

A computer vision based approach for reducing false alarms caused by spiders and cobwebs in surveillance camera networks

Ramya Hebbalaguppe, B.E. (Hons)

A thesis submitted as a requirement for the degree of Master of Engineering in
Electronic Engineering

Supervisors:

Prof. Noel E. O'Connor and Prof. Alan F. Smeaton

School of Electronic Engineering, Dublin City University

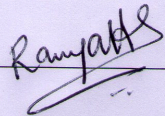
January 21, 2014



Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Engineering is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____



ID No.: 11211953

Ramya S. M. Hebbalaguppe

Date: 21 - 01 - 2014

Acknowledgements

I express my deep thanks to my principal supervisor, Prof. Noel E. O'Connor, for encouraging me to choose the research topic of my choice and for being always supportive of my interests. Your invaluable suggestions, patience and friendliness will always be remembered. Many thanks to Prof. Alan Smeaton for co-advising my thesis and also for recruiting me back in August 2011. I would like to thank my internal examiner, Dr. Robert Sadleir and external examiner, Dr. Xavier Giro'-i-Nieto for detailed corrections and suggestions which were very useful to improve the thesis work. It would not have been possible to write this masters thesis without the help and support of the many kind people around me, my mentor, Dr. Kevin McGuinness – for his agility, willingness to help me throughout my masters and my friend, Jogile Kuklyte, for her helping nature and for developing the annotation tool. I express my gratitude for suggestions by Netwatch team members, Dr. Cem Direkoglu, Dr. Jiang Zhou and Dr. Leonardo Gualano. Thanks to Niall Dorr, chief innovation officer at Netwatch for providing providing real world surveillance data. Many thanks to Dr. Rami Albatal, Dr. Praveen Pankajashan, Sam Dodge, and Abhilash Chandrashekhar, for your interesting technical discussions. Thanks to Enterprise Ireland grant IP-2011-0109 and Netwatch Security Systems Pvt. Ltd for funding the research work. Thanks to coursera founders, Prof. Andrew Ng and Prof. Daphne Koller for your noble ideas for setting up MOOC - your courses have made my dream of life long learning easy with reachable milestones every week. Many thanks to Prof. Ramakrishna Kakarala for his support in the past. I would like to thank Margaret Malone and Deirdre Sheridan from the CLARITY administrative staff for their constant help.

Above all, I am indebted to my husband, Seshan Srirangarajan for his personal support and also for setting an example of commitment and hard work addressing one's duties. Also, to our parents: Sowbagya Murthy, Vathsala Srirangarajan,

H.R. Srirangarajan and H. R. Sugnana Murthy. Many thanks to my ever inspiring friend, Vani Murthy for exuding positivity and hope everyday. Special thanks to my uncle, Krishna Murthy H. R. , for influencing me as a better human being, for which my mere expression of thanks does not suffice. My brother's and sisters' families have given me their unequivocal support throughout. I would also like to thank my friends - Preethi Krishnan, Sushrutha Bhandage, Niranjana Nagaraju and Chaitana Tigadi for being there for me.

Abstract

The main aim of the thesis is to explore computer vision based solutions to the reduction of false alarms in surveillance networks. More specifically, the problem of false alarms triggered by spiders, which contributes to a substantial percentage of nuisance alarms, is addressed. In an automated surveillance setup in which motion events trigger alarms, the percentage of false alarms raised by spiders can range from 20 – 50% depending on the season of the year, lighting conditions, camera type and other environmental factors. These alarms not only (a) increase the workload of human operators validating the alarms but also (b) increase labor costs associated with regular cleaning of the lens to avoid frequent build up of spiders/cobwebs. In this thesis, a novel and an economical method to reduce the false alarms caused by spiders is proposed by building a spider classifier intended to be part of the video processing pipeline for intruder detection systems. The proposed method, which uses a feature descriptor obtained by early fusion of image blur and texture, is suitable for real-time processing and yet comparable in performance to more computationally costly approaches like SIFT/RootSIFT with bag of visual words aggregation. The performance of the binary classifiers developed based on several visual features is comprehensively investigated. The proposed method can eliminate 98.5% of false alarms caused by spiders with a false positive rate of less than 1%, thereby reducing the workload of the surveillance personnel validating the alarms. This also optimises the usage of police resources, especially in situations where the event triggered due to the spider is not dismissed by an operator in time, resulting in police notification. The classifier confidence score also provides cues for prioritising events to be addressed and could be further used to actuate a mechanical wiper which might be used in clearing the spider webs remotely.

List of Publications¹

- R. Hebbalaguppe, K. McGuinness, J. Kuklyte, C. Direkoglu, L. Gualano, J. Zhou and N. E. O'Connor. **"An investigation of visual features to detect spiders/cobwebs in surveillance camera networks."**, 10th IEEE International Conference on Advanced Video and Signal based Surveillance, Krakow, Poland, 27-30 Aug'13. ²
- R. Hebbalaguppe, K. McGuinness, R. Albatal, and N. E. O'Connor. **"Reducing False Alarms in Visual Surveillance Systems based on Spider Classification."**, *Withheld*: EURASIP Journal of Image and Video Processing, Special Issue on Animal and Insect behavior in Image sequences, 15 Jan'13 ³
- J. Kuklyte, K. McGuinness, R. Hebbalaguppe, C. Direkoglu, L. Gualano, and N. E. O'Connor. **"Identification Of Moving Objects in Poor Quality Surveillance Data."**, IEEE International Workshop on Image and Audio Analysis for Multimedia Interactive services, Paris, France, 3-5 Jul'13.
- R. Hebbalaguppe, K. McGuinness, J. Kuklyte, G. Healy, N. E. O'Connor and A. Smeaton. **"How Interaction methods Affect Image Segmentation: User Experience in the Task"**, IEEE User Centric Computer Vision, Tampa, Florida, 17 -18 Jan'13.

¹Publications during M.E.

²Accepted for publication, however the paper is being withdrawn for reasons of commercial sensitivity (for patent purposes) and the paper will be resubmitted elsewhere.

³Article received positive first round reviews ("accept with minor revision"), but the paper is being withdrawn for reasons of commercial sensitivity (for patent purposes)

Abbreviations and Acronyms

| | |
|-------------|--|
| AHA | American Homeowner Association |
| BoVW | Bag of Visual Words |
| CCTV | Closed-Circuit Television |
| CPBD | Cumulative Probability of Blur Detection |
| HSV | Hue Saturation Value |
| IR | Infrared |
| JPEG | Joint Photographic Experts Group |
| LBP | Local Binary Patterns |
| LBPV | Local Binary Patterns with Variance |
| OEM | Original Equipment Manufacturer |
| LED | Light Emitting Diode |
| RBF | Radial Basis Function |
| RGB | Red Green Blue |
| ROC | Receiver Operating Characteristics |
| SIFT | Scale Invariant Feature Transform |
| SVM | Support Vector Machine |

Contents

| | |
|---|-------------|
| List of Figures | iv |
| List of Tables | viii |
| 1 Introduction | 1 |
| 1.1 Nuisance alarms in video surveillance | 4 |
| 1.2 Motivation | 10 |
| 1.3 Objectives | 13 |
| 1.4 Structure of thesis | 13 |
| 1.5 Conclusion | 14 |
| 2 Spider-based nuisance alarm reduction: a review | 16 |
| 2.1 Introduction | 16 |
| 2.2 Chemical based solutions | 17 |
| 2.3 Hardware based solutions | 18 |
| 2.4 Computer vision based solutions | 20 |
| 2.5 Conclusion | 22 |
| 3 Computer vision based spider and spider web detection | 24 |
| 3.1 Introduction | 24 |
| 3.2 Problem formulation | 26 |
| 3.3 Desirable characteristics for real time operation | 30 |

| | | |
|----------|--|-----------|
| 3.3.1 | Classification accuracy | 30 |
| 3.3.2 | Computation time | 31 |
| 3.3.3 | Receiver Operating Curve (ROC) | 31 |
| 3.3.4 | Classifier confidence score | 32 |
| 3.4 | Feature Extraction | 33 |
| 3.4.1 | Introduction | 33 |
| 3.4.2 | Cues for visual feature extraction | 34 |
| 3.4.3 | Descriptor fusion | 35 |
| 3.4.4 | Feature normalisation | 38 |
| 3.5 | Investigation of visual features | 38 |
| 3.5.1 | Intensity or Grayscale histograms | 40 |
| 3.5.2 | Optimised Haralick texture features | 42 |
| 3.5.3 | Cumulative Probability of Blur Detection | 45 |
| 3.5.4 | Blur histograms | 47 |
| 3.5.5 | Early fusion of Haralick texture features and CPBD | 48 |
| 3.5.6 | SIFT with BoVW | 48 |
| 3.5.7 | RootSIFT with BoVW | 53 |
| 3.5.8 | LBP variance | 53 |
| 3.5.9 | Early fusion of LBP Variance and CPBD | 55 |
| 3.6 | Classification | 55 |
| 3.6.1 | SVM Introduction | 55 |
| 3.6.2 | SVM classification setup | 58 |
| 3.7 | Conclusion | 62 |
| 4 | Dataset | 63 |
| 4.1 | Introduction | 63 |
| 4.2 | Annotation tool | 69 |
| 4.3 | Artefact reduction | 70 |

| | | |
|----------|---|------------|
| 4.4 | Evaluation dataset | 73 |
| 4.5 | Conclusion | 75 |
| 5 | Results | 76 |
| 5.1 | Introduction | 76 |
| 5.2 | Experimental set-up | 77 |
| 5.3 | Specific parameters used for feature extraction | 77 |
| 5.4 | Classifier setup | 80 |
| 5.5 | Classification accuracy | 81 |
| 5.6 | Computation time | 83 |
| 5.7 | ROC | 86 |
| 5.8 | Sample results | 91 |
| 5.9 | Field trial results | 93 |
| 5.10 | Discussion | 95 |
| 5.11 | Conclusion | 97 |
| 6 | Conclusions and future work | 99 |
| 6.1 | Conclusions | 99 |
| 6.2 | Research contributions | 101 |
| 6.3 | Future work | 102 |
| | Bibliography | 104 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | Illustration of true event handling: the case under consideration is a person trespassing – (a) the cameras with intrusion detection software protect a customer’s property; (b) shows the trespassing activity detected and the live footage is transmitted to the communication hub; (c) shows a team of intervention specialists carrying out appropriate action; (d) a customised audio warning is issued to prevent trespassing (e) if the intruders ignore the audio warnings, intervention specialists can quickly guide the respondents to the exact location of the intruders. (Image courtesy: Netwatch Security Systems, Ireland) | 2 |
| 1.2 | Illustration of false alarms triggered by spiders/cobwebs: Figures (a, b) illustrate the spiderwebs and spiders protruding out of surveillance camera hood – the support offered by the camera hood in most bullet type cameras makes an ideal habitat for spiders. Figure (c): spider infestation increases surveillance personnel workload/stress when a huge number of spider alerts are to be addressed. Figure (d): spiderweb build-up increases workload of the maintenance employee. | 6 |
| 1.3 | Validation of alarms in video surveillance | 9 |
| 1.4 | Spider Classification pipeline for video surveillance | 10 |

| | | |
|------|---|----|
| 2.1 | Literature survey on some existing chemical and hardware based solutions to overcome spider false alarms | 17 |
| 3.1 | Positive examples used in image classification | 27 |
| 3.2 | Negative examples used in image classification | 28 |
| 3.3 | Block diagram showing the various components of the proposed spider classification system | 29 |
| 3.4 | Feature extraction from images depicting a descriptor | 33 |
| 3.5 | Sample spider images for visual feature design | 34 |
| 3.6 | Illustration of <i>Early Fusion</i> of visual features | 37 |
| 3.7 | Illustration of <i>Late Fusion</i> of visual descriptors | 37 |
| 3.8 | Sample intensity histogram | 41 |
| 3.9 | The four directions of adjacency as defined for calculation of the Haralick texture features. The Haralick statistics are calculated for co-occurrence matrices generated using each of the four directions of adjacency. | 43 |
| 3.10 | Computation of CPBD metric | 45 |
| 3.11 | Early fusion of Haralick and CPBD features for image classification | 48 |
| 3.12 | SIFT detector showing the original image and difference of Gaussian filtering done at different scales | 49 |
| 3.13 | SIFT feature extraction (using VLFeat) | 50 |
| 3.14 | A bag-of-visual words model | 51 |
| 3.15 | Histogram representation of Visual Words | 51 |
| 3.16 | Bag-of-visual words for image classification | 52 |
| 3.17 | An illustration of LBP | 54 |
| 3.18 | An illustration of the SVM showing binary classification | 56 |
| 3.19 | Demonstration of Linear SVM used in binary classification | 57 |
| 3.20 | Visualization of RBF Kernel when σ is varied | 59 |

| | | |
|------|--|----|
| 3.21 | Illustration of variation of C parameter on decision function | 61 |
| 4.1 | A selection of triggered events, where each event comprises of three JPEG images. | 64 |
| 4.2 | Example of true events triggered by vehicles | 65 |
| 4.3 | Example of true events triggered by people | 66 |
| 4.4 | An example of true event triggered by animal | 66 |
| 4.5 | Examples of nuisance events | 67 |
| 4.6 | Screenshot of the annotation tool developed for creating the ground truth | 69 |
| 4.7 | Artefact reduction in <i>Quads</i> | 71 |
| 4.8 | An illustration of artefact removal on the images acquired from different camera sites | 74 |
| 5.1 | LBP Variance: Uniform patterns for $P = 8$ | 78 |
| 5.2 | A comparison of classification accuracy vs. Total execution time of visual descriptors | 84 |
| 5.3 | A comparison of ROC curves for all investigated visual features | 87 |
| 5.4 | A comparison of ROC curves for investigated visual features: CPBD, Haralick texture features and fusion of Haralick and CPBD | 89 |
| 5.5 | A comparison of ROC curves for investigated visual features: LBP Variance, CPBD and fusion of LBP variance and CPBD | 90 |
| 5.6 | True positives (spiders classified into <i>spider</i> category) and true negatives (non-spiders classified into <i>non-spiders</i> category) | 91 |
| 5.7 | Misclassification set: False positives (non-spiders classified as <i>spiders</i>) by our proposed algorithm | 92 |
| 5.8 | Misclassification set: False negatives (spiders classified as <i>non-spiders</i>) by our proposed algorithm. | 92 |

| | | |
|------|---|----|
| 5.9 | Field trial results: True positives (spiders classified as <i>spiders</i>) by the proposed algorithm | 93 |
| 5.10 | Field trial results: True negatives (non-spiders classified as <i>non-spiders</i>) by the proposed algorithm | 94 |
| 5.11 | Field trial results : False positives (non-spider classified as <i>spiders</i>) by the proposed algorithm | 94 |

List of Tables

| | | |
|-----|--|----|
| 3.1 | Feature combination results on the Brodatz dataset showing redundant feature combination not yielding improved classification accuracy while complementary feature fusion yielding improved classification accuracy. | 35 |
| 4.1 | Application of the Navier-Stokes equation from fluid dynamics to image inpainting. | 72 |
| 5.1 | Combination of (C, γ) obtained by grid search for tested feature vectors for image classification. | 80 |
| 5.2 | Comparison of the classification accuracies on tested approaches. . | 81 |
| 5.3 | Computation time for feature extraction and classification for each method (in milliseconds) | 83 |
| 5.4 | Training time taken by tested approaches. | 85 |

Chapter 1

Introduction

Applications of video surveillance are numerous; some of the interesting applied areas include detecting and tracking people (Dalal & Triggs 2005), vehicle monitoring and tracking (Maurin et al. 2005), surveillance event detection and recognition (Piciarelli & Foresti 2011), and crime prevention (Armitage et al. 1999). In a commercial security scenario such as monitoring a shopping center, parking lot, home security, etc., cameras aid in: (a) deterrence – where burglars may see the camera and then decide not to take the risk of committing a theft, (b) prosecution – burglars caught on camera are then prosecuted and most importantly, (c) monitoring and intervention – security personnel monitoring the area through a CCTV system may act on any suspicious behavior and thus prevent crime, e. g. by alerting the police or deploying security personnel to the location.

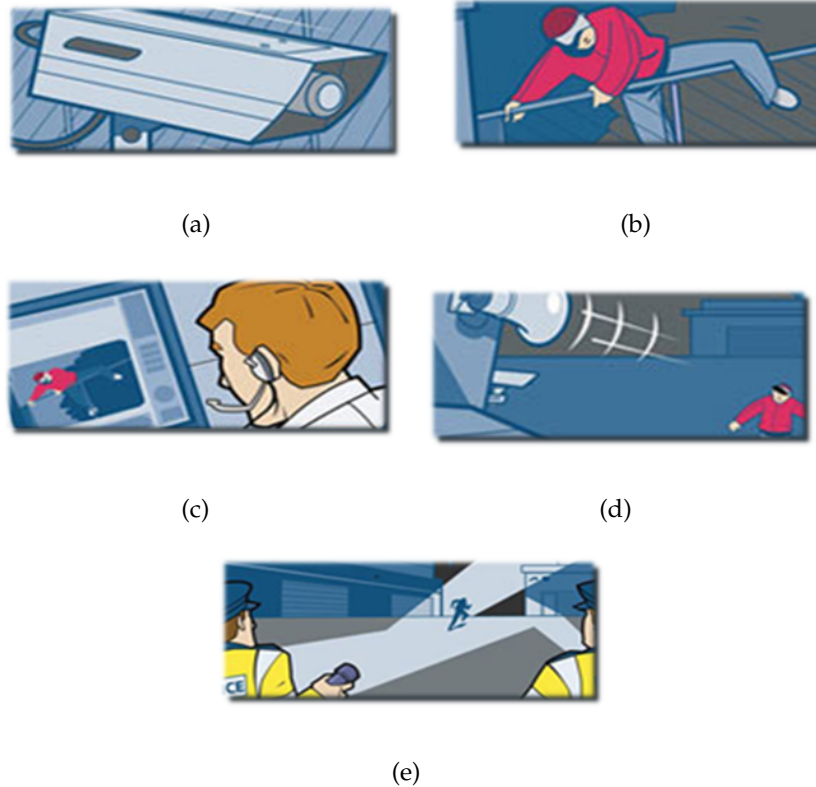


Figure 1.1: Illustration of true event handling: the case under consideration is a person trespassing – (a) the cameras with intrusion detection software protect a customer’s property; (b) shows the trespassing activity detected and the live footage is transmitted to the communication hub; (c) shows a team of intervention specialists carrying out appropriate action; (d) a customised audio warning is issued to prevent trespassing (e) if the intruders ignore the audio warnings, intervention specialists can quickly guide the respondents to the exact location of the intruders. (Image courtesy: Netwatch Security Systems, Ireland)

Given the abundance of security cameras and the fact that most of them are not monitored, there is a need to filter out unwanted information to save on human and material resources. The amount of footage can be drastically reduced by motion sensors that trigger recording only when motion is detected. Considering the ubiquity of video surveillance, automated video surveillance is an important

research area in the commercial sector in order to reduce the response time from occurrence to detection and subsequent handling of events.

To automate detection of events such as theft or other crimes, security companies can now offer networks of CCTV cameras equipped with *intrusion detection* software to protect their customer's property. Most security companies, including our industry partner Netwatch Security Systems, Ireland, offer such systems working on the principle that when a perimeter of the area being monitored is breached, a remote video response showing live footage of the break-in is transmitted from the installed CCTV system to the communication hub. Surveillance personnel in the communication hub monitor every aspect of the system on a constant basis, including each sensor (surveillance camera), the detection software, the phone lines, etc., to ensure that the system runs in the background without any need for intervention by the property owner. In the communication hub, motion triggered events in a scene are flagged to a human operator. Human operators, also called *intervention specialists* by Netwatch Security Systems, manually verify the motion triggered event. A detected event can be a *true event*: Netwatch Security Systems consider true events to be those in the camera field of view that might cause a potential hazard to the monitored environment. These are primarily triggered by people, vehicles and animals, crossing the surveillance camera field of view (Kuklyte et al. 2013). True event object classes in this thesis were identified with the involvement of experienced surveillance staff. A *false event* is an erroneous report of an emergency, causing unnecessary panic and/or bringing resources (such as emergency services) to a place where they are not needed. They are typically triggered by environmental changes like tree shake and vegetation movements due to wind, cloud movement, insects, spiders crawling over the lens, rain, snow, etc. Detecting true events gives a vital early warning, allowing the intervention specialists to take the next step, which is usually to warn off these intruders/criminals with a live audio warning through speakers

installed with the cameras. The local authorities are notified and the intervention specialists can quickly guide the respondents to the exact location of the intruders. Intervention specialists warn intruders to leave immediately via a personalised audio warning, thus preventing theft, vandalism, and other crimes. Netwatch Security Systems thus deploy systems that not only detect crime but also prevent crime. Figure 1.1 shows a pictorial depiction of true event handling where the example case considered is a person trespassing. In the case of a false event, the event is dismissed; however, these represent an undesirable increase in workload for intervention specialists.

1.1 Nuisance alarms in video surveillance

In many automatic intrusion detection systems that trigger events based on motion in a scene being monitored, surveillance personnel run the risk of being swamped by nuisance/false alarms. As mentioned earlier, nuisance alarm contributors fall into various categories like insects, foliage movement, lighting changes, rain, snow, etc. Ninety-four to ninety-nine percent of all police physical responses are caused by false surveillance camera alarm activations (Blackstone et al. 2005). In the year 2000, police responded to 36 million false calls at an estimated cost of \$1.8 billion (Blackstone et al. 2005). The American Homeowner Association (AHA) reports that 98.8% of alarms are false and it costs the taxpayer \$62.04 each time police respond¹.

The false alarms result in an increase in physical response time by the police to real alarms as false alarms have the potential to divert emergency responders away from legitimate emergencies. There is a huge amount of effort involved in validating the false alarms given that a significant proportion of alarms are false and this results in increased operator stress in addressing all alarms in a timely

¹<http://www.ahahome.com/non/articles/99/101399.html>

manner. Surveillance technology has reached a stage where mounting cameras to capture video imagery is cheap, but finding available human resources to sit and watch video footage is expensive (Collins et al. 1999). So there needs to be a trade-off between rejection of false alarms and addressing all alarms in a timely manner.

In this work, we gathered a representative sample of events by manually annotating events triggered from 12 monitored sites having a total of 275 cameras. Results showed that: 35% of alarms were triggered by spiders/webs, 30% by people, 4% by animals, 23% by vehicles, and 8% were due to other sources. The figures indicate that the percentage of spider based false alarms is significant and that there is a clear urgency to address this issue. The ability to detect and suppress alarms caused by spiders/insects could help to dramatically reduce false alarm rates. Analysis and extensive discussion with intervention specialists indicated that false alarms are triggered by spiders when they crawl over the surface of a surveillance camera lens or when the spiderwebs/cobwebs (cobwebs are abandoned spiderwebs) shake due to wind. There are also situations when hundreds of cameras that repeatedly trigger spider false alerts are turned off temporarily. In these extreme situations the site is left unmonitored. Hence, reduction of false alarms is a key problem for an efficient automated/semi-automated video surveillance system. In contrast with true event handling as shown in Figure 1.1, Figure 1.2 depicts false alarm handling that would result in operator stress and an increase in surveillance workload.



(a)

(b)



(c)

(d)

Figure 1.2: Illustration of false alarms triggered by spiders/cobwebs: Figures (a, b) illustrate the spiderwebs and spiders protruding out of surveillance camera hood – the support offered by the camera hood in most bullet type cameras makes an ideal habitat for spiders. Figure (c): spider infestation increases surveillance personnel workload/stress when a huge number of spider alerts are to be addressed. Figure (d): spiderweb build-up increases workload of the maintenance employee.

Guardian Alarm Systems² and York regional police³ suggest that spiders are one of the main contributors to false alerts especially when they climb on motion detectors⁴. An online false alarm awareness course offered in Florida advises that the face of the detector be kept clean to avoid spider alerts⁵.

