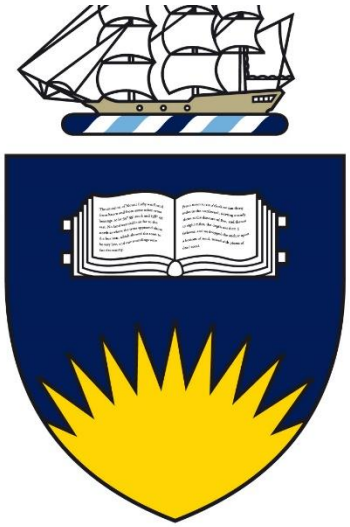

Flinders
UNIVERSITY

School of Computer Science, Engineering and Mathematics

Crowd Counting Through Head Detection

Submitted by

Manoj Jayaram

2131148

Studying Master in Biomedical Engineering

Supervised by Paul Gardner-Stephen

Submitted to the School of Computer Science, Engineering and Mathematics in the Faculty of Science and Engineering in partial fulfilment of the requirements for the degree of Master in Engineering (Biomedical) at Flinders University-Adelaide, Australia

Submitted on June 10th. 2016

Disclaimer

I certify that this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Manoj Jayaram

Date: 24/08/2016

Acknowledgement:

I would like to express my sincere thanks to my supervisor Paul Gardner-Stephen without whom I could never have accomplished my project. I am very thankful to all members of engineering services for their support. My gratitude to Finders University for providing me all the guidelines to accomplish my work. Special thanks to my uncle, aunt and parents who believed in me and finally to all my professors, lecturers and friends who directly or indirectly help me finish my project.

Abstract

When large numbers of people gather at a specific place or location for a specific purpose, it is defined as Mass Gathering. It is important to understand the event variables that might affect the health of people present during such mass gatherings. Crowd density is one of the most important variables to be considered during mass gatherings. Crowd density is defined as the number of people per unit area. The uncontrolled crowd density might lead to injury or illness, depending on the number of people. Hence it is important to keep track of the number of people present in a given area.

Devices used to count the total number of people in a given location are known as people counters. This type of system plays a vital role in recording the total number of visitors. Different types of sensors can be used for this purpose. Video cameras are one of the most accurate and reliable sensors that can be used for crowd calculation.

A significant drawback of video cameras is that they are heavy and expensive as compared to other potential sensors, and that they often require complex calibration in order to obtain usable results. A further undesirable aspect of high-resolution cameras is that they can compromise the privacy of the individuals who are being counted. Together, these factors have led us to the conclusion that there is value in creating crowd counting solutions that can use low-resolution cameras that are small, cheap, and sufficiently low-resolution that they represent no realistic risk to the privacy of the people being counted, even if the performance does not match the best results of the existing state of the art.

The primary goal of this study is to attempt to measure crowd density (people per area). As this is a complex challenge, we are concentrating on first step of it for now: counting people in a specific example context. Generalising the findings to measuring crowd density in general, while an admirable goal, is beyond the scope of this thesis. Within this study, we aim to make a prototype on MATLAB for the real time counting of people by using a video processing technique that is designed to be robust enough that it can work with low-resolution cameras, and simple enough that it can operate in real-time using small low-cost embedded processors. Thus, this study focuses on identifying and prototyping a suitable computationally affordable and feasible

solution, rather than advancing the state of the art in image processing through the generation of novel image processing algorithms, per se.

The method explored in this study uses neural network based regression analysis, and does not require any calibration of the camera. The image is obtained from a streaming camera in real time. Different experiments were conducted and the results proved that the method proposed in this study is intrinsic in its simplicity and powerful in its accuracy. It has the capability to give good results even when the quality of the camera is low and therefore it does not unnecessarily compromise with the privacy of people being counted.

Table of Contents

List of Figures

1 Introduction:

This thesis forms a component of the project 'Environmental Monitoring at events'. The aim of this project is to understand the relation between event variables and patient presentations during mass gatherings (Cassidy, Dix, and Jenkins, 1983). The rate of patient injury or illness can be minimized by improving the knowledge of the key features of events. Mass gatherings can be understood as an interrelationship between the biomedical, environmental and psychosocial domains (Arbon P, 2004). Our aim here is to collect the data of different key variables at mass gatherings that may influence Patient Presentation Rate (PPR) and Transport to Hospital Rate (THR). The data were collected using various electronic sensors, as well as photography, and manual collection methods that were deployed at events held across Australia during 2015/2016. The data include crowd density, temperature, wind-speed, humidity and crowd mobility.

In the current era, the number and variety of mass gatherings in society continues to grow (Arbon P, 2004). Mass gatherings can happen due to various social events like sport, concert, festivals and protests. Mass gatherings are unpredictable, as it can be more hazardous than expected and can result in the occurrence of massive injury and illness. During mass gatherings several factors like weather, event duration, crowd density, seating arrangements, use of alcohol and drugs, and the location of the events such as indoor/outdoor event influence the rate of injury and illness (Franaszek J, 1986). Hence, it is important to observe and keep in check all these factors to keep injury rates at a minimum. From all of the factors mentioned above, this thesis focuses on a single variable: crowd density. Crowd density is defined as the total number of people present in a given area. As accurately measuring crowd density for a given area is complex, we need to count the number of people and that specific area of the place where gathering occurs to measure the crowd density. Therefore, as a first step, this thesis focuses on counting the number of people in a given place, using only low-cost camera hardware, and relatively computationally simple algorithms, to ascertain the feasibility of measuring crowd density in this way.

The capability to be acquainted with the knowledge of the total number of people present in a place has always come up with the interest in different applications such as health, education,

transportation, robotics and commerce. To obtain such type of data and information from a source that is reasonably cost effective, efficient and easy to use is a challenge for engineers of the current era. Different sensor systems can be used for this purpose. In most of the cases they achieve almost the same results, but cost of the system varies. This kind of information can be useful for different purposes such as mass gathering events, tourists count estimation and security. The use of video cameras for the purpose of counting or tracking the people in a crowd has increased significantly over the past few years. The major reason for the rapid increase in the usage of this technique is the advancement in computer technology and algorithms of image processing. Many people used different techniques to count or track people (Damien Lefloch, 2008). Based on their complexities these methodologies can be classified into following three categories

- To improve the method of region tracking based on the texture and colour features of pixels for some scheme of classification.
- By using different human models and by incorporating 2D human appearances (Anon., 2006).
- By making a 3D model by using multiple cameras (John Krumm, July 1, 2000,).

While comparing the above methodologies it can be seen that the 3D model using multiple cameras gives more accurate results compared to the other two because it can rebuild the scene more accurately and remove occlusion problems. It also has a drawback of using a complex algorithm that is difficult to implement. On some occasions, this system requires a calibrated complex camera and it is unable to operate in real-time because the 3D models are very slow. To overcome these problems, the proposed strategy in this study uses the other two categories.

1.1 Motivation

The below table contains the list of some disasters happened during Mass gatherings. We can observe that in most of the events overcrowding was the primary reason. This may be due to overselling of tickets, more people trying to gain access or more people entering after the start of the event. In order to avoid these issues the crowd must be monitored continuously and immediate action should be taken as soon as the crowd density level exceeds the required level.

Date and place	Event	Disaster	Casualties
May 1985 Ibrox stadium, Bradford, UK	Football Match	Overcrowding at the barriers	66 Deaths and 766 injured (Name 2016)
July 1990 Mecca, Saudi Arabia	The Haj	Overcrowding and rush towards exit resulting in a stampede	1400 Deaths (History.com, 2009)
May 1992 Bastia, Corsica	Football match	Overcrowding of temporary stand which resulted in the collapse	17 Deaths and 1900 injured (Inc, 2012)
October 1998 Gothenburg, Sweden	Night club	Overcrowding at the venue. Stairway fire at emergency exit	63 Deaths and 213 injuries (Summers, 2011)
June 2000 Roskilde, Denmark	Music festival	Overcrowding and crowd surge	9 Deaths (Fricke, 2000)
Jan 2001 Sydney, Australia	Music festival	Overcrowding and surge at front stage	1 Death (Vincent, 2013)
April 2001 Johannesburg, South Africa	Football matches	Overcrowding	47 Deaths (Inc, 2012)
Feb 2004 Miyun District, China	Firework event	Overcrowding has resulted in a stampede	37 Deaths and 24 injured (McDonell and wires, 2015)
March 2010 Uttar Pradesh, India	Temple	High crowd density Stampede	60 Deaths (Shah Singh, 2010)
July 2010 Duisburg, Germany	Love parade Music festival	High crowd density Stampede	21 Deaths and 510 injured (Connolly, 2010)

Table 1: Mass gathering disasters due to overcrowding

Counting group of people and the estimation of density contributes to the crowd management and safety observation. This type of system can be deployed at event to observe the crowd and to check the number of people present at the event. It is also used by the Law enforcement agencies to detect unusual activities in a mob (Saleh, Suandi, and Ibrahim, 2015). Such systems are also useful for finding the total number of travellers, which is a key factor in the planning the development of public infrastructures.

Similarly, if we look at some advantages of this system, it can be used to measure social impact and importance of political rallies. For the sake of calibration of video camera, the person should have access to the camera model specification. In some cases, this could be a problem if the model of the camera is unknown. That's why in this thesis we focus on a methodology in which we do not have to calibrate the video camera.

Confidentiality of the general public is also an important theme in this thesis. When we talk about the technology that involves the video recording of people, it often generates concern and anxiety within the general public. Another problem with the video streaming technique is that it requires high bandwidth. This problem can be mitigated when we are able to detect the number of persons on low quality images collected from the videos. The use of low-resolution images solves both problems simultaneously: bandwidth requirements (and hardware cost) are reduced, while the possibility of compromising privacy through the identification of specific persons in the imagery is also mitigated.

Furthermore, by concentrating on the procedure or algorithm that needs a low speed and by using a lone camera for each site, this methodology could open the possibility to implement on embedded systems such as Field Programmable Gate Array (FPGA's), thus helping to satisfy the desire to create a low-cost and affordable system that has the capability to count the total number of people on the site and send or transmit the runtime statistical data of the crowd to the administrator. This thesis is motivated by the opportunity to avoid the difficulties with the run time video process in counting the total number of people. Also to make system which is fast, economical and can be used in different environments.

1.2 Problem Description

The number of people present in a given area often varies constantly during a mass gathering. It is important to find the number of people present in a given place at a given time. Nowadays, plenty of research has been published to solve this type of problem, which is “Counting or tracking people using video camera”. This task is not as simple as it seems, there are some scenarios that are very difficult to resolve even today, even with the high-speed computers. This is because that algorithm defines the boundaries of difficulty of the approach of tracking and detection by operating in real time. Counting gets easier if there are no disturbances present in a given place. The disturbances can be a tree, light pole or a bench. If the field of view is plain, like a soccer field which does not have any disturbances, then it gets easier to keep track of all the people present in the field.

1.3 Research Aims and Contributions

The main aim of the current study is to demonstrate a low-cost, computational and relatively simple method that could use low-resolution cameras to count the number people in a crowd, thus designing a system taking into account the privacy concerns of the people, as well as an economic imperative. We have developed a prototype on MATLAB for the real time counting of people by using image processing. This technique is really simple as it can even work using low-resolution cameras. The method proposed in this study does not require any calibration of the camera. The number of people present in the image was predicted by Neural Network regression.

All images were taken from the top with an angle close to 90 degrees and the camera was held in the hand. We used images which contain people, ranging from one till 5 in numbers. Some images were used to train the neural network and others were used for prediction. This technique can also be used to predict a larger number of people by using database consisting of more people.

This system can be incorporated in places like markets, parks, hotels and malls. Instead of using people detection or people tracking to avoid privacy issue, we have used a regression-based method which predicts the number of people present in an image without exposing their identity. This is a cost effective and an efficient approach. Low resolution images contribute to the cost effectiveness of this project and will reduce the overall budget of project installation and maintenance.

1.4 Research Questions

Is it possible to develop a low cost, crowd counting system? If so:

- Can it be used to count the number of people in real-time?
- Can that system work in areas with a low-density of people?
- Does that system protect the privacy of the people counted?
- Can the system use only low-cost light-weight camera equipment, such as internet cameras or smart-phones?
- Can the system be easily realised as a simple to use system?

1.5 Thesis structure

Chapter 2 provides the information regarding the previous works done on counting people. It explains the advantages and disadvantages and why there is need for new system to be developed and the difference between the previous works and our intended work.

Chapter 3, 4 and 5 describes in detail about the methodology, features, mode of training data, algorithms used and application of neural networks to count the number of people present in the image.

Chapter 6 describes the method of obtaining the datasets and the difficulties faced in obtaining it. It also includes the information regarding the device used for capturing the data and how the data was divided for training and testing.

Chapter 7 describes how head counting is done using neural network and results obtained on testing data

Chapter 8 describes the applications of this system in the real world followed by Chapter 9 which concludes this thesis with the advantages and disadvantages of this system followed by the future work on how the system can be developed further more accurate results.

