

University of Stirling

Faculty of Natural Sciences

Division of Computing Science and Mathematics

Title of your project

Author:
Your Name

Supervisor:
Dr. Deepayan Bhowmik

**Dissertation submitted in partial fulfillment for the degree of
Master of Science in *(insert your degree title)***

September 202x



**UNIVERSITY of
STIRLING**



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

Abstract	i
Attestation	ii
Acknowledgements	iii
1 Introduction	1
1.1 General Context and motivation	1
1.1.1 Counting the crowd.	2
1.1.2 Types of the crowd based on density	2
1.1.3 Detecting the motion in the frame	2
1.1.4 Identifying the types of motion	2
1.2 Aim and Objectives	3
1.3 Achievements	3
1.4 Overview of Dissertation	3
2 Background	4
2.1 Optical Flow	4
2.1.1 Corner Detection	4
2.1.2 Lucas Kanade Algorithm	5
2.2 Convolutional Neural networks	5
2.2.1 Convolution Networks	5
2.2.2 Pooling	5
2.2.3 Fully Connected Layers	5
2.2.4 Notable CNN Architectures	5
3 State-of-the-Art	6
4 Technical Chapters (change this to something appropriate)	7

4.1	First Section	7
4.1.1	First Subsection	7
4.1.2	Second Subsection	7
4.2	Bulleted and Numbered Lists	8
4.3	Figures and Captions	8
4.4	Source Code	10
5	Results and Discussions	11
5.1	Dataset	11
5.2	Experimental setup	11
5.3	Results	11
5.4	Discussions	12
6	Conclusions & Future Work	13
6.1	Conclusions	13
6.2	Future Work	13
	Appendix 1	15
	Appendix 2 – User guide	16
	Appendix 3 – Installation guide	17

List of Figures

4.1	Highly Technical Diagram	9
4.2	Highly Technical Diagram two	9

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 the spatio-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 (MI) 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

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).

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 Convolutional Neural networks

2.2.1 Convolution Networks

2.2.2 Pooling

2.2.3 Fully Connected Layers

2.2.4 Notable CNN Architectures

3

State-of-the-Art

Summarise current knowledge and what others have done in the various topics of your dissertation – in the application area and in the various technologies that you might have used or did use.

Do a good literature survey and maximise academic references. All references must be entered in your *reference.bib* file and call the variable name here similar to this example, Bhowmik and Feng [11] discussed Blockchain here. Their method explained how watermarking can be used for content authentication purposes [11].

You can create separate sections for various sub topic.

4

Technical Chapters (change this to something appropriate)

Note: This part of the dissertation will normally be expanded to be a *series of chapters*.

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.

4.1 First Section

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

4.1.1 First Subsection

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

4.1.2 Second Subsection

And, as required, more subsections.

4.2 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 if 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

4.3 Figures and Captions

As an example of a figure, consider Figure 4.1. Captions are entered using the `figure` environment (read the previous section for information about environments in general). The code

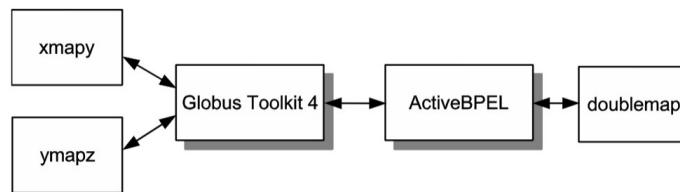


Figure 4.1: Highly Technical Diagram

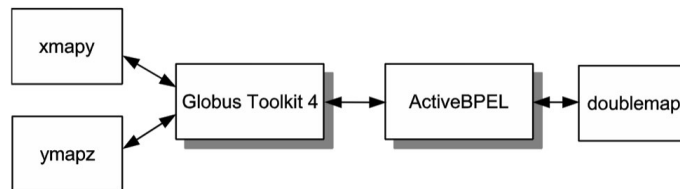


Figure 4.2: Highly Technical Diagram two

```

\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 T_EX 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 T_EX 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, T_EX 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. 4.1 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.

4.4 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 4.1: 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 4.1.

5

Results and Discussions

5.1 Dataset

Description of the dataset(s)

5.2 Experimental setup

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

5.3 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.

5.4 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.

6

Conclusions & Future Work

6.1 Conclusions

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

6.2 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] D. Bhowmik and T. Feng, "The multimedia blockchain: A distributed and tamper-proof media transaction framework," in *IEEE International Conference on Digital Signal Processing (DSP)*, 2017, pp. 1–5.

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***.