The research reported in this thesis investigates image processing/computer vision techniques to automatically determine if an image sequence contains a spider/spiderweb, resulting in a novel approach to identify spider-based false alarms in a surveillance context. At the time of submission of this thesis, there are no documented studies of any attempt to reduce false alerts by spiders in surveillance systems using computer vision. Unlike other methods, the proposed spider classification algorithm, which is designed to be a part of the video analytics system itself, can distinguish between events triggered by *spiders* and those triggered by other causes such as people, vehicles, and animals belonging to the *non-spider* category. Potential benefits include decreasing the number of false alarms via automatic event classification (by classification into *spider* and *non-spider* classes), facilitating event prioritisation (by providing cues if events contain spiders and if they do, whether they could be ignored or addressed with a lower priority) leading to efficient use of intervention specialists' time. The proposed method also assists the human operator by associating a confidence score to the detected events. Support vector machines with probabilistic outputs produced using a variant of Platt's method, are used to produce these confidence scores from image features (Platt 1999). These confidence scores can then be used to filter events that have high probability of being caused by spiders or spiderwebs, while ensuring true events are very unlikely to be classified incorrectly. Furthermore, a confidence score could be used to trigger a mechanical wiper blade for cleaning

²<http://guardianalarm.com/customer-service/preventing-false-alarms>

³<http://www.yrp.ca/default.aspx?pg=f32b2f79-b29b-44a9-a141-25a6a48cbf8c>

⁴<http://www.carolinasecurity.com/REDUCEFA.pdf>

⁵<http://www2.colliersheriff.org>

the area over the lens if a mechanical solution would be implemented in future; or to initiate a manual cleaning operation by a surveillance employee only on the cameras infested by spiders rather than for all the cameras monitored. The amount of effort needed for manual cleaning of lens is significant considering that there are for example, 25,000 cameras deployed by *Netwatch Security Systems*.

Figure 1.3 provides an overview of the typical processing pipeline used by Netwatch Security Systems for alarm validation. Figure 1.4 describes where the proposed spider alarm classification fits into the video processing pipeline. The pipeline suggested in Figure 1.4 will not only reduce the number of spider false alarms but it will also enable intervention specialists to respond quickly when an alarm is triggered. This is because the spider false alarm reduction pipeline determines when non-spider (human/vehicle/animal) activity has triggered the alarm. Therefore intervention specialists can be sure it is an intruder and not a spider or another similar insect in front of the surveillance camera which has triggered the alarm.

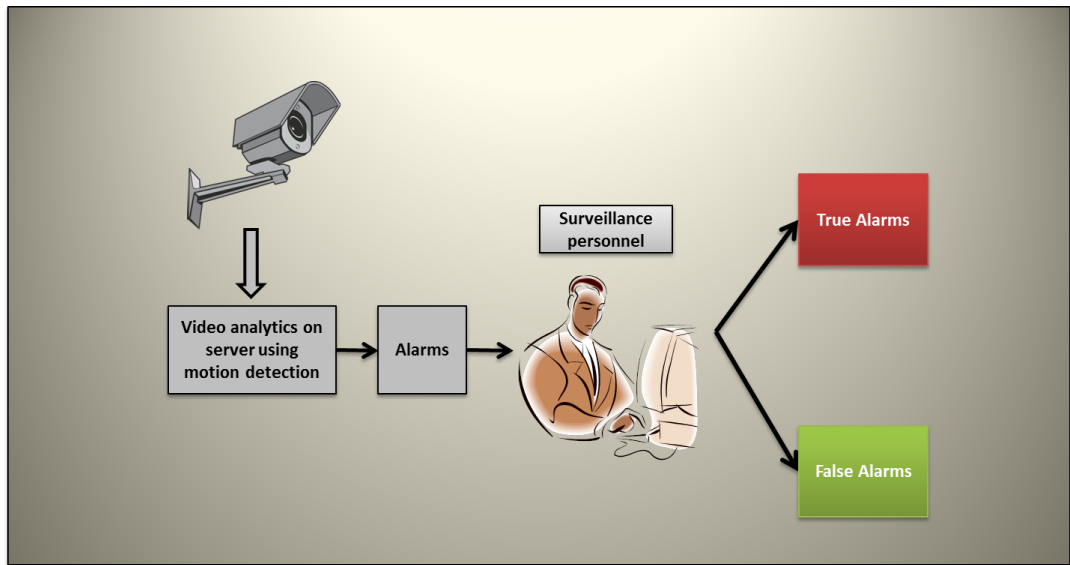


Figure 1.3: Illustration of a typical alarm validation process in video surveillance: A surveillance operator validates all the incoming alarms generated by video analytics software. The intervention specialist addresses the true alarms by taking necessary actions and by dismissing the false alarms.

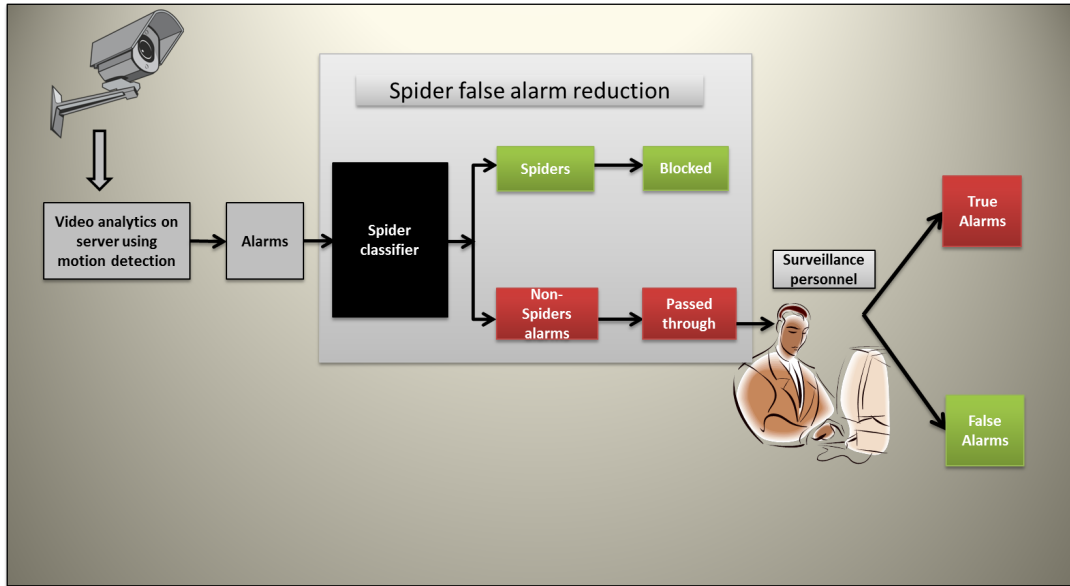


Figure 1.4: Illustration of validation of alarms with the spider false alarm reduction processing in the pipeline: A surveillance operator validates the non-spider alarms which are filtered out by the spider classification pipeline, while the spider false alarms are blocked thereby reducing the workload on a human operator.

1.2 Motivation

The research reported in this thesis explores economical solutions for the reduction of false alarms caused by spiders and spider webs. The solution could be generically applied to insects close to the surveillance camera lens. The following reasons motivate the research to classify incoming events into the *spiders* and *non-spiders* categories using computer vision technology:

1. *Reduction in intervention specialist workload:* False alarm rates were calculated by manual inspection of approximately 6,000 images gathered from Net-watch Security Systems during winter and summer seasons. Annotated data suggests that spider related alarms contribute to 20–50% of false alarms trig-

gered in outdoor surveillance systems. The percentage of false alerts might be surprising until one considers the presence of over a million spiders per hectare (Bristowe & Smith 1971) globally excluding Antarctica⁶. The sheer volume of false alarms raised by spiders in surveillance systems therefore means that there is huge human effort involved in event validation. The ability to detect and suppress alarms caused by spiders and insects could, therefore, have a large impact on the reduction of false alarm rates and on reduction of intervention specialists' work stress.

2. *Reduction in maintenance personnel workload:* Spider classification algorithms can further reduce the workload on maintenance employees whose job it is to manually clean the surveillance camera lens surface to prevent frequent build-up of spider webs. For example, Netwatch Security Systems has 25,000 cameras deployed in Ireland. With the plans to grow the business, the increase of workload to frequently clean the lens is not sustainable in this context. It is typically a twice yearly duty of a surveillance company to clean all external camera enclosures. Other alternative solutions, for example, asking the customers clean their own cameras on site are not desirable as security companies need to offer competitive services to retain customers, and maintenance is often offered along with the deployment of security cameras.
3. *Economical impact:* With the cost of site preparation (maintenance cost) often exceeding the cost of the detection equipment (surveillance cameras), the cost of employing intervention specialists to validate alarms will increase with time. This is because security companies like Netwatch Security Systems wish to expand their current business across the globe. New hiring has to be done to address all alarms in a timely manner. Circumstances arise

⁶<http://www.meadowtreasures.com/spiderfacts.htm>

however where it is difficult, excessively expensive or time does not allow for the preferred site preparation. Spider detection/classification could help facilitate optimal usage of available resources. Police and first responders receive many calls annually caused by surveillance camera alarm activations in both business premises and residences and a large majority of these turn out to be false. New technology could help cut down on spider triggered false alarms and ensure that the police are only called in cases of a genuine emergency.

4. *Commercial potential:* Spider false alarm detection/classification using computer vision technology when incorporated as a part of video processing pipeline of surveillance cameras can significantly reduce surveillance personnel workload. The proposed algorithm is not specific to Netwatch Security Systems. This algorithmic solution could be generically used to reduce spider alerts on data from different OEMs and service providers. Hence this embedded surveillance software solution, is expected to have a strong commercial potential.
5. *Ecological benefit:* From an ecological point of view, spiders have their role in environmental balance by controlling pests in the ecological chain. For the most part, spiders are harmless and generally beneficial in keeping the insect populations in check. The availability of prey such as houseflies and other pests in cities and the presence of lighting and warmth in parking lots make lamp posts or surroundings of surveillance cameras provide ideal habitats for spiders (Lizzy 2012). Using chemical sprays to deter them from the lens surface is not the best way to solve the problem noting the fact that the spiders and webs reoccur even with spider deterrent sprays.

1.3 Objectives

The objectives of the thesis are as follows:

1. To develop a novel approach to identify spiders and spider webs within a field of view of a standard security camera. Specifically, visual features are investigated to detect spiders/cobwebs in surveillance camera networks and combine features with an SVM framework for classification.
2. To compare the proposed method against the existing hardware and chemical solutions, highlighting the problems with the existing methods and how they can be overcome with the proposed method.
3. To evaluate the proposed computer vision algorithm in terms of (a) classification accuracy, (b) computation time, (c) training time, and (d) receiver operating characteristics (ROC) curve. This is followed by comparison with algorithms which have been developed for similar classification problems.
4. To perform a field trial to determine the accuracy of the spider classification algorithm in real deployments.

1.4 Structure of thesis

The remainder of this thesis is organised as follows:

Chapter 2 reviews the state-of-the-art in spider false alarm reduction. This chapter also highlights why the current methods – chemical, electronic and hardware solutions – are not effective solutions in practice. It also emphasises that computer vision based spider detection is a cost effective method.

Chapter 3 presents where the proposed algorithm fits as a part of a video processing pipeline for surveillance. Section 3.1 discusses how computer vision can be exploited in accomplishing the task of reducing false alarms and formulates this

as an image classification problem. It also addresses why image classification was chosen as oppose to object (spider) recognition. This is followed by a discussion on the selection of parameters which are central to algorithm operation in terms of – classification accuracy, computation time, and ROC curve. The visual feature design for spider classification and also a description of alternative visual feature extraction methodologies for comparison with the proposed method is addressed in this Chapter. In this chapter we also discuss machine learning frameworks for image classification.

Chapter 4, presents the dataset used in experiments that was obtained from real surveillance cameras deployed in Ireland by – Netwatch Security Systems Pvt Ltd. Section 4.2 describes the dataset used for training the algorithm based on a manual annotation process using a custom annotation tool. Section 4.3 discusses the artifact removal procedure employed for data preprocessing to remove artifacts introduced by third party software.

Chapter 5 focuses on system configuration for running experiments, simulation details, specific parameters used for extracting features, experimental results of classification accuracy, execution time and ROC curve. The proposed approach is also compared with the state-of-the-art visual features.

Finally, in Chapter 6 the thesis is concluded by discussing the novelty of the proposed method and outline some suggestions for future work.

1.5 Conclusion

This chapter discusses the applications of video surveillance highlighting its growing ubiquity. A key research area in automated/semi-automated surveillance is false alarm reduction. False alarms triggered by spiders contribute to a significant percentage of alarms. This thesis investigates surveillance technology which can distinguish whether an alarm is true (animals, people and vehicles) or a false

alarm triggered by spiders and insects close to the surveillance camera. For distinguishing spider and non-spider alarms, a computer vision solution is proposed that classifies images into *spider/non-spider* categories.

Chapter 2

Spider-based nuisance alarm reduction: a review

2.1 Introduction

False alarms triggered by spiders are not a new problem. Many security companies invest huge financial resources in cleaning operations to get a clear view of the scene being monitored. It has been shown in various deployments that the use of bullet cameras with built-in IR LEDs can dramatically increase the false alarm rate generated by the video analytics software, especially during hot and humid seasons (Honeywell 2010). The heat from the IR light attracts spiders and insects. When a spider or insect crawls across the camera faceplate it will often appear as a white, bright blob in the camera field of view (Honeywell 2010) (Lizzy 2012). Other reasons conducive to formation of spider webs are humidity and low light.

Traditionally, solutions proposed to circumvent this problem fell into two main categories. The first involved labor intensive manual cleaning of cobwebs such as broom sweep/vacuum cleaning or using aerosol sprays chemically formulated to help deter spiders from nesting. This approach was generally used for camera housings and around motion sensors. The second category includes additional

camera hardware or a change in entire camera units to reduce the formation of spiders/spiderwebs. Figure 2.1 shows some existing chemical and hardware based solutions. Moving away from the aforementioned methods, we propose solutions which are more economically viable for both legacy and current camera installations.

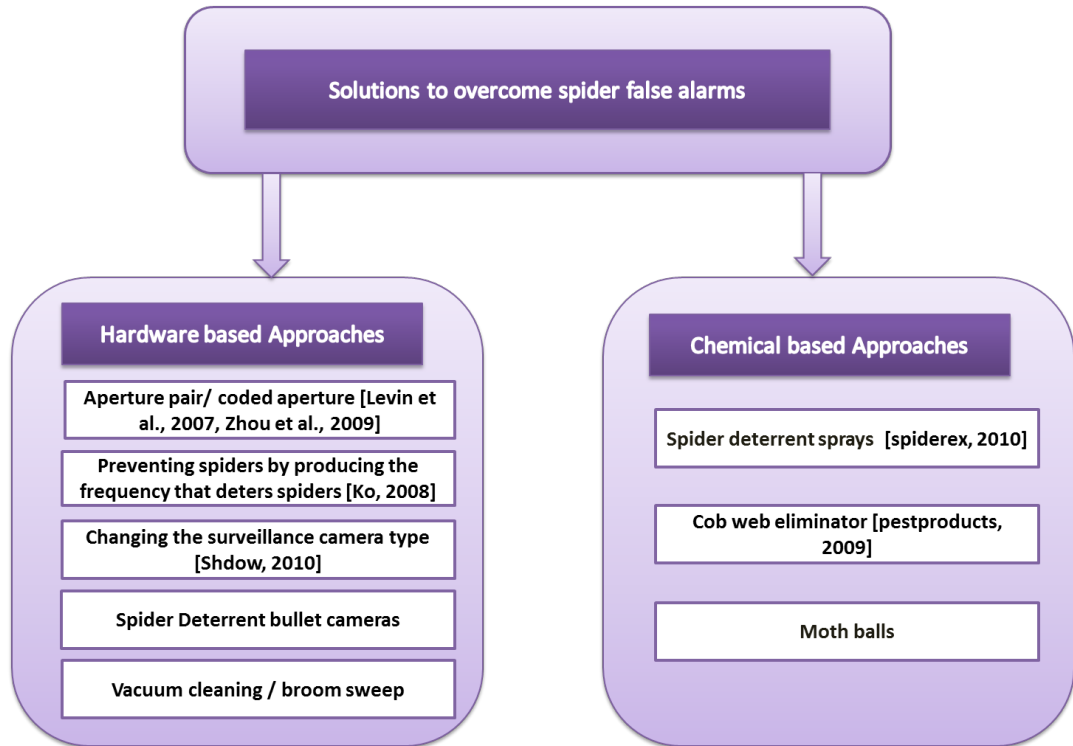


Figure 2.1: Illustration of a taxonomy of some existing chemical and hardware based solutions to overcome spider false alarms

2.2 Chemical based solutions

The false alarms triggered by spiders in surveillance were historically dealt with by using spider deterrent sprays¹². The spider deterrent sprays claim to reduce spider infestation along with problems spiders cause including spider webs and

¹<http://www.spiderex.co.uk>

²<http://www.pestproducts.com/spider.htm>

build up of any spider related material. However, using sprays to manually clean the surveillance camera lenses is expensive, monotonous, and involves significant human effort. Some sprays might cause staining on the camera lens resulting in the need for regular cleaning. Netwatch Security Systems reported that spiders reoccur frequently even with the usage of spider deterrent sprays.

Mounting old-fashioned odour mothballs close to surveillance camera lens can do a good job keeping the spiders at bay (Powell 1993) (Lawrinson 2006) (Roselle 1954). It involves placement of mothballs in a plastic bag around the lens, however the smell of moth balls is annoying especially in situations where CCTV cameras are mounted indoors. Mothballs have also been found to pose health risks to humans as they chemically consist either of flammable *naphthalene* or *para-dichlorobenzene*, both of which have a strong, pungent smell³. The International Agency for Research on Cancer (IARC) has classified mothballs as carcinogenic and neurotoxic (WHO 2007). In addition to previously discussed problems, there is of course still human effort involved in placement of mothballs and hence it is not economical. To summarise, chemical based solutions are labor-intensive and hence expensive.

2.3 Hardware based solutions

Mostly, the spiders are seeking prey and warmth in the proximity of the surveillance camera lens. This means that the distance between spiders and camera lens is often very small which falls outside the depth-of-focus in the video data captured by surveillance cameras (Hart 1996). This causes the spiders/webs to appear out-of-focus in the form of saturated or dark blobs in night and day situations respectively. This is because surveillance cameras seldom have large depth-of-field to render an object close to the lens unless it uses a prime or fixed fo-

³<http://www.thriftyfun.com/tf20777950.tip.html>

cal length lens⁴. A fixed lens camera with high resolution can do the job and costs less than a vari-focal lens camera to render all objects in focus in a surveillance setup.

- *Use of depth from defocus to detect spiders:* Typically spiders are seen a few centimeters from the surveillance camera lens surface unlike true event candidates like humans, animals, and vehicles. Hence, to find whether a triggered event is suggestive of insects/spiders or a true event (people, animal and vehicle), estimation of depth from camera to the object would be beneficial. However, it is difficult to detect depth from a monocular camera view. Techniques such as depth from defocus would require the installation of new hardware, i.e., it would require solutions like the implementation of aperture pair (Levin et al. 2007) or coded aperture to determine depth (Zhou et al. 2009). This would require modifications to thousands of cameras already deployed by the security industry, not to mention that security cameras with such features are unavailable in the market.
- *Change of camera type:* Changing the camera type from bullet type camera to dome shaped camera may facilitate the reduction in the formation of webs (Shdow 2010). The bullet type camera attracts spiders who build their webs between the top cover that protrudes out 2-3 centimeters from the glass and the bottom of the camera. Any camera with infra red will attract insects but dome cameras have a flat or round surface that makes it hard for a spider to build a web. However, Netwatch Security Systems has reported that the number of false alerts remained unaltered even with the change in camera type. As spiders are poikilothermic i.e., their body temperature varies with the ambient temperature, usage of thermal cameras would not help to detect spiders (Scholander, Flagg, Walters & Irving 1953)

⁴<http://www.securitycameraking.com/security-camera-lenses-145-ctg.html>,
<http://www.dpreview.com/forums/thread/3020935>

(Anderson 1970) (Kotiaho, Alatalo, Mappes & Parri 1996). In any case, thermal cameras tend to be more expensive than non-thermal counterparts.

- *A security camera capable of preventing spiders by generating frequencies that deter pests or spiders:* There are alternative hardware solutions devised to tackle spider false alarms. A security camera capable of preventing spiders by generating frequencies that deter pests or spiders is proposed in (Ko 2008). This describes a security camera capable of preventing spiders by exterminating spiders which would like to settle in front of the camera. A bullet type surveillance camera is proposed accessorised with speakers, humidity sensors, and temperature sensors. The speakers output ultrasonic waves at 20KHz - 50KHz and sets a humidity/temperature sensor to values that deter spiders in order to facilitate a clear field of view.

All these solutions would involve replacement of currently deployed cameras by new ones which would be expensive owing to the high cost of replacement of entire camera units at this point. Leaving economy aside, dumping thousands of legacy or even the current state-of-the-art surveillance cameras just for the advantage of reducing spider alerts would result in e-waste buildup.

2.4 Computer vision based solutions

While a large number of video analysis techniques have been developed specifically for investigating events in applications centered around humans such as detecting and tracking people (Dalal & Triggs 2005), real-time tracking of the human body (Wren et al. 1997), human face detection in complex backgrounds (Yang & Huang 1994), vehicle monitoring and tracking (Maurin et al. 2005), surveillance event detection and recognition (Armitage et al. 1999), and crime prevention (Piciarelli & Foresti 2011), very little attention has been paid to the analysis of

image sequences involving insects and more specifically spiders and spider webs. Most approaches to insect identification for environmental monitoring and ecological data analysis use a well-lit sophisticated microscope⁵. A computer vision approach with specialised hardware for automated rapid-throughput taxonomic identification of stoney larvae and anthropods is presented in a constrained lab environment in (Larios et al. 2007). Unlike this approach, the proposed work specifically focuses on spiders and spider webs close to the surveillance camera lens in cameras deployed in challenging environments.

Interpreting spiders with an image sequence can be challenging due to varying environment conditions like rain and snow, varying illumination conditions (day and night situations), heavily compressed low resolution images, and temporally sparse datasets (i.e., limited number of key frames). Furthermore, erratic spider movements in successive image frames make it difficult for analysis of spider shape and structure as do varying viewpoints based on how the surveillance camera is mounted and in some situations shaking or movement of the pole on which the camera is mounted. It is also worth noting that spiders too close to the lens are outside the camera's depth-of-field; hence spiders tend to appear defocussed.

Techniques such as masking zones of a surveilled area cannot be used as spiders tend to occupy almost the entire image or they jump erratically throughout the camera field of view. In such an approach, an object that first appears within a masked zone is not considered to be a reportable object until the object leaves the zone (Brodsky & Lin 2004). This technique can be applied to fixed objects like a tree shaking in a particular location in the field of view or situations where curtains get blown when an air-conditioner is turned on or a digital clock in a scene where only the digits change. However, this approach cannot be generalised for spiders/spider webs. As the images from CCTV might be of low contrast

⁵<http://web.engr.oregonstate.edu/tgd/bugid/>

and they might not have strong features, template matching (Lewis 1995) at a lower resolution does not work due to high intraclass variability in the shape and structure of spiders.

Visual features which are descriptive of a spiders are used to train a classifier using a large dataset taken from multiple low resolution (quality) and low cost cameras during both day and night. The next chapter describes and justifies the features used to discriminate between spider/webs and the *non-spider* categories. We compare our proposed visual features against state-of-the-art local features.

At the time of submission of this thesis, there are no documented studies dealing with spider detection/recognition or classification using computer vision technology in a surveillance setup. Computer vision for detecting spider alarms was chosen mainly because of important factors like economical viability and others as listed in Section 1.2. The proposed spider classification algorithm in a surveillance setup using real surveillance data is the first of its kind.

2.5 Conclusion

The purpose of this chapter was to review and discuss the state-of-the-art in spider-based false alarm reduction in surveillance camera networks. To date, false alarms triggered by spiders are typically addressed by chemical and hardware approaches. The *chemical based solutions* involve cleaning the exterior of surveillance cameras infested by spiders using spider deterrent sprays. The *hardware based solutions* require replacement of entire camera units or installation of additional hardware. These solutions were found to be expensive for surveillance industry deployments and would involve significant human effort. Therefore, an alternate solution is proposed that uses computer vision and machine learning for detecting spider-based alarms. The main factors motivating the decision are economical viability through reduction of human effort both in maintenance

of surveillance cameras and event handling. In addition to these benefits, the proposed solution is anticipated to have strong commercial potential.