2 Literature Review

This chapter we will demonstrate the research and background to the technologies of counting people. Since this thesis requires the counting of heads without tracking and any kind of face detection, we will try to restrict the research to the narrow field of head counting. We will start with the research in the area of head counting techniques to the functional technologies adapted to this area. We will also discuss the research in regression based techniques used in head counting. The existing research provided helps to draw out the significance of the solution proposed in our system which is stated in detail after the shortcomings of these existing systems are cited.

2.1 Existing Work

2.1.1 Face Detection and Motion Counting

Sweeney and Gross (2005) proposed a solution for motion detection and counting of people. They used publicly available, inexpensive cameras. Each camera has a database for capturing images and face detection mechanism is applied to count the number of faces. The high and low swarm of people is estimated through time stamp ***t*** on each image ***X*** by counting the number of faces in each image The system detects an unusual number of masses in a location with the help of webcams. Counting the number of faces has imprecise results than counting the number of heads as the backs of people are more distinct in the image.

2.1.2 Pedestrian Counter

Pedestrian counter is a basic strategy that can be used by many applications. For every person, counting the number of heads in a constantly waving swarm of people is different. Lin, Chen, and Chao (2001) proposed an idea that was an alternative of judgement every person has in cloud density. The approach used wavelet templates and vision-based techniques. It could perform counting heads in any background without any previous image reference. The information that characterizes the contour of the head was evaluated through wavelet template and used (Gavin, 2016). That information is processed by a classifier SVM. The resulting output that is detection performance after being processed by these two techniques is shown to be acceptable. The further lack of detection is compensated by a vision-based technique that makes use of the size

and positions of the detected frames. However the estimating results are not quite accurate due to the kernel factor. And though it has been claimed to count number of heads without reference image, but an image sequence is needed in order to count the number of heads in case of a moving crowd.

Pedestrian counters are categorized on the basis of sensors like ultrasonic sensors, infrared sensors, laser scanners, video cameras, piezoelectric sensors, microwave radar etc. Each of these has its own advantages, usages and downsides. Beam break principle is a most commonly used technique used to detect motion. A beam of IR light is sent to the receiver across the way which is sensitive to that same light. When something passes between the emitter and receiver, and it is not transparent to IR, then the 'beam is broken' and the receiver will let you know. In the past, the pedestrian's counters usually used turnstiles or gate type counters which are mechanical counters.

2.1.3 Regression methods

Two Bayesian regression methods were compared in (Chan and Vasconcelos, 2012). Estimating the approximate size of the homogeneous crowds has been approached by using the mixture of dynamic-texture motion model. The lower level features have been extracted from each segmented area and through Bayesian regression estimate of the number of people per area has been made. Two Bayesian regression models are analysed in which the first includes the Gaussian Process Regression (GPR) (which has a limited real-valued outputs which does not match the discrete counts) and Bayesian Poisson Regression (BPR) (which comprises of prior distribution on model's linear weights. These two crowd counting regressions-based methods are compared and evaluated on a large pedestrian data sets with very unique pedestrian traffic and camera views and outlines.

- In less dense condition GPR is more accurate than BPR
- In most dense condition BPR is more accurate than GPR

There is a limitation for crowd counting in the regression mentioned above which is for each specific viewpoint, it requires training. This training requirement hinders the ability of the crowd counting system during parades. It includes several factors:

- Due to motion, there are segment changes.
- Change in the appearance of dense crowds.
- Changes in persons viewing due to wearing heels.

The future works include viewpoints trainings, performance of Bayesian counting from sparse crowds.

2.1.4 Texture based method

Chan, Liang, and Vasconcelos (2008) proposed a texture-based method using contiguous region of coherent motion for counting moving people in videos. It also addresses the issue about the maintenance of privacy in tracking people for the purpose of counting. Some studies estimate this relationship between density and low-level features or counting head by training a regression model. These models work in global form as they learn a single regression function for the whole dataset of images or videos. But this strategy takes into account an assumption that all images have same density that is not true for most of the images. However, to deal with this aforementioned problem, the regression models work in local pattern as they divide the image into the cell and perform regression analysis on each cell separately as seen in Figure 1. After the calculation of density is done, the counting process is performed as shown in Figure 2.

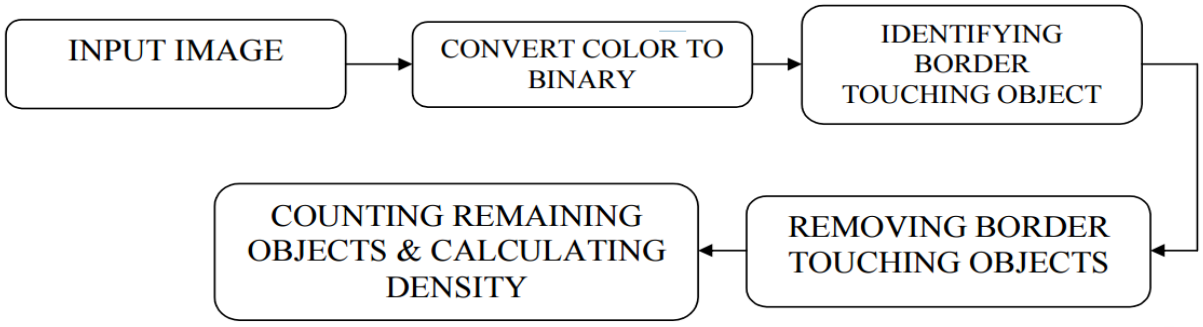

Figure 1: Block Diagram of cell count (Chan, Liang and Vasconcelos, 2008)

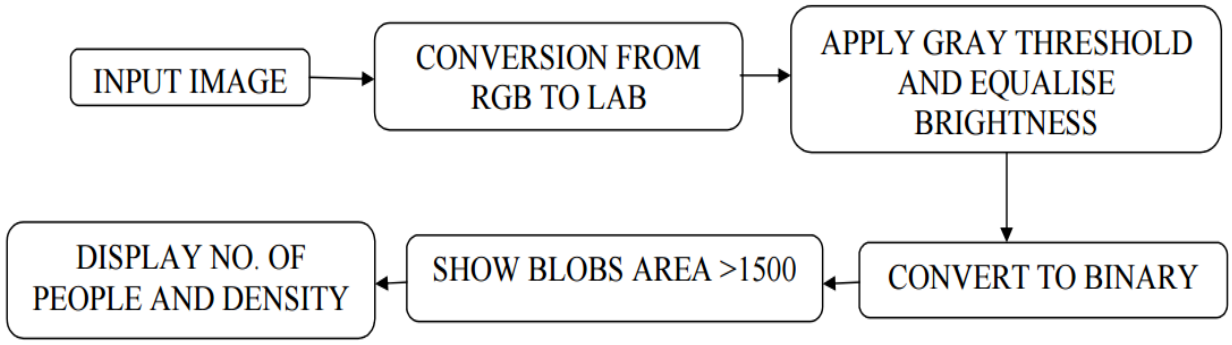

Figure 2: Block diagram of people count (Chan, Liang, and Vasconcelos, 2008)

Aziz et al. introduced a new technique which uses skeleton graph in order to differentiate and count people. Here, a single camera is used in a busy environment for head detection. After foreground estimation for each blob, the skeleton graphs are calculated which are present in the scene. To forecast a people's number and for head detection we sightsee the structural property of each blob. As independent state or partly blocked state each detected head in a skeleton shape is recognized and during tracking every state is updated (Aziz et al., 2016).

2.1.5 Head Counting Based on Feature-Based Regression

In a large and heavy crowd, precise and accurate counting becomes necessary as individual detection and tracking become impossible. Feature-based regression is another approach adopted by many researchers in intelligent crowd counting techniques. We discuss some of the techniques here:

A regression model will be used to estimate the number of people in the input image from the feature extracted image (Yeung Cho, et al., 1999). Some widely used regression models are linear, piecewise linear and neural network (Reis, 2014).

The basic framework of these models is:

1. Feature Extraction

2. Background subtraction

3. Crowd density estimation or count by a regression function

Kong and Gray (2006) introduced a view-point dependent invariant system to calculate the number of pedestrians, which can be easily deployed with some minor setup arrangements. The first step consists of feature extraction and normalization of images. It includes foreground region retrieval using background subtraction algorithm and by using an edge detector technique on the edges. An edge placement map is also being generated.

In another study based on the same domain, Stauffer and Grimson used foreground mask and adaptive mixture modelling for the background of each frame, in order to calculate a blob size histogram. The main assumption used to estimate the density of that image is that all the pedestrians in each frame must have the same size and lie on the same plane of horizontal ground. Hence, by linking this histogram, with the histogram obtained through edge alignment give better results in the form of cleaner image, as shown in Figure 3. After that, they estimate the Region of Interest (ROI) and its relative weights density map with the help of homograph related to calculation of relative pixel density. Moreover, this algorithm is also used to estimate

feature normalization, so that it gives the values of those features that are almost independent of the different camera view points and movements of the pedestrians on the ground plane. Training of the model is based on an offline neural network system that finds the possible relationship between the features and the total number of pedestrians in the given image. These experiments were conducted using two cameras of different specifications mounted at two different locations and different sites. The reliability and accuracy of the system provide strong evidence in the favour of used methodology of feature histogram instead of blob features and raw edges. By using improved computers the performance of such a system can be enhanced along with using a full covariance matrix and addition of prediction of each Gaussian will definitely lead to more robust tracking of lighting changes.

Figure 3: Original image and its mask

Kilambi et al. in their study in 2008 proposed a strategy for an algorithm to count number of people in group ignoring their individual recognition. A flow chart describing the basic outline of proposed algorithm is shown in Figure 4. The first step is the segmentation of the foreground region by using an adaptive mixture Gaussian method proposed by Atev, et al. The objects in real world ground plane those are smaller than human beings are filtered and removed through predictions. After removal of background small objects, in order to classify remaining regions as either group, individual people or vehicles foreground regions are being tracked by using the EKF pedestrian tracker method suggested by Masoud and Papanikolopous (2001). The proposed system cannot help read the stationary people in a scene. It needs to be extended accordingly. Methods to improve the accuracy near the horizon regions need to be investigated by using weighted estimates.

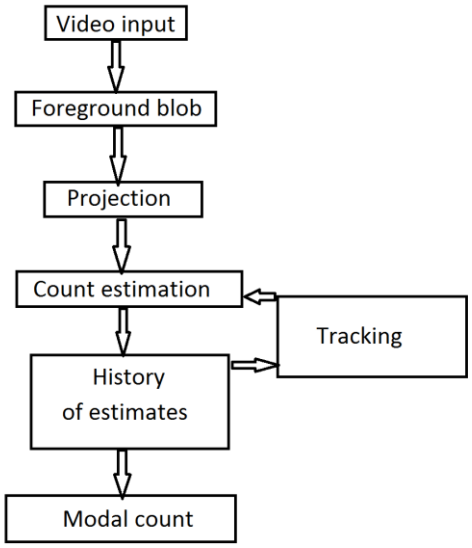

Figure 4: Flowchart of above explained method (Vasco Reis. 2014)

To estimate the number of people in a tracked groups two well-known approaches have been used. The first method used is Heuristic –based: in the combination with previous training algorithm, it uses the area covered by the projections on the ground plane. This strategy provides an efficient, simple and accurate solution for group counting. However, it is not as efficient as in the cases when size of the group differs from fixed head plane value of 160 cm and for the variable dynamics and the configuration of the group.

The second used method is shaping dependent, which uses the shape of the group's intersection with the ground along with the head plane projections and a function for cost minimization and to estimate the overall shape of the group.

Moreover, their system also deals with the splitting and merging of groups. This experiment was performed in real time, on P4 (Pentium IV 3.0 GHz PC) with different cameras at different heights and angles, along with different light intensities and locations. The problem occurred during the estimation of groups in the area away from the camera. This problem can be minimized by changing the ROI domain to ignore the area outside that specific range.

Both applied methods give equally efficient results, however, shape based estimator found to be more accurate as compare to the heuristic based method. For group estimation the error rate for heuristic method was 11.9% and for shape based method average error rate was 10.9%.

However, in terms of cost and time, heuristic method is less expensive and work with higher processing speeds.

To estimate the size of the crowd Chen et al. presented a privacy protection system to estimate the crowd size as privacy involving humans, is a collective issue in vision systems. This developed system is independent of object detection of feature extraction and doesn't generate a visual record of the people who are in groups in the current scene. Instead, they used the mixture of dynamic textures for segmentation of crowd that is moving in different ways (Chan and Vasconcelos, 2008). They also consider the effects of perception by linear insertion between two extremes of the scene before extracting features from segmented ROI.

The extracted values of segmented features obtained are as follows:

- Perimeter
- Area
- Perimeter-area ratio
- Perimeter edge alignment
- Interior edge features including edge orientation, texture features like homogeneity, Minkowski dimensions and total edge pixels.

Feature vector reversion based on Gaussian process has been used to estimate the number of people per segment. In this experiment a dataset of 49,885 pedestrian has been used. This system has been trained on 800 training frames and being tested on 1200 testing frames. The results of counting shows a deviation of 3 people from the ground-truth with 91% where the crowd is moving away from the camera and deviation of 2 people with 98% where the crowd is moving towards the camera. In addition to their study, they also highlighted the significance of multiple feature subsets for improving the overall performance of the developed system.

