University of Stirling

Faculty of Natural Sciences

Division of Computing Science and Mathematics

Classification of crowd motion using Computer Vision and ML

Author:
Chaitanya Kumar Reddy Alva

Supervisor:
Dr. Deepayan Bhowmik

Dissertation submitted in partial fulfillment for the degree of Master of Science in *BigData*

September 2020

Abstract

Summary of the dissertation ***within one page***.

This template starts the page numbering at the foot of this page. While you are printing drafts, you might find it useful to add the printing date and time into the footer – to help you, and your supervisor, tell which version is most current.

It is suggested that the abstract be structured as follows:

- Problem: What you tackled, and why this needed a solution
- Objectives: What you set out to achieve, and how this addressed the problem
- Methodology: How you went about solving the problem
- Achievements: What you managed to achieve, and how far it meets your objectives.

Attestation

I understand the nature of plagiarism, and I am aware of the University's policy on this.

I certify that this dissertation reports original work by me during my University project except for the following (*adjust according to the circumstances*):

- The technology review in Section 2.5 was largely taken from [17].
- The code discussed in Section 3.1 was created by Acme Corporation (`www.acme-corp.com/JavaExpert`) and was used in accordance with the licence supplied.
- The code discussed in Section 3.5 was written by my supervisor.
- The code discussed in Section 4.2 was developed by me during a vacation placement with the collaborating company. In addition, this used ideas I had already developed in my own time.

Signature: *(you must delete this, then sign and date this page)* **Date**

Acknowledgements

Acknowledge anyone that you wish to thank who has helped you in your work or supported you in any way: such as your supervisor, technical support staff, fellow students, external organisations or family. Acknowledge the source of any work that is not your own.

Contents

List of Figures

1 Introduction

Evolution of technology, artificial intelligence and robotics helped the world to achieve new targets in the field of security and surveillance. The combination of machine learning and surveillance emerged as a powerful tool to tackle crime, illegal activities and violent protests. In the recent years, we experienced many such activities that made us to understand the importance and necessity of automated video surveillance. With the help of computer vision, detecting people in the frame, counting the people in a dense scene, abnormal behaviour detection and motion analysis in surveillance videos is done without any manual intervention. Crowd motion analysis and abnormal behaviour detection have always been a challenging task in this field. Reason being the number of independent factors that define the motion of the individual. Analysing the motion of the crowd can avoid many voluntary or involuntary violence, riots, traffic jams and stampede.

1.1 General Context and motivation

As mentioned in [1], the main objectives of automated surveillance video analysis are continuous monitoring, reduction in laborious human task, object identification or action recognition and crowd analysis. This paper talks about detecting different types of crowd motion and abnormal behaviour tracking using CNN. Most of the study on analysing the crowd is done on the following areas.

- Counting the crowd.
- Types of the crowd based on density.
- Detecting the motion in the frame.
- Identifying the types of motion.

1.1.1 Counting the crowd.

Counting the crowd is a very important in order to maintain the safety and security. It helps to plan the events, traffic and the capacity of any situation. But, counting dense crowd is a difficult task. As mentioned in [2], more than 17% of the total papers written on crowd analysis are published on crowd counting. For example, [3] generalise different types of crowd counting and different algorithms used in the past while proposing a new approach of using the statistics of thespatio-temporal wavelet sub-bands. [4] uses a multi source (identifying different parts of the body in the frames from different algorithms) and Markov Random Field to count the people in the dense crowd.

1.1.2 Types of the crowd based on density

It is important to categorise the type of the crowd to understand the dynamics of the motion. Moore [5] suggests, the crowd can be treated as particles in fluid dynamics and the crowd is of 3 types, microscopic, mesoscopic and macroscopic based on the density. Microscopic view of crowd through a hydrodynamic lens implies understanding the flow of every individual in crowd and this is specific to limited number of individuals in the frame. Mesoscopic view implies more number of people in a frame. Macroscopic view implies the frame filled with people. The personal and interaction forces in each case are different which in turn drive the motion of the crowd. To further explain, the interaction force is very less in a microscopic view but very high in macroscopic view.