Chapter 3

Computer vision based spider and spider web detection

3.1 Introduction

Of all the human senses, vision is probably the richest in content. It is estimated that more than 50% of the cortex, the surface of the brain, is devoted to processing visual information (Allyn 2012, Govindu 2013). Inspired by human vision and its underlying neural mechanisms, *computer vision*, as a discipline, covers a wide variety of methods for interpretation and analysis of visual data using a computer. The original goal of computer vision was to understand a single image of a scene, by identifying objects, their structure and spatial arrangement. This was extended to understanding image sequences and video data. Object recognition in image data is analogous to event recognition in video data (Haering & Lobo 2001).

While event classification is mostly applied in web video search, consumer video management and smart advertising (Jiang et al. 2012), the events of interest in this thesis are false alarms triggered by spiders. The first step for reduction of false alarms triggered by spiders is to detect spiders and spider webs in a scene.

Reduction of false alarms triggered by spiders is better formulated as an image classification problem rather than a recognition and localisation problem. Localisation may not be appropriate given that spider webs tend to occupy almost the entire camera field of view and also as spiders tend to be too close to the camera lens, they appear as large defocussed blobs. Image classification on the other hand provides a confidence score that could be used to trigger manual/automatic cleaning or to aid surveillance personnel decision making.

An image is classified according to its visual content. For example, classification may be used to find if an image contains a vehicle or not. In this case, classification suggests whether the image contains either spiders/spider webs. The main steps to follow for image classification include:

1. Manual labeling of images into *spider* and *non-spider* categories.
2. Separation of available data into *training* and *test* data (typically 70% of data is allocated for training and the remaining 30% is testing data) .
3. From the training set, a visual classifier for the two classes is built by extracting discriminative visual features from images in the *spider* category and *non-spider* category.
4. Assessment of performance of the classifier on *test* data by computing various metrics such as classification accuracy, computation time, and receiver operating characteristics curve.

Netwatch Security Systems provides three consecutive frames one second apart which represent an event, which is an industry standard format. The three images corresponding to an event are played in succession to form a three frame video, this video along with the 3 consecutive frames are termed a *quad* by Netwatch. The quads used in this thesis were captured from 275 camera views at different locations covering both indoor and outdoor scenarios. The low

temporal and spatial resolution of the format is a “real-world” challenge often not considered in the research community. The event is triggered by simple frame differencing. Chapter 4 further discusses the dataset supplied by our industry partner. Although the data is spatially compressed and temporally sparse, the algorithm proposed could be applied more generically to surveillance data from any security industry.

3.2 Problem formulation

To develop a spider classifier, the problem is formulated as a binary classification task. The manually annotated training data set takes the form

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \quad (3.1)$$

where $x_i \in X$ is a vector of feature values computed for a test image i and $y \in \{0, 1\}$ is the binary label of example i . Positive examples are images belonging to the *spiders* category and negative examples are the *non-spider* category comprising people, animals and vehicles in our dataset. Figures 3.1 and 3.2 show representative images from spider and non-spider classes.

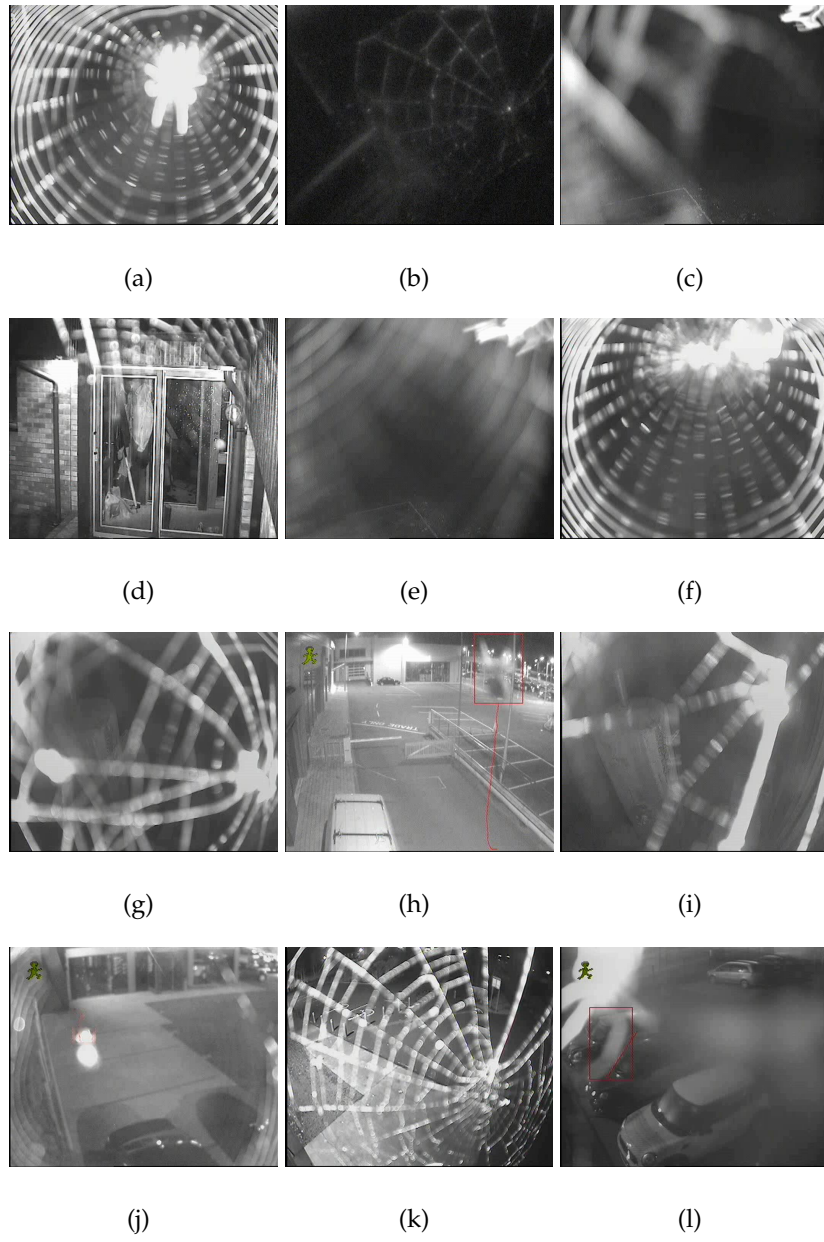


Figure 3.1: Positive examples used in image classification: *spider class* comprising of spiders and spider webs.



Figure 3.2: Negative examples used in image classification: *non-spider class* comprised of animals, people and vehicle.

A function $f : X \rightarrow \{0, 1\}$ is learnt to map every test image in X to a class label. Figure 3.3 demonstrates the proposed method for learning and predicting spider web images. It is mainly organised in two phases: learning (blocks to the left of the classifier) and classification (blocks to the right).

In the learning pipeline, the dataset is annotated into *spider* and *non-spider* classes by a human operator. Visual features from annotated images are extracted based on the assumption that texture and blur features contribute to discriminative capabilities for spider classification as discussed in Section 3.4. The features are then normalised. The extracted features along with class labels are then fed to a learning algorithm: a SVM (support vector machine) framework was used. A classifier model is built based on the features discriminative of *spider* and *non-spider* classes from the training data.

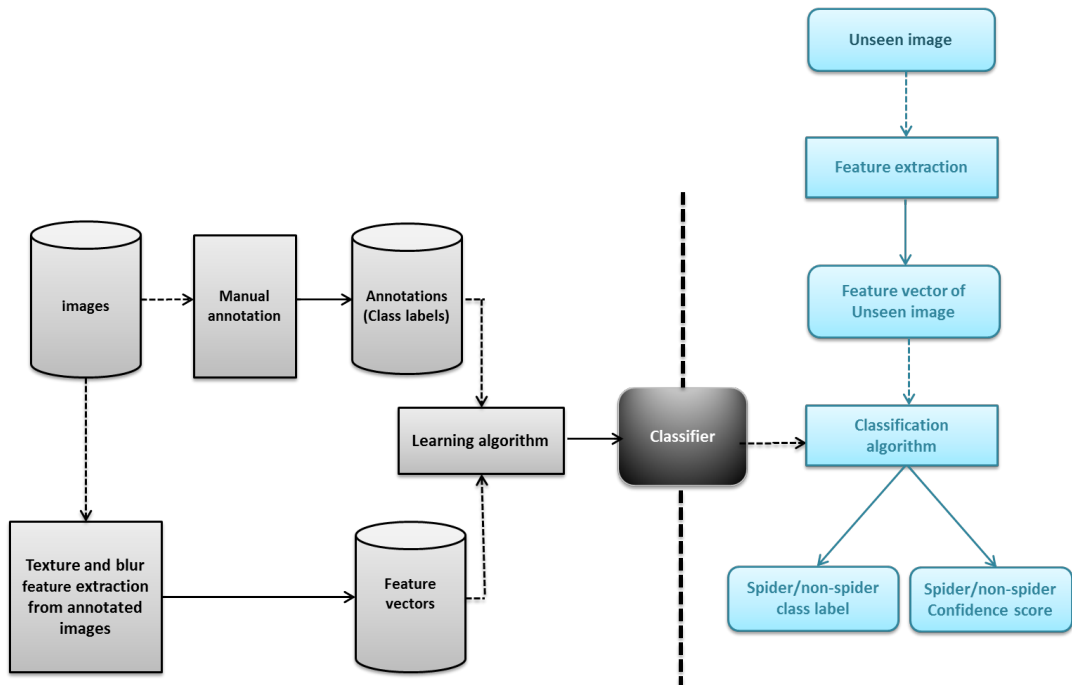


Figure 3.3: Block diagram showing the various components of the proposed spider classification system: a vertical dashed line separates the *Learning* and *Classification* phases.

During the testing phase, features from a previously unseen image are extracted. The feature vectors obtained are fed to the classification algorithm. The classification algorithm outputs class labels where class 1 corresponds to spider

class and class 0 corresponds to non-spider class. A probabilistic SVM (Platt 1999) also outputs the class probabilities to provide clues for surveillance operators for event prioritisation. Section 3.6.1 provides further details on Platt's probabilistic SVM.

3.3 Desirable characteristics for real time operation

Before either the selection of existing visual features or a decision to design new visual features, it is important to bear in mind some desirable characteristics for the target application. Based on these considerations, a descriptor suitable for spider classifier application is proposed.

3.3.1 Classification accuracy

Classification accuracy is a measure of true detections, specifically it is the proportion of correct predictions (both true positives and true negatives) of all the examples considered.

$$ClassificationAccuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.2)$$

where,

TP: true positives - spiders/spider webs classified into the *spiders* category

TN: true negatives - non-spiders classified into the *non-spiders* category

FP: false positives - non-spiders classified into the *spiders* category

FN: false negatives - spiders classified into the *non-spiders* category

Higher classification accuracy is always desirable. However in this application a very low number of false positives is an additional requirement given the consequence of misclassifying a real event as a spider (missing a potentially

hazardous event). The accuracy score is useful in scenarios where equal number of positive and negative samples would be used to train a classifier. In situations when the same number of *spider* and *non-spider* training samples are unavailable for training, i.e., the two classes are of very different sizes, Mathew's correlation coefficient could be considered (Powers 2011).

3.3.2 Computation time

Computation time is the sum of the time taken for visual feature extraction and classification, where an image is predicted to belong to either the *spider* and *non-spider* category. It is preferable that computation time be minimal. A low computation time should be an additional design goal for choosing existing or designing new visual descriptors which work in real-time or near real-time applications.

For this particular application, in consultation with Netwatch Security Systems it was decided that the decision making process should take no more than a second. The time taken should be reasonably small as the eventual aim is to design an algorithm that would be a part of a real-time video processing pipeline. Netwatch Security Systems suggested that surveillance personnel take from 45 seconds to slightly over a minute on average for manual event classification depending on day-night situations, complexity of events, and other factors.

3.3.3 Receiver Operating Curve (ROC)

A ROC is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The ROC shows the true positive rate plotted against the false positive rate while the probability threshold is varied. This allows the selection of to pick an appropriate value for the threshold (Swets 1996). ROCs are very important considering how important the recall is in

the sense of not missing *real* events or *non-spider* events in our classification task. If we consider spiders as nuisance events, our intent is to minimise false positives (non-spider or real events being marked as spiders) while maximising the true positives. Picking an operating point on this curve with low false positive rate may reduce the absolute accuracy, but will reduce the probability of a real event being misclassified.

3.3.4 Classifier confidence score

Surveillance personnel are not only interested in the class labels (spider \rightarrow 1 and *non-spider* \rightarrow 0) but also classifier confidence score in the result (i.e., the degree of its belief that the output should belong to the *spider* category). To support surveillance personnel, a confidence score can be obtained during classification with an SVM using Platt's probabilistic framework. Platt obtains SVM probabilities by training the parameters of an additional sigmoid function to map the SVM outputs into probabilities (refer to Section 3.6.1 for further details on probabilistic SVM (Platt 1999)).

3.4 Feature Extraction

3.4.1 Introduction

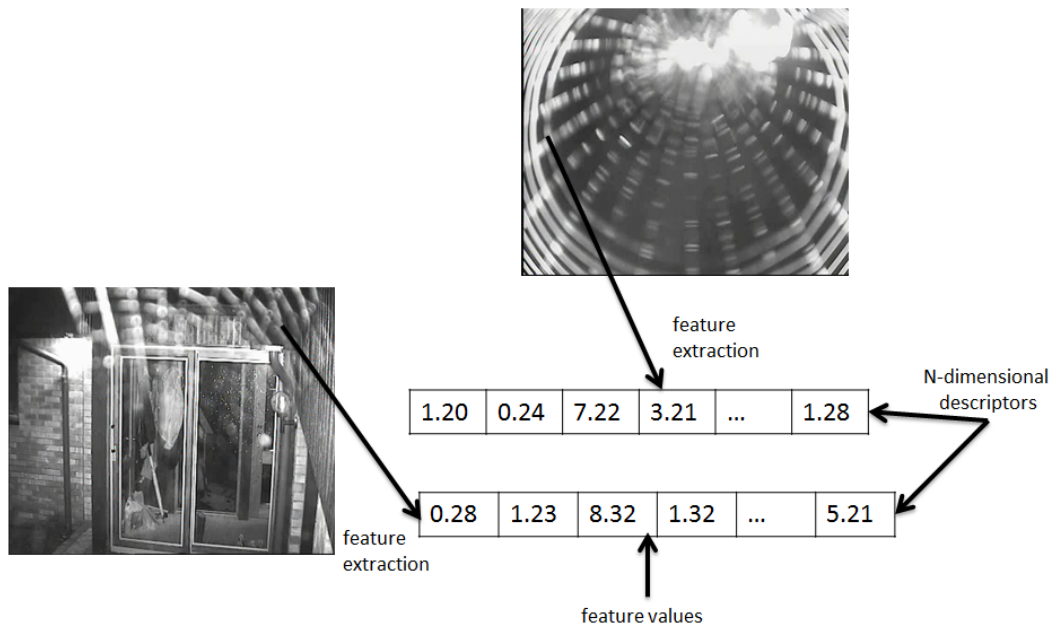


Figure 3.4: Feature extraction showing images represented using a descriptor. A descriptor is a fixed array of numbers also known as a *feature vector*. A set of features that describes one case (i.e., a row of predictor values) is called a vector. The *dimension* corresponds to the number of values in a descriptor.

Feature extraction consists of transforming generic arbitrary data, such as text or images, into numerical features usable for machine learning. The features are functions of the original measurement variables used in classification (Philpot 2011). Feature extraction also reduces the dimensionality by reducing the amount of redundant data to be processed, at the same time describing the data with sufficient accuracy. Figure 3.4 depicts visual feature extraction from images. An image feature is a distinguishing primitive characteristic or attribute of an

image. Some features are natural in the sense that such features are defined by the visual appearance of an image, while other artificial features result from specific manipulations of an image (Pratt 1978).

Our objective in this thesis is to extract features that are sensitive to the presence of spiders and spider webs while preferably remaining invariant to other variations in image content.

3.4.2 Cues for visual feature extraction

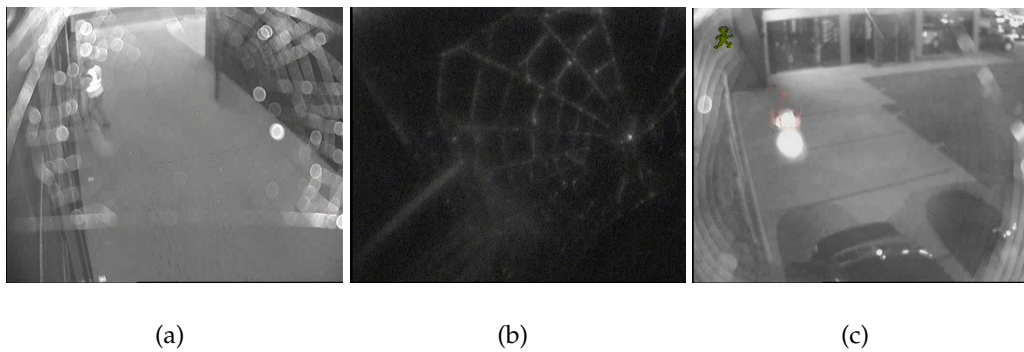


Figure 3.5: Sample spider images for visual feature design

The first computation step in both the learning phase and the classification phase is to perform visual feature extraction. For the visual feature descriptor design, combining contextual information such as the presence of spider and spider webs' proximity to surveillance camera lens surface should be considered. Figure 3.5 shows the typical appearance of spiders/webs in a surveillance setup. Observation of the coarse regular pattern found in the webs motivate to investigate statistical texture features. In addition to the texture information, an extent of image blur is chosen as another dominant feature considering that spiders appear blurry. Features that possess rotation invariance are investigated due to the fact that a spider or a spiderweb can occur in different orientations. No motion features (e.g. optical flow) were considered, since the smoothness constraints of

optical flow computation are usually violated when using only three images, each spaced one second apart.

3.4.3 Descriptor fusion

| Channels | SIFT | SPIN | RIFT | SIFT+SPIN | RIFT+SPIN | SIFT+SPIN +RIFT |
|----------|----------|----------|----------|-----------|-----------|--------------------|
| HS | 89.2±1.0 | 86.1±1.1 | 82.7±1.0 | 93.7±0.8 | 89.8±1.1 | 94.2±0.9 |
| LS | 94.9±0.7 | 87.9±1.0 | 88.5±0.9 | 94.7±0.8 | 91.4±0.9 | 95.2±0.7 |
| HS+LS | 94.4±0.7 | 90.2±1.0 | 89.6±1.0 | 95.4±0.7 | 92.8±0.8 | 95.9±0.6 |

Table 3.1: Feature combination results from (Zhang et al. 2007) on the Brodatz dataset. Because the RIFT and SIFT visual features provide similar information, the combination of the two does not yield greater performance. On the other hand, combining the SPIN feature with either results in improved accuracy. This Figure also shows different types of feature detectors: HS → Harris , LS →Laplacian and HS+LS → the combination of the two.

Work reported in (Gehler & Nowozin 2009) and (Weijer & Schmid 2006) motivate the idea of combining complementary descriptors to create a more discriminative descriptor that will work well in a wider variety of situations. There has been lot of research done in the area of descriptor fusion and it has been proven that fusion of complementary descriptors yields better results than the individual feature alone (Gehler & Nowozin 2009). The simplest way to combine descriptors would be to concatenate feature vectors and then use the combined vector through the same matching or classification procedure. Table 3.1 for example shows that fusing complementary descriptors for texture classification provides better discriminatory capabilities than using a single descriptor.

Descriptor fusion or descriptor combination could be achieved in many ways. The simplest is early fusion where the feature vectors from different image descriptors are concatenated into a single feature vector and then passed to the matching procedure or classification. Late fusion, also termed decision level fusion, involves merging of classification scores at a decision level.

The feature $X = \{X_a; X_b\}$, where X_a is a feature vector corresponding to blur and X_b , a feature vector corresponding to texture. These two features are complementary in nature and hence fusing these two features has merits over using a single feature. The merits of early descriptor fusion over using a single visual feature is detailed in Chapter 5.

Figures 3.6 and 3.7 illustrate early and late fusion schemes for visual descriptor fusion. (Ayache et al. 2007) investigated different fusion schemes derived from the classical early and late fusion schemes when using an SVM classifier for detecting pre-defined concepts in an image. They showed all fusion methods performed better on an average than a single feature in the concept detection task of TRECVID06. Normalised early fusion was found to be a good way to balance the influence of individual features. Hence, an early fusion strategy is used for classification although early fusion will generate feature vectors with larger dimensions. For some classifiers, this might imply more processing during the training. On the other hand, late fusion somehow assumes independence between the components of different feature vectors, as they are considered separately.

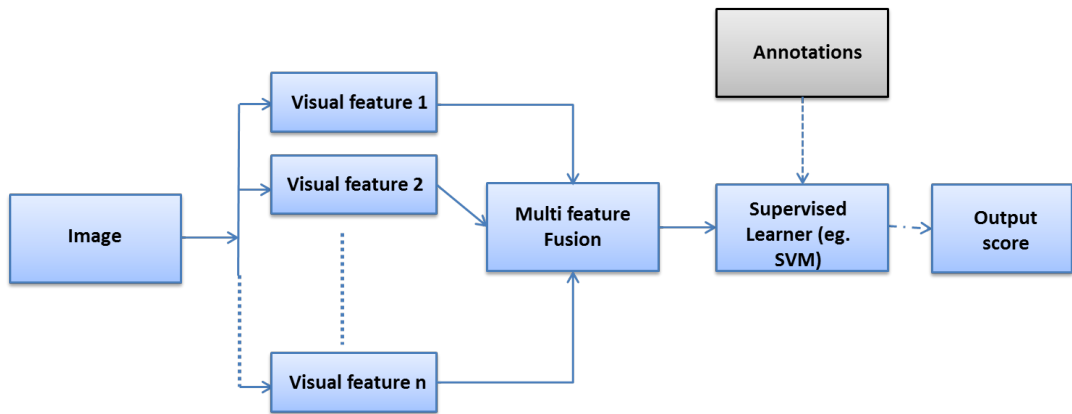


Figure 3.6: An illustration of the *Early Fusion scheme* as applied to visual features extracted from a single image. Multifeature fusion can refer to concatenation of feature vectors. The output score corresponds to class labels and probability.

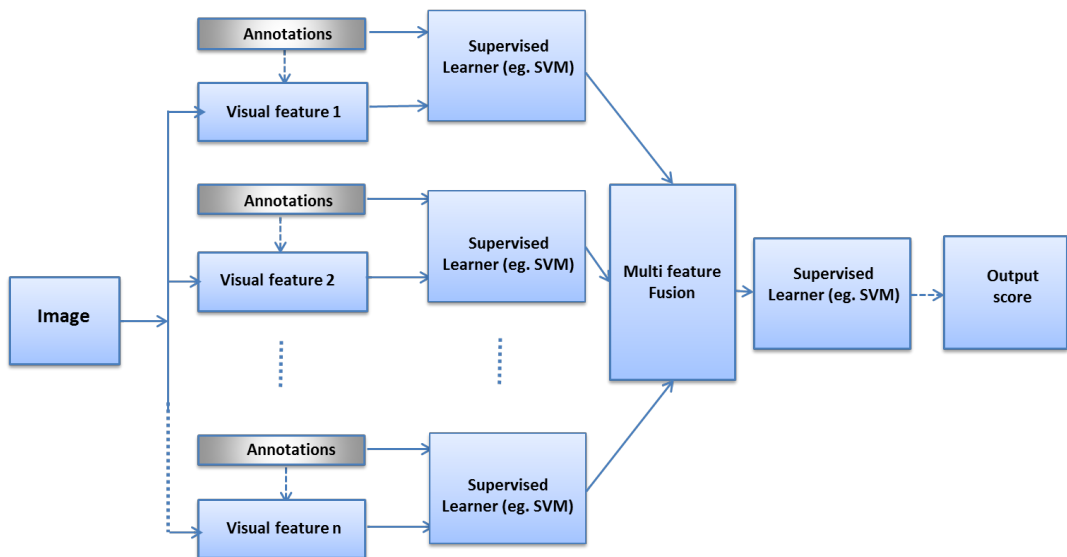


Figure 3.7: An illustration of the *Late Fusion scheme* as applied to visual features extracted from a single image. The output score corresponds to class labels and probability.