In 2013, Ryan et al. designed a scene-invariant crowd counting algorithm which uses local features to observe the size of the crowd (f. g.5). The uniqueness of their work was that they

scaled the solution of this problem to different environment and the turning it as scene invariant as compared to the other methods. They trained the system on more than one view point using camera calibration, that make the system intelligent enough to work for any new camera without any further training. Moreover, Depnman (2009) in his study on the same problem, used a method in which foreground segmentation method proposed by functions in the YCbCr 4:2:2 colour space and gave some invariance to the change in light intensity. In this method, camera calibration has been done to reimburse for the changing position of the camera before extracting features from the segmented regions. Local features like perimeter, area, perimeter and HOG have been calculated to find the approximation of the number of people in the group. At the end they adopted a Gaussian regression process to estimate the crowd density.

2.2 Drawbacks in Existing systems

From the above discussed methods, most of the head counting methods are dependent upon detection (Sweeney and Gross, 2005) (Lin, Chen, and Chao, 2001) (Gavin, 2016) - either face detection or body detection. Some methods are highly sensitive to light condition or other constraints (Conte et al., 2010). Some methods work perfectly fine for small gatherings, but when the crowd size becomes large, they give misdetections and hence wrong count (Gupta, Gupta, and Tiwari, 2011) (Singh et al., 2003). Some methods are based on person tracking or tracing, which is again not a favourable constraint for massive gatherings.

The above limitations are problem to our thesis as the main aim of this thesis is to develop a crowd counting system that is able to count the number of people in real time, cost effective and can operate with low resolution cameras to avoid the problems of privacy concerns. The image is taken from a video recording of a live event from vertical angle (close to 90 degrees) and counting is done based on the number of heads present in the image. The system should be able to give accurate results even when the crowd size becomes large and does not involve any kind of people tracking.

3 Methodology

First step was to develop a database of images with different number of visible heads. Next, these images were divided into categories, each category had images containing a specific number of heads. After that, we extracted features from each category and fed them to a Neural Network along with corresponding targets (head count in that image) to train a regression model. Whenever a new image is tested, its extracted features are sent to the NN model, which predicts the head count.

The above explained methodology is expressed in pictorial form in Figure 5.

3.1 Overall Methodology

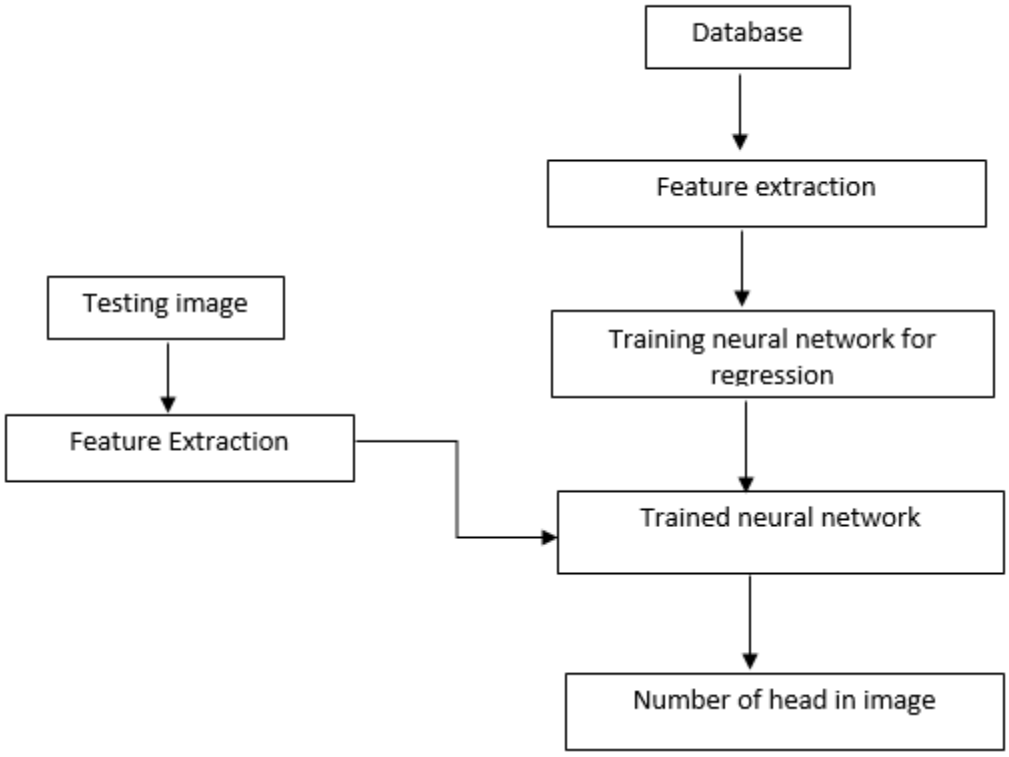

Figure 5: Overall process of head counting

3.2 Flowchart of Feature Extraction Process

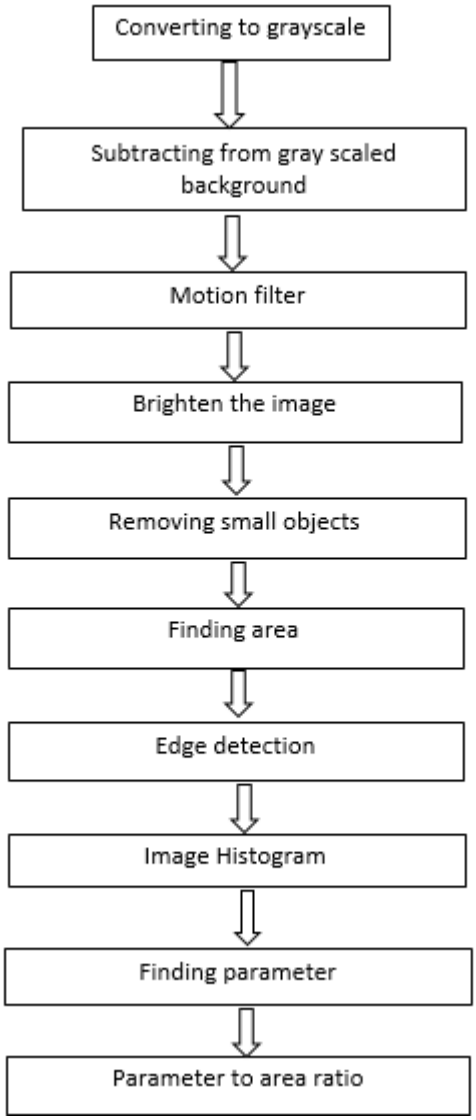

Figure 6: Feature extraction process

Figure 6 shows the flowchart of the feature extraction process. In the feature extraction process, we first need to convert the corresponding image to grey scale mode and deduct the background image from it so that we get a complete greyscale image. After this, we applied the motion filter to detect movements followed by the brightening the image and deleting unidentified small objects in the image. The exact area (which contained the number of heads of people) was identified in the image and later edge detection was applied onto it. This whole feature extraction

helps make the parameter to area ratio of the image so therefore we needed to extract the image histogram from it followed by finding the parameter.

3.3 Image Acquisition

For image acquisition a mobile phone camera was used. It was a low resolution camera which helped us achieve the privacy concerns of the crowd that was under supervision. The specifications of the camera and the method by which the data has been obtained are explained in detail in Chapter 6. Figure 7 displays the image of the mobile phone used while figure 8 displays a few images from our dataset.

Figure 7: Mobile phone used for capturing video

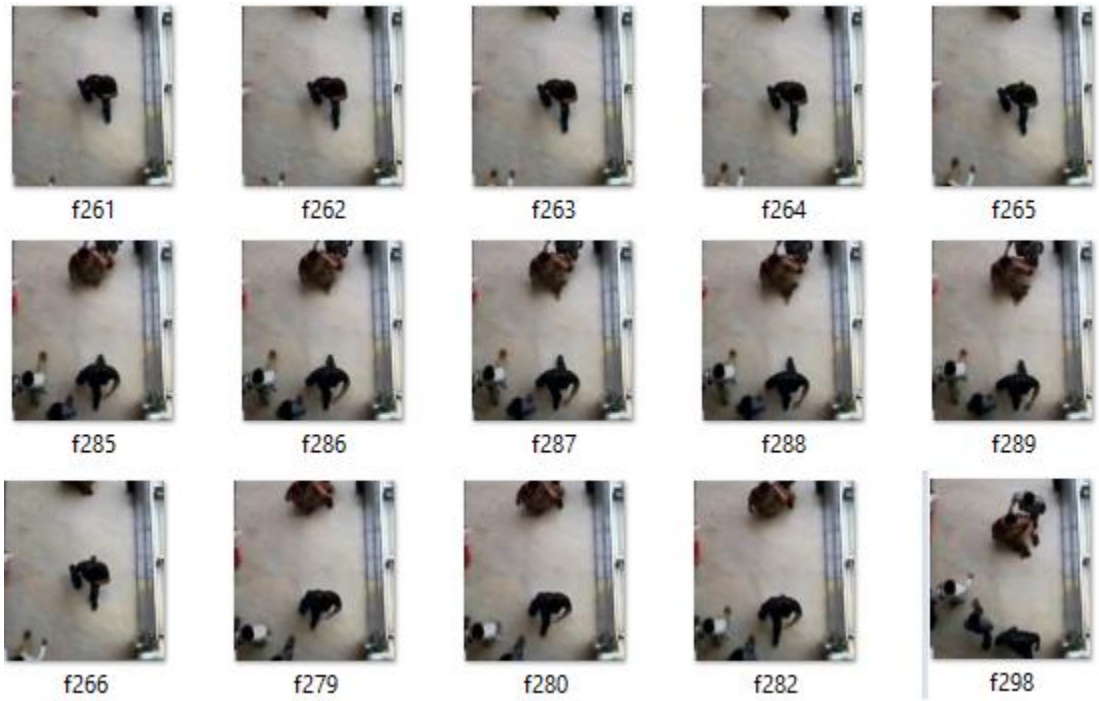

Figure 8: Few images from our dataset

3.4 Converting to greyscale:

Greyscale basically is simply a range of grey shades (only one colour) without actual colour. In greyscale images, the darkest shade is black, which shows the total absence of light either reflected or transmitted. Similarly, the lightest shade in greyscale is white that shows the total reflection or transmission of light at all visible wavelengths. Between the darkest and lightest shades of grey, all other shades are represented by equal brightness of red, green and blue colours (primary colours of RGB image) for transmitting light in an image, or it can be represented by three primary pigments e.g. Cyan, Magenta and Yellow colours for reflecting light in an image.

All the images acquired from mobile camera are colour images that consist of primary monochrome layers of RGB.

Figure 9: RGB image

In order to find some features of the image, we used a greyscale image (Figure 10) by converting RGB image (Figure 9). The greyscale image is a single layer image in which the greyscale shades depends on all light intensity values. The greyscale image after conversion is shown as below:

Figure 10: Greyscale image

3.5 Subtracting background

For feature extraction, we require the background image that was taken from the camera and there was no person in that image as shown below. Figure 11a shows the background image we obtained.

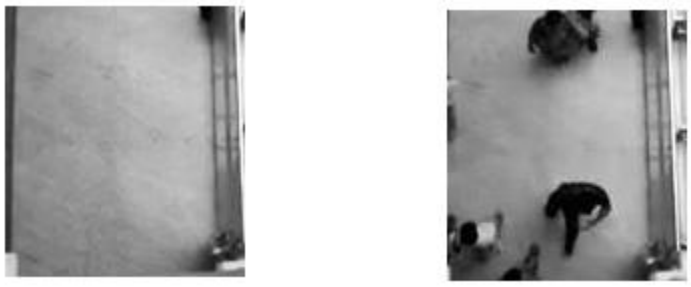

Figure 11: (a) Grey-scale background (b) Grey-scale scene

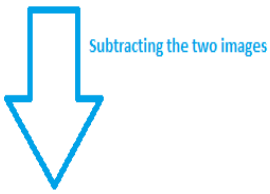

Figure 12: Background subtracted image

In order to find the number of heads present in an image, we converted the background image from RGB to grey-scale and subtracted it from the image that contains people (foreground image). The resultant image obtained after subtraction is shown in Figure 12.

In order to detect motion, the system must learn how to locate a person on the scene (big variations in the background scene). The reference image that represents the background part of the scene is used for the background estimation algorithm. For detecting moving objects and

their tracking, this background image was used and it will further be also used to separate the foreground and background.

After completion of background estimation, the algorithm is used to separate the foreground and the background (the foreground just represents the big variation between the current background and the image of the video camera) (Sahoo, AK. 2013).

The first task using this algorithm is to make a new image (background subtracted) resulting from the absolute difference between the current frame and current background. The resultant image is a grey-scale image. So, for segmentation, this image must be transformed into a binary image (i.e. Separation of the background and the foreground). A threshold must interfere when transforming a grey-scale image (255 levels) into a binary image (2 level). So, by setting the threshold value, all the pixels whose values are less than the defined threshold, will be viewed as the background of the scene (value 0). This technique will be helpful to remove “noisy” pixels in the image that represents the shadow of moving objects, in most of the cases. In a grey-scale, mostly the shadow of an object doesn't change a lot of features (colour) of the pixel, that’s why background subtraction has a small value.