1.1.3 Detecting the motion in the frame

Detecting the motion in the frame can be done either by training a model which involves feeding the motion images into a CNN architecture or without training by just tracking every point in the frame using optical flow. Santoro [6] did optical flow computation with the help of Shi-Tomasi Corner Detection and Lucas–Kanade algorithm to detect the motion of the crowd. Where as [7] uses motion information Images (MII) to train a CNN model for the motion and abnormality detection

1.1.4 Identifying the types of motion

Identifying the types of the motion can be a very useful in order to understand the crowd behaviour, planning an event, avoiding traffic jams and predicting the abnormal motion. Wei [8] trained 2 VGG16 CNN architecture models to detect the type of the crowd whether it is homogenous, heterogeneous or violent crowd. [9] studies the stability with the help of Tylor's theorem and Jacobean matrix and identifies the crowd motion to be of 5 generic types i.e. Lanes, arc/circle, fountainheads, bottlenecks and blocks.

1.2 Aim and Objectives

Define the scope and objectives of your project.

1.3 Achievements

Summarise what this project has achieved. Avoid terms like I achieved this or that.

1.4 Overview of Dissertation

Briefly overview the contents of what follows in the dissertation.

2 Background

Computer vision evolved from many complex theories, algorithms and models. This paper mainly talks about the video surveillance. This section helps to understand the required technical details confined to this area.

2.1 Optical Flow

Optical flow can possibly be one of the most important concepts of computer vision. Optical flow is used to find the pattern in the movement of the objects from one frame to another. This is widely used in the fields like robotics, image processing, motion detection, object segmentation etc. Videos are the series of images. These images can be independent from one another. But, in the real time, a video captures consecutive change in the pixels in certain duration of time. There are many algorithms which discuss the relation between these pixels in two different frames. [10] discuss various types of optical flow algorithms and evaluates them. This paper concludes that Lucas Kanade Algorithm is best among the other 8 optical flow algorithms.

Optical flow diagrams are usually denoted by the vectors pointing the change from frame F1 to frame F2. But in real time, it is easy to concentrate on only those points which provide more insights. For example movement of the hand from F1 to F2 changes hundreds of pixels and can be redundant. Rather it is simple and more appropriate to see the flow of only those pixels at the corner of the hand. Thus, Corner detection algorithms are used to reduce the complexity and improve the performance of the algorithms.

2.1.1 Corner Detection

This paper trails 2 types of corner detection techniques to check the best possibility for the model.

- Shi-Tomasi Corner detection.
- FAST Corner detection.

Shi Tomasi Corner detection algorithm is similar to Harris Corner Detector. it is widely used in detecting the interest points and feature descriptors. Interest points can be corners edges and blobs and are invariant to rotation, translation, intensity and scale changes. Only difference in harris corner detection and Shi Tomasi corner detection is the computed R value (used to detect the corner). FAST (Features from Accelerated Segment Test) on the other hand uses a different technique to predict not only the corners but also the edges based on the colour intensity and the threshold.

2.1.2 Lucas Kanade Algorithm

In the conclusion of [10], we can see that Lucas Kanade Algorithm is the one of the best algorithm to detect the optical flow. The assumption of Lucas Kanade algorithm is the flow of the local neighbourhood of the pixel is constant. It combines all the information from the surrounding pixels and often solves the inherent ambiguity of the optical flow equation. It is also considered to be less sensitive to the noise.

2.2 Density based clustering

Clustering in general is combining a group of similar objects based on their similarities like shape, angle, magnitude and position. In order to reduce the memory consumption of the CPU/GPU it is important to consider those points which are critical to the analysis. Thus, clustering the points and vectors based on the position and direction helps to combine the similar points and vectors to predict the movement of the crowd. In this paper we have considered using 2 types of density based clustering.

- DBSCAN.
- OPTICS.

2.3 Convolutional Neural networks

Convolutional neural networks are the advanced concept of neural networks which gives computers the ability to understand the images and videos. CNNs currently are being used in a wide range of application like Robotics, Face detection, Crowd detection, Weather study, Advertising, Environmental studies etc. Every neurone in the CNN has the learnable weights. They are initialised with random weights and can be trained to develop a model. CNN are comprised of below 3 topics.

- Convolution Networks (ConvNets).
- Pooling.
- Fully Connected Layers.

2.3.1 Convolution Networks (ConvNets)

Convolution Networks also called as ConvNets is the process of changing the pixels of the image using filters. The image is a matrix of pixels and a filter/kernel is used to alter the pixel value with matrix multiplication. This filter is applied on the whole image by striding through the image. There are different filters for different types of results.