3.4.4 Feature normalisation

Feature normalisation follows feature extraction. Feature normalisation/scaling is a method used to standardise the range of independent variables or features. The simplest method is rescaling the range of features to the range $[0, 1]$ or $[-1, 1]$. Scaling to $[0,1]$ is achieved as follows

$$x' = \frac{(x - m_i)}{(M_i - m_i)} \quad (3.3)$$

where x is the original value, x' is the scaled value, and M_i, m_i are the maximal and minimal values of the i^{th} attribute respectively.

The main advantage of scaling is to avoid attributes with greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculations (Juszczak et al. 2002). Feature normalisation was performed on all the descriptors investigated in Section 3.5.

3.5 Investigation of visual features

To a researcher in the field of computer vision, with so many varieties of local image descriptors already available, selection of a particular image feature can prove to be a daunting task with no easy or deterministic way to choose which descriptor is the best for a particular application. Semantic concept classification is similar to concept *spider classification*. It comprises of (1) data annotation, (2) feature extraction, (3) training a classifier, and (4) determining if the trained classifier is able to judge the existence of a semantic concept by analysing a visual feature extracted from a previously unseen image (Naphade & Smith 2004). However, our data is temporally sparse, spatially compressed, and with artificial artifacts/overlays (see Section 4.3 for details on the available dataset). The approaches discussed in semantic concept classification cannot just be applied

to the *spider* concept as they are designed to be generic across multiple concepts. The case under consideration is so specific that the features used are more adapted to one specific *spider* concept, while in bigger collections there are hundreds of concept to classify.

The following feature descriptors and their combination are evaluated based on particular characteristics of spiders and spider webs, specifically extent of blur and a texture particular to spider webs. Based on the knowledge that spiders close to the lens appear blurry, two feature descriptors encompassing blur information considered for feature extraction were: cumulative probability of blur (CPBD) and blur histograms. From that understanding of spiderwebs are found to have coarse texture properties, important texture features considered were: Haralick texture features, LBP Variance, SIFT/BoVW, and RootSIFT/BoVW. Some examples of spiders and spiderwebs show strong overexposure and underexposure of light as opposed to the non-spider category motivating us to use intensity or grayscale histograms. Thus, the features investigated are:

1. Intensity/ Grayscale histograms
2. A blur/sharpness metric based on the cumulative probability of blur detection (CPBD) (Narvekar & Karam 2011)
3. Blur histograms – a histogram of blur values on a 8×8 grid over the image, computed using CPBD (Narvekar & Karam 2011).
4. A well-known statistical method based on gray tone spatial dependencies for image classification called the Haralick texture descriptor. We have used the fast Haralick features described in (Miyamoto & Merryman 2005).
5. Early fusion of easily computable basic statistical features – optimised Haralick and CPBD.

6. A rotation invariant Local Binary Pattern Variance descriptor (LBP variance) (Zhenhua Guo & Zhang 2010)
7. Early fusion of LBP variance and CPBD.
8. SIFT with Bag of Visual Words (Lowe 1999, Sivic & Zisserman 2003) as typically used in literature for semantic concept classification.
9. RootSIFT with Bag of Visual Words (Arandjelovic & Zisserman 2012) – another popular approach for concept detection.

SIFT and RootSIFT were investigated in order to compare against the state-of-the-art approaches for semantic concept detection. The following subsections describe the descriptors listed previously in more detail.

3.5.1 Intensity or Grayscale histograms

An intensity histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. Mathematically an intensity histogram shows gray levels in the range $[0, L - 1]$ and a discrete function

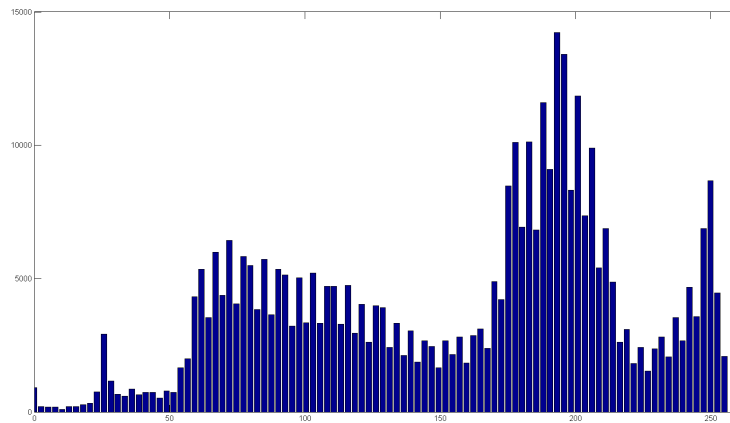
$$h(r_k) = n_k \tag{3.4}$$

represents an intensity histogram, where r_k is the k^{th} gray level and n_k is the number of pixels in the image having gray level r_k .

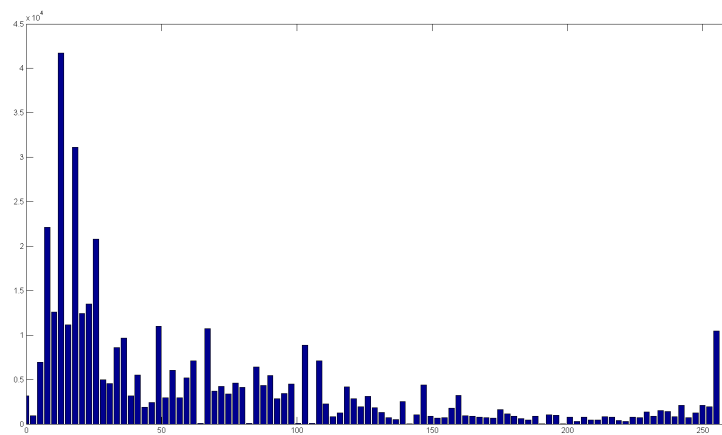
For an 8-bit grayscale image there are 256 possible intensities. The histogram of an 8-bit image can be thought of as a table with 256 entries, or bins, indexed from 0 to 255. In bin 0 we record the number of times a gray level of 0 occurs; in bin 1 we record the number of times a grey level of 1 occurs, and so on, up to bin 255.



(a) *non-spider* event triggered by a car (b) *spider* event triggered by a crawling spider



(c) 100 bin histogram of image (a)



(d) 100 bin histogram of image (b)

Figure 3.8: Sample intensity histogram of *spider* and *non-spider* category.

Grayscale histograms were chosen based on visual inspection of images in the spider category; the spiders appear as dark defocussed blobs during day light and white defocussed blobs during the night. This means the images appear either underexposed or overexposed. Histograms are typically used for thresholding, but in this case they are used feature vector which contains discriminatory information of *spider* and *non-spider* class. The number of bins was varied (50 to 250 in steps of 10) and the best classifier performance was achieved when 100 bins were used. Figure 3.8 shows a sample histogram with 100 bins obtained for images belonging to *spider* and *non-spider* categories. The histograms show that the *non-spider* image is either overexposed or normally exposed¹ i.e., with richer contrast than the *spider* image which is underexposed.

3.5.2 Optimised Haralick texture features

A method to describe statistical textural properties in blocks of image data in the spatial domain is proposed in (Haralick et al. 1973) . Statistical methods usually analyse the spatial distribution of gray values, by computing local features at each point in the image, and deriving a set of statistics from the distributions of the local features. Depending on the number of pixels defining the local feature, statistical methods can be further classified into first-order (one pixel), second-order (two pixels) and higher-order (three or more pixels) statistics (Ojala & Pietikinen 2012). Haralick et al. (1973) compute a set of gray-tone spatial-dependence probability distribution matrices and suggest a set of textural features extracted from these matrices. A gray-level co-occurrence matrix given is by G (Equation 3.5), which forms the basis of statistical texture features.

¹<http://www.luminous-landscape.com/columns/determining-exposure.shtml>

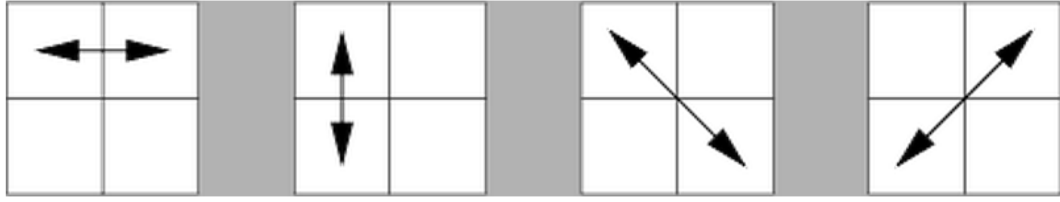


Figure 3.9: The four directions of adjacency as defined for calculation of the Haralick texture features. The Haralick statistics are calculated for co-occurrence matrices generated using each of the four directions of adjacency.

$$G = \begin{pmatrix} p(1, 1) & p(1, 2) & \cdots & p(1, N_g) \\ p(2, 1) & p(2, 2) & \cdots & p(2, N_g) \\ \vdots & \vdots & \ddots & \vdots \\ p(N_g, 1) & p(N_g, 2) & \cdots & p(N_g, N_g) \end{pmatrix} \quad (3.5)$$

where $P(i, j)$ is the relative frequency with two neighboring resolution cells. Figure 3.9 shows adjacency can be defined to occur in each of four directions in a 2D square pixel image (horizontal, vertical, left and right diagonals). Since rotation invariance is a primary criterion for any features used with these images, invariance was achieved for each of these statistics by averaging them over the four directional co-occurrence matrices (Boland 1999).

The significant features extracted from G are: homogeneity measured by angular second moment given by f_1 , linear structure, contrast measured by a difference moment of that matrix given by f_2 and the number of edge boundaries

present and the complexity of an image given by f_3 :

$$f_1 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{P(i, j)^2}{R}, \quad (3.6)$$

$$f_2 = \sum_{i=0}^{N_g-1} n^2 \left\{ \sum_{|i-j|}^n \frac{P(i, j)}{R} \right\}, \quad (3.7)$$

$$f_3 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} [ijP(i, j)/R] - \mu_x\mu_y}{\sigma_x\sigma_y}, \quad (3.8)$$

where, N_g is the number of quantised gray tones or distinct gray levels, and $P(i, j)$ is the relative frequency within two neighbouring resolution cells and μ_x, μ_y, σ_x , and σ_y are the means and standard deviations of marginal distributions associated with $P(i, j)/R$ and R is a normalising constant.

The 13 significant texture features out of 28 for fast calculation of Haralick features as described in (Miyamoto & Merryman 2005) are chosen. Table 2 in (Miyamoto & Merryman 2005) contains the other 10 formulae used which take into account a variety of entropy measures. Although computationally heavy, optimised code improves the computation speed of the feature calculation phase by a factor of 20 and construction of co-occurrence matrices by 20% by using a recursive blocking algorithm, scalar replacement and removal of redundancies.

3.5.3 Cumulative Probability of Blur Detection

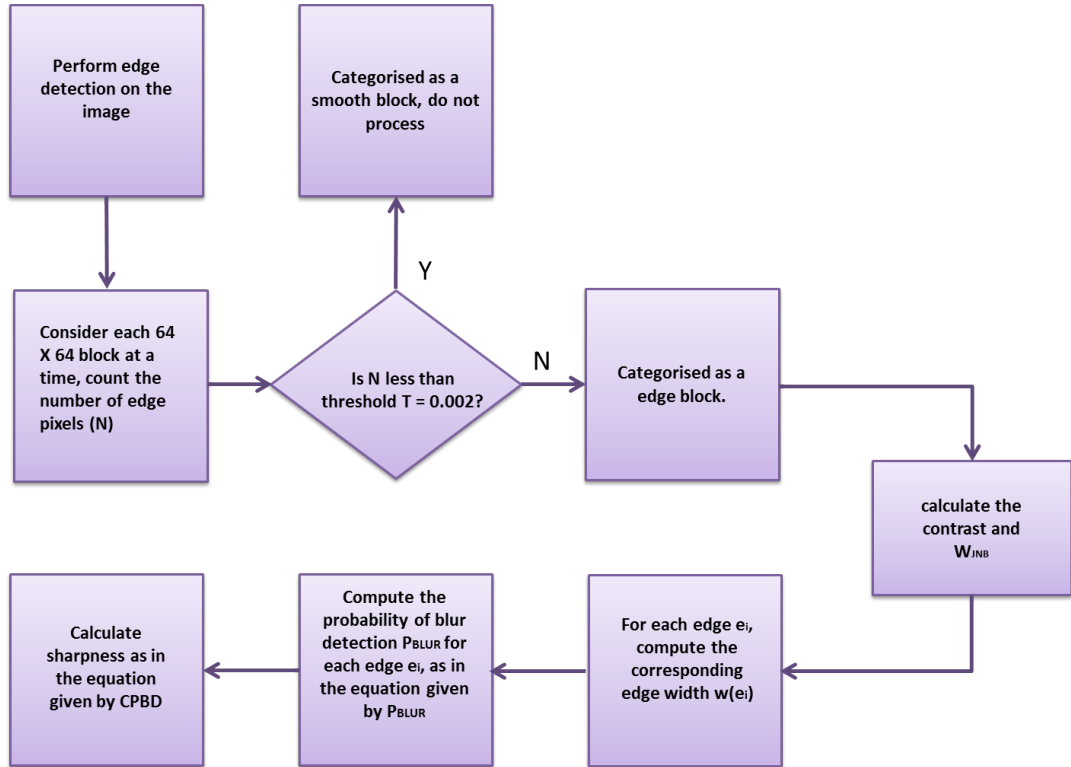


Figure 3.10: Block diagram illustrating the computation of the CPBD metric.

Image blurring can arise from a variety of sources – atmospheric scatter, lens defocus, optical aberration etc. As evident from the images representing the *spider class* in Figure 3.1, spiders/spider webs are most likely closer to the surveillance camera lens, which means outside the depth-of-focus as most surveillance cameras are focused at infinity. Hence, spiders and webs appear defocused and blurry.

A blur metric based on Cumulative Probability of Blur Detection (Narvekar & Karam 2011) is chosen as it is non referential in nature. This means that the system does not need a baseline to measure against. The metric is evaluated by taking into account the Human Visual System (HVS) response to blur distortions. The descriptor is intended to produce results with a very good correlation with

subjective scores especially for images with varying levels of perceived foreground and background blur. This metric uses no reference information from other images unlike full reference blur metrics like the structural similarity index (Wang et al. 2004). Since a human attention model is taken into consideration in development of the metric, it is anticipated this metric to correlate well with human blur perception.

Most blur detection is based on measuring the width of edges in an image. The CPBD metric performs edge detection as well, but instead of simply averaging the edge widths, it postulates that the blur around an edge is more or less noticeable depending on the local contrast around that edge. It derives a human perceptible threshold called *Just Noticeable Blur (JNB)*, which can be defined as the minimum amount of perceived blurriness around an edge given a contrast higher than the *Just Noticeable Difference (JND)*. It defines another edge width, called the JNB edge width, which is based on the local contrast around the edge. The probability of blur detection at an edge, for a given contrast, takes the form of a psychometric function which can be modeled as follows:

$$P_{BLUR} = P(e_i) = 1 - \exp\left(-\left|\frac{w(e_i)}{w_{JNB}(e_i)}\right|^\beta\right) \quad (3.9)$$

where w_{JNB} is the JNB edge width which depends on local contrast and $w(e_i)$ is the measured width of edge e_i and β is obtained by means of least squares fitting.

Figure 3.10 shows the block diagram summarising the calculation of the CPBD sharpness metric. The image is first divided into 64×64 blocks and then each block is characterized as an edge block or non-edge block as described in (Ferzli & Karam 2009). The non-edge blocks are not processed further, whereas, for each edge block, the width of each edge in the block is determined. The probability of blur detection at each edge is estimated using Equation 3.9, in which w_{JNB} depends on the contrast C of the edge block to which the edge belongs. It should

be noted that when $w(e_i) = w_{JNB}(e_i)$ then $P_{BLUR} = 63\% = P_{JNB}$. It follows that the blur is not detected at an edge if $P_{BLUR} \leq P_{JNB}$. Finally, the cumulative probability of blur detection is calculated as:

$$CPBD = P(P_{BLUR} \leq P_{JNB}) = \sum_{P_{BLUR}=0}^{P_{JNB}} P(P_{BLUR}) \quad (3.10)$$

where $P(P_{BLUR})$ denotes the value of probability distribution function at a given P_{BLUR} .

3.5.4 Blur histograms

The Probability of Blur Detection (PBD) histogram is accumulated into a single scalar value in CPBD (Narvekar & Karam 2011). However, useful information is lost by discarding the blurry edge values in a block (64 bins). Normalised PBD histograms retain the sharpness/blur information to achieve better classification results than CPBD. Thus blur histograms are also investigated in Chapter 5.

3.5.5 Early fusion of Haralick texture features and CPBD

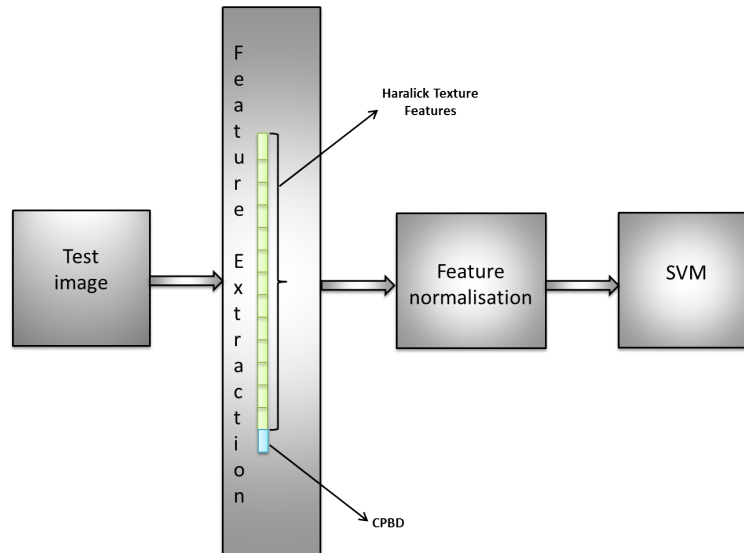


Figure 3.11: Early fusion of CPBD and Haralick features for image classification. The feature extraction block shows a concatenation of two feature vectors.

The optimised Haralick texture features and the CPBD blur measure provide complementary information about the image content. As such, fusing the descriptors is likely to provide more relevant information to the classifier and produce a higher-accuracy result. A simple early fusion strategy in which simply concatenates the feature vectors obtained from CPBD (Narvekar & Karam 2011) and optimised Haralick texture features (Miyamoto & Merryman 2005) is proposed. Since CPBD is only a scalar, this simply increases the overall dimension of the feature vector by one. Figure 3.11 illustrates early fusion of the two feature vectors.

3.5.6 SIFT with BoVW

The scale-invariant feature transform descriptor (SIFT) proposed by Lowe describes the local shape of a region surrounding a key point using edge orientation

histograms. In the current work, the difference of Gaussians key point detector is used to detect the key points in the images (Lowe 1999). A SIFT keypoint is a circular image region with an orientation. It is described by a geometric frame of four parameters: the keypoint center coordinates x and y , its scale (the radius of the region), and its orientation (an angle expressed in radians). Key points are defined as maxima and minima of the result of a Difference of Gaussians (DoG) function applied in scale space to a series of smoothed and resampled images as shown in Figure 3.12.

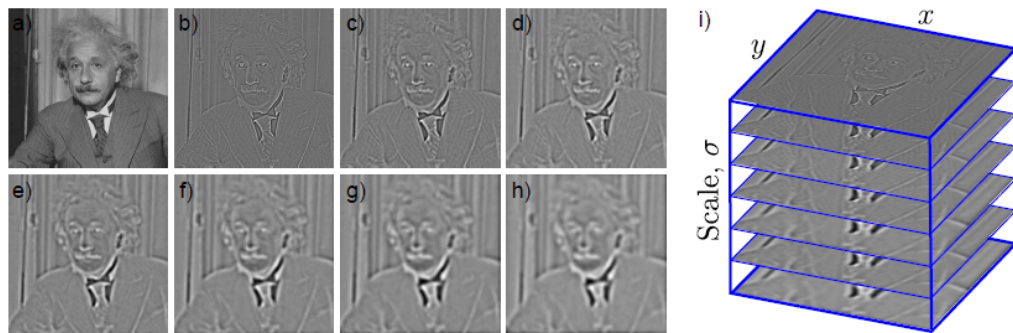


Figure 3.12: The SIFT detector. a) Original image. b-h) The image is filtered with difference of Gaussian kernels at a range of increasing scales. i) The resulting images are stacked to create a 3D volume. Points that are local extrema in the filtered image volume are considered to be candidates for interest points (Prince 2012).

The SIFT descriptor is a spatial histogram of the image gradient. The SIFT descriptor is assigned to each key point and built to be invariant against shift, rotation and lighting intensity changes, i.e. the gradient direction and the relative gradient magnitude remain the same under the different changes. Use of Histogram equalisation/stretching in SIFT/RootSIFT will not improve the performance because of inherent luminance invariance associated with these algorithms.

Figure 3.13 (b) shows the SIFT keypoint overlay on the test image in yellow and Figure 3.13 (c) shows feature description on a 4×4 grid using image gradient direction in green. This uses a popular VLFeat library (Vedaldi & Fulkerson 2008).

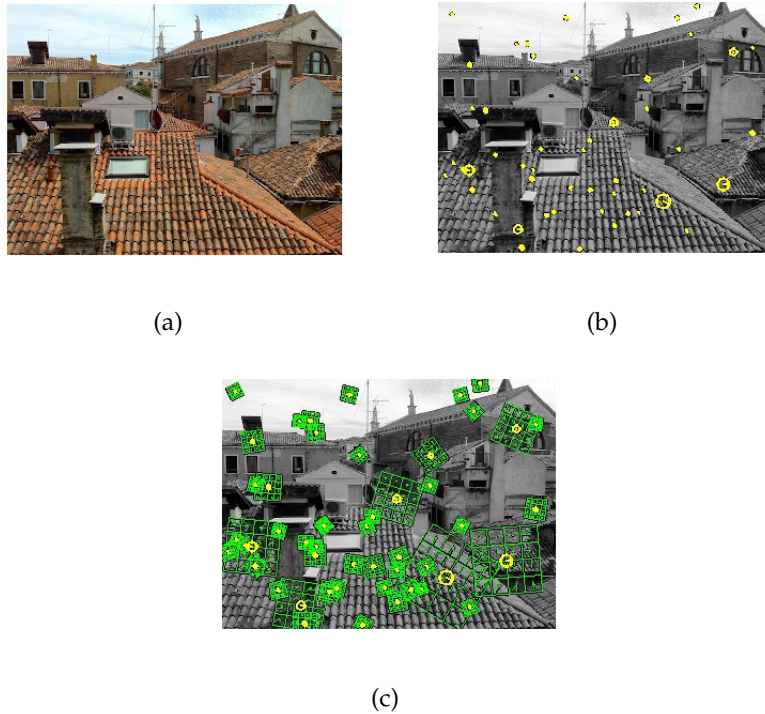


Figure 3.13: Demonstration of SIFT feature extraction (using VLFeat open source computer vision Library (Vedaldi & Fulkerson 2008)): Subfigure (a) is the original image. Subfigure (b) This image is transformed into grayscale and shown with 50 random SIFT keypoints overlaid. Subfigure (c) The image on the right is the SIFT descriptor overlay over the gray scale image.

The Bag of Words (BoW) model is traditionally used in document classification and represents a sparse vector of the frequency of words from a dictionary. In text analysis, a bag corresponds to a document, whilst words corresponds to the keywords. This was extended to images and termed *Bag of Visual Words (BoVW)*. BoVW represents a sparse vector of occurrence counts of elements of a vocabulary

of local image features. Figures 3.14 and 3.15 visually explain an image and its corresponding representation using a histogram of visual words.