This first step plays an important role in the people counting algorithm to make a pixel-by-pixel absolute difference between the two images, i.e. background and foreground images. The new resultant image shows all the differences between these two images. If the resultant image is not an empty image (black image) then it means that there is a modification in the two consecutive frames that shows that there are some people in the scene. An unwanted noise will appear if the camera quality is low, so there will be possible pixel changes in two consecutive images even if the background is not changed or if there is no one present in an image (noise is random, fluctuation of pixel value is due to poor quality of cameras and scanners). Due to this difference, the two consecutive frames are not exactly the same, but are very similar to each other. If there is any noise in the video camera, then the “noisy” pixels will have close grey-scale values. A threshold is the best way to filter information such that it can separate the values that are lower than the threshold and the values which are greater than the threshold values. It is necessary to introduce the threshold value in order to decide the existence of motion.

An absolute difference of two pixels is a difference between pixel intensities. As a result of this it gives an absolute result. The result corresponds to the variation between the two pixel values.

Two different techniques are used for thresholding. These techniques are given below:

- In the first technique thresholding is done on the difference value between two-pixel values. Firstly difference is calculated and then this difference is compared with the threshold value. In case the difference is less than the threshold, then the value will be assigned a zero (black) value. An alternative method can be considered, in which a larger grey-scale value is selected from the difference of the pixel's value. Difference value and threshold value are compared and if the value is greater than the threshold, 1 is assigned to it and if it is less than the threshold, then zero is assigned.
- In the second technique whole image pixel values are involved in the thresholding process. All pixel values are summed up and compare with the threshold value if the value is less than the threshold value than the resulted image is empty.
-

3.6 Motion filter

The images are taken from the camera mounted at the top so people are moving and camera capture the moving images of the people. In order to remove this motion effect and make image clearly visible we applied motion filter. Motion filter is the type of filter through which the process of convolution of the image, approximate the linear motion of the camera. The resultant image obtained after removing motion effect of relative movement of the camera and person is shown below in Figure 13.

Figure 13: Image obtain after applying motion filter

3.7 Brighten the image

The image obtained from motion filter is good enough, but it is not bright. So we have to make measures in order to brighten up the image. Here image brightening means making the black area darker and white area brighter. Thresholding is performed at this stage. Each pixel value is compared with some predefine threshold and if the pixel value is greater than the assigned value, then one is assigned to it, however, if the value is less than the threshold then zeroes value is assigned. Figure 14 shows the brightened image after removing the dark contents.

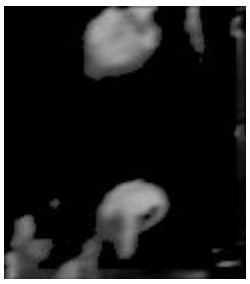

Figure 14: Brighten image obtain after removing darker contents

3.8 Removing small objects or noise- Morphological opening

Morphology in terms of image processing describes as a variety of image processing techniques that deals with the morphology of the image or with the shape of the features in a specific image. Morphological operations are usually used to enhance the quality of the image by removing imperfections after segmentation. In other words, it can be employed as a branch of biology that deals with different forms and the structure of living organisms i.e. plants and animals.

In mathematical terms, morphology opening is described as the erosion of set A with structuring element B followed by the dilation of this set by a structuring element B:

$$A \bullet \mathrm{B} = (\mathbf{A} \ominus \mathbf{B}) \oplus \mathbf{B} \quad (1)$$

Where $\ominus$ and $\oplus$ denote erosion and dilation, respectively (Tcheslavski, 2009).

Along with the closing, opening is being used as a benchmark for noise removal from an image in the field of image processing and computer vision. Morphological opening of an image removes small unwanted objects (consider as noise) from the foreground (typically taken as the dark pixels over RGB or grey background) of an image and place them in the background. These techniques

can escort to detect specific shapes in an image. Opening of an image is mostly used to find the things for which a specific structuring element can fit (e.g. Edges, corners).

One can think of *B* sweeping around the inside of the boundary of *A*, so that it does not extend beyond the boundary, and shaping the *A* boundary around the boundary of the element.

- Opening is idempotent,

$$(A \bullet \mathrm{B}) \bullet \mathrm{B} = \mathrm{A} \bullet \mathbf{B} \tag{2}$$

- Opening is increasing,

$$A \subseteq B \ \ then \ \ (A \bullet \mathrm{B}) \ \subseteq \mathrm{C} \bullet \mathbf{B} \tag{3}$$

- Opening is anti-extensive

$$(A \bullet \mathrm{B}) \ \subseteq \mathrm{A} \tag{4}$$

- Opening is translation invariant.
- Opening and closing satisfy the duality

$$(A \bullet \mathrm{B}) = (A^C \bullet \boldsymbol{B^C})^C \tag{5}$$

We remove all the objects that have pixels which are less in number than a certain defined limit, for the more visibility of the white areas. After this, only those area(s) are left where portion of white area(s) is as per defined limit and these area(s) are representing the person in image.

The image obtained after undesired object, removing is shown in Figure 15.

Figure 15: Image obtained after removing small objects

This image is now ready for feature extraction.

4 Feature Extraction

Features are being extracted from the given image. Features plays a very vital role in the image classification, as on the basis of the features extracted, the result of distinguishing the image is deduced and we are able to classify the images into different classes. The information obtained from the features concern the present edges, shapes and image contents etc. There are different feature extraction techniques available and each technique of feature extraction abstracts the image information based on the type and design of the image. Selection of feature extraction technique depends on the application for which we are extracting features and for what purposes we are using it.

There are four types of features extracted here:

- Detected blobs Area.
- Detected blobs Perimeter.
- Detected blobs ratio of Perimeter to area.
- Detected blobs Histogram of edges (or outline).

4.1 Area of blobs

From the resultant image of previous steps which contains only the large objects as per the defined limit and small objects have been removed, to find out the area of blobs, the area of only white pixels has been calculated.

4.2 Edge detection

Edge detection is defined as a set of mathematical models that works to detect those points in an image where the brightness of image changes abruptly, more sharp or discontinuous. The points where the brightness of image changes sharply are organized into a pattern termed as edges.

In other words, we can say that edge detection is an image processing technique that finds the boundaries of objects present in an image. It is used for data extraction, feature detection from an image and image segmentation in a wide range of areas such as image processing, machine vision and computer vision (Umbaugh, 2010).

Some basic algorithms of edge detection include Prewitt, Sobel, Robert, Canny and fuzzy logic methods.

2D images extracted from a 3D scene can be classified into viewpoint dependent and viewpoint independent edge extraction. A *viewpoint independent edge* extraction usually deals with the inherent properties of the 3D objects, such as surface shape and marking. On the other hand, *viewpoint dependent edge* detection may change with the change in the viewpoint, hence reflects the geometry of the scene, such as objects obstructing one another (Lindeberg, 1998).

Detection of ideal step edges of a natural image is not as likely as usually considered in the literature. These detections are typically affected by one or several of the following effects:

- Penumbral blur
- Focal blur
- Shading on a smooth object

Gaussian smoothed edge method or error function has been used by several researchers in their study as an extension of the ideal step edge model to model the effects of blurry edge in real life applications (Zhang & Bergholm, 1997). So, in this way a 1D image f which has exactly one edge placed at $x = 0$ can be modelled as follows

$$f(x) = \frac{I_r - I_l}{2}\left(erf\left(\frac{x}{\sqrt{2\sigma}}\right) + 1\right) + I_l \quad (6)$$

Here, in the image at the left side of the edge, the intensity is $I_l = \lim f(x)$ and at the right side of the edge it is $I_r = f(x)$. The scale parameter for the edge σ is called blur scale. In an ideal situation this scale parameter should be adjusted in a way that it's based on image quality to protect the true edge of the image from any damage.

In edge detection we basically find the discontinuity of all the areas. Discontinuity can be said as the region where the change in the brightness is sharp. Image is brightened at first and then edge detection technique is applied. The result obtained after applying the edge detection technique is as shown below in Figure 16.

Figure 16: Image obtained after edge detection

4.3 Pictorial representation of steps followed

Figure 17: (a) RGB background (b) RGB scene

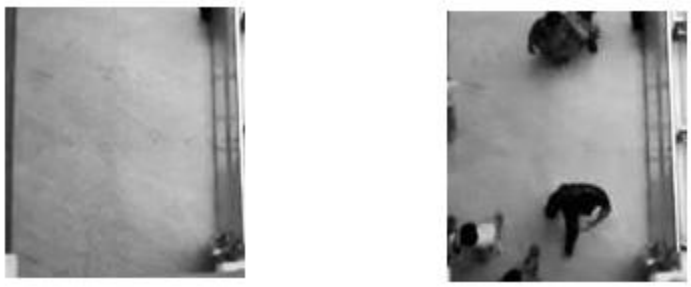

Figure 18: (a) Grey-scale background (b) Grey-scale scene

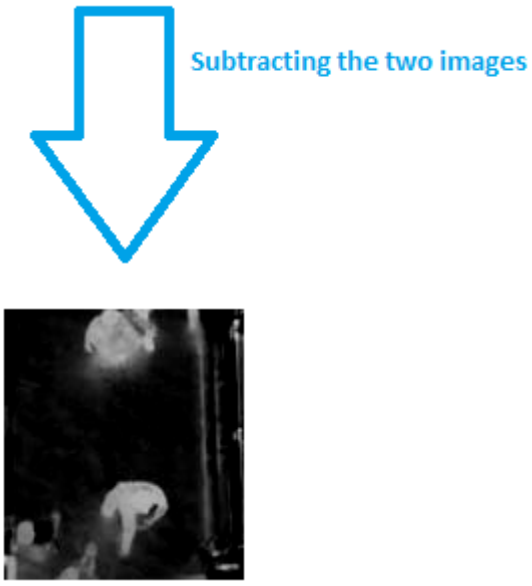

Figure 19: Result of background subtraction

Figure 20: Motion filter applied

Figure 21: Threshold based segmentation

Figure 22: Removal of extra pixels from image boundary

Figure 23: Edge detection

4.4 Image histogram

Image histograms are based on the individual intensity where the number of pixels are graphically represented. Each intensity value is represented by the bins, which are the building blocks of the histogram. It is usually estimated by computing all pixels of an image and based on their intensity each pixel is assigned to the bin. This histogram is an important tool to inspect the image thoroughly. They identify the background and grey value range in the image instantly as well as quantization of noise and clipping in the image can also be spotted immediately. In other words, the tonal distribution of digital images is represented by an image histogram. Based on the tonal value number of pixels are plotted in the histogram. By looking at the histogram of a specific image one can visualize the intensity of all pixels of an image (Freeman, 2005).

Image histograms are built in programs in many modern digital cameras. Photographers can use them to visualize the intensity of the captured image and the impact of blown out light or blacked out shadow on the pixel quality of image.

The horizontal axis of the graph represents the intensity variations, while the vertical or y-axis represents the number of pixels of that particular intensity. The right hand side of the graph shows light and white areas while the left side of the horizontal axis represents the black or dark areas and the middle area of graph shows medium grey tone. However, the vertical axis represents the size of the area that is measured for each of the given intensity zone (Martin, 2007). The histogram can simply be found by scanning images into the signal pass and running count of the number of pixels that are present for each intensity value.

After finding edges in the image we find the histogram of that edge detected image.

4.5 Parameter of the detected edges

After edge detection, all the white pixels are counted for estimating boundaries of the image.

4.6 Parameter to area ratio

It is the 4th component of feature vector.

$$perimeter\ to\ area\ ratio = \frac{no\ of\ white\ pixels\ in\ edge\ perimeter}{total\ no\ of\ white\ pixels} \quad (7)$$

Final extracted feature vector will be a row vector consisting of

i. Area of blobs
ii. Perimeter of blobs
iii. Area to perimeter ratio
iv. Histogram of edge-detection

This feature vector will be then provided to a linear regressive model.

5 Linear Regression

The relation in-between a scalar dependent variable y and one or more descriptive variable denoted as Xi is unfolded by the Linear Regression technique of statistical modelling (Nicholsan and Gibson, 2016)Regression model denoted by the common equation which is:

$$y = \beta_0 + \sum \beta_i X_i + \varepsilon_i \tag{8}$$

Here, parameter vector is represented by βi, whose components are called effects, or regression coefficients. βi is the main focus in the methods of Inference in linear regression and statistical estimation. Error term, noise or disturbance term is denoted by εi. The factors which affect the dependent variable y are stopped by this error term variable, however, it does not stop those variables which effect xi. M5's method was used in Linear Regression, a technique which is used in feature selection, but to remove those features with lowest standardized coefficient till no enhancement is identified in the estimation of the error. For people counting linear regression is used in modern methods. Meta- algorithm, also called Additive Regression and Bootstrap aggregating (Breiman, 1996) are the two methods used to improve the linear Regression. In current study Artificial Neural Network model has been used for regression analysis.