2.3.2 Pooling

Pooling is the process of reducing the size of the image with the help of a filter. The pooling is usually of 2 types, Average pooling and Max pooling. Image is reduced to a desired size by the filter by taking the average of the pixels or Max pixel depending on the pooling technique.

2.3.3 Fully Connected Layers

Fully connected layers are the neural networks which has the 1D array of the ConvNets as the inputs and a series of different hidden layers which are fully connected. The output of these networks are the classification nodes which can either be integers or One Hot encoded values predicting the classification. The prediction of the classification is usually done by SoftMax (Picks the highest probability node).

2.3.4 Notable CNN Architectures

There are few CNN architectures which are available as the modules. These modules are pre-trained and can be directly implements provided the inputs and outputs are exactly same as expected by the module. PyTorch library can be explored to find the list and implementation of these networks. The list of these modules is shown below.

- AlexNet
- VGG
- ResNet

- SqueezeNet
- DenseNet
- Inception v3
- GoogLeNet
- ShuffleNet v2
- MobileNet v2
- ResNeXt
- Wide ResNet
- MNASNet

3 State-of-the-Art

Crowd motion analysis deals with a combination of computer vision, Image processing and machine learning techniques. This section provides in depth knowledge of related work and state of the art techniques by explaining the recent and ground braking scientific papers in this field. More specifically the motivation introduction and the techniques used in these papers.

3.1 Crowd analysis using optical flow

The papers [6] [11] gives a generic and very efficient way of tracking the crowd motion in the videos. As shown in the figure 3.1 below, the process of crowd tracking is done in 4 steps. KLT feature tracker is used to do the optical flow estimation.KLT feature tracker is the combination of Shi- Tomasi corner detection and the famous Lucas Kanade optical flow algorithm. Shi-Tomasi corner detection technique is used to identify the interesting corners of the frame. A section of surrounding pixels of these points are also added to the tracking. This step is followed by tracking the points in the consequent frames. The tracking is done with the help of Lucas Kanade algorithm. In addition to the change in the points, magnitude and the angle of the vector is also calculated at this point of time.

Now that all the vectors are derived from frame fk and fk+1, block partitioning is done on the whole frame. The frame is divided in to multiple blocks and the vectors in the specific blocks are clustered based on the angle and magnitude. This clustering is done with the help of DBSCAN algorithm. All the points which are considered to be one group are marked with single vector. Any person leaving the crowd or joining the crowd is considered as separate or single block accordingly. The result of this paper are shown in the figure 3.2. The person behind the crowd is considered as separate group and the tracking is done multiple times to get the flow.

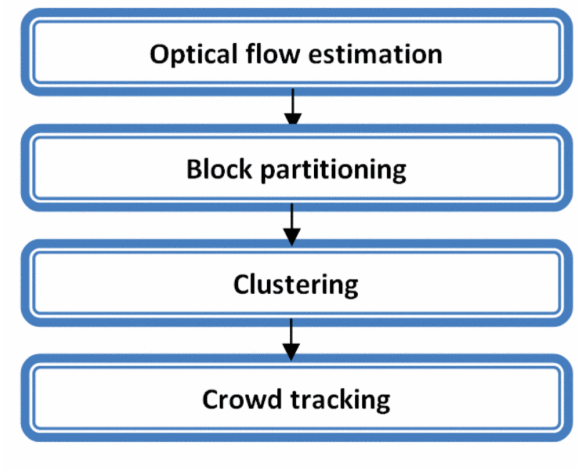

Figure 3.1: Data Flow Diagram

Figure 3.2: Density based partitioning and crowd tracking

The paper [12] is an interesting use of optical flow to detect the dominant motion in the crowded scenes. This paper suggests a combination of both Shi Tomasi corner detection algorithm and FAST corner detection to identify the interesting points in the frame. Keeping track on interesting points in multiple frames, the trajectory is captured. New feature points are added in every 5 frames to handle the load. The new feature points which are close to the old points are discarded. By getting the trajectories of all the points, a new clustering framework Longest Common Subsequences (LCSS) is introduced. With the help of this framework, multiple trajectories are compared for the matching points and the dominant path is captured by clustering trajectories.