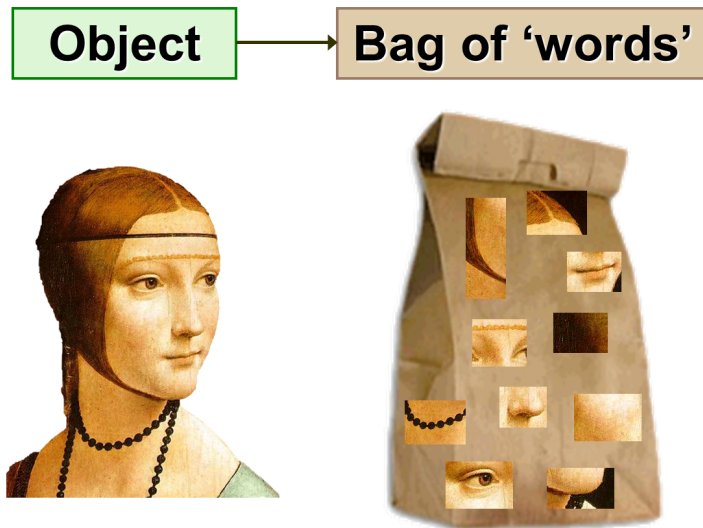


Figure 3.14: A bag-of-visual words model (source: “Recognizing and Learning Object Categories” by Li Fei Fei, Rob Fergus, and Antonio Torralba, ICCV, 2009).

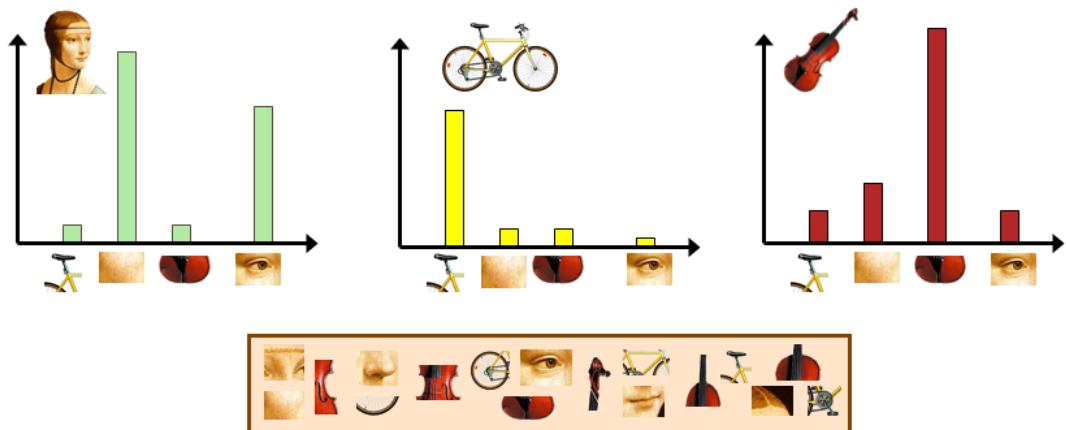


Figure 3.15: A histogram representation of Visual Words (source: “Recognizing and Learning Object Categories” by Li Fei Fei, Rob Fergus, and Antonio Torralba, ICCV, 2009).

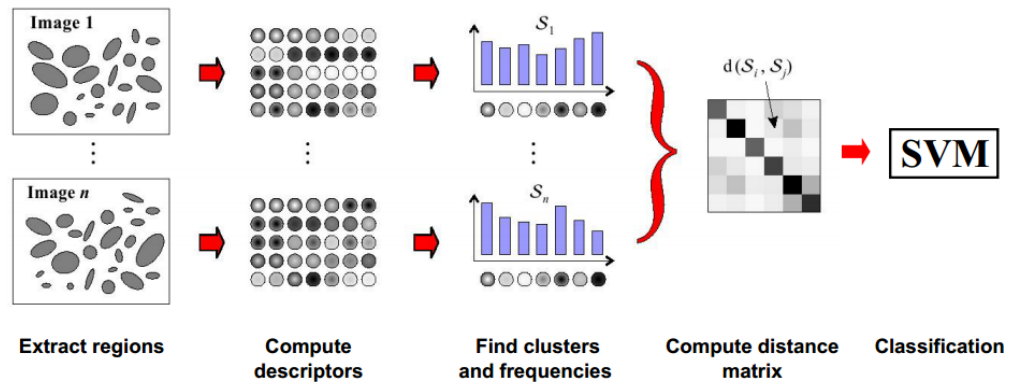


Figure 3.16: Bag-of-visual words for image classification. The steps in a Bag of Visual Words model are (a) Extraction of keypoint regions using Difference of Gaussians, (b) The region surrounding the keypoints are then described using the SIFT descriptor, (c) Since there are variable number of keypoints in every image, a fixed length histogram is used to represent an image using clustering, (d) A distance matrix (i.e., two dimensional array of distances computed from $N \times N$ matrix, where N is the number of points) is computed (e) An SVM is used for image classification based on the distance matrix

A fixed length descriptor is desirable for efficient classification, but images generally produce different numbers of SIFT key points. A bag of visual words approach (Sivic & Zisserman 2003) is used to aggregate the variable number of SIFT descriptors for an image into a fixed length histogram. This is done by first clustering the descriptors for all images in the training set to produce a codebook. Clustering is a common method for learning a visual vocabulary or codebook. Given this codebook, a visual word histogram descriptor is calculated for an unseen image which needs to be classified. Then each SIFT descriptor from that image is assigned to the nearest cluster centre in this codebook and the corresponding index in the histogram is incremented. Figure 3.16 shows the SIFT/BoVW approach used for image classification.

3.5.7 RootSIFT with BoVW

It is well known for problems such as texture and image classification, that using Euclidean distance to compare histograms often yields inferior performance compared to using measures such as χ^2 or Hellinger. A SIFT descriptor was originally designed to be used with Euclidean distance. Calculating Euclidean distance in the feature map space is equal to calculating the Hellinger distance in the original space as detailed in (Arandjelovic & Zisserman 2012) (Vedaldi & Zisserman 2012). Therefore, the performance of SIFT histogram for image classification can be boosted by using a better distance measure based on a Hellinger Kernel.

RootSIFT is simply a $L1$ norm of SIFT vectors followed by an element-wise square root of the SIFT descriptor (Arandjelovic & Zisserman 2012).

$$RootSIFT = \sqrt{\frac{SIFT}{sum(SIFT)}} \quad (3.11)$$

3.5.8 LBP variance

The LBP (Local Binary Patterns) operator is one of the best performing local texture descriptors and is widely used in texture classification (Ojala et al. 1994). LBP characterises the spatial structure by comparing a pixel with its neighbours.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (3.12)$$

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

where g_c represents the center pixel and $g_p(p = 0, 1, 2 \dots p - 1)$ denotes its neighbour on a circle of radius R , and P is the total number of neighbours. The

neighbours not falling within the radius can be estimated by bilinear interpolation.

Figure 3.17 shows calculation of LBP.

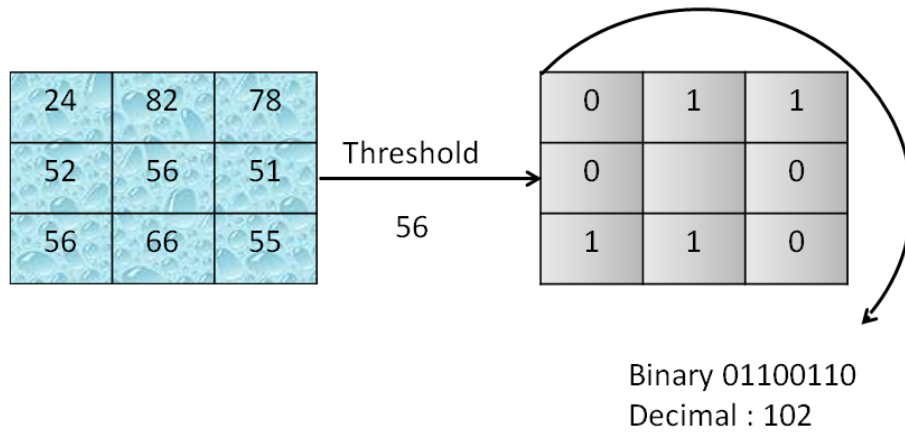


Figure 3.17: Illustration of LBP where, $P = 8$ and $R = 1$. The basic idea of Local Binary Patterns is to capture the local structure in an image by comparing each pixel with its neighborhood. If the intensity of the centre pixel is greater than or equal to its neighbour, then it is denoted with a 1 and 0 if not. The binary pattern is then used as an LBP code

Image texture is known to have two orthogonal properties – contrast and spatial structure. Contrast is affected by gray scale value changes while the spatial structure is affected by rotation. A rotation invariant measure VAR is introduced to incorporate the local image texture if gray scale invariance is not required.

$$VAR_{P,R} = \frac{1}{P} \sum_{p=0}^{P-1} (g_p - \mu)^2, \mu = \frac{1}{P} \sum_{p=0}^{P-1} g_p \quad (3.13)$$

Experimental results show that the performance of LBP variance is superior to LBP alone (Ojala et al. 1994). Local Binary Pattern (LBP) (Ojala, Pietikainen & Harwood 1994) features have the drawback of losing global spatial information, while global features preserve little local texture information. In LBP Variance,

an alternative hybrid scheme, globally rotation invariant matching is performed which is required for spiderweb classification (Zhenhua Guo & Zhang 2010). LBP variance (LBPV) is proposed to characterise the local contrast information in the one-dimensional LBP histogram (Zhenhua Guo & Zhang 2010). The LBP codes are computed on sample points on a circle of radius specified by a user – in the experiments, LBP was computed on an (8,1) neighborhood (where 8 corresponds to number of neighbours and 1 corresponds to radius) and a uniform rotation invariant LBP scheme was chosen for mapping (see Figure 3.17).

3.5.9 Early fusion of LBP Variance and CPBD

The feature vectors obtained from CPBD and LBP variance are fused using an early fusion scheme. Early fusion potentially results in having better discriminatory capability than using LBP, Variance and blur information independently (Narvekar & Karam 2011) and (Zhenhua et al. 2010).

3.6 Classification

3.6.1 SVM Introduction

A Support Vector Machine (SVM) is a powerful algorithm based on Vapnik-Chervonenkis statistical learning theory. Applications of SVM include classification, regression and anomaly detection. An SVM has strong regularisation properties. Regularisation refers to the generalisation of the model to new data. The advantages are as follows: SVM models have similar functional form to neural networks and radial basis functions, both of which are popular data mining techniques. However, neither of these algorithms has the well-founded theoretical approach to regularisation that forms the basis of SVM (Vapnik 2000) (Milenova et al. 2005).

Practically, a classification task involves separating data into training and testing sets. Each instance the training set contains class labels and the features or observed variables. The goal of an SVM is to produce a model based on the training data which predicts the target values of the test data given only the test data attributes (Hsu et al. 2003). Figure 3.18 presents an overview of an SVM for binary classification.

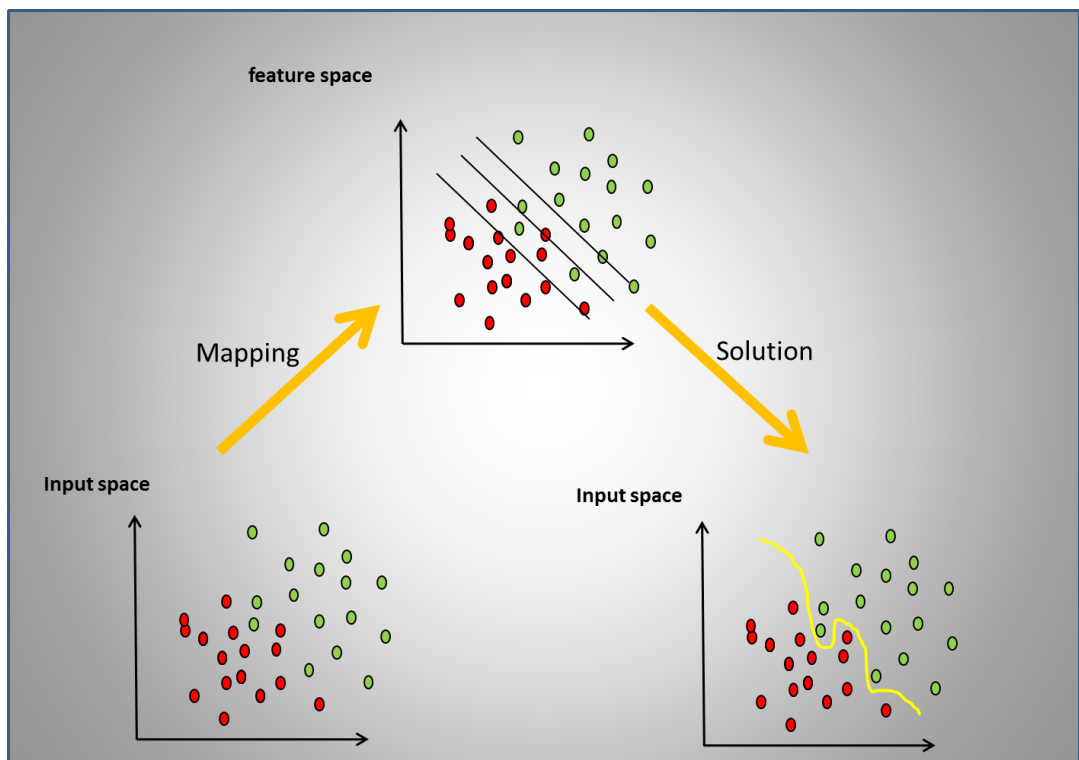


Figure 3.18: An illustration of the SVM showing binary classification

The input space is transformed to the feature space where the data is separated into two classes. The goal of SVM modeling is to find the optimal hyperplane that separates clusters of vectors in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other side of the plane. Using the terminology from the SVM literature, a predictor variable is called an attribute, and a transformed attribute that is used

to define the hyperplane is called a feature. The task of choosing the most suitable representation is known as feature selection².

To illustrate SVM operation, Figure 3.19 shows binary classification using a linear SVM depicting the support vectors and decision boundary.

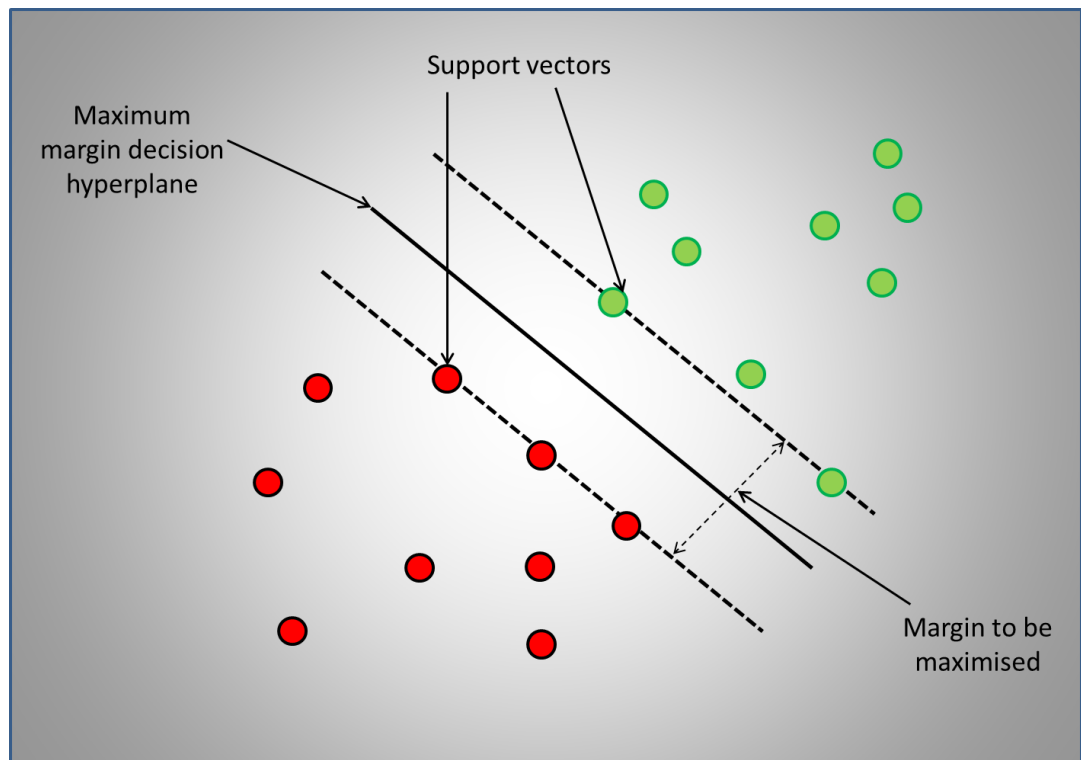


Figure 3.19: An example of a linear SVM showing 5 support vectors against the margin of a classifier where green circles → positive vectors and red circles → negative vectors.

The SVM defines the criterion to look for a decision surface that is maximally far away from any data point (Manning, Raghavan & Schütze 2008). This distance from the decision surface to the closest data point determines the margin of the classifier. This method of construction necessarily means that the decision function for an SVM is fully specified by a subset of the data which defines the

²<http://www.dtrek.com/svm.htm>

position of the separator i.e., the vectors near the hyperplane. These points are referred to as the support vectors (Manning et al. 2008).

With the knowledge that SVMs are extensively used in image classification, Platt (1999) and Lin. et al. (2007) propose support vector machine classifiers and a variation to produce probability outputs . Platt scaling basically fits a sigmoid³ on top of the SVM decision values to scale to the range of [0, 1], which can then be interpreted as a probability.

Mathematically, given the training examples $x_i \in \mathbb{R}^n, i = 1, 2, \dots, l$, labeled by $y_i \in \{+1, -1\}$, a binary SVM computes a decision function $f(x)$ such that $sign(f(x))$ can be used to predict the label of any test example x . Instead of predicting the label, many applications like ours would require posterior class probability $Pr(y = 1|x)$. Platt (1999) proposes approximating the posterior with a sigmoid function given by

$$Pr(y = 1|x) \approx P_{A,B} \equiv \frac{1}{1 + exp(Af + B)} \quad (3.14)$$

where $f = f(x)$, A denotes slope of the curve and B denotes the offset from the decision surface separating the two classes.

3.6.2 SVM classification setup

The binary visual classifier was trained for two classes, *spider* (positive) and *non-spider* (negative), using the previously described features. In the subsequent evaluation of the classifier, the soft margin SVM implementation provided by LIBSVM with a Radial Basis Function (RBF) kernel was used (Chang & Lin 2011). The soft-margin method will choose a hyperplane that splits the examples as cleanly as possible, while still maximising the distance to the nearest cleanly split

³A sigmoid function is a mathematical function having an S shape, it is a special case of logistic function

examples ⁴. An RBF kernel nonlinearly maps samples into a higher dimensional space so it, unlike the linear kernel, it can handle the case when the relationship between class labels and attributes is nonlinear. Normally, a Gaussian is used as the RBF kernel.

An RBF is characterised by two parameters : C and γ . The goal is to identify good values for (C, γ) so that the classifier can accurately predict unknown/test data. To find the optimal parameters for C and γ , a grid search is performed for optimal values of C and γ using ten fold cross-validation.

An RBF kernel (Vert et al. 2004) on two samples x and x' is given by

$$K(x, x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\sigma^2}\right) \quad (3.15)$$

where $\|x - x'\|_2^2$ is the squared Euclidean distance between the two feature vectors and $\gamma = -\frac{1}{2\sigma^2}$.

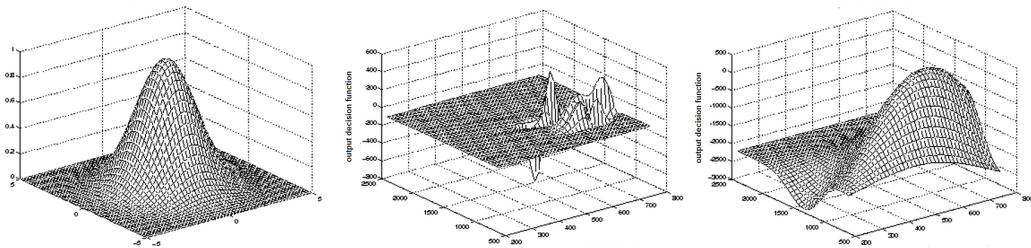


Figure 3.20: An example of RBF Kernel when σ is varied. The mesh plot at the center shows an RBF kernel when σ is small. The plot on the right shows an RBF kernel with larger value of σ for a smoother decision surface and more regular decision boundary. An RBF with large σ will allow a support vector to have a strong influence over a larger area. (Example from (Chin 1999))

Figure 3.20 shows a RBF Kernel when σ is varied. Intuitively, the γ parameter defines how far the influence of a single training example reaches, with low

⁴In a hard-margin SVM, a single outlier can determine the boundary, which makes the classifier overly sensitive to noise in the data.

values meaning far reach and high values meaning near reach as γ is inversely related to σ . The C parameter trades off misclassification of training examples against simplicity of the decision surface. Figure 3.21 shows that a low value for C makes the decision surface smooth, while a high value for C aims at classifying all training examples correctly (ScikitLearn 2010), (Chin 1999).

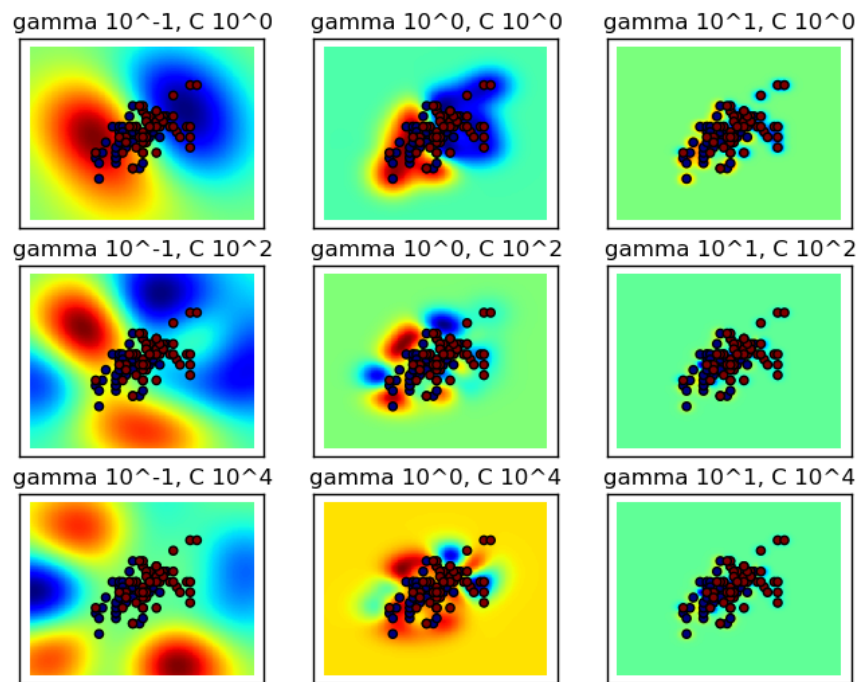


Figure 3.21: Visualisation of the decision function as cost parameter C is varied using `scikit-learn`. The C parameter trades off misclassification of training examples against simplicity of the decision surface. The C parameter tells the SVM optimisation how much misclassification of training examples is allowed. For large values of C , the optimisation will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimiser to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

3.7 Conclusion

In this chapter a spider classification pipeline is proposed to address false alarms triggered by spiders. The pipeline comprises of visual feature extraction and classification blocks. The cues for detecting spiders/spider webs are discussed. An investigation for various image descriptors was carried out proposing a new descriptor for classifying images into *spider* and *non-spider* class based on image texture and blur characteristics. This chapter also discussed the SVM classification framework and its variation to provide confidence scores. These confidence scores can then be used to filter events that have high probability of being caused by spiders or spiderwebs, while ensuring that true events are very unlikely to be classified incorrectly.

Chapter 4

Dataset

4.1 Introduction

The dataset used in this thesis was gathered from CCTV surveillance footage from Netwatch Security Systems, a well-known Irish surveillance company. Netwatch provides remote CCTV System monitoring and protection for business premises. The company uses video analytics to detect events that are passed to a human operator for manual verification. The existing analytics software generates three images taken one second apart for every event triggered, where the event triggering mechanism is based on motion calculated by frame differencing. The three images corresponding to an event are played in succession to form a three frame video, this video along with the 3 consecutive frames are termed a *quad* by our industry partner. The quads were captured from 275 camera views at different locations covering both indoor and outdoor scenarios. Intervention specialists categorise events as *true* or *false* based on visual inspection of these quads.