5.1 Artificial Neural network

Brain is a decision system, (Ailamaki, et al., 2012)) in human beings that has neurons connected in a complex way. To handle complicated problems related to humans, the performance of the brain is more powerful and faster than any computer processor. The communication in between the layers of neurons is in a parallel pattern. The previous layer sends output lines to each neuron and the next layer receives input lines from each neuron, this is how parallel communication works. To take proper decisions neurons must learn and memorize how to send and receive information from the system.

To solve various difficult problems like posture, motion, mathematical calculations, etc., the brain has powerful decision abilities. By learning and memorizing pervious cases/problems that are similar to the given problem, new problems can be solved. The training and memorizing of neurons to solve various problems and perform different tasks under varying conditions are used

to duplicate the underlying functionality of the brain, it is explained in the literature that mimicked the brain's neuron (Sieu, et al., 2014). (1) Input layer, (2) Middle or Hidden layer(s), and (3) Output layers are the three main parts of ANN involved in the intelligent mathematical process.

The system's inputs are the part of the first layer that is an input layer (Hudson, 2001.) The "input layer" is simply a vector of the inputs. The essential unit of the ANN called neurons are presented in the second layer that is also called hidden layer. The main mathematical calculations to process the inputs and providing the proper outputs occurs inside the neurons. In the real brain, which is biological neuron (Bullinaria, 2004), a line of values from the previousis sentnd to the next layer are sent and received by the neuron in the hidden layer. The weight value of the channel (i.e., line) that carries the value to and from the neuron is the main factor on which the received and sent values depend. The value that is multiplied by the carried value (i.e., multiplying the weights value by the coming value from the previous neuron) before passing the result to the next neuron is called the weight of the channel. We can change the weight value by changing the intended task to be performed; its value can be decided for learning and memorizing to do that task (Hajek, 2005).

5.1.1 Mathematical Model:

A nerve cell is noted by the estimation of the load and the standard of the association between associating in nursing information. Positive qualities assign simulative associations, however the negative weight values replicate repressive associations. The real movement in the nerve cell is shown by the 2 subsequent elements.

The internal activity of somatic cells is represented as:

$$v_k = \sum_{j=1}^{p} w_{kj} x_j \tag{9}$$

5.1.2 Recurrent ANNs:

In this, the input associations are not present. The approximations (effort) of the unit knowledge associated degree method has defined the ANN that can move towards the state as gradual state. In gradual state, the initialization doesn't allow for change. The alteration in initiation estimation of yielded neurons are used in some applications (Che, et al., 2011).

5.1.3 Training

The training is the important factor in the formation of ANN model (Nicholsan and Gibson, 2016) as in the overall performance of the model there is some dependency upon it as well. In training process there are two types:

1. Supervised
2. Unsupervised

In the first one, we provide the training data and know about both inputs and outputs. In general it can be used both as regression and classification.In the second type, we know about the inputs but the outputs are unknown [23]. In 2011 (Jayalakshmi & Santhakumaran, 2011) discussed in detail about the normalization, as normalization can be done in multiple ways, the standard method used is to compress them. Here, we pass the inputs from sigmoidal function. Next to normalization is the optimization, to optimize the network. In optimization process the weights of the neurons are changed and hence we optimize the network for the achievement of high accuracy for given training data.

5.1.4 Operation

The network development is started by passing the information to the block "F1", in the start the output of block "F2" is zero so the G1 and the G2 both get ON. The output of the block "F1" is compared with the given info, if its same then valid otherwise data is passed through other functions and a loop process start till valid result.

5.1.5 Initialization:

$$w_{ij}^{b}(0) = 1 \tag{10}$$

$$w_{ij}^{b}(0) = \frac{1}{1+N} \tag{11}$$

N= Neuron number in F1 block

M= Neuron number in F2 block

$$0 < i < N$$

$$0 < j < M$$

Threshold ρ is selected,

$$0 \leq p \leq 1$$

1. Input applied is x

2. In block "F2" activation values is calculated for "y_0" of neurons [40]

$$y_i' = \sum_{j=1}^{N} w_{ij}^{f}(t) x_i \tag{12}$$

3. winning neuron is selected and renamed "k"

$$0 \leq k \leq M$$

4. test is done at condition

$$\frac{w_k^b(t).x}{x.x} > \rho \tag{13}$$

Denotes the inner product, so go to step 7, or else go to step 6. [38]

Note that

$$w_k^b.x \tag{14}$$

Fundamentally is the inner product

$$x^*.x \tag{15}$$

$$0 < i < N$$

$$w_{kl}^{b}(t+1) = w_{kl}^{b}(t)x_1 \tag{16}$$

$$w_{lk}^{f}(t+1) = \frac{w_{kl}^{b}(t)x_l}{\frac{1}{2}+\sum_{i=1}^{N} w_{ki}^{b}(t)x_i} \tag{17}$$

5. All the neurons are re-enabled in block “F2” and jump to step 2.

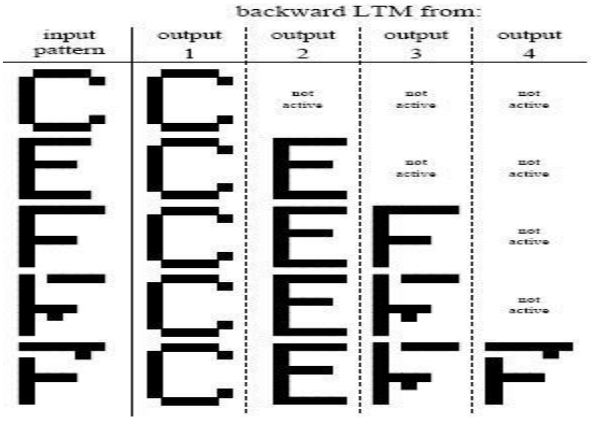

Figure 24: Letter pattern examples

5.1.6 Delta Rule

Meanwhile we are utilizing units with nonlinear enactment capacities, we need to summarize the delta guideline:

The enactment is a differentiable capacity of the cumulative information, given by

$$y_k^p = F(s_k^p) \tag{18}$$

In which

$$s_k^p = \sum_j w_{jk} j_j^p + \theta_k \tag{19}$$

To get the precise simplification of the delta rule as presented in the previous chapter [41], we fixed

$$\Delta_p w_{jk} = -\gamma \frac{\partial E^p}{\partial w_{jk}} \tag{20}$$

Ep = total quadratic error p

$$E^p = \frac{1}{2} \sum_{0=1}^{N_o} (d_o^p - y_o^p) 2 \tag{21}$$

$$\frac{\partial E^p}{\partial w_{jk}} = \frac{\partial E^p}{\partial s_k^p} \frac{\partial s_k^p}{\partial w_{jk}} \tag{22}$$

Second factor is

$$\frac{\partial s_k^p}{\partial w_{jk}} = y_j^p \tag{23}$$

So we define

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p} \tag{24}$$

We will get a remodeled recommendation which is proportional to the delta standard as described in the past section. Setting off a slope plummet on the mix-up surface on the off chance that compress the weight improvements as indicated by:

$$\Delta_p w_{jk} = \gamma \delta_k^p y_j^p \tag{25}$$

Set-up stays to make sense of what “δp” have to be for every unit “k” in system. An interesting result that we now determine is that there is a basic recursive calculation of “δ's” which is to be portrayed via causing mistake flags in contrary by system.

In order to process “δp” we relate series principle toward combined fractional subservient. As the result we obtain two elements, first one reflecting the adjustment in mix-up as an element of the yield of the unit and the other one is linking the adjustment in the yield as a part of variations in the information.

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p} = -\frac{\partial E^p}{\partial y_k}\frac{\partial y_k^p}{\partial s_k^p} \tag{26}$$

Second factor will be computed by

$$y_k^p = F(s_k^p) \tag{27}$$

So we obtain

$$\frac{\partial y_k^p}{\partial s_k^p} = F'(s_k^p) \tag{28}$$

This result was attained by delta rule (standard).

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p} = -\frac{\partial E^p}{\partial y_k}\frac{\partial y_k^p}{\partial s_k^p} \tag{29}$$

In the below equation

$$\frac{\partial y_k^p}{\partial s_k^p} = F'(s_k^p) \tag{30}$$

Obtained now is

$$\delta_o^p = \left(d_o^p - y_o^p\right)F_o{}'(s_o^p) \tag{31}$$

For output unit "o". In addition, if "k" is not a yield unit but somewhat a shrouded unit k = h, we don't quickly know the obligation of the unit to the yield blunder of the system. Hence, a mistake can be formed as a constituent of the net inputs from covered up to yield layer;

$$\mathrm{Ep} = \mathrm{Ep}(\mathrm{sp}_1, \mathrm{sp}_2, \ldots\ldots \mathrm{sp}_\mathrm{j} \ldots\ldots) \tag{32}$$

And we utilized this tenet to compose the model

$$\frac{\partial E^p}{\partial y_h^p} = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p}\frac{\partial s_o^p}{\partial y_h^p} = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p}\frac{\partial}{\partial y_j^p}\sum_{j=1}^{N_h} w_{ko} y_j^p = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} w_{ho} = -\sum_{o=1}^{N_o} \delta_o^p w_{ho} \tag{33}$$

Adding (substitution) of equation gives

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p} = -\frac{\partial E^p}{\partial y_k}\frac{\partial y_k^p}{\partial s_k^p} \tag{34}$$

$$\delta_h^p = F'(s_h^p)\sum_{o=1}^{N_o} \delta_o^p w_{ho} \tag{35}$$

The equation below provides repeated system for assuming the "δ's" for all units, it is used to spot the changes in weight as directed by mathematical statement. This system sets up the delta guideline for a feed forward system of non-linear units.

$$\delta_o^p = \left(d_o^p - y_o^p\right)F_o'(s_o^p) \tag{36}$$

5.2 Regression using ANN

Classification or regression can be done by using neural network (Nicholsan and Gibson, 2016). In short, the help in predicting the data or classifying the data after training of the network (M.JEEVAN BABU, 2001).

To convert the data which is continuous, logistic regression technique is usually used by the classifiers. It converts the data into binary variables i.e. 1 and 0. It usually group the data by defining certain inputs like age, height and weight and then we give these input to an already trained network and it gives and output accordingly (Jae H. Song, 2005).

Let us consider another example, if you have the age and distance data of school to home then you can predict that how much time an average student will take to come from school to home. This technique can be applied on number of scenarios.

As we have discussed earlier that it can also be classified therefore just in case if we want to see the results we can plot independent variables y to a continuous x.

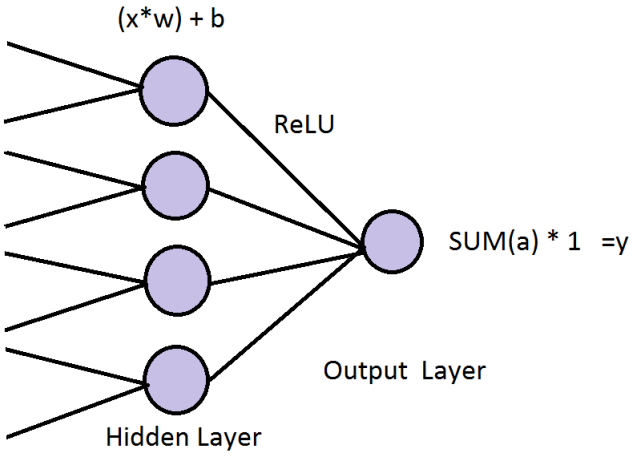

Figure 25: ANN Regression

As shown in Figure 25, ‘y’ is the output of the system where as ‘x’ represents the input. Features, which are represented by the x variable were passed to the network in the previous layer. Here x was multiplied with the corresponding weight, w (Razvan Pascanu, 2005).

An activation function was fed with a sum of products which are supplementary. When we talk about this case rectifier linear unit is an example of activation function. It does not penetrate on thin pitches as sigmoid activation functions that’s why it is highly useful. Because of this property it is mostly used.

Rectifier Linear Unit (ReLU) gives an activation at the output for each hidden nodes. These activations are added in the output node. This procedure allows the activation sum to be permitted through. Which means there will only be one node of neural network at the output which will carry out regression. Then the sum of previous layer will be multiply with the activation factor 1. The end result will be ‘y’.

We can simply compare the value of y with its ground truth value, in order to make the network learn or to implement the backpropagation. Similarly, the biases and weight of the network are modified. This process continues until the error is removed.

5.3 Use of a Levenberg-Marquardt NN for regression

Levenberg-Marquardt algorithm (Gavin, 2016) is like the quasi-Newton method was designed without computing Hessian matrix to approach second-order training speed. With the below formula, we can make the approximation for the Hessian matrix.

$$\mathbf{H} = \mathbf{J}^T\mathbf{J} \tag{37}$$

Slope can be calculated by using the formula given below

$$\mathbf{g} = \mathbf{J}^T\mathbf{e} \tag{38}$$

Jacobian matrix is represented by J. Jacobian Metrix, which represents the network error of first order derivatives in association with the weights and biases. The network error is represented by the vector e. To activate the Jacobian matrix back propagation technique is used. The major reason for using this technique is that it is very simpler as compare to hessian matrix.

