames, and the third shows the detected motion patterns. (Best viewed in colour)

the block size below $20X20$. Block size lower than $20X20$ lead to very slow execution of the proposed algorithms while block size larger than $40X40$ creates block boundary artifacts and false motion pattern detections in *inter-clustering* of flows. Therefore experiments were carried out with the optimal block size of $30X30$.

Similarly, all other thresholds are selected carefully according to the video being processed. To evaluate the performance, we compare our approach with the two other methods and the ground truth. The ground truth is manually generated from the motion flow field. As is evident from the Figure 5.3, our approach outperforms the two other methods used for crowd flow segmentation.

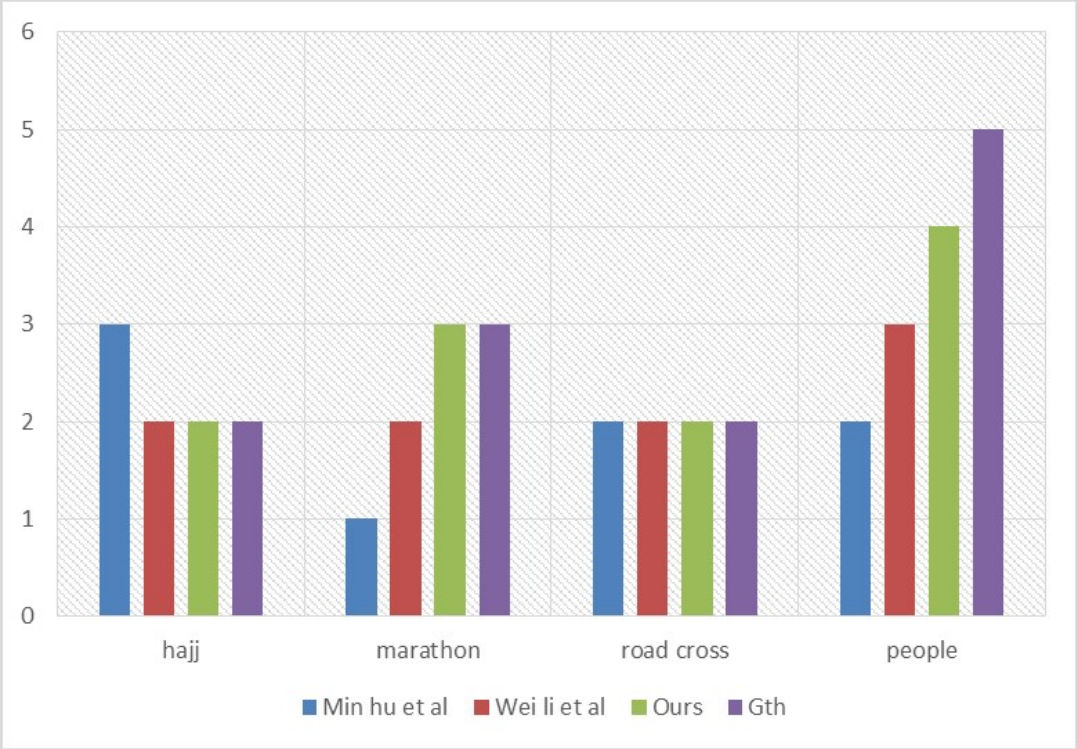


Figure 5.3: Comparison with state-of-the-art methods

5.5 Summary

In this study, we used a dataset that covers both high and low-density crowded situations, and we propose a framework that segments the crowd flow into different meaningful semantic regions. These kind of analysis are very useful for crowd managers in understanding crowd behavior. Moreover, this analysis also provides useful input to crowd simulation models for initialization and validation of their models. Our approach does not rely on the traditional methods of detection and tracking and rather adopts a holistic approach by employing optical flow, followed by a clustering algorithm. This approach can be applied to different situations, and it is independent of the camera view.

"I always wondered why somebody didn't do something about that; then I realised, I am somebody."

- Myra Hargrave Comiskey

6

Extracting Motion Information from Crowd Scenes

It is very critical for a computer vision system to capture accurate motion information for understanding the behaviour of the crowd. Thus the performance of the system is dependent upon how the motion is represented. The desired motion representation should generate long and most importantly reliable trajectories. These trajectories can then be used to describe the flow for the whole video. Traditionally, optical flow is used for computing the pixel-wise flow between the two frames [21, 105]. However, there are some inherent problems in the direct usage of optical flow for motion representation. Firstly, optical flow produces ambiguous results on the boundaries of conflicting flows and perform poorly in the case of very slow moving objects. Secondly, optical flow does not represent the long-range spatio-temporal motion representation required in many applications. Major parts of this chapter have been published in the paper titled *"Extracting descriptive motion information from crowd scenes"*, Saqib *et al*. [158].

6.1 Introduction and motivation

Crowd scene constitutes a significant number of people gathered together in one place. The analysis of crowd has many real-world applications in surveillance, public space design, crowd management and in the behavioural study of animals and marine life. The visual analysis of crowd includes various computer vision tasks such as object detection, tracking and using motion information for understanding the behaviour of the crowd. However, the crowd is a complex phenomenon, and the application of particular computer vision technique depends upon the structure and density of the crowd. Detection and tracking might work very well in the low-density crowd

because of the clear visibility of pedestrians in the crowd. However, in high-density crowd detector most often miss the detections or result in false-positive detections. Moreover, tracking is also not feasible in the high-density crowd. The tracker is lost easily and requires re-initialization at regular interval. The reason for suboptimal performance is that there is less number of pixels per person and inter-object occlusions, which make the detection and tracking unreliable solutions. Therefore in such scenarios, a holistic understanding of the scene is required. The study of crowd behaviour includes the study of flows, dominant flows/common pathways, and identification inflows/source and sinks/outflows. This study of behaviour can be beneficial as a preprocessing step in the understanding of crowd behaviour in extraordinary situations such as congestion.

Typically obtaining complete trajectories is a difficult task from crowd motion in the high-density crowd. Therefore to get complete trajectories a notion of tracklet is used to capture short-term motion. A tracklet is a fragment of trajectory obtained by the tracker for the object. As previously mentioned, detection of the object and their tracking does not perform well in a high-density crowd. In most of the previous studies, tracklets are extracted from feature points and subsequently tracked for the very short duration to generate tracklets. After tracklets generation, various approaches such as Markov Random Field (MRF) [217] are used to ensure spatio-temporal dependencies between the tracklets [217, 219]. The tracklets gives a robust high-level representation of the crowd and are less like likely to drift as compared to complete trajectories [97].

In this study, we propose a framework for crowd behavior by analyzing the flows in the crowd and describing the flows quantitatively. Our proposed framework used particle flow computed by the moving grid of particle with the optical flow initialized by the dynamical system [166]. Numerical integration of the dynamical system over the segment of the video is used to generate tracklets. The first step in the crowd analysis is to generate reliable long trajectories. Firstly, we make use of tracklets to generate short term reliable trajectories for a small segment of video and then use clustering both spatially and temporally to produce flows for the whole video. By using the resultant flows, we can identify source/inflow and outflow/sink of the scene. Once the reliable flows are achieved, we compute basic parameters like speed, density and direction of the dominant flows. These parameters can fully characterize pedestrian movement and measuring these will help the crowd managers in understanding overall motion in the scene. Once dominant flow are describe by these parameters, we can visually analyze speed-density relationship that serve basis for understanding the behavior of complex system.

6.2 Background and related work

There has been a body of work on the usage of tracklets for understanding the collective motion of pedestrians in the crowd. Zhou *et al*. [217] used Random Field Topic (RFT) to obtain semantic regions by non-parametric clustering of the tracklets. The RFT model ensures spatio-temporal coherence between the tracklets. The information about source and sink in the scene are given to the models as a highly semantic-level priors during the inference. However, our framework does not assume sources and sinks to be known a priori. In a variation of this approach, Zhou *et al*. formu-

lated the whole crowd with the agent-based model called Mixture model of Dynamic pedestrian Agents (MDA) [219]. In this model, each pedestrian is modeled as an agent with the three factors. The factors included the belief of starting and the destination position of the pedestrian, movement dynamics, and the timing of pedestrian entering the scene. The collective behavior is analyzed using agent after learning from real data. However, this agent-based approach is only suitable for the low-density crowd, where agents can easily model each pedestrian movement. Blei and Frazier [18] used Distance Dependent Chinese Restaurant Process (DD-CRP) to cluster the coherent tracklets in the first stage. In the second stage, the clusters are aggregated temporally to obtain pathways, or dominant motion flows in the scene. Next, a geometric analysis is applied on the pathway shape to determine the sources and sinks in the scene. Hassanein *et al*. [64] considered tracklet as a directed line segment and used line geometry to find similarity between the tracklets. Upon clustering, semantic regions are extracted which include source, sink, and common pathways. Wang *et al*. analysed motion patterns by clustering the hybrid generative-discriminative feature maps [185]. The extracted feature map associate motion pattern with low-level feature. Finally, hierarchical clustering is used to cluster feature maps into semantic regions.However, above all mentioned approaches only describe the crowd flows qualitatively. Meta-tracking is another technique employed to discover the scene structure in the crowd [81]. The Orientation Distribution Functions (ODFs) model is used to find out the direction of the flow at each pixel. A particle meta-tracking is used to obtain particle trajectories called meta-tracks. Finally, meta-tracks are clustered to get dominant motion flows in the crowd. This kind of motion representation is complex for extracting dense trajectories.