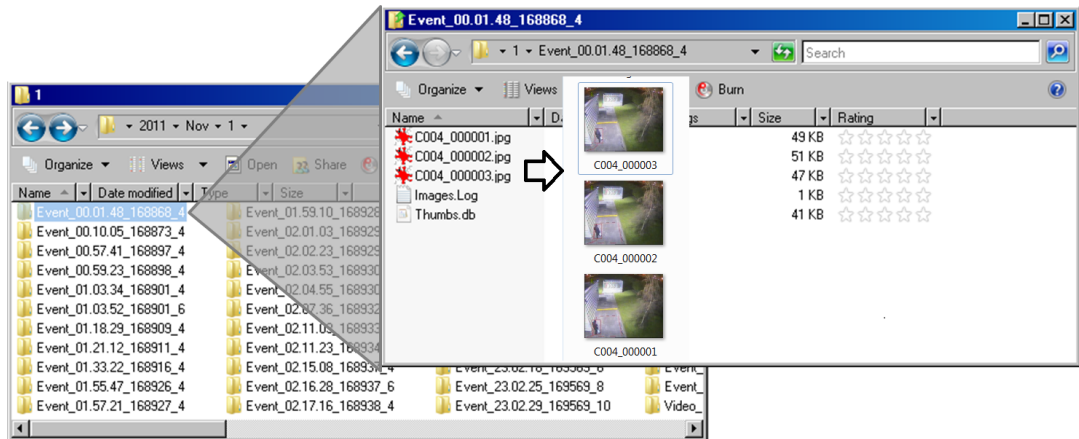


Figure 4.1: A selection of triggered events, where each event comprises of three JPEG images.

Figure 4.1 shows the structure of events detected by the existing software. In the example, the input directory is from the year 2011, '2011\Nov\1' where an example 'quad' named *Event_00.01.48_168868_4* has three images: *C004_000001.jpg*, *C004_000002.jpg* and *C004_000003.jpg*.

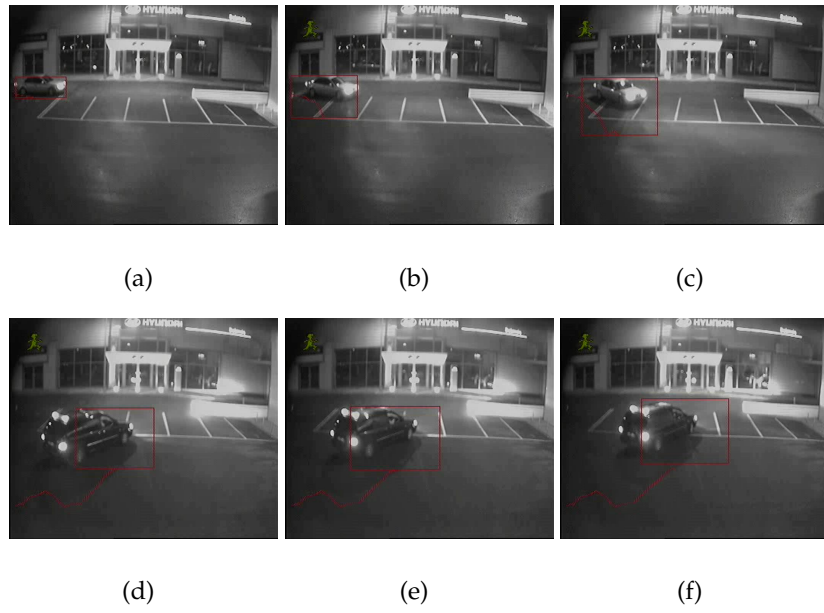


Figure 4.2: Example of true events: Each row corresponds to a detected event - comprising of three frames taken a second apart. Subfigures (a,b, and c) and (d,e, and f) respectively show two events triggered by movement of a vehicle in the scene

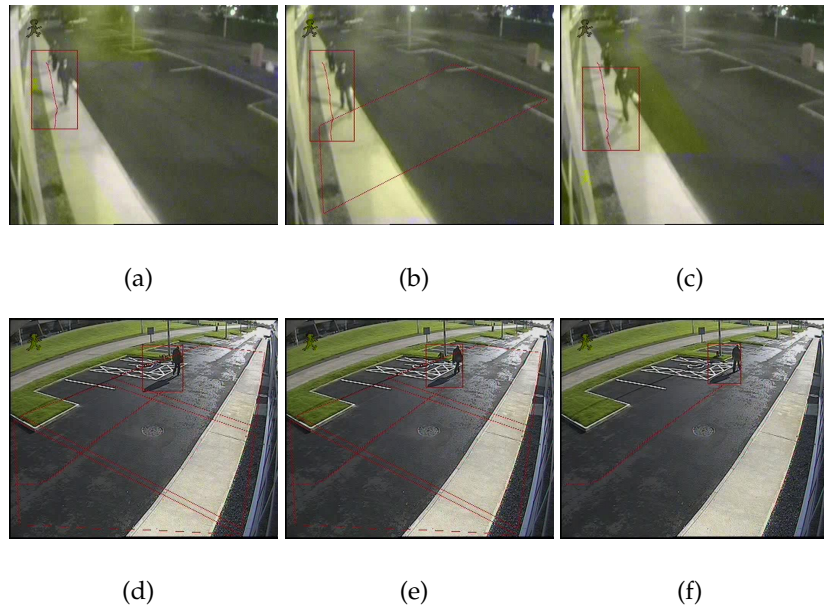


Figure 4.3: Example of true events: Each row corresponds to a detected event - comprising of three frames taken a second apart. Subfigures (a,b, and c) and (d,e, and f) respectively show two events triggered by people walking in the scene

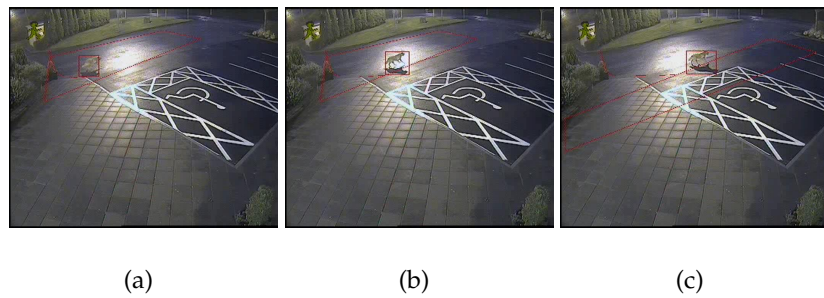


Figure 4.4: An example of true event triggered by animal: Subfigures (a,b, and c) shows an event triggered by a dog running in the area under surveillance

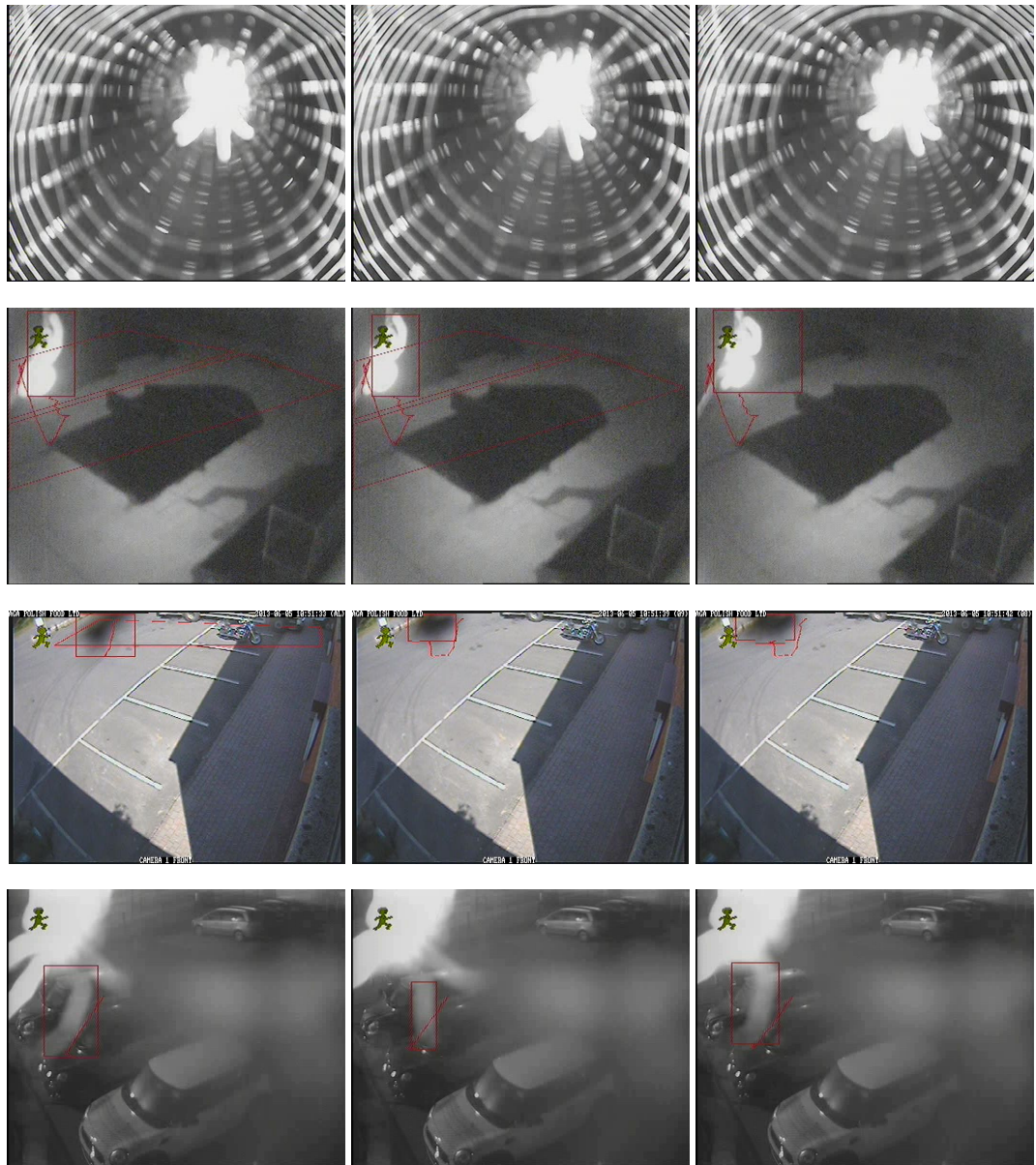


Figure 4.5: Examples of nuisance events: Each row corresponds to an event triggered; each event is comprised of three frames taken a second apart. All sample events shown are triggered by spiders crawling over the camera view or spider web shake due to wind.

Figures 4.2, 4.3, 4.4 and 4.5 show some representative samples of true and false detections for a variety of different events. From these samples of events, it can be

inferred that the three images are compressed spatially showing JPEG blocking artefacts. It can be noticed that the dataset is temporally sparse with just a single frame per second¹. Frames include artefacts introduced by the existing analytics system. The artefacts include a green man on the top left of the image and/or red boxes and trails in some events indicating the localisation and tracking for the assumed intruder. These three images are then passed to an intervention specialist for verification.

The artificial artefacts (image overlays) which take the form of trails, lines and a green man symbol pose challenges in video analysis, because most visual descriptors use gradient information from edges, lines and corners for feature description (i.e., information in artefacts are picked up for feature description along with actual features representing spiders; hence these do not accurately represent spider class images). Thus artefact reduction pre-processing is required to more accurately extract features and help to achieve better classification results.

¹Full frame rate is 30 frames/second but over 82% of video surveillance recording is at the rate of 6 frames/second - <http://ipvm.com/updates/1100>

4.2 Annotation tool

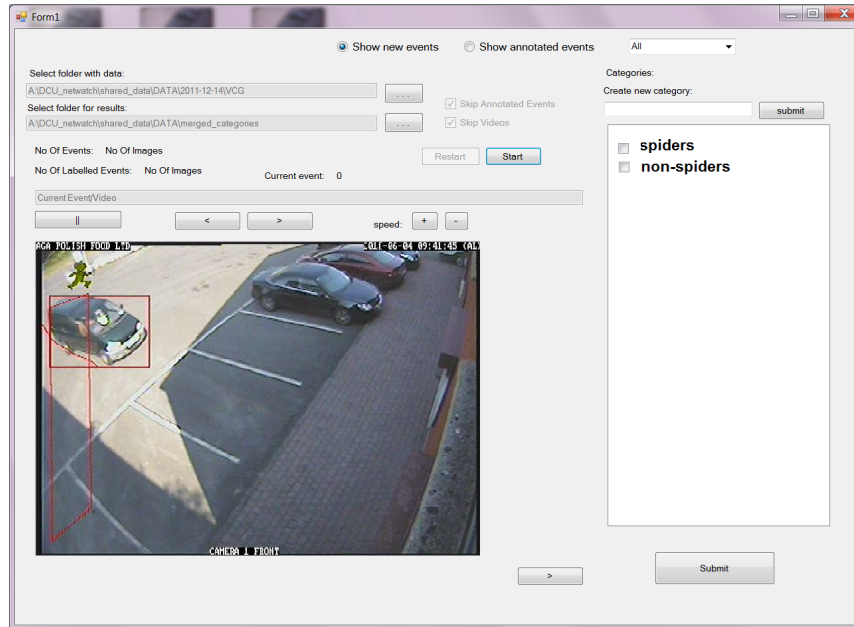


Figure 4.6: Screenshot of the annotation tool developed for creating the ground truth.

A supervised machine learning framework is used to learn a function that maps images into two classes - *spiders* and *non-spiders*. Supervised learning is a task of inferring a function from labeled training data (Mohri et al. 2012). Annotations provide evidence for the class label and the class label tends to globally describe each image.

For evaluation of the proposed algorithm, the dataset was selected from a large number of events created by the existing analytics software after manual annotation. Figure 4.6 shows a screen shot of the annotation tool developed specifically for this purpose. Annotation of images into *spider* and *non-spider* categories was carried out by an experienced surveillance technician with the aid of an industry intervention specialist.

Annotations belonged to two categories *spiders* and *non-spiders* where the spider class comprised of both spiders and spider webs. In some situations this class also included insects close to the surveillance camera lens. Non-spiders consist of true event contributors like people, animals and vehicles crossing the field of view.

4.3 Artefact reduction

As mentioned in Section 4.1, the dataset is temporally sparse and spatially compressed. The sample dataset also includes red boxes and trails² showing the approximate object location and trails of object motion introduced by third party software. In addition to these artefacts, a green man symbol is seen on the top left of event images. The green man symbol is a pictorial depiction of an intruder that always appears in the same location in the images, and thus was masked out and safely ignored.

Of course, classification could be simplified by processing the raw images, but in fact these are not available to Netwatch Security Systems. The specific requirement was to perform classification even in the presence of artefacts. For this reason, an artefact preprocessing step for bounding boxes and (trajectory) lines was introduced.

Figure 4.7 describes an artefact removal step that comprises of : *automatic artefact detection* and *artefact reduction*. An automatic artefact detection step was developed using the saturation channel information while we use an existing algorithm for artefact reduction using inpainting based on the Navier Stokes equation (Bertalmio, Bertozzi & Sapiro 2001).

²Third party video analytics in most cases introduces red bounding boxes and trails to indicate the location of intruder. However, some images were also found that had saturated blue and green artefacts also as shown in figure 4.8

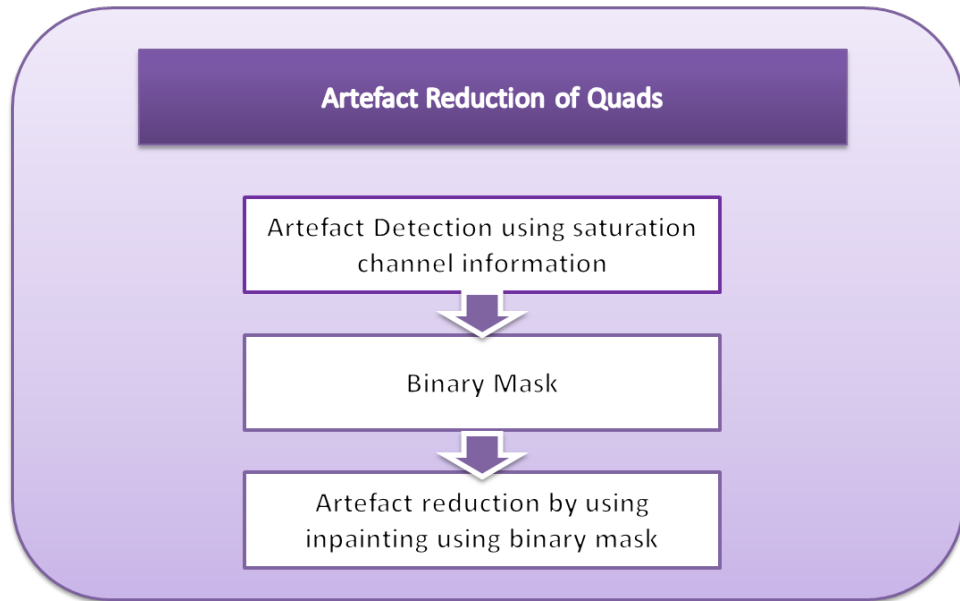


Figure 4.7: Artefact reduction in *Quads*.

1. *Automatic artefact detection*: In Figures 4.2, 4.3, 4.4 and 4.5, it can be noted that the artefacts are heavily saturated compared to the rest of image. In imaging, color saturation is used to describe the intensity of color in the image. An image is said to be saturated when it has overly bright colors. Visual inspection of the quads revealed that the artefact region is heavily saturated compared to the rest of the image. This was the motivation behind the usage of saturation channel cues for artefact detection. The input to the artefact detection phase is an image with artefacts, the output of the artefact detection phase is a binary mask. The white pixels in the binary mask indicate the artefact pixels.

First, the RGB image was converted into HSV space. Minimum windowing image processing on a 3×3 neighbourhood with a threshold of 0.1 was applied on the saturation channel, where the neighbourhood size and threshold were empirically chosen. For a 3×3 neighbourhood, if s is the saturation channel, then the saturation value at j, i^{th} pixel is obtained by performing minimum windowing as given by the condition below:

$$s(j, i) - \min([s(j - 1, i - 1), s(j - 1, i), s(j - 1, i + 1), s(j, i - 1), s(j, i + 1), s(j + 1, i - 1), s(j + 1, i), s(j + 1, i + 1)]) \geq 0.1$$

if this condition is true then the pixel value in the output binary mask is set to 1. The binary mask is morphologically dilated (Dougherty & Lotufo 2003) to make sure colour bleeding from artefacts due to heavy JPEG image compression is also considered for inpainting. In Figure 4.8, sub-figures (b, e, and h) show that the output of the automatic artefact detection phase, a binary mask which can then be used for image inpainting.

2. *Inpainting to reduce artefacts:*

| | |
|-----------------|-----------------------|
| Navier-Stokes | Image Inpainting |
| Stream function | Image intensity |
| Fluid Velocity | Isophote direction |
| Vorticity | Smoothness |
| Fluid viscosity | Anisotropic Diffusion |

Table 4.1: Application of the Navier-Stokes equation from fluid dynamics to image inpainting. Higher order partial differential equations are used for smooth interpolation along the artefact pixels. (source: (Bertalmio et al. 2001)).

Image inpainting involves filling in part of an image or video using information from the surrounding area. Applications include the restoration of damaged photographs and movies and the removal of selected objects (Bertalmio et al. 2000). Inpainting is traditionally used for filling in small

image gaps. Inpainting functions well for linear structures which can be thought of as one dimensional patterns, such as lines and object contours (Criminisi et al. 2003). After the user selects the regions to be restored either in paintings or photographs, the algorithm automatically inserts pixel data into the inpainting region. The fill-in is done in such a way that isophote lines (level lines) arriving at the regions boundaries are continued inside the inpainting region. The technique introduced here does not require the user to specify from which parts of the image the novel information is taken. Figure 4.1 demonstrates an example of Navier Stokes inpainting restoration of the photograph.

Inpainting was used for artefact removal using the binary mask automatically generated from the *artefact detection* phase. The inpainting algorithm introduces the importance of propagating both the gradient direction (geometry) and gray-values (photometry) of the image in a band surrounding the hole to be filled-in. The algorithm is designed to continue isophotes/level-lines while matching the gradient vectors at the boundary of inpainting region. The method is directly based on the Navier-Stokes equations³ for fluid dynamics, which has the advantage of proven theoretical and numerical results (Bertalmio et al. 2001). Sub-figures 4.8 (c, f, and i) show the image after artefact removal.

4.4 Evaluation dataset

As mentioned in Section 4.1, just three frames are used to determine if the event is triggered by a spider or not as this information is adequate for trained personnel to quickly and accurately tell whether the activity is potentially important or

³Table 4.1 describes the analogy of Navier's stokes equation to image inpainting.

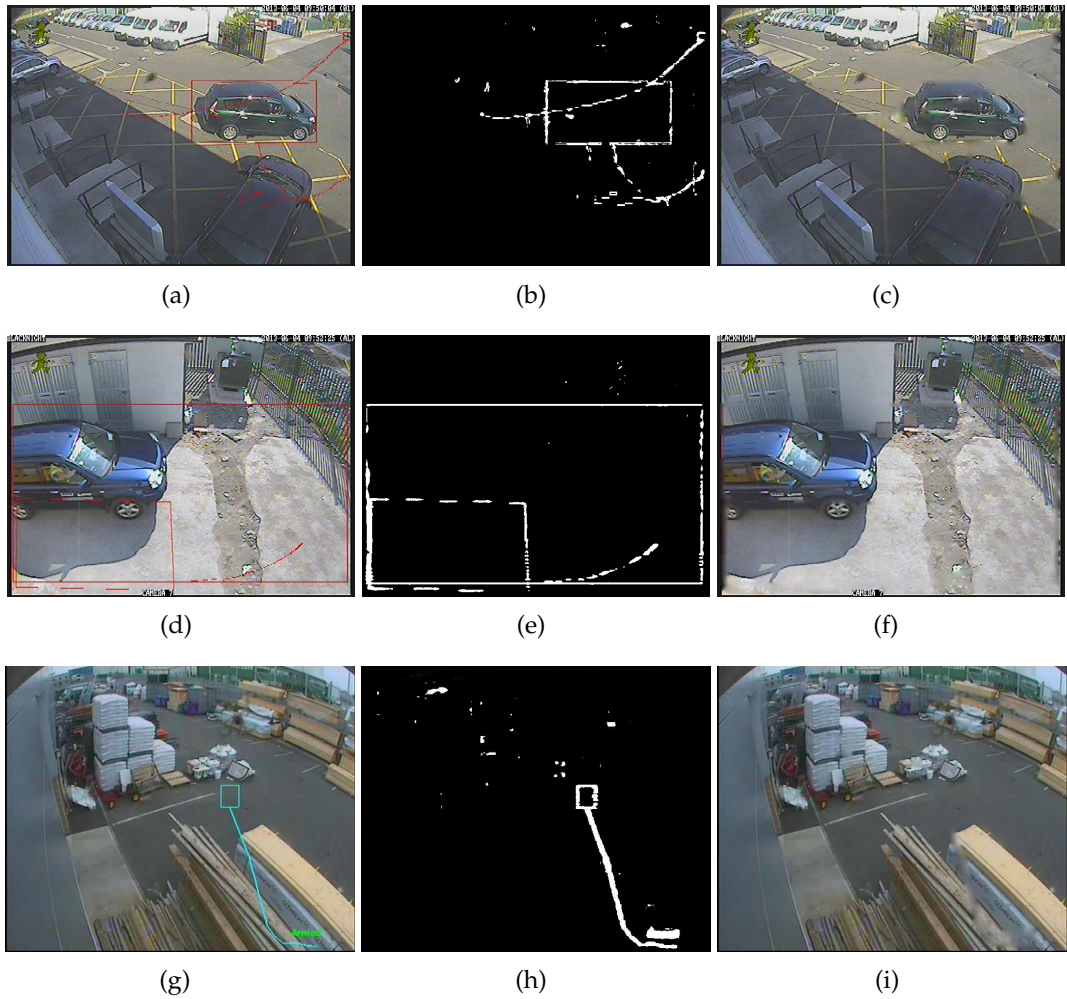


Figure 4.8: An illustration of artefact removal on the images acquired from different camera sites. The subfigures (a, d and g) show images with artefacts; subfigures (b, e and h) show the corresponding mask generated by using minimum windowing on the saturation channel, and subfigures (c, f and i) show images after artefact reduction

whether it can be safely disregarded. The data format was also chosen for use with the proposed false positive reduction technique as it it closely reflects the reality of the kind of surveillance data that is available in most real surveillance industry deployments – temporally sparse and spatially compressed.

2,273 images from spider related events were found via manual annotation. An equal number of non-spider events were randomly chosen producing a dataset

containing 4,546 images in total. The captured images have a resolution of 704×576 and were obtained from data captured in both indoor and outdoor environments with broad geographical distribution.