Levenberg -Marquardt algorithm is used to approximate the Hessian matrix. The following formula was used for this purpose.

$$\mathbf{x}_{k+1}=\mathbf{x}_k-[\mathbf{J}^T\mathbf{J}+\mu\mathbf{I}]^{-1}\mathbf{J}^T\mathbf{e} \tag{39}$$

Here Newton's method is used for Hessian matrix approximation, if the scalar term μ is set to zero. If we keep a large the value of μ with a small step size, it will become gradient descent. The major reason why we use newton method is that it has more accuracy as compared to other methods. Similarly, it also has a lowest error with the fast processing time. In order to improve and maximize the production of the function μ is increased otherwise it is decreased. So it can be said that the performance is reduced after every iteration.

5.4 Training of Neural Network

The training data that is used to train the model consisted of 77 images. The breakdown of these images with respect to persons present in a single image is given below

- 51 images containing a 1 person in an image.
- 5 images containing 2 persons in an image.
- 8 images containing 3 persons in an image.
- 13 images containing 5 persons in an image.

No data of images containing 4 persons has been provided for training purpose. Then we generated an ANN regression model form MATLAB App 'Functional fitting Neural Network'. A model that, we obtained after extensive training with the training images is shown below (fig. 27).

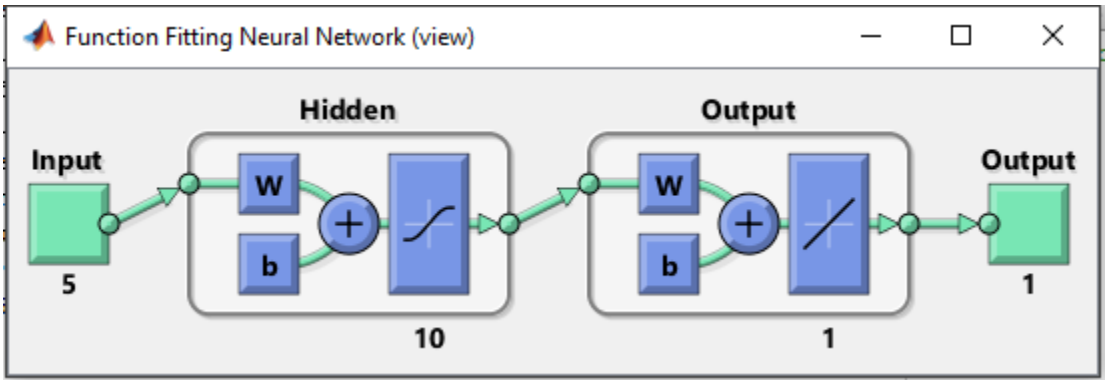

Figure 26: Neural network diagram

Figure 26 shows neural network diagram.The training data provided to neural network were further divided into training data, validation data and testing data. Training performance Graph of neural network for the training, testing and validation of the data with the mean squared error is shown in Figure 27.

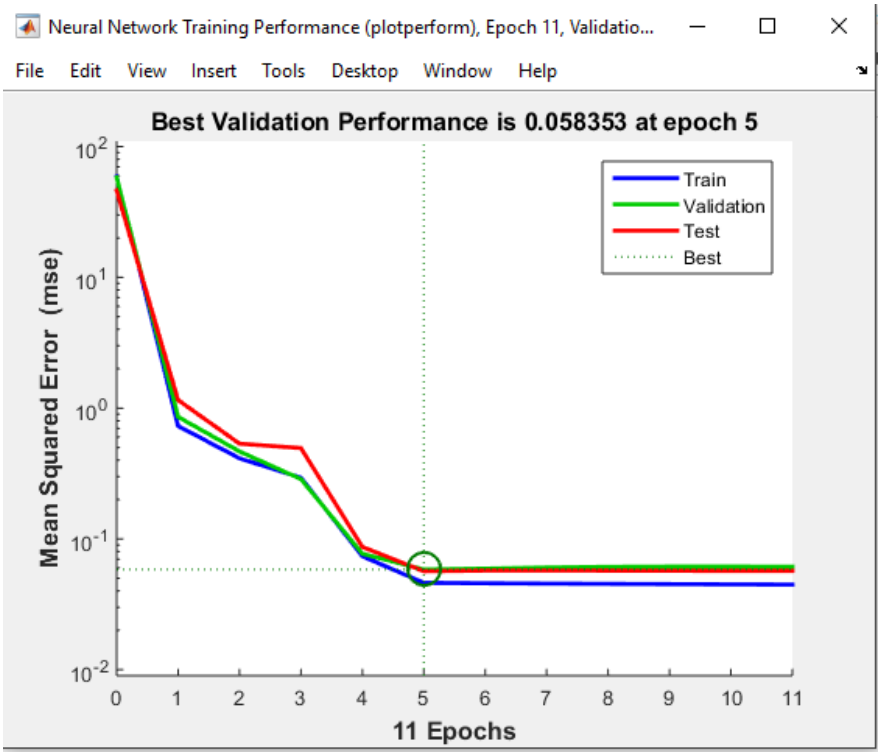

Figure 27: Performance evaluation graph of neural network

6 Creation of Training and Test Corpus

6.1 Creation of database

In order to obtain accurate results for this study, we have to train our data along with testing. For developing a neural network, we had to create a database which has been used for training and testing different images that were collected by the low resolution camera. The training and testing data were all composed of different images which means that, the training data images were not used in creating the testing database.

For creating the database, we had to collect different images of people suitable for our study. So therefore we used a camera which requires a few conditions regarding its setup as shown in Figure 28.

1. The camera location must be fixed
2. The camera should be mounted at the top of a room, e.g., on a ceiling at or close to 90 degrees in angle.

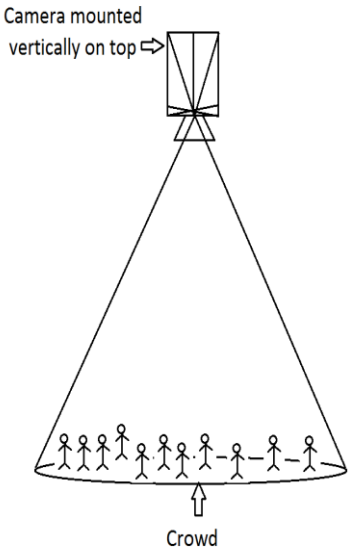

Figure 28: Placement of the camera

This is quite a reasonable assumption since IP cameras are all usually fixed.

The camera was held static and its resolution was kept low for two purposes:

1. To know the efficiency of the system via using a low cost camera so that the application could be easily affordable.
2. To make sure that the privacy of the people is not violated (their faces are not visible, just the top of their heads is focused and even that is also not perfectly clear, but it is clear enough to deduce how many people are present in the image).

6.2 Difficulties faced in creating the dataset

The major problem we faced at the beginning was the unavailability of a standard database that is required as per project's criteria. We planned to obtain datasets by recording the events held across Australia. By raising a fixed pole with camera at the place of the events, recording can be done. Unfortunately the event organizers turned down our request as it might pose a threat to the crowd present in the event. There was no freely available dataset on the Internet which meets the criteria mentioned above in 'Creating the database'. And due to privacy concerns, most buildings and malls in Australia do not give permission to take videos/images of people.

6.3 Creating dataset.

Due to the difficulties mentioned above, we created a dataset on our won by using a mobile phone camera. The steps are explained as follows.

- A person holding the mobile phone camera was made to stand in the first floor in a private place.
- The camera was held vertically to record the movement of the people in ground floor in a way where only heads are seen.
- 4 videos of 5 minutes duration were recorded by myself and helpers, will all the people involved including the images, consenting to the process prior to being involved. Some of the images taken from the video can be seen in Figure 29.

Figure 29: Few of the datasets created

6.4 Specifications of the camera

The images acquired by using the mobile phone are in RGB24 format with resolution 320x240 Pixel. The specifications of the camera are as shown in the below table.

Camera	Primary	13 MP, f/2.4, 29mm, face detection/laser autofocus, OIS, dual-LED flash.
	Features	1/33” sensor size, 1.12 µm pixel size, Geo-tagging, touch focus, panorama and HDR.
	Video	2160p@30fps, 1080p@30fps and HDR.

Table 2: Specifications of the camera used

6.5 Division of training and testing data

6.5.1 Training Data

- 51 images in Category one
- 5 images in Category two
- 8 images in Category three
- 12 images in Category four
- 13 images in Category five

6.5.2 Testing Data

- 6 images in Category one
- 0 images in Category two
- 4 images in Category three
- 2 images in Category four
- 7 images in Category five

In the entire video, there were only 5 frames in category 2, so we kept those 5 images for the training purpose.

7 Head counting through Neural Network (Testing of the model)

In order to obtain the finest model, we train a neural network several times on training images, then we provide an input image to count the number of heads. These images were used from the testing data. In the end, we verify whether it is calculating the number of heads present in the image correctly or not. The results were very promising. The overall performance of the trained model shows that accuracy of head calculation is quite satisfactory as the number of heads in training data were different from the number of heads in the testing samples, For example there were no image in the training sample contains four heads as present in testing image.

The testing images contain one, four and five persons and results are shown in Figure 30 with the number of heads predicted by the trained neural network.

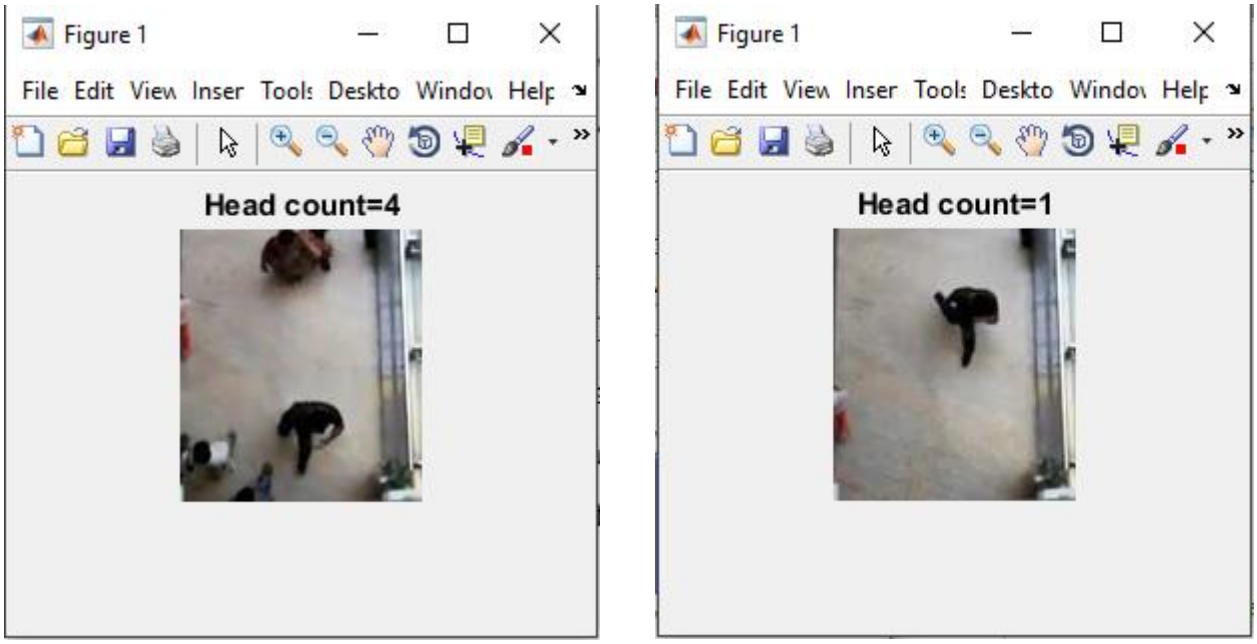

Figure 30: Result obtained for image with heads 4 and 1.