The raw data about path taken by pedestrian is considered noisy. For instance, Nedrich *et al*. used weak tracking system to identify source and sink through the mean shift clustering algorithm [130]. Manifold is a denoising algorithm which is also used for clustering of the data. Xu *et al*. proposed a shrinkage-based framework for unsupervised trajectory clustering using adaptive kernel-based estimation [204]. The framework can be viewed as the combination of mean-shift clustering and manifold base model. The center of the cluster represent manifold, and the structure of cluster shows motion patterns exhibited by the pedestrians in the crowd. Trajectories in the cluster are considered as noisy traversal of the manifold. Thus suppressing the noise and reconstructing manifold is seen as clustering of the trajectories.

Some approaches have been proposed to extract feature point for extracting trajectories called feature trajectories. The features are extracted using SIFT, HOG, and Histogram of Oriented Flow (HOF) or combination of various feature descriptors [171]. SIFT features are used to find correspondence between two frames for trajectories. Similarly, sun *et al*. used SIFT descriptor along with KLT tracker to track feature of the sampled points from dense SIFT descriptor for dense trajectories [170]. Nawaz *et al*. proposed an end-to-end approach for trajectory clustering in aerial videos [129]. Firstly, camera motion is compensated before clustering of trajectories obtained from the co-efficient of Discrete Wavelet Transform (DWT). In another approach motion is analyzed and performed by the clustering of the trajectories in the frequency domain [73]. However, these approaches are not able to capture whole motion information because only few

feature points are detected which are not sufficient for the extraction of dense trajectories. The most related work to our proposed work used KLT tracker to extract motion trajectories and used hierarchical clustering to obtain dominant flows [38, 41]. However, the approaches are not able to capture the long-term motion required for understanding the behavior of the crowd. Furthermore, these approaches does not identify source and sink locations of the dominant flows.

Together, these studies provide useful insight into the study of crowd flow regarding motion representation, similarity measures and crowd behavior qualitatively such as the identification of sources, sinks and dominant motion pattern or common pathways. However, previous studies did not quantify the speed, density, and direction of the flows in the crowd. Our proposed framework extract dense motion trajectories using tracklets as a basic motion unit. Most importantly, our work uses perspective normalization of the extracted flows to find the accurate measure of the density of the pedestrians in the flows. Furthermore, the flows are described by the speed and direction of dominant motion patterns.

6.3 Proposed methodology

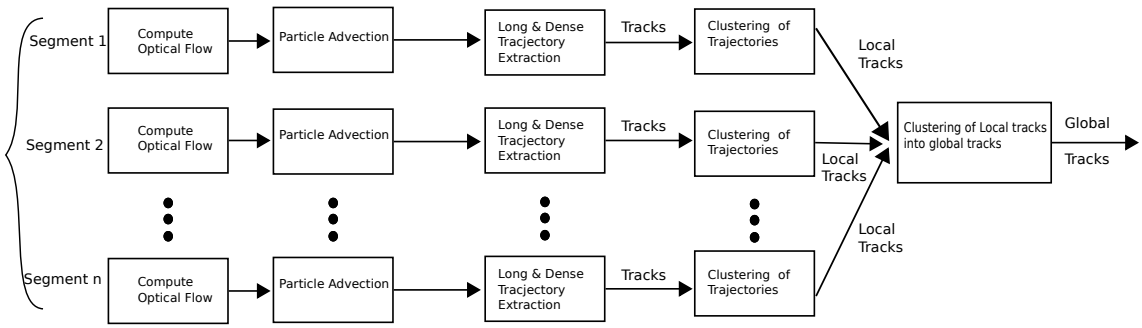


Figure 6.1: Block diagram of proposed framework for dense trajectories extraction

In this study, we propose a framework for describing flows exhibited by pedestrians in the crowd. In the first step, a stable, dense trajectories are generated in the crowded scene. Secondly, the generated trajectories are clustered together spatially and temporally to get dominant flows. These flows help at a later stage in the identification of sources and sinks for the behavior analysis. According to the proposed framework, input video is divided into multiple equal length segments as shown in the Fig 6.1. To generate reliable tracklets, the first frame of each segment is overlaid with the grid of particles. After initialization of the dynamical system, the particles are moved through the segment by optical flow [166]. The integration of the dynamical system gives reliable particle trajectories for particular video segment. The trajectories are then clustered into local tracks using Longest Sub-sequence (LCS) metric. Global tracks are obtained by applying hierarchical clustering algorithm to the local tracks. The following are the steps discussed in greater detail to extract descriptive motion information from the dominant flows.

6.3.1 Motion flow-field computation

Consider a feature point i in the frame of the video segment at time t is given by the following equation (6.1)

$$Z_{i(t)} = (X_{i(t)}, V_{i(t)}) \tag{6.1}$$

Where $Z_{i(t)}$, $X_{i(t)}$, $V_{i(t)}$ are the flow vector, location and velocity of the i^{th} feature point at time t respectively. Furthermore, $V_{i(t)} = (vx_{i(t)}, vy_{i(t)})$ incorporate both change in x and y directions. Similarly, $X_{i(t)} = (x_{i(t)}, y_{i(t)})$ is the initial location of the particle in the grid and $X_{i(t+1)} = (x_{i(t+1)}, y_{i(t+1)})$ is next location at time $t+1$ approximated for all particles in the grid by solving the dynamical given by eq (6.3). There is an angle θ_i associated with velocity vector $V_{i(t)}$ where $0° \leq \theta \leq 360°$. The motion flow field constitute all the flow vectors given by $\{Z_1, Z_2, \ldots, Z_m\}$ where m is the number of all foreground feature point in the frame. The dynamical system is initialized in which the velocity of feature point i is related to its optical flow at time t given by eq (6.2).

$$V_{i(t)} = F(X_{i(t)}) \tag{6.2}$$

6.3.2 Particle advection

In this step, a grid of particle is overlaid on the first frame and advect it on the optical flow field for each segment of the video. The grid resolution is kept at a lower resolution to make the process less computationally expensive. Let $X_{i(t)} = (x_{i(t)}, y_{i(t)})$ is the initial location of the particle in the grid. The next location $X_{i(t+1)} = (x_{i(t+1)}, y_{i(t+1)})$ can be approximated for all the particle in the grid by solving the following eq (6.3).

$$X_{i(t+1)} = F(X_{i(t)}) + X_{i(t)} \tag{6.3}$$

Let Ω_i is the trajectory of the particle, which is essentially the movement of a particle from one frame to another frame according to the flow map. The pair of the flow maps are computed for each particle in the video segment as in [166]. In the unstructured crowd, the particle may drift from the flow of pedestrian and background noise can also induce its effect in the particle flow. Therefore a threshold term B_i is included in eq (6.4) to remove erroneous particles out of the computation. As a result of particle advection short trajectories called tracklets are obtained.