To train classifiers and assess their performance, the dataset is partitioned into two sets: 70% of the data (3,182 images) is used for training and the remaining 30% (1,364 images) is used for testing the out-of-sample performance of the classifiers. Each set contains an equal number of positive and negative examples so that the expected error rate of a random classifier is 50%. The dataset comprising of 4,546 images are split into the ratio of 70%-30% (for training and testing respectively) in a random fashion 10 times to obtain 10 distinct variations of the data – 3,182 images for training and 1,364 for testing.⁴ The 10 sets of classification results are then averaged to produce a single estimate.

4.5 Conclusion

In this chapter the dataset provided by Netwatch Security Systems is discussed. Manual annotation of images was carried out with the help of a human operator experienced in the surveillance industry. An annotation tool was specifically developed for this purpose. This chapter also detailed on artefact (image overlay) reduction procedure as a specific requirement by Netwatch Security Systems to perform classification even in the presence of artefacts in the dataset introduced by third party software. Artefact reduction comprised of automatic artefact detection by using saturation channel cues and artifact reduction using image inpainting using Navier Stokes equations (Bertalmio et al. 2001).

⁴A LIBSVM tool - subset.py (Chang & Lin 2011) was used for this purpose to choose these random subsets of data in the ratio 70% of training and 30% for testing given dataset.

Chapter 5

Results

5.1 Introduction

This chapter elaborates on the results obtained using the visual features and their combination in an SVM classification framework. In this chapter an investigation is carried out to determine if combining descriptors yields an improved result for spider classification. Also, results related to the computational cost of feature combination for a "real-world" or near real-world application are presented. In the face of possible performance constraints it is desirable to know which descriptors contribute the most to improved computation cost. Therefore, the goal is to arrive at the feature combination that provides highest classification accuracy at lowest computation cost.

The chapter is divided into sections to address the following areas : (1) experimental setup; (2) specification of parameters used for feature extraction and image classification; (3) parameters important in terms of the choice of the proposed feature extraction method to measure using a combination of classification accuracy, computation time and ROCs; (4) a two dimensional chart which intuitively compares classification accuracy to total execution time taken. Finally some representative correct classifications (i.e., true positives and true negatives)

and mis-classifications (i.e., false positives and false negatives) of the proposed algorithm are presented.

5.2 Experimental set-up

All processing was performed on a 64-bit laptop PC running on Windows 7 platform with a 2.2 GHz Intel i7 processor and 8GB of RAM. All of the feature extraction algorithms were implemented using MATLAB except for SIFT/BoVW and RootSIFT/BoVW where the VLFeat C implementation of SIFT with the corresponding MATLAB wrapper were used (Vedaldi & Fulkerson 2008). Programming was mainly done in MATLAB. This included the evaluation of parameters in terms of classification accuracy, computation time, and the ROCs.

For field trials, the proposed method was then ported into the Python programming language. This is mainly because Python is free to use even for commercial products, portable, and fast to prototype¹. Prototyping in Python is made easy as it consists of an extensive standard library operating at lower computation load than MATLAB.

5.3 Specific parameters used for feature extraction

This section gives specific threshold and parameters used in the feature extraction phase. In all cases, parameters were selected based on experiments to obtain the best performing parameters.

- For deciding edge/non-edge blocks a *threshold* of 0.002 and a fitting parameter $\beta = 3.6$ was used in the case of CPBD.
- For fast implementation of Haralick features, Stefan Winzeck's implementation available on MATLAB central file exchange was used. The Haralick

¹because of its cross platform nature, it can also work for users that run Linux or mac OS X.

features like other features were normalised before classification. The Haralick texture features did not have a specific threshold parameter and hence parameter tweaking was not required.

- For LBP variance, LBP codes are computed on sample points on a circle of radius specified by a user – a radius of 1 was used. The LBP variance used was on an (8,1) neighborhood and a uniform rotation invariant LBP scheme. Figure 5.1 shows uniform patterns in LBP variance (Zhenhua Guo & Zhang 2010). In case of LBP variance, (4,1) neighbourhood and (12, 1.5) neighbourhood did not achieve higher classification accuracies compared to the uniform (8,1) neighbourhood scheme.

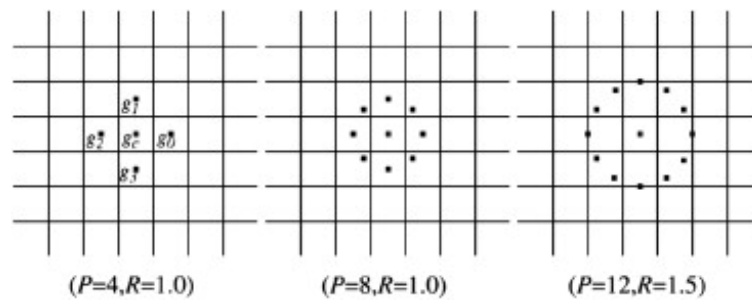


Figure 5.1: LBP Variance: Uniform patterns for $P = 8$.

- For the SIFT/BoVW implementation, an average of 1,567 SIFT points were computed per image. 100 clusters (100 visual words) which were chosen empirically were used in the codebook and k -means clustering was used to fit this to the training set. The VLFeat C implementation of integer k -means² with a MATLAB wrapper was used for clustering. In the case of SIFT/BoVW and RootSIFT/BoVW, 100 clusters were adequate to achieve exceptionally high classification accuracies of 98.9% and 99.2%. The training and classification time increases with increase in the number of clusters.

²Integer K-means (IKM) is an implementation of K-means clustering (or Vector Quantisation, VQ) for integer data.

This justifies the choice of using 100 clusters for image classification both for SIFT/BoVW and RootSIFT/BoVW.

- In the case of gray-scale histograms, 100 bins were used. The number of bins was varied (50 to 250 in steps of 10) and the best classifier performance was achieved when 100 bins were used
- RootSIFT is derived from SIFT by taking the L1-norm of SIFT feature vectors. This means, even in the case of RootSIFT/BoVW implementation an average of 1,567 SIFT points were computed per image. 100 clusters (visual words) were used in the codebook, which was fit to the training set using k -means.

All the code implementations are in MATLAB. However, SIFT/ RootSIFT implementations used a C implementation with MATLAB wrapper. This is because SIFT/BoVW implementations in MATLAB is significantly slower than the C language counterpart. The choice of using MATLAB was considered as MATLAB is found to be a powerful tool for prototyping and algorithm simulation. The main advantage of considering C language in future for the considered application is that the C programming language is a compiled low-level language known for its execution speed and efficiency in embedded systems (Fangohr 2004), (Huang et al. 2004).

5.4 Classifier setup

| Image descriptor | C | γ |
|-----------------------------------|--------|----------|
| CPBD | 32,768 | 8 |
| Haralick | 32,768 | 0.00048 |
| LBP Variance | 512 | 8 |
| Fusion of Haralick and CPBD | 512 | 8 |
| Fusion of LBP Variance and CPBD | 2,048 | 8 |
| SIFT with Bag of Visual Words | 2 | 2 |
| RootSIFT with Bag of Visual Words | 8 | 0.5 |
| Blur histogram | 32,768 | 8 |
| Intensity Histogram | 0.5 | 0.00003 |

Table 5.1: Combination of (C, γ) obtained by grid search for the investigated visual feature vectors for image classification.

A binary visual classifier is trained for two classes, *spider* (positive) and *non-spider* (negative), using the previously described features (see Section 3.5). Support vector machine classifiers and a variation of Platt’s method to produce probability outputs was used. In the experiments, the soft margin SVM implementation provided by LIBSVM with a Radial Basis Function (RBF) kernel was used. To find the optimal parameters for C and γ , a grid search on C and γ using ten fold cross-validation was performed. Exponentially growing sequences of C and γ was found to be the best method to identify good parameters (Chang & Lin 2011). Grid search was performed with C varying from from 2^{-5} to 2^{15} in steps of 2^2 , similarly γ was varied from 2^3 to 2^{-15} in steps of 2^{-2} . Table 5.1 lists the (C, γ) pairs that

achieve the highest cross-validation accuracy (for the purpose of experimental repeatability, the values of (C, γ) used in image classification are recorded).

5.5 Classification accuracy

| Image descriptor | Descriptor dimension | Classification accuracy |
|---------------------------------|----------------------|-------------------------|
| CPBD | 1 | 65.8% |
| Haralick | 13 | 91.6% |
| Fusion of Haralick and CPBD | 14 | 98.82% |
| LBP Variance | 10 | 98.5% |
| Fusion of LBP Variance and CPBD | 11 | 98.4% |
| SIFT/BoVW* | 100 | 98.9% |
| RootSIFT/BoVW* | 100 | 99.28% |
| Blur histogram | 64 | 82.5% |
| Intensity Histogram | 100 | 53% |

Table 5.2: Comparison of the classification accuracy using the image descriptors investigated. * indicate features implemented in C programming language with MATLAB wrapper.

Table 5.2 shows the classification accuracies (percentage of correct classifications) on the test data for each of the different types of features that were tested. The best performing methods are the fusion of Haralick and CPBD, SIFT/BoVW, and RootSIFT with BoVW which achieve comparable accuracies of 98.82% , 98.9%, and 99.2% respectively. The Haralick/CPBD descriptor has lower dimension when compared to SIFT and RootSIFT, but slightly higher than LBP variance.

It is observed that the performance of intensity histogram is comparable to a random Gaussian and hence it is not suitable for our application. CPBD and Blur histograms offer intermediate classification accuracy of 65.8% and 82.5% and hence were not considered further for spider classification.

It can be noted that classification accuracy increased by 7.22% when the Haralick descriptor is fused with the CPBD, which is just a scalar. However, Fusion of LBP variance with CPBD resulted in 0.1% decrease in classification accuracy when compared to the LBP variance descriptor alone. This indicates that Haralick texture features and CPBD contain complementary information. This is taken as a justification that combining descriptors can yield a much improved result if they contain complementary information.

5.6 Computation time

| Method | feature extraction (ms) | classification (ms) | total time (ms) |
|-----------------------------------|----------------------------|------------------------|--------------------|
| CPBD | 2,106 | 0.110 | 2,106.110 |
| Haralick | 31.2 | 0.250 | 31.450 |
| Fusion of Haralick and CPBD | 2,137 | 0.158 | 2,137.158 |
| LBP Variance | 2,464 | 0.368 | 2,464.368 |
| Fusion of LBP Variance and CPBD | 4,570.8 | 0.204 | 4,571.004 |
| SIFT with Bag of Visual Words | 5,928 | 0.622 | 5,928.62 |
| RootSIFT with Bag of Visual Words | 6,864 | 0.622 | 6,864.62 |
| Blur histogram | 2,402 | 0.71 | 2,402.71 |
| Intensity Histogram | 46.8 | 0.69 | 48.69 |

Table 5.3: Computation time for feature extraction and classification for each method (in milliseconds).

From Table 5.3, it can be noted that the Haralick texture and intensity histogram image features take the least computation time at 31.4 milliseconds and 48.6 milliseconds respectively³ However the classification accuracy of these two methods is 91.2% and 53% which quite low relative to other approaches. Fusion of Harlick and CPBD results in 98.82% classification accuracy with computation time of 2.1 seconds whereas fusion of LBP variance with CPBD results in similar classification accuracy at almost twice the computation time, 4.5 seconds. From Section 3.3.2, it can be recalled that Netwatch Security Systems suggested that surveillance personnel take 45 seconds to slightly over a minute on average for manual event

³Although fusion of Haralick texture features with CPBD used a slower MATLAB implementation, it outperformed SIFT/BoVW and RootSIFT/BoVW features which were implemented in C language.

classification. The variation is attributable to day–night situations, complexity of events, and other factors. This means that the computation time taken by the proposed method (Fusion of Harlick and CPBD) is reasonable for real-time application despite the fact that MATLAB implementation of the proposed method was not optimised which could lead to improved computational performance.

To provide a visualisation of the comparison of the different methods investigated, a two dimensional chart which compares classification accuracy to total execution time taken is presented in the Figure 5.2.

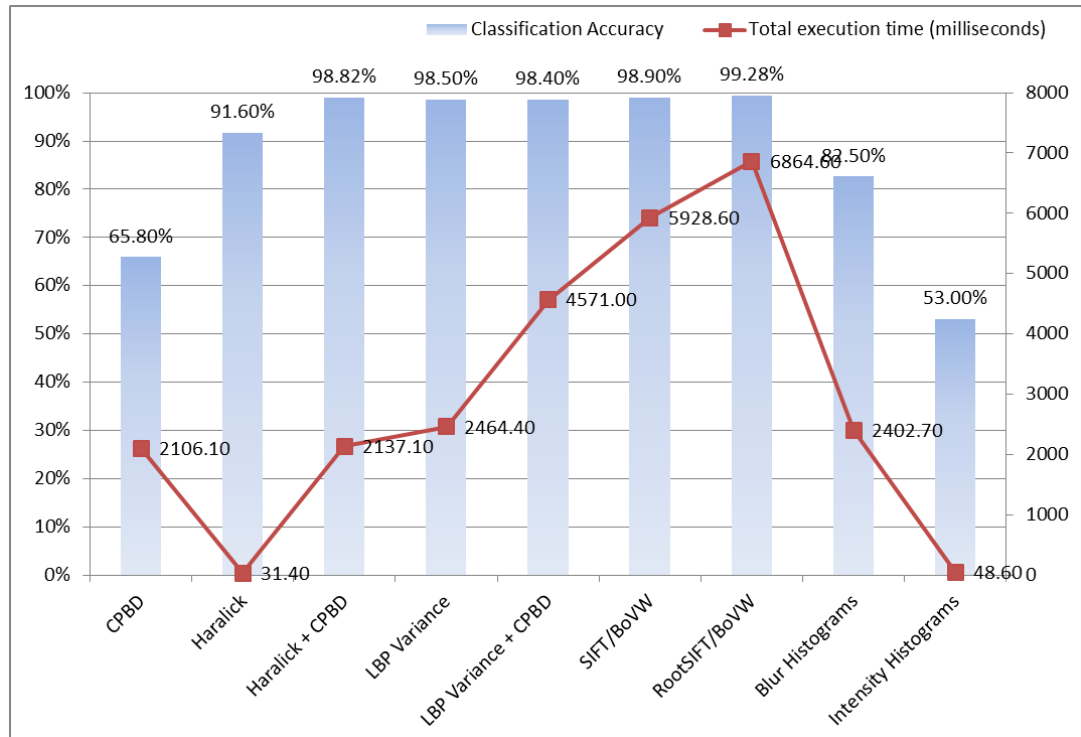


Figure 5.2: A comparison of classification accuracy vs. total execution time for different visual descriptors. Fusion of Harlick and CPBD offers 98.82% classification accuracy with computation time of only 2.1 seconds.

The spider classifier was trained using 70% of the dataset (3, 182 images) consisting of 4, 546 images. The training time that was recorded is shown in Table 5.4.

Considering the computation time⁴ and training times shown in Table 5.3 and Table 5.4, it can be concluded that the fusion of CPBD with Haralick texture features results in lower training and test times when compared with SIFT/BoVW and RootSIFT/BoVW, which is state-of-the-art for image classification. The proposed descriptor also outperforms the LBP variance and early fusion of LBP variance and CPBD methods in terms of computation time.

| Method | Training time for 3182 images |
|-----------------------------------|-------------------------------|
| CPBD | 59.407 seconds |
| Haralick | 3.182 seconds |
| LBP Variance | 0.999 seconds |
| Fusion of Haralick and CPBD | 4.995 seconds |
| Fusion of LBP Variance and CPBD | 6.968 seconds |
| SIFT with Bag of Visual Words | 6 hours |
| RootSIFT with Bag of Visual Words | ≈ 6 hours |
| Blur histogram | 44 seconds |
| Intensity Histogram | 10 seconds |

Table 5.4: Training time taken by each method. The SIFT/BoVW figure includes time taken for feature extraction and k -means clustering to generate the codebook.

The dataset comprising of 4, 546 images was partitioned into two sets: 70% of the data (3, 182 images) is used for training and the remaining 30% (1, 364 images) is used for testing the out-of-sample performance of the classifiers. Although training time is not a very important factor compared to execution time, it is still worth noting that machine learning performance is typically found to improve

⁴time taken for feature extraction + time taken for classification

with a model trained with a larger dataset. It is for this reason that training time was considered.

5.7 ROC

Receiver Operator Characteristic (ROC) curves are commonly used to present results for binary decision problems in machine learning (Davis & Goadrich 2006). ROC curves show how the number of correctly classified positive examples (spiders) varies with the number of incorrectly classified negative examples (humans, vehicles, and animals). The objective is to minimise false positives (non-spider events marked as spiders events) while maximising the true positives (spiders classified as spiders). The ROC curve shows the true positive rate plotted against the false positive rate while we vary a probability threshold, which assists selecting an operating point that appropriately balances the tradeoff between true and false positives for a particular application. Picking a value with low false positive rate may reduce absolute accuracy (proportion of correct classifications), but will reduce the probability of *non-spiders* classified as *spiders*. ROCs for all 9 features are described in Chapter 3. Figure 5.3 shows the comparison of ROCs for all tested approaches.

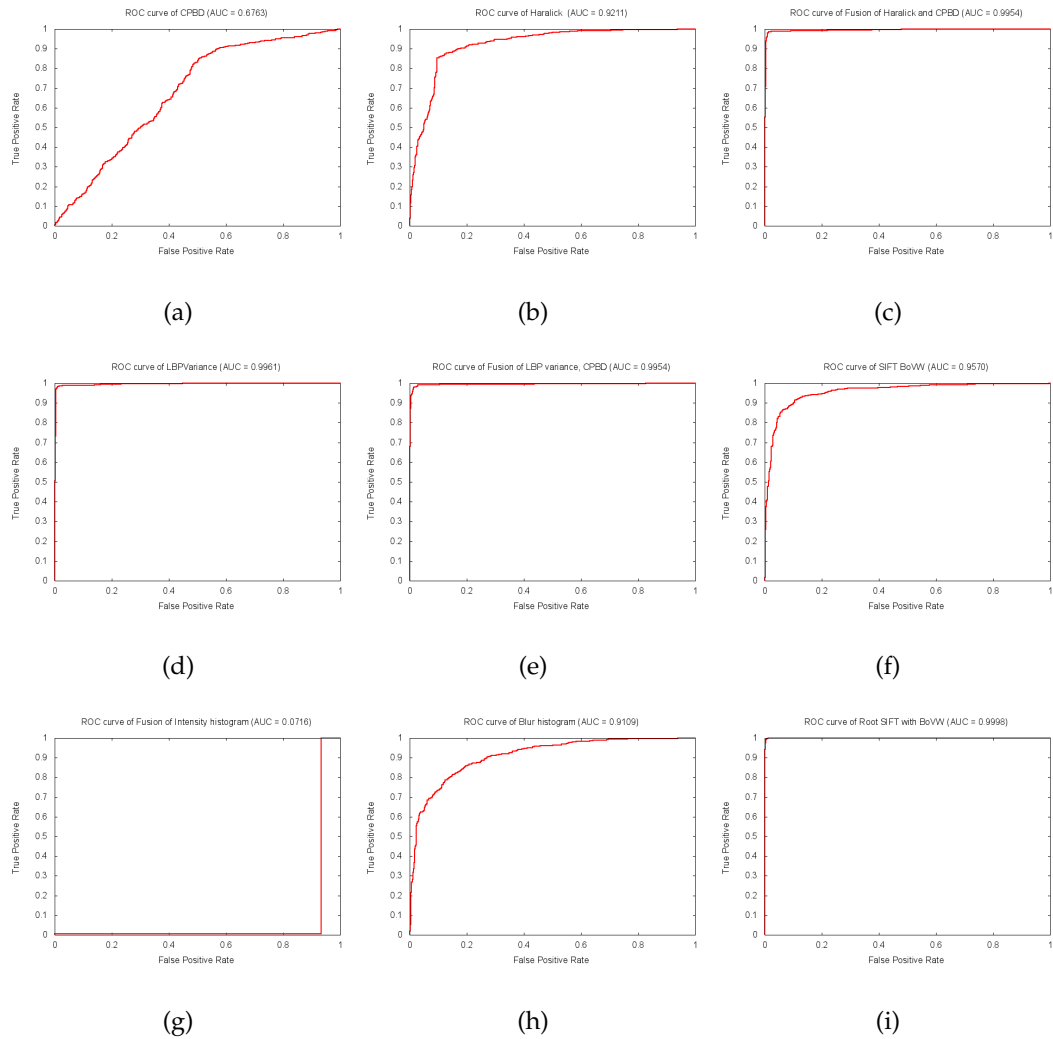
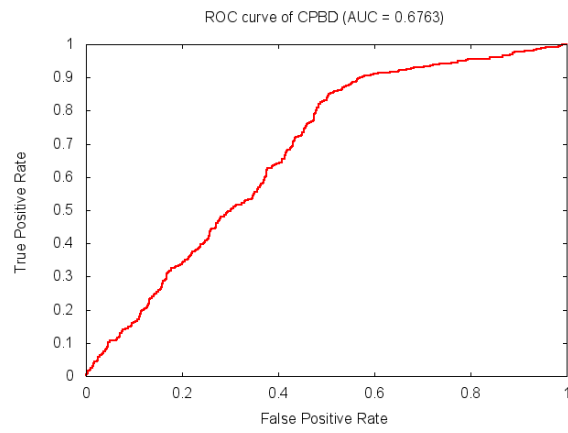


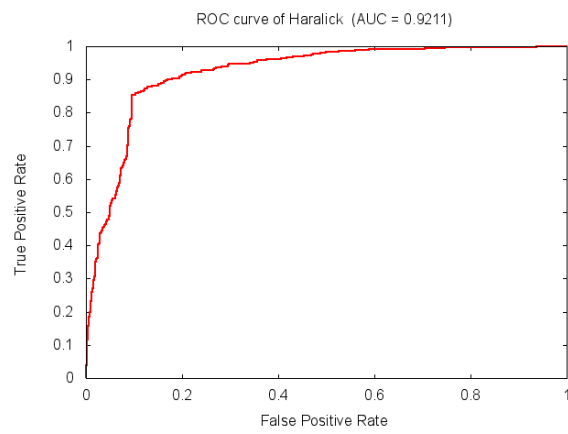
Figure 5.3: A comparison of ROC curves for all investigated visual features (a) CPBD feature (b) Haralick (c) Early fusion of the Haralick and the CPBD (d) LBP variance (e) Early fusion of the LBP variance and the CPBD (f) SIFT/ BoVW (g) Intensity histogram (h) Blur Histogram and (i) RootSIFT/ BoVW.

Figures 5.4 and 5.5 show the ROC curves for each of the tested methods and illustrate the merits of descriptor fusion. Note that Haralick/CPBD gives the lowest false positive rate in comparison with the state of the art SIFT/BoVW and RootSIFT/BoVW approaches but with lower computational cost. Improved classification performance is obtained for fusion of Haralick with CPBD features,

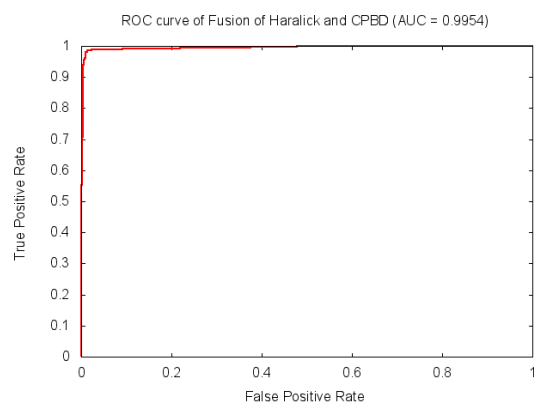
indicating that they are complementary. This is not the case for LBP variance with CPBD.