7.1 Results on testing data

Testing data was consisted of the following images Figure 31:

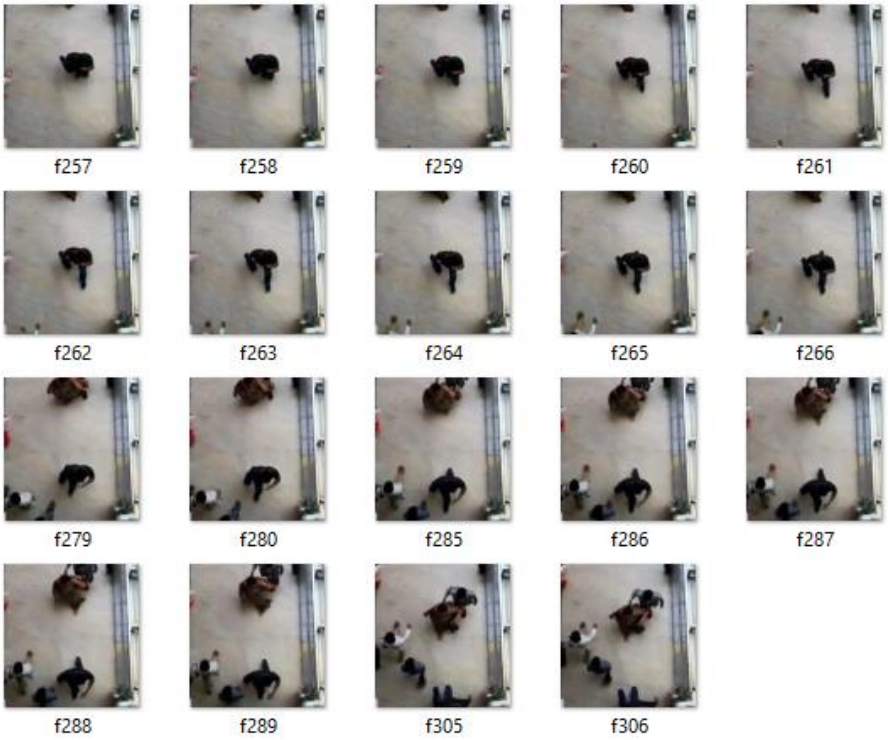

Figure 31: Testing Images

Where our system was trained on some images having 1, 2, 3 and 5 heads and tested on other images having 1, 2, 3, 4 and 5 heads:

Testing samples	Actual number of heads (peoples)	Number of head predicted by neural network
'f9.jpg'	1	1
'f10.jpg'	1	1
'f11.jpg'	1	1
'f17.jpg'	1	1
'f246.jpg'	1	1
'f247.jpg'	2	2
'f268.jpg'	3	3
'f270.jpg'	3	3
'f275.jpg'	3	3
'f278.jpg'	3	3
'f281.jpg'	4	4
'f282.jpg'	4	4
'f283.jpg'	5	5
'f284.jpg'	5	5
'f294.jpg'	5	5
'f297.jpg'	5	5
'f303.jpg'	5	5
'f307.jpg'	5	4
'f308.jpg'	5	5

Table 3: Trained on 1, 2, 3, 5 heads of training data- Testing on all testing data

As we can see from the above table, our system was able to detect the head count in almost all the test images correctly. If we had trained it on a small sized data, we may have not obtained that much accuracy.

When our system was trained on few images having 1, 3 and 5 heads and tested on other images having 1, 2, 3, 4 and 5 heads results are shown below:

Testing samples	Actual number of heads (peoples)	Number of head predicted by neural network
'f9.jpg'	1	1
'f10.jpg'	1	1
'f11.jpg'	1	1
'f17.jpg'	1	1
'f246.jpg'	1	1
'f247.jpg'	1	1
'f268.jpg'	3	3
'f270.jpg'	3	3
'f275.jpg'	3	3
'f278.jpg'	3	3
'f281.jpg'	4	3
'f282.jpg'	4	3
'f283.jpg'	5	4
'f284.jpg'	5	5
'f294.jpg'	5	5
'f297.jpg'	5	5
'f303.jpg'	5	5
'f307.jpg'	5	5
'f308.jpg'	5	5

Table 4: Trained on 1, 3, 5 heads of training data- Testing on all testing data

CASE #	MAE
1	0.052
2	0.176

Table 5: Mean Approximation Error (MAE) Evaluation for both cases

8 Application

The method proposed in this thesis can be used in different places. Some of its applications are given below:

- Mass gathering events: We can keep track of the people coming in and out and the total number of people present in the event. Depending on the event area, it is possible to understand whether the crowd density is larger than what the area can hold for. The social mindset of the people can be observed and the odd one can be picked out immediately. Any occurrence of signs of hazardous events can be found soon and immediate response can be taken to minimize the rate of injury and illness as much as possible. Furthermore, it can also have an immense importance in any disaster evacuation situations to see the evacuation routes that people take most often.
- Security: Security is also a great concern in present time. Security cameras can be placed inside and outside of buildings, airports, shopping complex, train stations and many other places. We can keep track of people who have entered the building and sensitive areas can be monitored continuously. Threats can be detected much faster and immediate action can be taken.
- Counting of people plays a vital role in surveillance systems. It can assist as a subtask in many stage processing tasks. The low level procedures are improved by people count robust estimation. Blob extraction can be used as robust estimation. In this problem, the people are estimated or the people density is estimated at the street or the platform. The output is usually a “fill rate” of the space expressed as a percentage. While in the solution of this problem location sensors are to be used.
- Some special sensors such as passive-optics directional, electro-optical and thermic sensors are used in the indoor applications where it detects the direction and passage of a person who has entered the building. After detecting the direction, the number of people who have entered and exited the building are counted with high accuracy. But the drawback is that, these sensors are expensive.

- The main advantage of the video cameras as compared to dedicated sensors is that we can use the video camera in tracking, recognition and density estimation of the crowd. Research is being done in a bulk under the umbrella of image processing over the objective of “people count estimation”.

9 Conclusion and Future Works

9.1 Conclusion

Using the above discussed techniques, we were able to develop a system which is capable of measuring the number of people present in an image derived from a low-resolution, low-cost camera, and to process the images quickly, in order to obtain a sufficiently accurate headcount. Results show a Mean Approximation Error (MAE) of 0.052, which is acceptably low given the hardware and computational constraints. Now we will discuss the pros and cons of this project.

Advantages:

The major advantages of this project are mentioned below:

- The system is very efficient and gives very good prediction rate and MAE of 0.052.
- Features used here are: histogram, perimeter of head blobs, area of white region, perimeter to area ratio. All these features are very basic in nature, yet give effective results.
- One of the major objectives of this project was to take care of the privacy of the people. In order to fulfil this requirement a low quality camera was used. Even if people look towards the camera, it would still be extremely difficult for an observer to identify them.
- Another factor that contributes towards ensuring the privacy of persons is the fact that the camera is installed from a high-angle vantage point, such that only heads are visible.
- This project is simple, cost effective, easy to install and maintain.
- No camera calibration is required.

Disadvantages:

There are a few limitations that must be taken into account. These limitations are given below:

- The camera must be mounted exactly at the top of the ceiling such that only the heads are visible, no portion of face/ body should be visible.

- This process is very much dependent on the background. This background is subject to change with the time. For example, lightning condition is different at different hours of the day. Also, the arrangement of furniture can be changed.
- We needed to subtract the background image from the camera so background image should be static throughout the experiment.
- The camera should be steady while capturing images.

9.2 Future work

In future, some improvements can be made in the presented method. The current work developed is very sensitive to the angle of the camera from where the images are taken and also the motion of the camera. The camera should be static in order to get static background. So to counter this problem, we can modify the system such that it updates the background image at regular intervals, which means that any change in lighting conditions or change/addition in furniture arrangement is no longer an issue.

We can also try different types of feature extraction methods like PCA, ICA, LBP, LQP etc. With the same neural network regression model and compare results. We can also use some other regression models like Bayesian Regression, Support Vector Machine regression etc. to compare the results with our proposed method.

With a little modification, this system has the potential to find application in measuring and reporting crowd density (number of persons per unit area) in a mass gathering, in real-time. This is particularly important in monitoring crowd flow in disaster evacuation situations. Crowd density can be found if the total area of the place is caught in camera view.

Similarly, in future, we can use this technique and place the camera on the entrance and exit doorways of the building to count the total number of people present in the building. By using this simple technique, we don't have to use the expensive sensors on the doors of the buildings to count the number of people.

Usually systems which give good accuracy, are expensive to implement and complex to understand and difficult to maintain, but our system is not only cost effective, but also gives accurate results ensuring that the privacy of the people is maintained.

In conclusion, the goal of demonstrating a basic system that can use low-resolution low-cost cameras, and yet is able to accurately measure crowd-density has been realised, and can form the basis of further explorations to optimise its functionality and generalise its usability to a wider range of environment

10 Appendix A – Description of contents of the appendix

```
Clear
clc
close all

features=[];
targets=[];

%%  storing features and targets of training images with 1 head
DB_folder = strcat(cd, '\trn tst\trn\ones');  %folder path
filenames = dir(fullfile(DB_folder, '*.jpg')); %this variable contains name properties(not content)
of all jpg files in DB_folder
total_ims = numel(filenames);%total number of images%total number of images
for im_no=1:1:total_ims %FOR ALL IMAGES OR FRAMES THAT ARE TO BE READ.....
full_name= fullfile(DB_folder, filenames(im_no).name);%image name
img=imread(full_name);%contains name of dir folder and all the whole files
feature_i=get_feature_m(img);%calling function to get feature vector
features=[features; feature_i];%concatenating feature vector in feature matrix
targets=[targets; 1];%concatenating target in target matrix
end

%%  storing features and targets of training images with 2 heads
DB_folder = strcat(cd, '\trn tst\trn\twos');%folder path
filenames = dir(fullfile(DB_folder, '*.jpg')); %this variable contains name properties(not content)
of all jpg files in DB_folder
total_ims = numel(filenames);%total number of images
for im_no=1:1:total_ims %FOR ALL IMAGES OR FRAMES THAT ARE TO BE READ
full_name= fullfile(DB_folder, filenames(im_no).name);%image name
img=imread(full_name);%contains name of dir folder and all the whole files
feature_i=get_feature_m(img);%calling function to get feature vector
features=[features; feature_i];%concatenating feature vector in feature matrix
targets=[targets; 2];%concatenating target in target matrix
end

%%  storing features and targets of training images with 3 heads
DB_folder = strcat(cd, '\trn tst\trn\threes');%folder path
filenames = dir(fullfile(DB_folder, '*.jpg')); %this variable contains name properties(not content)
of all jpg files in DB_folder
total_ims = numel(filenames);%total number of images
for im_no=1:1:total_ims %FOR ALL IMAGES OR FRAMES THAT ARE TO BE READ
full_name= fullfile(DB_folder, filenames(im_no).name);%image name
img=imread(full_name);%contains name of dir folder and all the whole files
feature_i=get_feature_m(img);%calling function to get feature vector
```

```
features=[features; feature_i];%concatenating feature vector in feature matrix
targets=[targets; 3];%concatenating target in target matrix
end

%%  storing features and targets of training images with 4 heads
% DB_folder = 'D:\head counting\manoj video\trn tst\trn\fours'; %folder path
% filenames = dir(fullfile(DB_folder, '*.jpg')); %this variable contains name properties(not
content) of all jpg files in DB_folder
% total_ims = numel(filenames);%total number of images
% for im_no=1:1:total_ims %FOR ALL IMAGES OR FRAMES THAT ARE TO BE READ
% full_name= fullfile(DB_folder, filenames(im_no).name);%image name
% img=imread(full_name);%contains name of dir folder and all the whole files
% feature_i=get_feature_m(img);%calling function to get feature vector
% features=[features; feature_i];%concatenating feature vector in feature matrix
% targets=[targets; 4];%concatenating target in target matrix
% end

%%  storing features and targets of training images with 5 heads
DB_folder = strcat(cd, '\trn tst\trn\fives');%folder path
filenames = dir(fullfile(DB_folder, '*.jpg')); %this variable contains name properties(not content)
of all jpg files in DB_folder
total_ims = numel(filenames);%total number of images
for im_no=1:1:total_ims %FOR ALL IMAGES OR FRAMES THAT ARE TO BE READ
full_name= fullfile(DB_folder, filenames(im_no).name);%image name
img=imread(full_name);%contains name of dir folder and all the whole files
feature_i=get_feature_m(img);%calling function to get feature vector
features=[features; feature_i];%concatenating feature vector in feature matrix
targets=[targets; 5];%concatenating target in target matrix
end

%% calling autogenerated script to train an ANN model
heads_nn

%% testing
img=imread('f281.jpg'); %test image
feature_i=get_feature_m(img);%getting feature vector
head_count=round( net(feature_i') )%rounding off (head count can only be a whole number)
imshow(img); %displaying image
t=strcat('Head count=', num2str(head_count));%head count
title(t)%%displaying head count
```

11 References

1. Sidenbladh, H., Black, M.J. and Fleet, D.J. (2000) *Stochastic Tracking of 3D Human Figures Using 2D Image Motion*. Available at: http://cs.gmu.edu/~zduric/it835/Papers/eccv00.pdf (Accessed: 23 August 2016).
2. MathWorks, T. (1994) *Trainlm*. Available at: http://au.mathworks.com/help/nnet/ref/trainlm.html (Accessed: 23 August 2016).
3. Nicholson, C. and Gibson, A. (2016) *Using neural networks with regression - Deeplearning4j: Open-source, distributed deep learning for the JVM*. Available at: http://deeplearning4j.org/linear-regression.html (Accessed: 23 August 2016).
4. Online, S. *Linear regression*. Available at: https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/linear_regression.pdf (Accessed: 23 August 2016).
5. Chan, A.B. and Vasconcelos, N. (2012) 'Counting people with low-level features and Bayesian regression', *IEEE Transactions on Image Processing*, 21(4), pp. 2160–2177. doi: 10.1109/tip.2011.2172800.
6. Arandjelovic´, O. (2008) *Crowd Detection from Still Images*. Available at: http://www.comp.leeds.ac.uk/bmvc2008/proceedings/papers/267.pdf (Accessed: 23 August 2016).
7. Atev, S., Masoud, O. and Papanikolopoulos, N. (2006) 'Practical mixtures of Gaussians with brightness monitoring', *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, , pp. 423–428. doi: 10.1109/ITSC.2004.1398937. (Atev, Masoud, and Papanikolopoulos, 2006)
8. Aziz, K., Merad, D., Iguernaissi, R., Drap, P. and Fertil, B. (2016) 'Head detection based on skeleton graph method for counting people in crowded environments', *Journal of Electronic Imaging*, 25(1), p. 013012. doi: 10.1117/1.jei.25.1.013012.
9. Belavkin, R.V. *Lecture 11: Feed-Forward Neural Networks*. Available at: http://www.eis.mdx.ac.uk/staffpages/rvb/teaching/BIS3226/hand11.pdf (Accessed: 23 August 2016).