$$X_{i(t+1)} = F(X_{i(t)}) + X_{i(t)} * B_i \tag{6.4}$$

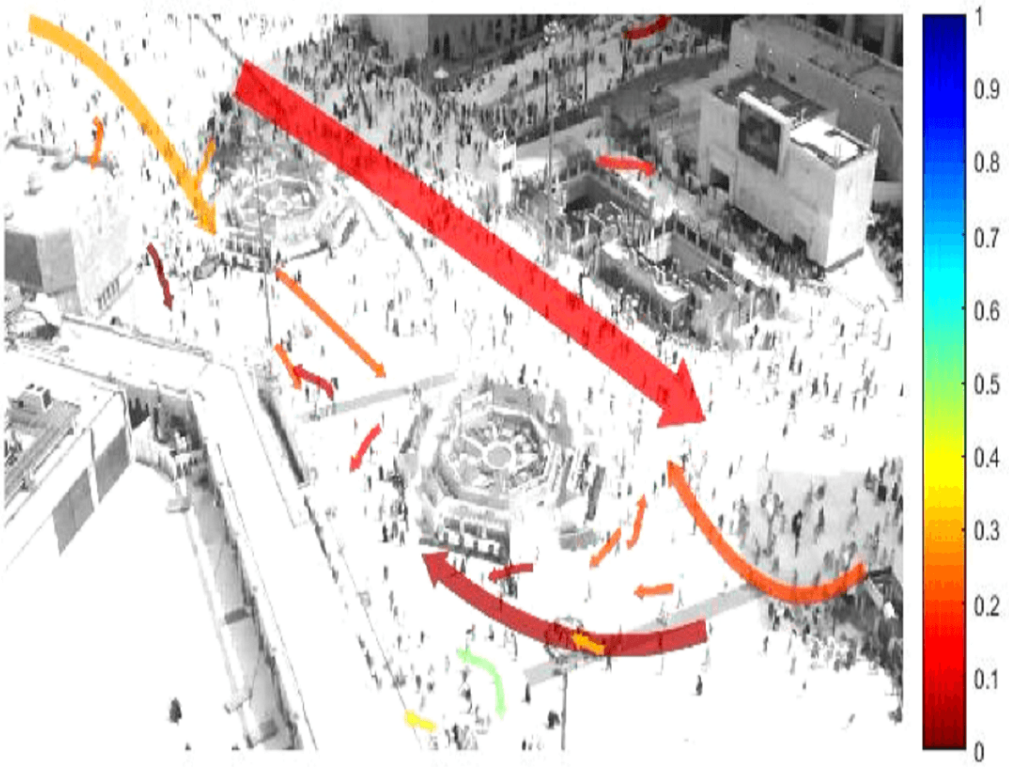
6.3.3 Clustering of tracklets

Tracklets only provide information about the local movements. Because the tracklets are short-term, therefore to obtain dense trajectories the tracklets are clustered together to generate dominant motion flows. The source and sink of the dominant flows are identified which help in describing the speed, density, and mean direction of the flows.

The first step in the clustering in getting the dense tracks is by extending shorter tracklets through the segments of the video. The assumption here is that these shorter tracklets are located spatially close together and can have plausible continuation if they can be connected to each other in a head-to-tail manner for a long trajectory construction. For the construction of long trajectory, the first tracklet called a query tracklet (that need to be extended) is compared with the potential candidates tracklets from the neighborhood of query tracklet's position on the grid. The best among the candidate is selected for the continuation of query tracklet. The selected candidate tracklet is now considered to be a query tracklet, and the same process is repeated until there are no more tracklets to be extended. Once the set of this tracklets are achieved, which are then connected by mean of K^{th} order least square polynomial regression. As shown in the Figure 6.1 after long trajectory construction, tracks are fed to the hierarchical clustering algorithm which clusters them into local tracks using Longest Common Sub-sequence as a similarity metric (LCS). The LCS is an efficient technique for matching tracks of unequal length to determine the longest common portion between the tracks. Finally, the local tracks are combined into the global tracks using the same hierarchical clustering algorithm.

Once the global track (dominant flows) are achieved, the next step is to compute basic parameters that describe the flows. The first parameter that describe the flow is average speed of the flow. Speed together with the density can serve a base for understanding the behavior of complex systems. Let G_1 and G_2 are the two dominant flows obtained after clustering. Let $G_i = \{t_1, t_2, ...t_n\}$, where t_j are the tracklets that belong to the dominant flow G_i. In order to find the average speed of G_i, we first compute the the speed of each tracklet t_j belongs to G_i. The instantaneous speed v_k of each point $p_i \; \varepsilon \; t_j$ can be computed as $v_k = \frac{\Delta_d}{\Delta_t}$, where Δ_d is the displacement among two consecutive points p_i and p_j that belong to t_j and Δ_t is the frame rate of video. The overall speed v of tracklet t_j can be computed by $v = \frac{1}{n} \sum_{k=1}^{n} v_k$ where n is the total number of points in t_j. In this way we compute the speed of every tracklet that belongs to G_i and finally average speed of $V_i = \frac{1}{N} \sum_{m=1}^{N} v_m$, where N is the number of tracklets belong to G_i.

In the next step, we compute the mean direction of the flow in the same way as average speed is computed. Here, we compute instantaneous flow f_k of each point $p_i \; \varepsilon \; t_j$. since f_k is a vector showing displacement u in x-dimension and displacement v in y-dimension, therefore we compute the angle $\theta_k = tan^{-}1 \; \frac{u}{v}$. The mean direction of $t_j = \frac{1}{n} \sum_{k=1}^{n} \theta_k$ and finally average direction of G_i becomes $\frac{1}{N} \sum_{m=1}^{N} \theta_m$. We compute the density D_i of G_i by $\frac{N}{M}$, where N is total number of tracklets belong to G_i and M is the total number of tracklets belonging to all dominant flows. The summarized representation of dominant flows is shown in Figure 6.3. It is obvious in the figure that multiple dominant flows with their respective descriptions are detected by our algorithm. From the figure, we can easily visualize the inverse relationship of speed-density. The red-thick and long arrow shows the main dominant flow which implies that density on this path is high relative to other paths in the scene.



(a) Hajj video sequence.

Figure 6.2: Summarized representation of dominant flows after the application of the algorithm. More red is the color the low is the speed of the flow while thicker the arrow represents the high density. (Best viewed in colour)

6.4 Experimental results and discussion

We have conducted several experiments to evaluate the performance of the proposed framework on publically available datasets. All the algorithms are implemented in MATLAB and the experiments are carried out on PC with the configuration of $2.7Ghz$ (core i7) processor and 8 GB of RAM. The datasets include both indoor/outdoor scenes comprising of structured/unstructured crowd of varying density level. The complete details of the datasets are shown in Table 6.1. The input to our proposed framework is the segments of the video. We have divided every video into the segments of 50 frames. The ground truth for the all the videos is established manually by the human observer so that obtained results can be compared to the ground truth.

The effectiveness of our proposed framework is compared with the relevant current approaches in the extraction of dominant flows/global tracks and also in the identification of sources and sinks of the dominant tracks. The extraction of accurate flows is vital to the understanding of the crowd scene. Our framework uses the extracted information to describe the speed, density, and mean direction of the dominant tracks. Firstly, we compare the global tracks/dominant flows generated by our framework with ground-truth dominant flows. Moreover, a similar comparison is drawn between the current approaches and ground -truth flows using a flow similarity metric. The flow

Table 6.1: Datasets for comparison

Sequence	Resolution	Location	Crowd Size
Airport [3]	110	110	5 to 34
Hajj [3]	120	120	3 to 24
Station [41]	100	100	0 to 42
Escalator [41]	100	100	0 to 42
Gallery [11]	100	100	0 to 42

similarity metric is used to measure the similarity between dominant tracks generated and the ground-truth dominant tracks. The flow similarity (Sim) is given by the (6.5)

$$Sim = \frac{\left(\sum_{i=1}^{N} \arg\max_{j\in[1,M]} \frac{LCS(Gtrack_j, Gth_i)}{Length(Gth_i)}\right)}{N} \tag{6.5}$$

Whereas N and m represent the total number of ground truth tracks and detected number of tracks respectively. The Figure 6.4 shows the performance of our method encoded with the colors. The darker the color, the more similar is the extracted global tracks with the ground truth flows. Similarly, a lighter color indicates lower resemblance of the detected global tracks with the detected one.

Secondly, The second metric identifies how closely the locations of sources/sinks of extracted trajectories lie together to the ground truth sources/sinks. For this comparison, merely using distance metric such as euclidean will not work as it is hard to related changes in the scene to the changes in the real-world.