Figure 3.3: Dominant motion detection results

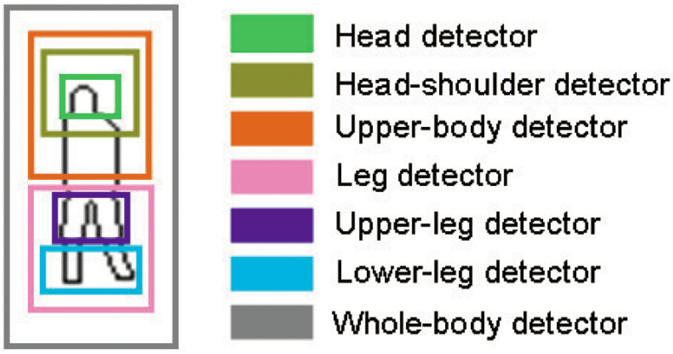

Figure 3.4: Multi Feature multi detector model

The results of this model are shown in the figure 3.3.

3.2 Crowd counting

Crowd counting is a difficult task and the accuracy of the proposed models depends on the scene. In case of large density of crowd, it is extremely hard to track the crowd for counting. Most recent paper [13] suggests that crowd counting can be done in 2 ways. The first is by using detection based models, where the crowd is being tracked with the help of body parts or the shape and the count is produced from the tracking model. The second is Regression based models, where the model predicts the number of the crowd with out tracking them. This model is based on developing a density map and estimating the count from the produced density map. A part from these 2 methods there is another method based on CNN, which also produced promising results. But, in the CNN models, there are cases where the system predicted various different objects as human heads causing a huge difference in the count. [13] deals with the combination of density based model and the CNN in order to fix the on going issue with the CNN models.

[14] is another interesting paper which can be grouped into the detection based models. In this paper, the writer creates a part-template tree with the human postures in different angles, poses and shapes. A hierarchical part-template matching algorithm is used to estimate human shapes and poses by matching local image. Multiple detectors are used to detect the multiple body shapes as shown in the figure 3.4. The segmentation is done based on these multiple detectors. Background subtraction is used to evaluate the model and produced promising results. [15] can be grouped under the regression based model of counting crowds. This paper suggests the motion segmentation of the crowd clustered based on the direction. The count on the each direction is estimated using the Gaussian process. This model is explained in the figure 3.5

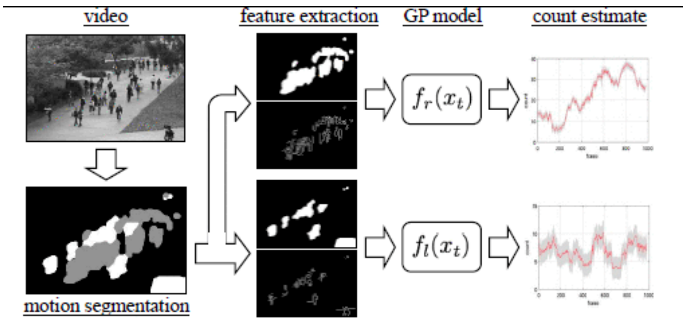

Figure 3.5: Regression Based crowd counting

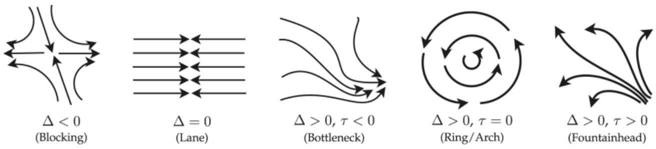

Figure 3.6: Types of motion based on the product and sum of eiganvalues

3.3 Motion detection and classification

In order to understand the behaviour of the crowd, it is important to detect the motion and also classify them. For example to check if the vehicles are moving in the right path, people following the suggested path, person walking in restricted area etc. The crowd motion can be categorised into different groups based on the scene. If the scene is to identify the anomaly in the crowd motion for example in the traffic, the motion can be grouped into Lanes, Arcs, blocks. In case of entering or exiting enclosed buildings the motion can be Bottlenecks or Fountainheads. In case of violent protest, it can be categorised into converging or diverging. In all the cases, it is important to study the type of the motion in order to tackle the situation.

[9] states that crowd motion can be categorised into 5 types: Lanes, Arcs, Bottlenecks, Fountainheads and Blocks. This paper suggests a model which can categorise the videos with out a training model. It used Particle advection to pick the interesting points in the frame and track these points as a spacio-temporal data. The motion is further produced in the equation using Taylor theorem. The stability of the motion is detected by calculating Jacobian Matrix. The eigenvalues of this matrix is used to classify the motion as shown in the figure 3.6.