10. Blum, A.L. and Rivest, R.L. (1993) 'Training a 3-node neural network is NP-complete', in *Machine Learning: From Theory to Applications*. Springer Science + Business Media, pp. 9–28.
11. Breiman, L. (1996) 'Bagging predictors', *Machine Learning*, 24(2), pp. 123–140. doi: 10.1007/bf00058655.
12. Bullinaria, J.A. (2015) *Biological Neurons and Neural Networks, Artificial Neurons*. Available at: http://www.cs.bham.ac.uk/~jxb/INC/l2.pdf (Accessed: 23 August 2016).
13. Chan, A.B., Liang, Z.-S.J. and Vasconcelos, N. (2008) *Privacy Preserving Crowd Monitoring: Counting People without People Models or Tracking*. Available at: http://visal.cs.cityu.edu.hk/static/pubs/conf/cvpr08-peoplecnt.pdf (Accessed: 23 August 2016).
14. Chan, A.B. and Vasconcelos, N. (2008) 'Modeling, clustering, and segmenting video with mixtures of dynamic textures', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5), pp. 909–926. doi: 10.1109/tpami.2007.70738.
15. Chen, K., Loy, C.C., Gong, S. and Xiang, T. (2012) *Feature Mining for Localised Crowd Counting*. Available at: http://www.eecs.qmul.ac.uk/~ccloy/files/bmvc_2012b.pdf (Accessed: 23 August 2016).
16. Chen, C.-H., Chang, Y.-C., Chen, T.-Y. and Wang, D.-J. (2008) 'People counting system for getting in/out of a bus based on video processing', *2008 Eighth International Conference on Intelligent Systems Design and Applications*, 3, pp. 565–569. doi: 10.1109/ISDA.2008.335.
17. Che, Z.-G., Chiang, T.-A. and Che, Z.-H. (2010) *Feed-Forward Neural Networks Training: A Comparision Between Genetic Algorithm and Back-Propogation Learning Algorithm*. Available at: http://www.ijicic.org/ijicic-10-03015.pdf (Accessed: 23 August 2016).
18. Conte, D., Foggia, P., Percannella, G., Tufano, F. and Vento, M. (2010) 'A method for counting people in crowded scenes', *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, , pp. 225–232. doi: 10.1109/AVSS.2010.78.

19. Conte, D., Foggia, P., Percannella, G., Tufano, F. and Vento, M. (2010) 'A method for counting moving people in video surveillance videos', *EURASIP Journal on Advances in Signal Processing*, 2010(1), p. 231240. doi: 10.1155/2010/231240.
20. Lefloch, D., Cheikh, F.A., Hardeberg, J.Y., Gouton, P. and Picot-Clemente, R. (2008) *Real-time people counting system using a single video camera*. Available at: http://spie.org/Publications/Proceedings/Paper/10.1117/12.766499 (Accessed: 23 August 2016).
21. Davies, A.C., Velastin, S.A. and Yin, J.H. (1995) 'Crowd monitoring using image processing',*Electronics & Communication Engineering Journal*, 7(1), pp. 37–47. doi: 10.1049/ecej:19950106.
22. Denman, S.P. (2013) *Improved detection and tracking of objects in surveillance video*. Available at: http://eprints.qut.edu.au/29328/ (Accessed: 23 August 2016).
23. Engelbrecht, A.P. (2007) *Computational intelligence: An introduction*. Available at: https://books.google.com.au/books?id=IZosIcgJMjUC&pg=PA501&lpg=PA501&dq=Optimization+and+global+minimization+methods+suitable+for+neural+networks.+p.+41&source=bl&ots=DvmuCaFiPg&sig=cJuqs5EQlj8l9mAf37uH23w3H-Q&hl=en&sa=X&ved=0ahUKEwiD3pa59NfOAhWCnZQKHRyEASsQ6AEIKTAB#v=onepage&q&f=false (Accessed: 23 August 2016).
24. Lin, S.-F., Chen, J.-Y. and Chao, H.-X. (2001) 'Estimation of number of people in crowded scenes using perspective transformation', *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 31(6), pp. 645–654. doi: 10.1109/3468.983420.
25. Gavin, H.P. (2011) *The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems*. Available at: https://scholar.google.com.au/citations?user=3yUXPoUAAAAJ&hl=en (Accessed: 23 August 2016).
26. Goyal, M. (2011) *Morphological Image Processing*. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.219.4602&rep=rep1&type=pdf (Accessed: 23 August 2016).

27. Günther, F. and Fritsch, S. (2012) *Neuralnet: Training of Neural Networks*. Available at: https://journal.r-project.org/archive/2010-1/RJournal_2010-1_Guenther+Fritsch.pdf (Accessed: 23 August 2016).
28. Gupta, B., Gupta, S. and Tiwari, A.K. (2010) *Face Detection Using Gabor Feature Extraction and Artificial Neural Network*. Available at: https://www.researchgate.net/publication/267223531_Face_Detection_Using_Gabor_Feature_Extraction_and_Artificial_Neural_Network (Accessed: 23 August 2016).
29. Hightower, J. and Borriello, G. (2001) 'Location systems for ubiquitous computing', *Computer*, 34(8), pp. 57–66. doi: 10.1109/2.940014.
30. Hudson, J.F.P. (1969) *Piecewise Linear Topology*. Available at: http://www.maths.ed.ac.uk/~aar/papers/hudson.pdf (Accessed: 23 August 2016).
31. Jayalakshmi, T. and Santhakumaran, A. (2011) 'Statistical normalization and back Propagationfor classification', *International Journal of Computer Theory and Engineering*, , pp. 89–93. doi: 10.7763/ijcte.2011.v3.288.
32. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M. and Shafer, S. (2000) 'Multi-camera multi-person tracking for EasyLiving', *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, , pp. 10–3. doi: 10.1109/VS.2000.856852
33. Kilambi, P., Ribnick, E., Joshi, A.J., Masoud, O. and Papanikolopoulos, N. (2008) 'Estimating pedestrian counts in groups', *Computer Vision and Image Understanding*, 110(1), pp. 43–59. doi: 10.1016/j.cviu.2007.02.003.
34. Kong, D., Gray, D. and Tao, H. (2006) 'A viewpoint invariant approach for crowd counting', *18th International Conference on Pattern Recognition (ICPR'06)*, 3, pp. 1187–1190. doi: 10.1109/ICPR.2006.197.
35. Kumar, P. (2016) *Object counting and density calculation using Matlab*. Available at: https://www.scribd.com/doc/305009952/Object-Counting-and-Density-Calculation-Using-Matlab (Accessed: 23 August 2016).
36. Lindeberg, T. (1996) 'Edge detection and ridge detection with automatic scale selection', *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, , pp. 465–470. doi: 10.1109/CVPR.1996.517113.

37. Babu, J. and Vani, S. (2011) *Performance Analysis of Classifying Unlabeled Data from Multiple Data Sources*. Available at: http://ijcsit.com/docs/Volume%202/vol2issue4/ijcsit2011020472.pdf (Accessed: 23 August 2016).
38. Masoud, O. and Papanikolopoulos, N.P. (2001) 'A novel method for tracking and counting pedestrians in real-time using a single camera', *IEEE Transactions on Vehicular Technology*, 50(5), pp. 1267–1278. doi: 10.1109/25.950328.
39. Montufar, G., Pascanu, R., Cho, K. and Bengio, Y. (2005) *On the Number of Linear Regions of Deep Neural Networks*. Available at: https://arxiv.org/pdf/1402.1869.pdf (Accessed: 23 August 2016).
40. Reis, J.V.D. dos (2014) *Image Descriptors for Counting People with Uncalibrated Cameras*. Available at: https://sigarra.up.pt/feup/pt/pub_geral.show_file?pi_gdoc_id=371540 (Accessed: 23 August 2016).
41. Ryan, D., Denman, S., Fookes, C. and Sridharan, S. (2014) 'Scene invariant multi camera crowd counting', *Pattern Recognition Letters*, 44, pp. 98–112. doi: 10.1016/j.patrec.2013.10.002.
42. Cassidy, S.L., Dix, K.M. and Jenkins, T. (1983) 'Evaluation of a testicular sperm head counting technique using rats exposed to dimethoxyethyl phthalate (DMEP), glycerol?-monochlorohydrin (GMCH), epichlorohydrin (ECH), formaldehyde (FA), or methyl methanesulphonate (MMS)', *Archives of Toxicology*, 53(1), pp. 71–78. doi: 10.1007/bf01460003.
43. Saleh, S.A.M., Suandi, S.A. and Ibrahim, H. (2015) 'Recent survey on crowd density estimation and counting for visual surveillance', *Engineering Applications of Artificial Intelligence*, 41, pp. 103–114. doi: 10.1016/j.engappai.2015.01.007.
44. Bansal, A. and Venkatesh, K.S. (2015) 'People Counting in High Density Crowds from Still Images',*International Journal of Computer and Electrical Engineering*, 7, pp. 316–324. doi: 10.17706/ijcee.2015.7.5.316-324.

45. Liu, Q. and Peng, G. (2007) 'A robust skin color based face detection algorithm', *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, 2, pp. 525–528. doi: 10.1109/CAR.2010.5456614.
46. KaewTraKulPong, P. and Bowden, R. (2002) 'An improved Adaptive background mixture model for real-time tracking with shadow detection', in *Video-Based Surveillance Systems*. Springer Science + Business Media, pp. 135–144.
47. Stauffer, C. and Grimson, r W.E.L. (1999) *Adaptive background mixture models for real-time tracking*. Available at: http://www.ai.mit.edu/projects/vsam/Publications/stauffer_cvpr98_track.pdf (Accessed: 23 August 2016).
48. Warner, B.A. (2002) 'Basic engineering data collection and analysis', *The American Statistician*, 56(1), pp. 76–77. doi: 10.1198/tas.2002.s130.
49. Sweeney, L. and Gross, R. (2005) *Mining images in publicly-available cameras for homeland security (PDF)*. Available at: https://www.semanticscholar.org/paper/Mining-Images-in-Publicly-Available-Cameras-for-Sweeney-Gross/60c6d533d16625789484a97b1ec7996cd3ff06a1/pdf (Accessed: 23 August 2016).
50. Umbaugh, S.E. (2010) *Digital image processing and analysis: Human and computer vision applications with CVIPtools, Second edition*. Available at: https://books.google.com.au/books/about/Digital_Image_Processing_and_Analysis.html?id=UQTMw5uoGHgC&source=kp_cover&redir_esc=y (Accessed: 23 August 2016).
51. Cho, S.-Y., Chow, T.W.S. and Leung, C.-T. (1999) 'A neural-based crowd estimation by hybrid global learning algorithm', *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 29(4), pp. 535–541. doi: 10.1109/3477.775269.
52. Zhang, J., Tan, B., Sha, F. and He, L. (2011) 'Predicting pedestrian counts in crowded scenes with rich and high-dimensional features', *IEEE Transactions on Intelligent Transportation Systems*, 12(4), pp. 1037–1046. doi: 10.1109/tits.2011.2132759.
53. Fricke, D. (2000) *Nine dead at Pearl Jam concert*. Available at: http://www.rollingstone.com/music/news/nine-dead-at-pearl-jam-concert-20000817 (Accessed: 23 August 2016).

54. Vincent, P. (2013) *Guilt over Australian fan's death felt like 'murder': Limp Bizkit*. Available at: http://www.smh.com.au/entertainment/music/guilt-over-australian-fans-death-felt-like-murder-limp-bizkit-20131003-2uukt.html (Accessed: 23 August 2016).
55. McDonell, C. correspondent S. and wires (2015) *Overcrowded Shanghai waterfront leads to NYE stampede killing 36*. Available at: http://www.abc.net.au/news/2015-01-01/35-dead-in-shanghai-new-year-stampede/5995536 (Accessed: 23 August 2016).
56. Video and stampede, I. temple (no date) *More than 60 killed in India temple stampede*. Available at: http://edition.cnn.com/2010/WORLD/asiapcf/03/04/india.temple.deaths.uttarpradesh (Accessed: 23 August 2016).
57. Connolly, K. (2010) *Festivalgoers killed in stampede at Love Parade in Germany*. Available at: https://www.theguardian.com/world/2010/jul/24/love-parade-festival-tunnel-stampede (Accessed: 23 August 2016).