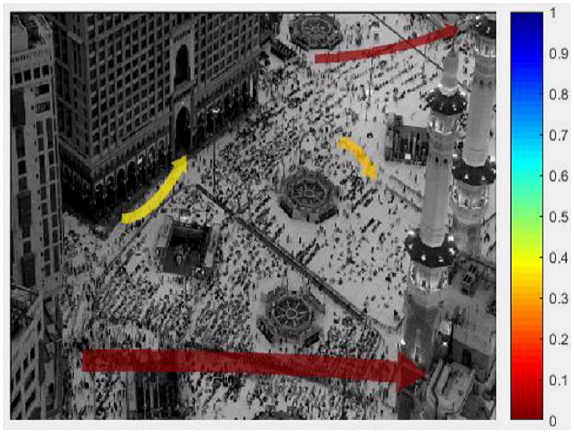
Therefore, we have used an association matrix to relate the locations of sources and sinks by their joint probability distribution. We have constructed an association matrix for ground-truth, our method, and the state-of-art methods. The measure of *Kullback-Leibler (KL) divergence* also referred to as relative entropy is adapted to compute the difference between the association of matrices our methods and other baseline methods with the ground truth matrices. The Kullback-Leibler formula is given below in (6.6)

$$D_{KL}(AM_{Gth}||AM_{Gtrack}) = \sum_{i} AM_{Gth}(i) \ln \frac{AM_{Gth}(i)}{AM_{Gtrack}(i)} \tag{6.6}$$

The comparative analysis is shown in the Fig 6.4 The figure explores the difference between the relative entropies of our method with other methods encoded with color. The smaller value of the entropies for our method indicates that the sources and sinks locations are closed to the locations established by the ground-truth entropies as compared to the other state-of-the-art methods.

6.5 Summary

we have proposed a framework for extracting descriptive motion information from the crowd scenes. The framework makes sure that reliable, dense and trajectories are extracted. The descriptive information includes speed, density, and mean direction of dominant flows.

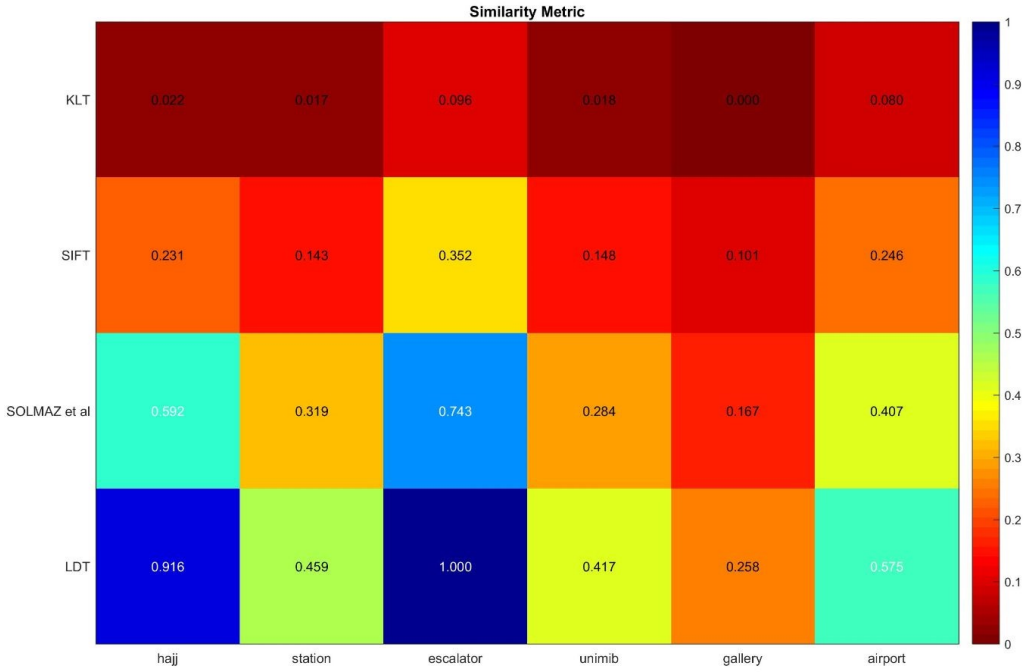


(a) Gallery video sequence.

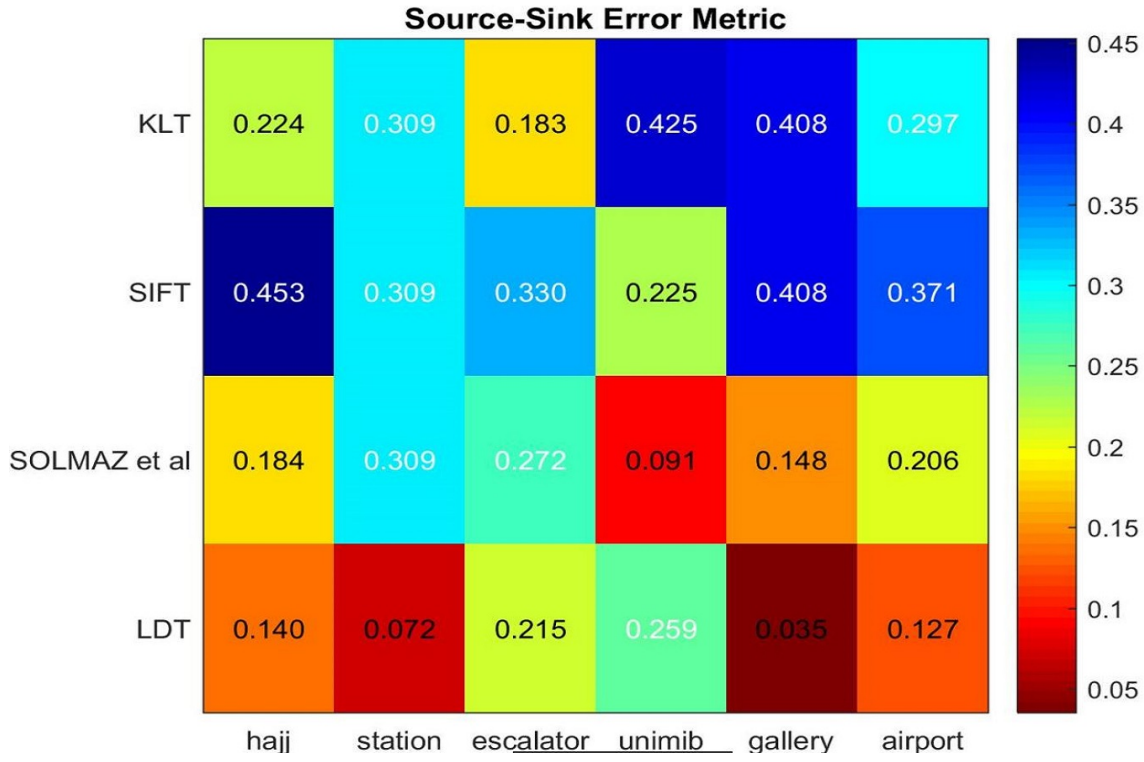
(b) Airport video sequence.

(c) Unimib video sequence.

Figure 6.3: Summarized representation of dominant flows after the application of the algorithm. More red is the color the low is the speed of the flow while thicker the arrow represents the high-density. (Best viewed in colour)



(a)



(b)

Figure 6.4: Top row shows similarity metric comparison of our Long Dense Trajectories (LDT) with base line trajectories and Similarity metric comparison of LDT with other similar methods. Bottom row shows comparison error metric of our LDT with base line trajectories and comparison error metric of LDT with other similar methods (Best viewed in colour)

"Our greatest weakness lies in giving up. The most certain way to succeed is always to try just one more time."

- Thomas A. Edison

7

Conclusions and Future work

This chapter concludes the thesis and provides some direction for future work. The aim of the thesis is to explore different aspects of the pedestrian crowd. Those aspects include crowd counting, flow analysis, and the study of crowd behavior. This study aim is also to come up with the automatic analysis of crowd videos. I have adopted/proposed several computer vision and deep learning approaches to build intelligent solutions for understanding the crowd dynamics.

In the first part of the thesis, I have adopted classical feature descriptors such as LBP, GLCM, GOCM, etc. for capturing the texture exhibited by the crowd. I concluded, that texture feature alone cannot represent the crowd properly. Moreover, the texture feature is crowd blind and will not classify crowd when presented with some arbitrary texture pattern. A crowd is a complex phenomenon; a single hand-crafted feature is not very helpful. Therefore a combination of several features has been evaluated. When the crowd is sparse, the current state-of-the-art object detectors is an excellent alternative for crowd counting and motion analysis as it also provides the localization. I used Faster-RCNN to detect and localize person and then improved the performance by using the proposed Motion Guided Filter. The MGF can suppress false positive and recover missed detections using spatio-temporal information.

In the second part of the thesis, I adopted the holistic approach of representing the crowd flow with the dense optical flow. Following the extraction of optical flow, a hierarchical clustering algorithm was used to find dominant motion patterns in the crowd. Moreover, motion information from crowd scenes such as average speed, density, and mean direction is extracted - the motion information provide useful insight into crowd motion analysis.