Identifying the type of the crowd is key to understand the type of the action, scene of the incident and actions of the people. This paper [8] categorise to crowd into 3 types based on a model called BMO model (Behaviour, Mood and Organisation model). This model is a rule based model which helps

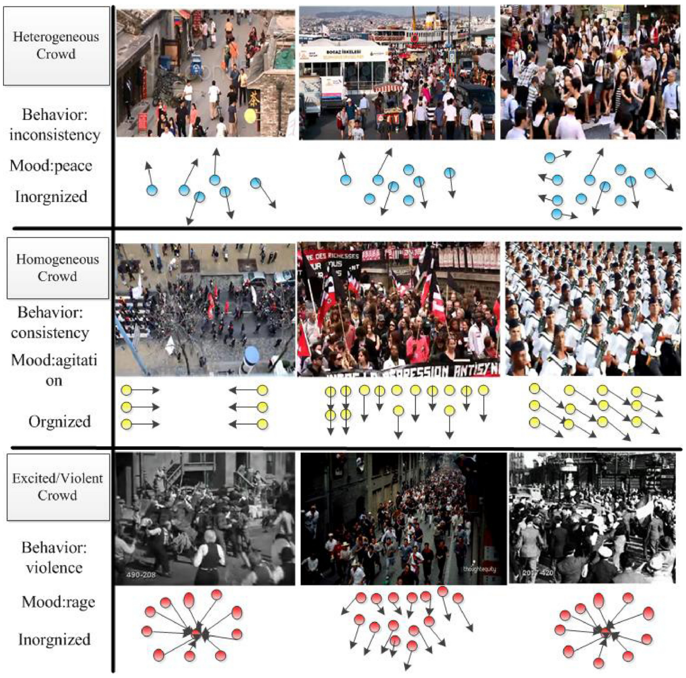

Figure 3.7: Crowd categorisation using 2 deep networks

to detect the crowd motion to be grouped into Heterogenous, Homogenous and Violent crowds. To implement this model, they trained 2 very deep VGG networks and combined the FCN layers of both the networks to predict the type of the crowd. The inputs to these networks are the motion map and key frame. The results and the implementation of this model are shown in the figure 3.7

3.4 Anomaly detection in the crowd motion

Another very interesting and key papers published are focused on the Anomaly detection. This type of papers focus on identifying unexpected behaviour in the frame. For example the paper [16] focuses on 3 types of anomalies, namely: Point Anomaly which points a single object in the frame. This can be an unexpected motion or sudden change in the magnitude of single object. The second type id the collective anomaly, where most of the objects in the frame experience sudden drift in the direction and velocity. This type of anomaly is usually found in the riots, explosions etc. The third type of anomaly is contextual anomaly which can be unexpected shaped object in the frame etc. The figure 3.8 explains the different types of the anomalies clearly.

This paper implements the model with the footages from the stable surveillance cameras and techniques like background reduction. It also proposes a new way of gathering the features like direction, change in the points, dis-

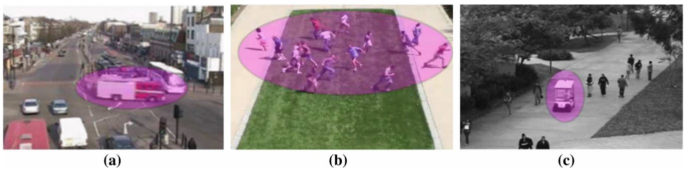

Figure 3.8: Types of anomalies

tance etc. These features are further classified using k-means clustering and distance calculation to predict the motion to be expected or unexpected.

4 Feature Detection

Feature detection using open CV is the a very useful and important technique is most of the areas which deals with images and videos. Each image or the frame is the combination of pixels and each each pixel is the number that represents a colour. For a computer it is extremely difficult to understand the difference between these numbers and thus, feature detection is a complex yet interesting topic to understand. This paper tries to implements 3 different corner detection techniques 1) Harris Corner detection, 2) Shi Tomasi corner detection. 3) FAST algorithm for corner detection. The advantages and disadvantages of these techniques are discussed further.

4.1 Harris Corner Detection

This corner detection technique was first introduced by Chris Harris & Mike Stephens in their paper [17] in 1988. Idea behind this technique is to find the difference between the intensity for a displacement of (u, v) in all directions. The mathematical equation 4.1 for the same is given below.