(a)

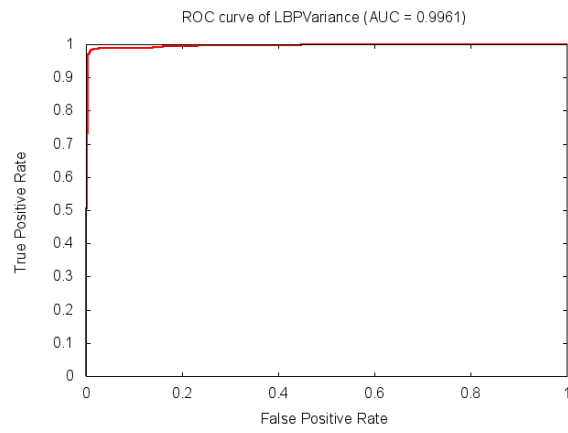


(b)

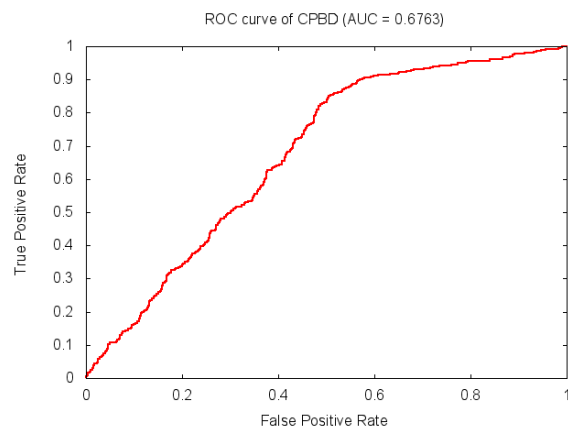


(c)

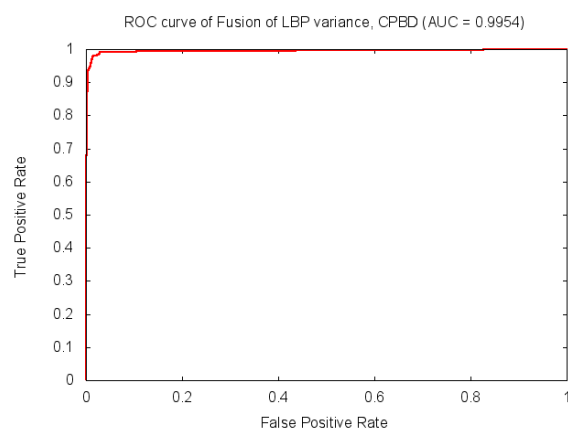
Figure 5.4: A comparison of ROC curves for visual features. ROC curve for (a) CPBD , (b) Haralick texture features, and (c) Fusion of Haralick and CPBD.



(a)



(b)



(c)

Figure 5.5: A comparison of ROC curves for visual features. ROC curve for (a) LBP Variance , (b) CPBD, and (c) Fusion of LBP variance with CPBD

5.8 Sample results



(a)

(b)

(c)



(d)



(e)



(f)

Figure 5.6: Subfigures (a), (b), and (c) contain images that were categorised as true positives (spiders classified as spiders) by the proposed algorithm; images (d), (e), and (f) show true negatives (non-spiders classified as non-spiders).



(a)

(b)

Figure 5.7: False positives (non-spiders classified as spiders) by the proposed algorithm.



(a)

(b)

Figure 5.8: False negatives (spiders classified as non-spiders) by the proposed algorithm.

Figures 5.6, 5.7, and 5.8 show some examples of correctly classified images and misclassified images by the proposed algorithm. Figure 5.7 shows false positives (non-spiders classified as spiders); it should be observed that reflections and lighting produce an effect very similar in appearance to a spider web, which

explains the misclassification. Figure 5.8 shows false negatives (spiders classified as non-spiders); it appears that the extremely low contrast in these images may be responsible for the classifier failing to recognise the spiders correctly.

5.9 Field trial results

A preliminary field trial in collaboration with Netwatch Security Systems was performed on a test site with 12 camera views without retraining the algorithm for that site. Events triggered from the site were passed through the spider classification pipeline. On inspection of the classified dataset it was observed that 9% of the quads were filtered into *spider* class. Figures 5.9 and 5.10 show some correct classification results during field trial. Figure 5.11 shows an event where a *non-spider* is classified into *spider* class, as the motion of the person was occluded by a spider web. Thus, preliminary studies show promising results in terms of reduction of overall false alarm rates whilst at the same time reducing an intervention specialist's workload.

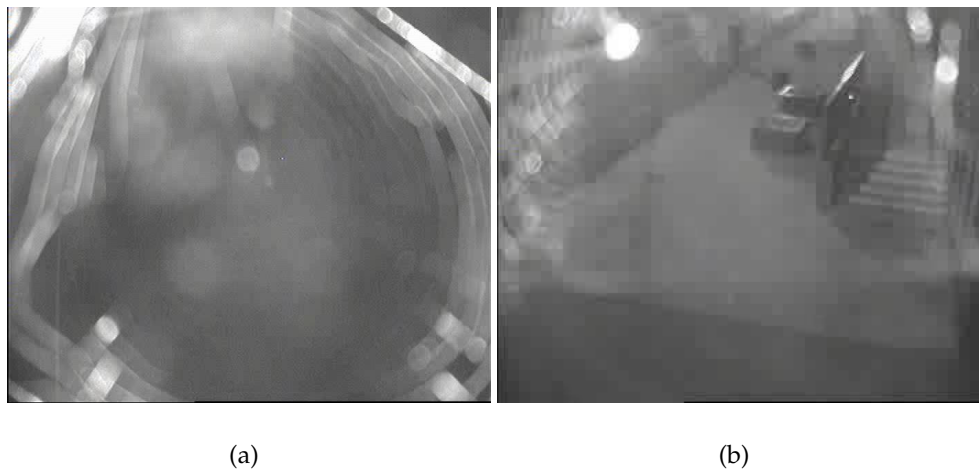


Figure 5.9: Field trial results: Spiders classified as *spiders* by the proposed algorithm.



(a)

(b)

Figure 5.10: Field trial results : non-spiders classified as *non-spiders* by the proposed algorithm.



Figure 5.11: Field Trial result: A person appearing within a spiderweb is classified into the *spiders* category by the proposed algorithm. However, the confidence score generated by the proposed algorithm could potentially be used to trigger a lens cleaning event.

5.10 Discussion

The proposed descriptor, which fuses easily computable Haralick texture features and a blur metric based on cumulative probability of blur detection, produced a classification accuracy of 98.82%, which is comparable with the more computationally expensive SIFT/BoVW and RootSIFT/BoVW descriptors whose classification accuracies were 98.9% and 99.2%.

There is a clear merit in fusion of complementary features as this results in better classification rates as seen in Table 5.2. Note that the individual Haralick and CPBD classification rates were 91.6% and 61.8% while fusion of those features increased the classification rates to 98.82%. Although fusion of LBP variance with CPBD gave similar classification results, the computational time was found to be almost twice as high compared to the proposed fusion method. Classification accuracies of Blur histograms, intensity histograms and CPBD were significantly lower and hence those features did not meet the requirement of reaching a very high classification accuracy.

The ROC curves show that the proposed method can achieve a classification accuracy of 98.82% with only 0.5% false positive misclassification (*non-spiders* classified as *spiders*). The probability threshold could be reduced to trade-off for a much higher classification accuracy if more false detections are permitted.

Performance of RootSIFT with BoVW is significant reaching accuracy of 99.2%. Most of the processing effort for these algorithms comes from point detection phase rather than the feature extraction phase. The bottleneck is in the keypoint detection stage using Difference of Gaussians and not in histogram creation phase. It is for this reason the descriptors such as FAST were proposed (Wagner et al. 2008). In contrast to the classic SIFT approach; Wagner et al. (2008) use the FAST corner detector for feature detection in mobile phones. Fast Retina Keypoint (FREAK) aims to make descriptors faster to compute on embedded devices by op-

timising the keypoint description stage (Alahi et al. 2012). Uniform sampling may increase the computation time but this may also reduce classification accuracy.

Based on observations of the images the proposed algorithm could also be applied for other insects close to the lens and events caused by rain and snow in addition to spider false alarms. Although the experimental dataset is spatially compressed and temporally sparse, the algorithm proposed could be applied generically to surveillance data from any security industry and the proposed method could potentially improve classification results particularly for better quality data without artifacts that does not require artifact reduction pre-processing.

The results obtained from a preliminary field trial show 9% of the data gathered in the trial was filtered out as spiders. However, it is worth noting that the field trial was conducted only on a single site with 9 camera views. The field trial data was captured only for 2 days unlike the data from the training set, that covered different seasons, geographic locations and camera views. Training data provided by Netwatch Security Systems, i.e., the dataset used for training a model was acquired from 12 monitored sites having a total of 275 cameras. The percentages of false alarms triggered by spiders varied from 20% to 50% on the original dataset provided. This is the likely explanation for the discrepancy in percentages from the training data and field test data. Further field trial results need to be obtained to test the algorithm rigorously.

Although spider and spider web “events” seem to be very different in visual terms, the following reasons justify considering them in the same class : (a) both spiders and spider webs appear blurry as both are seen close to the lens surface; (b) also in most cases, spiders and spiderwebs occur together in a given field of view. In this case, the coarse regular pattern found in spiderwebs/cobwebs and blur descriptive of spiders are fused as they carry complementary information for both spiders and spiderwebs; (c) experimental results such as higher classification accuracies and close to ideal ROCs possibly suggest appropriate choice of features

and the categories. In other words, combining both spiders and spider webs into single class seems to make sense as both tend to coexist.

Finally, it is important to comment on the robustness of the proposed algorithm in the event of criminals learning about the computer vision technology to detect spiders. It is extremely difficult for humans to simulate the presence of spiders and at the same time mask their own presence. Hence, it would be a very unlikely situation where a person is able to circumvent the system by generating some pattern that would resemble a spider or a web.

5.11 Conclusion

This chapter provided details of the experimental setup, specific parameters used for feature extraction, and the classifier setup. The focus was on the results obtained from the proposed visual descriptor. A performance comparison of the proposed descriptor with the state-of-the-art descriptors was presented in terms of classification accuracy, computation time, training time, and receiver operating characteristics. Based on these results, a visual descriptor that fuses easily computable Haralick texture features and a blur metric based on cumulative probability of blur detection was selected as the best choice. This approach produced a classification accuracy of 98.82%, which is comparable with the more computationally expensive SIFT/BoVW and Root SIFT with Bag of Visual Words descriptors whose classification accuracies were 98.9% and 99.2% respectively. This highlights the benefits of fusion of complementary features. Although fusion of LBP variance with CPBD gave similar classification results, the computational time was found to be almost twice as high compared to the proposed fusion method. The ROC curves show that the proposed method can achieve a classification accuracy of 98.82% without any false positive misclassifications (non-spiders

classified as spiders). The probability threshold could be reduced to trade-off for higher classification accuracy if more false detections are permitted.

Based on observations of the images we believe that the proposed algorithm could also be applied for other insects close to the lens and events caused by rain and snow in addition to spider false alarms. A field trial on a test site was performed to test the efficiency of the proposed method. The results show that 9% of the data generated at this site was filtered out as spiders.

Chapter 6

Conclusions and future work

6.1 Conclusions

This thesis focused on the use of computer vision for false alarm reduction in surveillance camera networks and specifically addressed the false alarms triggered by spiders which can contribute to 20-50% of false alarms. A novel solution to this common problem facing the surveillance industry is proposed. The following is an overview of the research that has been described in this thesis.

Chapter 1 discusses false alarms in a video surveillance scenario. This led to the understanding of the need for developing a false alarm reduction pipeline. This chapter also introduced statistics of false alarms triggered by spiders from the data gathered by Netwatch Security Systems. The research was based on motivations such as the significant human effort involved in event handling and in lens cleaning operations, and the economical and environmental impact of existing methods.

In Chapter 2, the literature in the area of false alarms triggered by spiders is reviewed. The solutions that exist are mainly chemical based methods to clean the exterior of surveillance cameras using spider deterrent sprays and hardware based methods that required additional hardware or replacement of entire camera units.

Most solutions available in the literature require significant human effort and are expensive for surveillance industry deployments. A solution using computer vision for detecting spider alarms was chosen because of important factors such as economical viability and reduction in human effort.

In Chapter 3, the spider false alarm reduction problem is formulated as an image classification task. An investigation of various image descriptors was carried out to propose a new descriptor which could classify images into *spider* and *non-spider* classes. Observation of the coarse regular pattern found in the webs motivate the investigation of texture features. The idea of combining complementary descriptors to create a more discriminative descriptor that will work well in a wider variety of situations was explored, given the large intraclass variability in spider image sequences. In addition to the texture information, the extent of image blur as another dominant feature was investigated considering that spiders and spider webs appear blurry. An SVM classification framework and a variation that outputs confidence values was discussed.

Chapter 4 describes the data set provided by Netwatch Security Systems. It also illustrates positive and negative examples used for image classification. Manual annotation was carried out with the help of a human operator with experience in the surveillance industry. An annotation tool was specifically developed for this purpose. This chapter also provides details on artefact reduction process as a specific requirement by Netwatch Security Systems to perform classification even in the presence of artifacts introduced by third party software.

Chapter 5 discusses the results obtained from the proposed visual descriptor. A comparison of performance of the proposed descriptor with the state-of-the-art descriptors in terms of classification accuracy, computation time, training time, and ROC curve is presented.

A visual descriptor, which fuses easily computable Haralick texture features and a blur metric based on cumulative probability of blur detection was proposed.

This produced a classification accuracy of 98.82% and is comparable with the more computationally expensive SIFT/BoVW and Root SIFT with Bag of Visual Words descriptors whose classification accuracies were 98.9% and 99.2% respectively. This underlines the benefits of fusion of complementary features contributing to better classification accuracies. Although fusion of LBP variance with CPBD gave similar classification results, the computational time was found to be almost twice that of the proposed fusion method. The ROC curves show that the proposed method can achieve a classification accuracy of 98.82% with less than 1% false positives (spiders classified as non-spiders). The probability threshold could be reduced to achieve higher classification accuracies if more false detections are permitted.

Based on observations of the images we believe that the proposed algorithm could also be applied to insects close to the lens and events caused by rain and snow in addition to spider false alarms. The chapter concluded by illustrating the classification results from the proposed method. Finally, the developed algorithm underwent a preliminary field trial at one site with 12 cameras. Promising results were obtained by Netwatch Security Systems when the algorithm was deployed in practice.

6.2 Research contributions

The key contributions of this thesis are:

- A novel approach to reduce false alarms triggered by spiders is presented. This is intended to reduce human operator effort, maintenance cost, and operator stress involved in validating false alarms triggered by insects, spiders, and flies close to the lens. The algorithm is also intended to help optimise usage of police resources especially in situations when false alarms

triggered by spiders if not dismissed in time result in police being notified or cameras being switched off.

- At the time of submission of this thesis, there are no documented studies that attempted to reduce false alerts by spiders in surveillance systems using computer vision. A visual descriptor using various computer vision and machine learning techniques is presented.
- The proposed method is evaluated against widely used visual features, and compared against other state-of-the-art methods in terms of accuracy, ROC curve, and computation time. The thesis showed that the proposed method achieves state-of-the-art performance with lower computational cost.
- From field trial results, it can be concluded that the proposed pipeline has commercial potential for development of novel video surveillance software among different OEM's.

6.3 Future work

In the future the spider classification algorithm needs to be converted from a working prototype into a real world application as the current MATLAB/python implementation of the feature extractors need to be ported to C and optimisation will have to be done to improve training and classification time.

The performance of RootSIFT with BoVW is significant reaching 99.2% classification accuracy on the experimental dataset used of course at the cost of computational overhead. The current implementation of RootSIFT/BoVW uses gradient information for keypoint detection. A further investigation needs to be carried out to determine if a uniform sampling of the image for keypoint detection yields lower computation time.

Online learning needs to be incorporated in the future based on a recently uncovered industry requirement, the result of which will be an incrementally and dynamically trained classification framework to recognise the same events when they reoccur at some point in the future.

This algorithm could be tested on different OEMs supplying different surveillance camera types. It would be worthwhile to deploy the current algorithm with 25,000 cameras already deployed by Netwatch Security Systems. It would be interesting to study spiders as a function of weather and then superimpose GPS coordinates of surveillance cameras for better visualisation of spider alarms in an urban setting.

This latter suggestion is a particularly novel idea and could be a useful tool beyond surveillance in the area of environmental monitoring in the study of spider fauna in an urban setting. One such example is the study of spider population. This is of interest to the research community as they prevent population explosion of pests through predation. For example, *The Antwerp Spider Research Project* aims to study spider fauna of Antwerp's inner city area (Keer 2008) and the use of the existing widely deployed CCTV infrastructure to assist in this worthy endeavour is an intriguing possibility.

Bibliography

- Allyn, W. G. (2012), 'The minds eye', <http://www.rochester.edu/pr/Review/V74N4/0402.brainscience.html>.
- Anderson, J. F. (1970), 'Metabolic rates of spiders', *Comparative Biochemistry and Physiology* **33**(1), 51–72.
- Arandjelovic, R. & Zisserman, A. (2012), Three things everyone should know to improve object retrieval, *in* 'Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on', IEEE, pp. 2911–2918.
- Bertalmio, M., Bertozzi, A. & Sapiro, G. (2001), Navier-stokes, fluid dynamics, and image and video inpainting, *in* 'IEEE Computer Vision and Pattern Recognition, 2001.', Vol. 1, pp. I-355–I-362 vol.1.
- Boland, M. V. (1999), 'Materials and methods, murphy lab', http://murphylab.web.cmu.edu/publications/boland/boland_node26.html.
- Bristowe, W. S. & Smith, A. (1971), *The world of spiders*, Collins.
- Brodsky, T. & Lin, Y.-T. (2004), 'Object blocking zones to reduce false alarms in video surveillance systems'. US Patent App. 10/969,720.
- Chang, C.-C. & Lin, C.-J. (2011), 'LIBSVM: A library for support vector machines', *ACM Trans. Intell. Syst. Technol.* **2**(3), 27:1–27:27.

- Chin, K. (1999), 'Support vector machines applied to speech pattern classification', *Mphil. In Computer Speech and Language Processing, Cambridge University Engineering Department* .
- Dalal, N. & Triggs, B. (2005), Histograms of oriented gradients for human detection, *in 'In CVPR'*, pp. 886–893.
- Davis, J. & Goadrich, M. (2006), The relationship between Precision-Recall and ROC curves, *in 'Proceedings of the 23rd international conference on Machine learning'*, ICML '06, ACM, New York, NY, USA, pp. 233–240.
- Dougherty, E. R. & Lotufo, R. A. (2003), Hands-on morphological image processing, SPIE Bellingham.
- Fangohr, H. (2004), A comparison of c, matlab, and python as teaching languages in engineering, *in 'Computational Science-ICCS 2004'*, Springer, pp. 1210–1217.
- Ferzli, R. & Karam, L. J. (2009), 'A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb)', *Image Processing, IEEE Transactions on* **18**(4), 717–728.
- Gehler, P. & Nowozin, S. (2009), On feature combination for multiclass object classification, *in 'Computer Vision, 2009 IEEE 12th International Conference on'*, pp. 221 –228.
- Govindu, V. M. (2013), 'Computer vision e1 216', <http://www.ee.iisc.ernet.in/new/people/faculty/venu/cv/>.
- Haering, N. & Lobo, N. D. (2001), *Visual event detection*, Kluwer Academic Publishers.
- Hart, D. (1996), *The camera assistant: a complete professional handbook*, Focal Press.

- Honeywell (2010), 'Video analytics with external infrared illumination - application note', http://www.honeywellvideo.com/documents/Video_Analytics_External_IR_Illumination_Application_Note.pdf.
- Keer, K. V. (2008), 'The antwerp spider research project (asop)', http://www.arachnology.be/antwerpen/pages/asop_english.html.
- Ko, Y. H. (2008), 'Security camera capable of preventing spiders'. US Patent App. 12/253,585.
- Kotiahio, J., Alatalo, R. V., Mappes, J. & Parri, S. (1996), 'Sexual selection in a wolf spider: male drumming activity, body size, and viability', *Evolution* pp. 1977–1981.
- Lawrinson, J. (2006), *Bye Beautiful*, Penguin UK.
- Lewis, J. (1995), Fast template matching, *in* 'Vision Interface', Vol. 95, pp. 15–19.
- Lizzy, L. (2012), 'The sydney morning herald: City dwelling spiders getting all warm and fuzzy - and bigger', <http://www.smh.com.au/national/city-dwelling-spiders-getting-all-warm-and-fuzzy--and-bigger-20121202-2ap1k.html>.
- Lowe, D. G. (1999), Object recognition from local scale-invariant features, *in* 'The Proceedings of the Seventh IEEE International Conference on Computer Vision', Vol. 2.
- Manning, C. D., Raghavan, P. & Schütze, H. (2008), *Introduction to information retrieval*, Vol. 1, Cambridge University Press Cambridge.
- Miyamoto, E. & Merryman, T. (2005), 'Fast calculation of haralick texture features', *Available from: ece. cmu. edu/~ pueschel/teaching/18 799B CMU spring05/material/eizan tad. pdf*.

- Naphade, M. R. & Smith, J. R. (2004), On the detection of semantic concepts at trecvid, *in* 'Proceedings of the 12th annual ACM international conference on Multimedia', ACM, pp. 660–667.
- Narvekar, N. D. & Karam, L. J. (2011), 'A No-Reference Image Blur Metric Based on the Cumulative Probability of Blur Detection (CPBD)', *IEEE Transactions on Image Processing* **20**(9), 2678–2683.
- Ojala, T., Pietikainen, M. & Harwood, D. (1994), Performance evaluation of texture measures with classification based on kullback discrimination of distributions, *in* 'Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on', Vol. 1, pp. 582–585 vol.1.
- Ojala, T. & Pietikinen, M. (2012), 'Texture classification', <http://www.cse.oulu.fi/CMV/Research>.
- Philpot, W. (2011), 'Digital image processing'.
- Piciarelli, C. & Foresti, G. (2011), 'Surveillance-oriented event detection in video streams', *Intelligent Systems, IEEE* **26**(3), 32–41.
- Platt, J. C. (1999), Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, *in* 'Advances in Large margin Classifiers', MIT Press, pp. 61–74.
- Powell, P. (1993), 'Household pest control'.
- Powers, D. (2011), 'Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation', *Journal of Machine Learning Technologies* **2**(1), 37–63.
- Pratt, W. K. (1978), 'Image feature extraction', *Digital Image Processing: PIKS Scientific Inside, Fourth Edition* pp. 535–577.

- Prince, S. J. (2012), *Computer vision: models, learning, and inference*, Cambridge University Press.
- Roselle, R. E. (1954), 'Nebraska 4-h entomology club manual: Extension circular 16-01-2'.
- Scholander, P., Flagg, W., Walters, V. & Irving, L. (1953), 'Climatic adaptation in arctic and tropical poikilotherms', *Physiological Zoology* **26**(1), 67–92.
- ScikitLearn (2010), 'Support vector machines: Radial basis function parameters, scikit learn', http://scikit-learn.org/0.13/auto_examples/svm/plot_rbf_parameters.html.
- Shdow (2010), 'Shdow security, ultimate digital video surveillance solutions', http://www.shdowsecurity.com/things_you_should_know_before_purchasing_a_video_security_system.php.
- Sivic, J. & Zisserman, A. (2003), Video Google: A text retrieval approach to object matching in videos, in 'IEEE International Conference on Computer Vision', Vol. 2, pp. 1470–1477.
- Swets, J. A. (1996), *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers.*, Lawrence Erlbaum Associates, Inc.
- Vapnik, V. (2000), *The nature of statistical learning theory*, springer.
- Vedaldi, A. & Fulkerson, B. (2008), 'VLFeat: An open and portable library of computer vision algorithms', <http://www.vlfeat.org/>.
- Vedaldi, A. & Zisserman, A. (2012), 'Efficient additive kernels via explicit feature maps', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **34**(3), 480–492.

- Weijer, J. V. D. & Schmid, C. (2006), Coloring local feature extraction, *in* 'European Conference on Computer Vision'.
- WHO (2007), *IARC Monographs On the evaluation of carcinogenic risks to humans*, Vol. 89, World Health Organization.
- Yang, G. & Huang, T. S. (1994), 'Human face detection in a complex background', *Pattern recognition* **27**(1), 53–63.
- Zhang, J., Marszałek, M., Lazebnik, S. & Schmid, C. (2007), 'Local features and kernels for classification of texture and object categories: A comprehensive study', *International journal of computer vision* **73**(2), 213–238.
- Zhenhua Guo, L. Z. & Zhang, D. (2010), 'Rotation invariant texture classification using lbp variance (LBPV) with global matching', *Pattern Recognition* **43**(2010), 706–719.