7.1 Summary of the thesis

- Chapter 2 presents an extensive literature review about state-of-the-art methods published related to crowd dynamics using computer vision and machine learning approaches. The crowd dynamics covers different aspects of crowd analytics which include crowd counting, crowd motion analysis, crowd behavior analysis, and crowd anomaly detection. It was found in the literature that a large body of work has been published using hand-crafted features to represent the crowd — pros and cons of these approaches discussed in greater detail. The high-density crowd can be broadly classified into structured and unstructured crowds. In a structured crowd, the motion of the people is predictable and goal oriented while in an unstructured crowd, the motion of the people is unpredictable and random. The first type of analysis is to find the count and density of the crowd. Traditionally, the counting analysis of crowds falls into two categories. 1. Counting by regression 2. Counting by detection Counting by regression is most suitable for a very high-density crowd where the features extracted are directly mapped to density using various regression techniques such as Gaussian Process Regression, Ridge Regression, and Bayesian Process Regression. In a regression-based analysis, feature map is normalized for perspective distortion before counting. With the current success in deep learning for object detection tasks, a person detector can be efficiently used for low-density crowd counting with the benefit of localization.

- Chapter 3 describe the proposed framework for crowd counting using texture feature. This framework uses the fusion of different texture features for crowd counting. According to the framework, images are divided into blocks and blocks are further divided into overlapping cells to capture more texture as crowd is a complex phenomenon. The texture features are then extracted and normalized for perspective before classification. Experimental analysis proved that fused texture features have superior performance compared to the texture feature when used alone. The focus of the second part of the chapter is using state-of-the-art object detectors for the task of head detection. The choice of head detection for detecting the number of people is a viable option because it is visible most of the time even when other body parts are fully occluded. However, the feature extracted for head detection is not discriminative enough to be used alone for person detection. Experimental results show that the head detection task is complicated. However, region-based object detector works well on different movie dataset.

- In chapter 4, counting framework for counting the pedestrians in the crowd is proposed. The proposed frame using state-of-the-art object detector to detect a pedestrian. However, the detector performance is degraded due to occlusion, deformation, and illumination, etc. To overcome the limitations of pedestrian detectors, a Motion Guided Filter was proposed that exploits spatial and temporal information between consecutive frames of the video to recover missed detections. The framework is based on Deep Convolution Neural Network (DCNN) for crowd counting in the low-to-medium density videos. Various state-of-the-art network architectures have been employed namely Visual Geometry Group (VGG16),

Zeiler and Fergus (ZF) and VGGM in the framework of a region-based DCNN for detecting pedestrians. After pedestrian detection, the proposed MGF is used to suppress multiple detections and recover misdetections. The performance is evaluated on three publicly available datasets. The experimental results demonstrate the effectiveness of the approach which significantly improves the performance of state-of-the-art detectors.

- In chapter 5, a hierarchical clustering approach for crowd motion analysis was proposed, the aim of which is to detect different dominant motion patterns in real-time videos. A motion field is generated by computing the dense optical flow. The motion field is then divided into blocks. For each block, an Intra-clustering algorithm for detecting different flows within the block. While Inter-clustering algorithm for clustering the flow vectors among different blocks. The performance is evaluated on different real-time videos. Experimental analysis shows that our approach can cluster accurately dominant motion patterns as compared to other proposed approaches.

- An important contribution that automated analysis tools can generate for management of pedestrians and crowd safety is the detection of conflicting large pedestrian flows: this kind of movement pattern, in fact, may lead to dangerous situations and potential threats to pedestrian's safety. For this reason, detecting dominant motion patterns and summarizing motion information from the scene are inevitable for crowd management. In this chapter, a framework is developed that extracts motion information from the scene by generating point trajectories using particle advection approach. The trajectories obtained are then clustered by using an unsupervised hierarchical clustering algorithm, where the similarity is measured by the Longest Common Sub-Sequence (LCSS) metric. The achieved motions patterns in the scene are summarized and represented by using color-coded arrows, where speeds of the different flows are encoded with colors, the width of an arrow represents the density (number of people belonging to a particular motion pattern) while the arrowhead represents the direction. This novel representation of a crowded scene provides a clutter-free visualization which helps the crowd managers in understanding the scene. Experimental results show that our method can be effectively used for crowd management.

7.2 Future research

Various approaches for crowd analytics were investigated in this thesis with the objective of automating the visual analysis of the crowd. Several areas are identified for further research which is summarised as follow.

1. Various action recognition and pose estimation approaches in conjunction with temporal information can be used for better understanding of crowd behavior in low-to-medium density crowd videos.

2. Due to viewpoint variation, the performance of the existing system can be extended if the crowd is covered by multiple cameras. This will require a higher level of correlation of multiple cameras for the analysis of crowd.

3. Another extension to the crowd counting framework is to modify current state-of-art-the object detectors to detect very small objects that can be used for counting in very high-density crowd situations.

4. The detection framework could be extended in conjunction with multi-target tracking for more dense and complex videos.

BIBLIOGRAPHY

[1] http://www.cvg.reading.ac.uk/pets2009/a.html. 7

[2] S. Ali. *Taming Crowded Visual Scenes*. PhD thesis, Orlando, FL, USA, 2008. AAI3377797. 5

[3] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007. 7, 22, 88

[4] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, June 2007. 24, 25, 26, 77

[5] S. Ali and M. Shah. *Floor Fields for Tracking in High Density Crowd Scenes*, pages 1–14. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. 24

[6] E. L. Andrade, S. Blunsden, and R. B. Fisher. Modelling crowd scenes for event detection. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 175–178. IEEE, 2006. 22

[7] E. L. Andrade and R. B. Fisher. Simulation of crowd problems for computer vision. In *First International Workshop on Crowd Simulation*, volume 3, pages 71–80, 2005. 22

[8] P. Arbelez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–335, June 2014. 19, 58

[9] M. Arif and S. Daud. Counting of people in the extremely dense crowd. 2012. 15, 56

[10] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Interactive object counting. In *European Conference on Computer Vision*, pages 504–518. Springer, 2014. 16, 56

[11] S. Bandini, A. Gorrini, and G. Vizzari. Towards an integrated approach to crowd analysis and crowd synthesis: A case study and first results. *Pattern Recognition Letters*, 44:16–29, 2014. 88

[12] A. Bansal and K. Venkatesh. People counting in high density crowds from still images. *arXiv preprint arXiv:1507.08445*, 2015. 15, 55

[13] J. Barandiaran, B. Murguia, and F. Boto. Real-time people counting using multiple lines. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on*, pages 159–162. IEEE, 2008. 16, 20

[14] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994. 22

[15] Y. Benabbas, N. Ihaddadene, and C. Djeraba. Motion pattern extraction and event detection for automatic visual surveillance. *J. Image Video Process.*, 2011:7:1–7:15, Jan. 2011. 26

[16] Y. Benabbas, N. Ihaddadene, T. Yahiaoui, T. Urruty, and C. Djeraba. Spatio-temporal optical flow analysis for people counting. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 212–217. IEEE, 2010. 17, 20

[17] P. Berens and M. J. Velasco. The circular statistics toolbox for matlab. *MPI Tech. Rep.*, 184:1–21, 2009. 77

[18] D. M. Blei and P. I. Frazier. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, 12(Aug):2461–2488, 2011. 83

[19] L. Boominathan, S. S. Kruthiventi, and R. V. Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 640–644. ACM, 2016. 17, 56

[20] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part IV*, chapter High Accuracy Optical Flow Estimation Based on a Theory for Warping, pages 25–36. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. 74

[21] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. *Computer Vision–ECCV 2010*, pages 282–295, 2010. 81

[22] X. Cao, Z. Wang, Y. Zhao, and F. Su. Scale aggregation network for accurate and efficient crowd counting. In *European Conference on Computer Vision*, pages 757–773. Springer, Cham, 2018. 18, 57, 68

[23] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, July 2012. 19, 58

[24] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7, June 2008. 7, 15, 16, 56

[25] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008. 14, 21, 55, 66, 67

[26] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7, June 2008. 31, 32, 36

[27] A. B. Chan, M. Morrow, and N. Vasconcelos. Analysis of crowded scenes using holistic properties. In *Performance Evaluation of Tracking and Surveillance workshop at CVPR*, pages 101–108, 2009. 19, 57

[28] A. B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):909–926, May 2008. 14, 17, 26

[29] A. B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):909–926, 2008. 19, 57

[30] A. B. Chan and N. Vasconcelos. Bayesian poisson regression for crowd counting. In *2009 IEEE 12th International Conference on Computer Vision*, pages 545–551, Sept 2009. 21