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2 \tag{4.1}$$

Window function *w(x, y)* is either a rectangular window or gaussian window which gives weights to pixels underneath. The corners are detected by maximising the *E(u, v)* which means maximising the second term by using Tylor's theorem as shown in the equations 4.2 & 4.3

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{4.2}$$

where

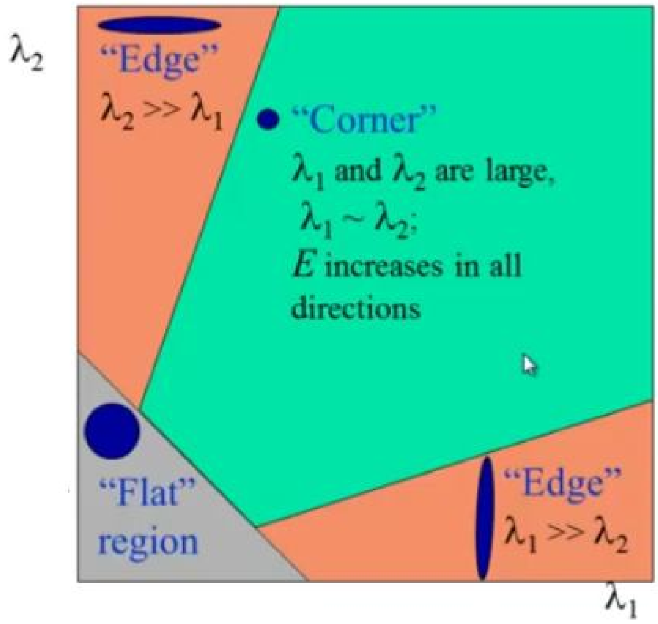

Figure 4.1: Harris corner detection using eigenvalues

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \tag{4.3}$$

Here, I_x and I_y are image derivatives in x and y directions respectively. *R* value is calculated from the eigenvalues of the matrix *M* from the equation 4.4

$$R = det(M) - K(trace(M))^2 \tag{4.4}$$

where

- det(M) = $\lambda_1 \lambda_2$.
- trace(M) = $\lambda_1 + \lambda_2$.
- λ_1 and λ_2 are the eigenvalues of M.
- k - Harris detector free parameter in the equation.

edge, corner and flat region in the image are detected with the help of eigenvalues as shown in the figure 4.1

4.2 Shi Tomasi Corner Detection

Shi Tomasi Corner detection was first proposed by J. Shi and C. Tomasi in the paper [18] in 1994. This approach is a small modification to the Harris Corner detection in calculating the *R* value. As mentioned before the *R* value is calculated by 4.4 But, as per Shi Tomasi Corner detection, the *R* value is calculated by minimising the product of eigenvalues as shown in 4.5.

$$R = min(\lambda_1, \lambda_2) \tag{4.5}$$

If this value is greater than the threshold, then it is considered as the corner.

4.3 FAST Algorithm for Corner Detection

5 Density Based Clustering

6 Lucas-Kanade Algorithm

7 Convolutional Neural Networks

8 Machine Learning

9 Implementation

The technical body of the dissertation consists of a number of chapters (just one here, but there will usually be more). Follow a logical structure in how you present your work. This will usually be the phases of the software development cycle, the modules of your system, etc. ***However, please do not write your dissertation to read like a diary.***

Include a chapter demonstrating what you have achieved and how your system is used in practice – for example showing a typical session as a series of pasted in screen shots, with an accompanying commentary.

You ***should*** also include a chapter explaining how you obtained feedback from your “customer” or potential users of your system, what feedback you actually obtained, and your analysis and comments.

9.1 First Section

Subdivide your text into sections, using the `\section` command.

9.1.1 First Subsection

If necessary, also use subsections. Subsections are entered using the `\subsection` command (all these heading styles are self-numbering).

9.1.2 Second Subsection

And, as required, more subsections.

9.2 Bulleted and Numbered Lists

Note: This section begins with the code `\section{Bulleted and Numbered Lists}` in the `.tex` file.

Bulleted or numbered lists are entered using the `itemize` and `enumerate` environments, respectively. An **environment** in LaTeX is a block of code in between a `\begin` and `\end` command. For example, the code

```
\begin{itemize}
    \item Up
    \item Down
    \item Left
    \item Right
\end{itemize}
```

would produce the following list:

- Up
- Down
- Left
- Right

The indentation is not necessary (the pdf will look the same even it the `.tex` file does not use indents), but it helps make the code easier to read.