[31] A. B. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *IEEE Transactions on Image Processing*, 21(4):2160–2177, April 2012. p, q, 15, 16, 56, 63, 64, 65, 67, 68

[32] A. B. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *Image Processing, IEEE Transactions on*, 21(4):2160–2177, 2012. 21

[33] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014. 45, 47, 48, 50

[34] D. Y. Chen and P. C. Huang. Visual-based human crowds behavior analysis based on graph modeling and matching. *IEEE Sensors Journal*, 13(6):2129–2138, June 2013. 27

[35] K. Chen, S. Gong, T. Xiang, and C. Change Loy. Cumulative attribute space for age and crowd density estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2467–2474, 2013. 69

[36] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In *In BMVC*. q, 8, 16, 56, 64, 69

[37] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In *BMVC*, volume 1, page 3, 2012. p, 15, 55, 63, 65

[38] A. M. Cheriyadat and R. J. Radke. Detecting dominant motions in dense crowds. *Selected Topics in Signal Processing, IEEE Journal of*, 2(4):568–581, 2008. 22, 84

[39] A. M. Cheriyadat and R. J. Radke. Detecting dominant motions in dense crowds. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):568–581, Aug 2008. 24, 74

[40] W. Chongjing, Z. Xu, Z. Yi, and L. Yuncai. Analyzing motion patterns in crowded scenes via automatic tracklets clustering. *China Communications*, 10(4):144–154, April 2013. 23

[41] W. Chongjing, Z. Xu, Z. Yi, and L. Yuncai. Analyzing motion patterns in crowded scenes via automatic tracklets clustering. *china communications*, 10(4):144–154, 2013. 84, 88

[42] Y. Cong, H. Gong, S.-C. Zhu, and Y. Tang. Flow mosaicking: Real-time pedestrian counting without scene-specific learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1093–1100. IEEE, 2009. 20

[43] D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento. A method for counting moving people in video surveillance videos. *EURASIP Journal on Advances in Signal Processing*, 2010:5, 2010. 15, 55

[44] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 19, 32, 42, 57

[45] A. C. Davies, J. H. Yin, and S. A. Velastin. Crowd monitoring using image processing. *Electronics & Communication Engineering Journal*, 7(1):37–47, 1995. 14, 55, 66, 67

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 41, 59

[47] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014. 19, 57

[48] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. 2009. 19, 57

[49] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *Int. J. Comput. Vision*, 51(2):91–109, Feb. 2003. 26

[50] G. Eibl and N. Brandle. Evaluation of clustering methods for finding dominant optical flow fields in crowded scenes. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec 2008. 74

[51] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 47

[52] M. Everingham, A. Zisserman, C. K. Williams, L. Van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorkó, et al. The pascal visual object classes challenge 2007 (voc2007) results. 2007. 45

[53] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 42

[54] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 19, 57, 58

[55] J. Ferryman and A. Shahrokni. Pets2009: Dataset and challenge. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–6. IEEE, 2009. p, q, 63, 64, 65, 66, 69

[56] L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht. Learning to count with regression forest and structured labels. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2685–2688. IEEE, 2012. 16, 56

[57] H. Fradi and J.-L. Dugelay. A new multiclass svm algorithm and its application to crowd density analysis using lbp features. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 4554–4558, Sept 2013. 30

[58] H. Fradi, X. Zhao, and J.-L. Dugelay. Crowd density analysis using subspace learning on local binary pattern. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pages 1–6, July 2013. 14, 30

[59] W. Ge and R. T. Collins. Marked point processes for crowd counting. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2913–2920. IEEE, 2009. 19, 57

[60] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 43, 59

[61] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 43, 50

[62] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014. 59

[63] R. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979. 14, 30, 33

[64] A. S. Hassanein, M. E. Hussein, and W. Gomaa. Semantic analysis for crowded scenes based on non-parametric tracklet clustering. In *IJCAI*, pages 3389–3395, 2016. 83

[65] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 41, 42

[66] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995. 24

[67] J. L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991. 23

[68] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 22, 60

[69] Y.-l. Hou and G. K. Pang. Automated people counting at a mass site. In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, pages 464–469. IEEE, 2008. 14, 55

[70] M. Hu, S. Ali, and M. Shah. Detecting global motion patterns in complex videos. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–5. IEEE, 2008. 22, 23

[71] M. Hu, S. Ali, and M. Shah. Learning motion patterns in crowded scenes using motion flow field. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–5, Dec 2008. 22, 24, 75

[72] Q. Hu, P. Wang, C. Shen, A. van den Hengel, and F. Porikli. Pushing the limits of deep cnns for pedestrian detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(6):1358–1368, 2018. 19, 57

[73] W. Hu, X. Li, G. Tian, S. Maybank, and Z. Zhang. An incremental dpmm-based method for trajectory clustering, modeling, and retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 35(5):1051–1065, 2013. 83

[74] S. Huang, X. Li, Z. Zhang, F. Wu, S. Gao, R. Ji, and J. Han. Body structure aware deep crowd counting. *IEEE Transactions on Image Processing*, 27(3):1049–1059, 2018. 18, 57, 68

[75] S. Huang and D. Ramanan. Expecting the unexpected: Training detectors for unusual pedestrians with adversarial imposters. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2017. 19, 57

[76] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2554. 15, 55

[77] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2554, 2013. 16

[78] N. Ihaddadene and C. Djeraba. Real-time crowd motion analysis. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008. 75

[79] M. Jalali Moghaddam, E. Shaabani, and R. Safabakhsh. Crowd density estimation for outdoor environments. In *Proceedings of the 8th International Conference on Bioinspired Information and Communications Technologies*, BICT '14, pages 306–310, ICST, Brussels, Belgium, Belgium, 2014. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 14, 30

[80] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 45, 59

[81] P.-M. Jodoin, Y. Benezeth, and Y. Wang. Meta-tracking for video scene understanding. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 1–6. IEEE, 2013. 83

[82] D. Kang, D. Dhar, and A. B. Chan. Crowd counting by adapting convolutional neural networks with side information. *arXiv preprint arXiv:1611.06748*, 2016. 17, 18, 56, 57

[83] S. Khan, G. Vizzari, S. Bandini, and S. Basalamah. Detecting dominant motion flows and people counting in high density crowds. *Journal of WSCG*, 22(1):21–30, 2014. 14, 55

[84] S. D. Khan, G. Vizzari, S. Bandini, and S. Basalamah. Detecting dominant motion flows and people counting in high density crowds. *Journal of WSCG*, 22(1):21–30, 2014. 75, 78

[85] P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos. Estimating pedestrian counts in groups. *Computer Vision and Image Understanding*, 110(1):43–59, 2008. 15, 55

[86] D. Kim, Y. Lee, B. Ku, and H. Ko. Crowd density estimation using multi-class adaboost. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 447–451, Sep 2012. 14, 31

[87] J. Kim and K. Grauman. Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2921–2928, June 2009. 26

[88] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1187–1190. IEEE, 2006. 14, 55

[89] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 1187–1190, 2006. 21

[90] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1446–1453, June 2009. 23, 25

[91] L. Kratz and K. Nishino. Spatio-temporal motion pattern models of extremely crowded scenes. In *Machine Learning for Vision-Based Motion Analysis*, pages 263–274. Springer, 2011. 23

[92] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 42, 43, 45

[93] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *null*, pages 878–885. IEEE, 2005. 19, 57

[94] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *Advances in neural information processing systems*, pages 1324–1332, 2010. 15, 55, 68

[95] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan. Scale-aware fast r-cnn for pedestrian detection. *IEEE Transactions on Multimedia*, 20(4):985–996, 2018. 19, 57

[96] M. Li, Z. Zhang, K. Huang, and T. Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008. 20

[97] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan. Crowded scene analysis: A survey. *IEEE transactions on circuits and systems for video technology*, 25(3):367–386, 2015. 82

[98] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(1):18–32, Jan. 2014. 26

[99] W. Li, J. H. Ruan, and H. A. Zhao. Crowd movement segmentation using velocity field histogram curve. In *2012 International Conference on Wavelet Analysis and Pattern Recognition*, pages 191–195, July 2012. 75

[100] Z. Lin and L. S. Davis. Shape-based human detection and segmentation via hierarchical part-template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):604–618, 2010. 19, 58

[101] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. o, 43, 45, 47, 48, 50, 51, 59

[102] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 41

[103] D. G. Lowe. Object recognition from local scale-invariant features. In *iccv*, page 1150. Ieee, 1999. 42

[104] C. Loy, S. Gong, and T. Xiang. From semi-supervised to transfer counting of crowds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2256–2263, 2013. 15, 20

[105] W.-C. Lu, Y.-C. F. Wang, and C.-S. Chen. Learning dense optical-flow trajectory patterns for video object extraction. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 315–322. IEEE, 2010. 81

[106] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981. 22

[107] P. Luo, Y. Tian, X. Wang, and X. Tang. Switchable deep network for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 899–906, 2014. 19, 57

[108] R. Ma, L. Li, W. Huang, and Q. Tian. On pixel count based crowd density estimation for visual surveillance. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 1, pages 170–173 vol.1, Dec 2004. 32

[109] W. Ma, L. Huang, and C. Liu. Advanced local binary pattern descriptors for crowd estimation. In *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, volume 2, pages 958–962, Dec 2008. 14, 30

[110] W. Ma, L. Huang, and C. Liu. Crowd estimation using multi-scale local texture analysis and confidence-based soft classification. In *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, volume 1, pages 142–146, Dec 2008. 21, 30

[111] W. Ma, L. Huang, and C. Liu. Crowd density analysis using co-occurrence texture features. In *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*, pages 170–175, Nov 2010. 21, 30

[112] W. Ma, L. Huang, and C. Liu. Estimation of crowd density using image processing. *Computer Sciences and Convergence Information Technology*, pages 170–175, 2010. 14, 55

[113] Z. Ma and A. Chan. Crossing the line: Crowd counting by integer programming with local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2539–2546, 2013. 17, 20

[114] Z. Ma and A. B. Chan. Crossing the line: Crowd counting by integer programming with local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2539–2546. 15, 55

[115] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1975–1981, June 2010. 25

[116] M. Manfredi, R. Vezzani, S. Calderara, and R. Cucchiara. Detection of static groups and crowds gathered in open spaces by texture classification. *Pattern Recogn. Lett.*, 44(C):39–48, July 2014. 14, 31, 34

[117] J. Mao, T. Xiao, Y. Jiang, and Z. Cao. What can help pedestrian detection? In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6034–6043. IEEE, 2017. 19, 57

[118] A. Marana, L. Costa, R. Lotufo, and S. Velastin. On the efficacy of texture analysis for crowd monitoring. In *Computer Graphics, Image Processing, and Vision, 1998. Proceedings. SIBGRAPI '98. International Symposium on*, pages 354–361, Oct 1998. 14, 20, 30

[119] A. Marana, L. da Fontoura Costa, R. Lotufo, and S. Velastin. Estimating crowd density with minkowski fractal dimension. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3521–3524 vol.6, Mar 1999. 14, 20, 30

[120] A. N. Marana, M. A. Cavenaghi, R. S. Ulson, and F. L. Drumond. *Advances in Visual Computing: First International Symposium, ISVC 2005, Lake Tahoe, NV, USA, December 5-7, 2005. Proceedings*, chapter Real-Time Crowd Density Estimation Using Images, pages 355–362. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. 14, 30, 33

[121] A. N. Marana, S. Velastin, L. Costa, and R. Lotufo. Estimation of crowd density using image processing. 1997. 14, 55, 66, 67

[122] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor. Fully convolutional crowd counting on highly congested scenes. *arXiv preprint arXiv:1612.00220*, 2016. 17, 56

[123] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *European Conference on Computer Vision*, pages 720–735. Springer, 2014. 44, 50

[124] R. Mehran, B. E. Moore, and M. Shah. *A Streakline Representation of Flow in Crowded Scenes*, pages 439–452. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. 24, 26

[125] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942. IEEE, 2009. 22, 23

[126] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942, June 2009. 24, 25, 26

[127] B. E. Moore, S. Ali, R. Mehran, and M. Shah. Visual crowd surveillance through a hydrodynamics lens. *Commun. ACM*, 54(12):64–73, Dec. 2011. 24

[128] S. M. Mousavi, S. O. Shahdi, and S. Abu-Bakar. Crowd estimation using histogram model classificationbased on improved uniform local binary pattern. *International Journal of Computer and Electrical Engineering*, 4(3):256, 2012. 14, 30

[129] T. Nawaz, A. Cavallaro, and B. Rinner. Trajectory clustering for motion pattern extraction in aerial videos. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 1016–1020. IEEE, 2014. 83

[130] M. Nedrich and J. W. Davis. Learning scene entries and exits using coherent motion regions. In *International Symposium on Visual Computing*, pages 120–131. Springer, 2010. 83

[131] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, Jul 2002. 14

[132] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, Jul 2002. 30, 33, 42

[133] D. Onoro-Rubio and R. J. López-Sastre. Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, pages 615–629. Springer, 2016. 17, 56

[134] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2056–2063, 2013. 19, 57

[135] W. Ouyang, X. Zeng, and X. Wang. Modeling mutual visibility relationship in pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3222–3229, 2013. 19, 57

[136] O. Ozturk, T. Yamasaki, and K. Aizawa. Detecting dominant motion flows in unstructured/structured crowd scenes. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3533–3536, Aug 2010. 74

[137] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3253–3261, 2015. 68, 69

[138] R. Raghavendra, A. Del Bue, M. Cristani, and V. Murino. *Abnormal Crowd Behavior Detection by Social Force Optimization*, pages 134–145. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 26

[139] H. Rahmalan, M. Nixon, and J. Carter. On crowd density estimation for surveillance. In *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, pages 540–545, June 2006. 14, 21, 30

[140] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016. o, 43, 45, 47, 48, 50, 51

[141] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017. 59

[142] H. Ren and T. B. Moeslund. Abnormal event detection using local sparse representation. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pages 125–130, Aug 2014. 27

[143] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. o, 19, 41, 43, 44, 45, 46, 47, 48, 49, 50, 51, 58, 59

[144] X. Ren. Finding people in archive films through tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. l, o, 46, 51

[145] M. Rodriguez, S. Ali, and T. Kanade. Tracking in unstructured crowded scenes. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1389–1396, Sept 2009. 12, 74

[146] M. J. Roshtkhari and M. D. Levine. An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. *Computer Vision and Image Understanding*, 117(10):1436 – 1452, 2013. 26

[147] E. Rosten and T. Drummond. *Machine Learning for High-Speed Corner Detection*, pages 430–443. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. 24

[148] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992. 61

[149] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 42

[150] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *Digital Image Computing: Techniques and Applications, 2009. DICTA'09.*, pages 81–88. IEEE. 15, 56

[151] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *Digital Image Computing: Techniques and Applications, 2009. DICTA'09.*, pages 81–88. IEEE, 2009. 21

[152] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Scene invariant multi camera crowd counting. *Pattern Recognition Letters*, 44:98–112, 2014. 14, 20

[153] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Scene invariant multi camera crowd counting. *Pattern Recognition Letters*, 44:98 – 112, 2014. Pattern Recognition and Crowd Analysis. 16, 56

[154] D. Ryan, S. Denman, S. Sridharan, and C. Fookes. An evaluation of crowd counting methods, features and regression models. *Computer Vision and Image Understanding*, 130:1–17, 2015. 69

[155] D. B. Sam, S. Surya, and R. V. Babu. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 6, 2017. 17, 18, 56, 68

[156] M. Saqib, S. D. Khan, and M. Blumenstein. Texture-based feature mining for crowd density estimation: A study. In *Image and Vision Computing New Zealand (IVCNZ), 2016 International Conference on*, pages 1–6. IEEE, 2016. 19, 29, 57

[157] M. Saqib, S. D. Khan, and M. Blumenstein. Detecting dominant motion patterns in crowds of pedestrians. In *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*, volume 10225, page 102251L. International Society for Optics and Photonics, 2017. 73

[158] M. Saqib, S. D. Khan, N. Sharma, and M. Blumenstein. Extracting descriptive motion information from crowd scenes. In *2017 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2017. 81

[159] M. Saqib, S. D. Khan, N. Sharma, and M. Blumenstein. Person head detection in multiple scales using deep convolutional neural networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2018. 29

[160] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3):271–304, 2005. 22

[161] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang. Crowd counting via adversarial cross-scale consistency pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5245–5254, 2018. 18, 57, 68

[162] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, Jun 1994. 24

[163] C. Shiyao, L. Nianqiang, and L. Zhen. Multi-directional crowded objects segmentation based on optical flow histogram. In *Image and Signal Processing (CISP), 2011 4th International Congress on*, volume 1, pages 552–555. IEEE, 2011. 75

[164] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 42, 45, 47, 48, 49, 50, 59, 64, 67