If the `enumerate` environment is used instead, the bullets are replaced by numbers. For example, the code

```
\begin{enumerate}
    \item Up
    \item Down
    \item Left
    \item Right
\end{enumerate}
```

produces the list

1. Up
2. Down
3. Left
4. Right

9.3 Figures and Captions

As an example of a figure, consider Figure 9.1. Captions are entered using the `figure` environment (read the previous section for information about environments in general). The code

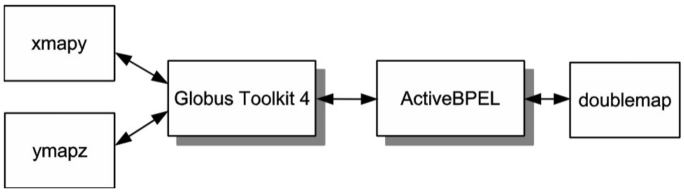

Figure 9.1: Highly Technical Diagram

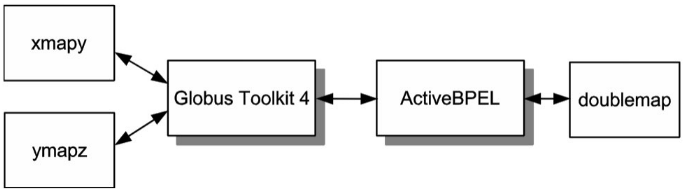

Figure 9.2: Highly Technical Diagram two

```
\begin{figure}[h]
    \center\includegraphics[width=12cm]{image.png}
    \caption{Highly Technical Diagram}
    \label{mylovelydiagram}
\end{figure}
```

will produce the following figure **if the file *image.png* is in the same folder as your `.tex` file**.

The `[tb]` direction after the beginning of the environment causes the figure to be placed "here" in the text (at least approximately – sometimes TEX will move the figure slightly if the spacing does not work well in exactly the given location). For large figures, use `[t]` or `[b]` instead to place the figure at the top or bottom of a page. You can also leave off the `[h]` entirely to have TEX make its best guess for where the figure should go.

The `\includegraphics` command puts an image file from your computer into your finished pdf. **If there is no file with the given name in the folder with your `.tex` file, your document will not compile at all.** The bracket text `[width=12cm]` is optional; without it, TEX will use the normal size of the image. Sometimes this will be far too large, so it is a good idea to specify a width directly.

Figures have automatic numbering, and it is possible to make cross-references to figures. The code `\Fig{mylovelydiagram}` will create a link to Fig. 9.1 in the text with the number of that figure. You can change the text "mylovelydiagram" to be anything you want – it never appears in the final pdf.

9.4 Source Code

To include programming source code in your document, use the `lstlisting` environment. The LaTeX code

```
\begin{lstlisting}[language=Python, frame=single]
    def factorial(n):
        if n == 0:  return 1
        else:  return n * factorial(n-1)
\end{lstlisting}
```

produces the following in the pdf:

Listing 9.1: Some Python code

```
def factorial(n):
        if n == 0: return 1
        else: return n * factorial(n-1)
```

You can change `language=Python` to `language=Java`, etc., for different programming languages. The `frame=single` can be removed if you do not want the border around your code snippet. See https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings for syntax coloring and other option. You can reference the listing with the command, `\ref{lst:label}`, as in see listing 9.1.

10 Results and Discussions

10.1 Dataset

Description of the dataset(s)

10.2 Experimental setup

Say what is the experimental set up, parameters that were used.

10.3 Results

Stand back and evaluate what you have achieved and how well you have met the objectives. Evaluate your achievements against your objectives in Section 1.2. Demonstrate that you have tackled the project in a professional manner.

The previous paragraph demonstrates the use of automatic cross-references: The "1.2" is a *cross-reference* to the text in a numbered item of the document; you do not type it as `1.2` but by using the `\Sec` command. The number that appears here will change automatically if the number on the referred-to section is altered, for example, if a chapter or section is added or deleted before it. Cross-references to section are entered with the `\ref` command just like for figures. The TEX code above reads

```
Evaluate your achievements against your objectives
in section \ref{objectives sec}.
```

For this to work, the code for the text on page **??** must read

```
\section{Scope and Objectives} \label{objectives sec}
```

As with figure labels, the text inside of `\label` and `\Fig` never appears in the final pdf; you can make it whatever you want as long as you use the same text in each to complete the reference.