[165] V. A. Sindagi and V. M. Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1879–1888. IEEE, 2017. 18, 57

[166] B. Solmaz, B. E. Moore, and M. Shah. Identifying behaviors in crowd scenes using stability analysis for dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2064–2070, 2012. 25, 82, 84, 85

[167] B. Solmaz, B. E. Moore, and M. Shah. Identifying behaviors in crowd scenes using stability analysis for dynamical systems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(10):2064–2070, Oct. 2012. 26

[168] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):747–757, Aug. 2000. 14

[169] V. B. Subburaman, A. Descamps, and C. Carincotte. Counting people in the crowd using a generic head detector. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 470–475. IEEE, 2012. 19, 20

[170] J. Sun, Y. Mu, S. Yan, and L.-F. Cheong. Activity recognition using dense long-duration trajectories. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 322–327. IEEE, 2010. 83

[171] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2004–2011. IEEE, 2009. 83

[172] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 42

[173] Y. Tian, P. Luo, X. Wang, and X. Tang. Pedestrian detection aided by deep learning semantic tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5087, 2015. 19, 57

[174] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 19, 42, 58

[175] H. Ullah and N. Conci. Crowd motion segmentation and anomaly detection via multi-label optimization. In *ICPR workshop on Pattern Recognition and Crowd Analysis*, 2012. 75

[176] H. Ullah, M. Ullah, M. Uzair, and F. Rehman. Comparative study: The evaluation of shadow detection methods. *International Journal Of Video & Image Processing And Network Security (IJVIPNS)*, 10(2):1–7, 2010. 16, 56

[177] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60(2):79–87, 1988. 60

[178] T. Van Oosterhout, S. Bakkes, and B. J. Kröse. Head detection in stereo data for people counting and segmentation. In *VISAPP*, pages 620–625, 2011. 20

[179] J. J. Van Wijk. Image based flow visualization. *ACM Transactions on Graphics (ToG)*, 21(3):745–754, 2002. 23

[180] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *null*, page 734. IEEE, 2003. 19, 57

[181] T.-H. Vu, A. Osokin, and I. Laptev. Context-aware cnns for person head detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2893–2901, 2015. l, o, 40, 43, 46, 47, 50, 51

[182] E. Walach and L. Wolf. Learning to count with cnn boosting. In *European Conference on Computer Vision*, pages 660–676. Springer, 2016. 17, 18, 56, 68, 69

[183] B. Wang, M. Ye, X. Li, F. Zhao, and J. Ding. Abnormal crowd behavior detection using high-frequency and spatio-temporal features. *Machine Vision and Applications*, 23(3):501–511, 2012. 25

[184] C. Wang, X. Zhao, Z. Wu, and Y. Liu. Motion pattern analysis in crowded scenes based on hybrid generative-discriminative feature maps. In *2013 IEEE International Conference on Image Processing*, pages 2837–2841, Sept 2013. 23

[185] C. Wang, X. Zhao, Z. Wu, and Y. Liu. Motion pattern analysis in crowded scenes based on hybrid generative-discriminative feature maps. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 2837–2841. IEEE, 2013. 83

[186] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009. 19, 57

[187] X. Wang and W. Ouyang. A discriminative deep model for pedestrian detection with occlusion handling. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3258–3265. IEEE, 2012. 19, 57

[188] Y. Wang, H. Lian, P. Chen, and Z. Lu. Counting people with support vector regression. In *Natural Computation (ICNC), 2014 10th International Conference on*, pages 139–143. IEEE, 2014. 14, 55

[189] Z. Wang, H. Liu, Y. Qian, and T. Xu. Crowd density estimation based on local binary pattern co-occurrence matrix. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, pages 372–377, July 2012. 14, 30, 33

[190] Z. Wang, H. Liu, Y. Qian, and T. Xu. Crowd density estimation based on local binary pattern co-occurrence matrix. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, pages 372–377, July 2012. 21

[191] J. R. Williamson. Gaussian artmap: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, 9(5):881 – 897, 1996. 24

[192] B. Wu, F. N. Iandola, P. H. Jin, and K. Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *CVPR Workshops*, pages 446–454, 2017. 59

[193] S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2054–2060. IEEE, 2010. 23

[194] S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2054–2060, June 2010. 25

[195] S. Wu, H. S. Wong, and Z. Yu. A bayesian model for crowd escape behavior detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(1):85–98, Jan 2014. 27

[196] S. Wu, H. Yang, S. Zheng, H. Su, Y. Fan, and M.-H. Yang. Crowd behavior analysis via curl and divergence of motion trajectories. *International Journal of Computer Vision*, 123(3):499–519, 2017. 25

[197] S. Wu, Z. Yu, and H.-S. Wong. Crowd flow segmentation using a novel region growing scheme. In P. Muneesawang, F. Wu, I. Kumazawa, A. Roeksabutr, M. Liao, and X. Tang, editors, *Advances in Multimedia Information Processing - PCM 2009*, volume 5879 of *Lecture Notes in Computer Science*, pages 898–907. Springer Berlin Heidelberg, 2009. 74

[198] X. Wu, G. Liang, K. K. Lee, and Y. Xu. Crowd density estimation using texture analysis and learning. In *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, pages 214–219, Dec 2006. 21, 30

[199] L. Xiaohua, S. Lansun, and L. Huanqin. Estimation of crowd density based on wavelet and support vector machine. *Transactions of the Institute of Measurement and Control*, 28(3):299–308, 2006. 14, 19, 55, 57

[200] L. Xiaohua, S. Lansun, and L. Huanqin. Estimation of crowd density based on wavelet and support vector machine. *Transactions of the Institute of Measurement and Control*, 28(3):299–308, 2006. 21, 30

[201] F. Xiong, X. Shi, and D.-Y. Yeung. Spatiotemporal modeling for crowd counting in videos. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 5161–5169. IEEE, 2017. 18, 57, 68, 69

[202] G. Xiong, X. Wu, Y. L. Chen, and Y. Ou. Abnormal crowd behavior detection based on the energy model. In *Information and Automation (ICIA), 2011 IEEE International Conference on*, pages 495–500, June 2011. 26

[203] H. Xu, P. Lv, and L. Meng. A people counting system based on head-shoulder detection and tracking in surveillance video. In *Computer Design and Applications (ICCDA), 2010 International Conference on*, volume 1, pages V1–394. IEEE, 2010. 20

[204] H. Xu, Y. Zhou, W. Lin, and H. Zha. Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4328–4336, 2015. 83

[205] H. Yang, Y. Cao, S. Wu, W. Lin, S. Zheng, and Z. Yu. Abnormal crowd behavior detection based on local pressure model. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–4. IEEE, 2012. 27

[206] H. Yang, H. Su, S. Zheng, S. Wei, and Y. Fan. The large-scale crowd density estimation based on sparse spatiotemporal local binary pattern. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6, July 2011. 14, 30

[207] Y. Yang, J. Liu, and M. Shah. Video scene understanding using multi-scale analysis. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1669–1676, Sept 2009. 23

[208] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 42, 45, 47, 48, 50, 59, 64, 67

[209] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015. 17, 56, 68

[210] J. Zhang, B. Tan, F. Sha, and L. He. Predicting pedestrian counts in crowded scenes with rich and high-dimensional features. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1037–1046, 2011. 14, 55, 66, 67

[211] L. Zhang, L. Lin, X. Liang, and K. He. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, pages 443–457. Springer, 2016. 19, 57

[212] S. Zhang, C. Bauckhage, and A. B. Cremers. Informed haar-like features improve pedestrian detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 947–954, 2014. 19, 57

[213] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Occlusion-aware r-cnn: Detecting pedestrians in a crowd. *arXiv preprint arXiv:1807.08407*, 2018. 19, 57

[214] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016. 17, 56, 68

[215] T. Zhao, R. Nevatia, and B. Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE transactions on pattern analysis and machine intelligence*, 30(7):1198–1211, 2008. 19, 57

[216] B. Zhou, X. Wang, and X. Tang. Random field topic model for semantic region analysis in crowded scenes from tracklets. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3441–3448, June 2011. 23

[217] B. Zhou, X. Wang, and X. Tang. Random field topic model for semantic region analysis in crowded scenes from tracklets. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3441–3448. IEEE, 2011. 82

[218] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2871–2878, June 2012. 23

[219] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2871–2878. IEEE, 2012. 82, 83

[220] L. Zhu, Y. Chen, and A. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *IEEE transactions on pattern analysis and machine intelligence*, 32(6):1029–1043, 2010. 19, 58

[221] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014. 19, 43, 58