10.4 Discussions

Analyse your results and discuss it by including your insight. For example why the results are behaving like this, why there is an outlier etc.

11 Conclusions & Future Work

11.1 Conclusions

Summarise what you have achieved. Again do not say I achieved this. Say what the project has achieved.

11.2 Future Work

Explain any limitations in your results and how things might be improved. Discuss how your work might be developed further. Reflect on your results in isolation and in relation to what others have achieved in the same field. This self-analysis is particularly important. You should give a critical evaluation of what went well, and what might be improved.

Bibliography

[1] G. Sreenu and M. S. Durai, “Intelligent video surveillance: a review through deep learning techniques for crowd analysis,” *Journal of Big Data*, vol. 6, no. 1, p. 48, 2019.

[2] G. Tripathi, K. Singh, and D. K. Vishwakarma, “Convolutional neural networks for crowd behaviour analysis: a survey,” *The Visual Computer*, vol. 35, no. 5, pp. 753–776, 2019.

[3] D. Bhowmik and A. Wallace, “Statistical t+2d subband modelling for crowd counting,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1533–1537.

[4] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, “Multi-source multi-scale counting in extremely dense crowd images,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2547–2554.

[5] B. E. Moore, S. Ali, R. Mehran, and M. Shah, “Visual crowd surveillance through a hydrodynamics lens,” *Communications of the ACM*, vol. 54, no. 12, pp. 64–73, 2011.

[6] F. Santoro, S. Pedro, Z.-H. Tan, and T. B. Moeslund, “Crowd analysis by using optical flow and density based clustering,” in *2010 18th European Signal Processing Conference*. IEEE, 2010, pp. 269–273.

[7] C. Direkoglu, “Abnormal crowd behavior detection using motion information images and convolutional neural networks,” *IEEE Access*, vol. 8, pp. 80 408–80 416, 2020.

[8] X. Wei, J. Du, Z. Xue, M. Liang, Y. Geng, X. Xu, and J. Lee, “A very deep two-stream network for crowd type recognition,” *Neurocomputing*, vol. 396, pp. 522–533, 2020.

[9] B. Solmaz, B. E. Moore, and M. Shah, “Identifying behaviors in crowd scenes using stability analysis for dynamical systems,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 10, pp. 2064–2070, 2012.

[10] B. Galvin, B. McCane, K. Novins, D. Mason, S. Mills *et al.*, “Recovering motion fields: An evaluation of eight optical flow algorithms.” in *BMVC*, vol. 98, 1998, pp. 195–204.

[11] N. Rosandić, T. Petrović, R. Tušek, Z. Kučiš, A. Štruklec, M. Liashuha, and A. Alatas, “Crowd analysis by using optical flow and density based clustering,” 2019.

[12] A. M. Cheriyadat and R. J. Radke, "Detecting dominant motions in dense crowds," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 568–581, 2008.

[13] J. Chen, W. Su, and Z. Wang, "Crowd counting with crowd attention convolutional neural network," *Neurocomputing*, vol. 382, pp. 210–220, 2020.

[14] Z. Lin and L. S. Davis, "Shape-based human detection and segmentation via hierarchical part-template matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 604–618, 2010.

[15] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–7.

[16] S. D. Bansod and A. V. Nandedkar, "Crowd anomaly detection and localization using histogram of magnitude and momentum," *The Visual Computer*, vol. 36, no. 3, pp. 609–620, 2020.

[17] C. G. Harris, M. Stephens *et al.*, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[18] J. Shi *et al.*, "Good features to track," in *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE, 1994, pp. 593–600.

Appendix 1

You may have one or more appendices containing detail, bulky or reference material that is relevant though supplementary to the main text: perhaps additional specifications, tables or diagrams that would distract the reader if placed in the main part of the dissertation. Make sure that you place appropriate cross-references in the main text to direct the reader to the relevant appendices.

Note that you should ***not*** *include your program listings as an appendix or appendices.* You should submit one copy of such bulky text as a separate item, perhaps on a disk.

Appendix 2 – User guide

If you produced software that is intended for others to use, or that others may wish to extend/improve, then a user guide and an installation guide appendices are ***essential***.

Appendix 3 – Installation guide

If you produced software that is intended for others to use, or that others may wish to extend/improve, then a user guide and an installation guide appendices are ***essential***.