KU LEUVEN

Smart detection of seepage in river dikes based on thermal infrared images

Ir. Koen Wildemeersch

Thesis submitted for the degree of Master of Science in Artificial Intelligence, option Engineering and Computer Science

Thesis supervisor:
Prof. Dr. Ir. Dirk Vandermeulen and
Prof. Dr. Ir. Tinne Tuytelaars

Assessor:
Prof. Dr. Ir. Jaak Monbaliu and
Dr. Ir. Marco Pedersoli

Academic year 2013 – 2014

Preface

First of all I would like to thank prof. Dirk Vandermeulen and prof. Tinne Tuytelaars for supervising this thesis. Thank you very much for sharing your expertise and knowledge but also for sharing my excitement about infrared images, drones and the possibilities that the future may bring. Thank you for letting me work independently at the same time. Moreover, thank you for accepting a priori that the outcome of this thesis could never be perfect because of technical improvements yet to be made.

I would also like to thank the jury for reading the text. Thank you prof. Dirk Vandermeulen, prof. Tinne Tuytelaars, prof. Jaak Monbaliu and Dr. Marco Pedersoli for your input both during and outside the interim presentation and other contact moments.

This thesis research would not have been possible without the support of both Flanders Hydraulics Research and Antea Group who have allowed me to work on this thesis in such an independent way. Special thanks to ing. Klaas Pieter Visser and ir. Patrik Peeters for the support and brainstorming about the problem at hand. Also, special thanks to my many colleagues at Antea Group who were happy to reason with me and enrich me with their specific expertise. Notably I would like to thank ir. Koen Foncke and lic. Koen Vereycken for proof reading this text from the water related point of view.

I would also like to thank ir. Yves Yde (Alcatel-Lucent) and ir. Lieven Lemiengre (Sigasi) for reading the text from the artificial intelligence related point of view.

In the scope of this work two levee inspections were carried out to map the known leaks. Therefore I want to thank Herbert Vercauteren and Chris Rosiers from Waterwegen en Zeekanaal NV for their assistance during the levee inspection and sharing their knowledge about the levee with me. During the levee inspections I could also use a FLIR thermal infrared camera to make additional images. I would like to thank FLIR NV for that and especially ing. Koen Jacobs who took the patience to explain the camera's functioning to me and go through the images with me afterwards.

During this research I also met some interesting people who are or have been working on related issues. I would like to thank Drs. Sandra Fraikin (Provincie Zuid-Holland)

and ing. Alfred Bruyndonckx (NV De Scheepvaart) for providing me with thermal infrared images of levees, previous research findings and thoughts about practical applications.

Last - and most important to me - in this preface is my girlfriend and life companion who has been very understanding and supportive during the whole extent of these studies. Thank you for sharing my enthusiasm when I managed to draw a polygon around an area that was so obviously visible even without the need of it. Thank you for the high fives whenever the number of true positives increased and I was happy to find a leak. Thank you for bearing with me when I first spent so much time to compute features and then to throw many of them away.. It must not always have been easy.. but that I managed, is because of you.

Gent,
January 3, 2014

Ir. Koen Wildemeersch

Contents

Abstract

In this thesis an algorithm is developed to detect seepage in river levees based on thermal infrared images. From previous research it is known that thermal contrast (a cold levee body and warmer water leaking through the levee for example) can be a good, fast and non-destructive indicator of seepage which may eventually lead to levee failure. Early detection is essential in good levee management and this algorithm is developed to be a practical tool that can simplify and help the levee guards in keeping levees and what it protects safe. A supervised learning approach is chosen which allows this tool to continuously improve on new training and testing data. Such an approach is feasible because the image acquisition itself still needs to be improved and only few training and testing data was available at the time of research.

In a first step, suspicious areas (blobs) are detected in the thermal infrared image. To do so the image is pre-processed and the contours of suspicious areas are derived using maximally stable extremal regions. As a result of this step a collection of blobs becomes available. Using simple post-processing some of the blobs can be removed without much computational effort. In a second step, a set of features is derived that represent each of these blobs. Three kinds of features are implemented. A first kind that is based on the temperature measurements and a second kind that is based on image characteristics. These features are used to feed a classifier that separates the leaks from objects, image imperfections and other blobs that do not correspond to a leak. To decide on which features are best to be used, a recursive feature elimination algorithm guided by cross validation is implemented. It is inherent to this kind of problem that there is an imbalance between the number of blobs that are actually a leak and these that are not. This impedes the classification task at hand. To deal with this, different sampling techniques are implemented. Next to the imbalance in the data set, there is not much diversity in the set of true leaks because only few examples are available. Therefore, artificial samples are generated based on the derived blobs in step 1. In a third step a support vector classifier with linear kernel is implemented to perform the actual classification. Using a linear kernel has the disadvantage of only allowing relatively simple decision boundaries but the advantage of less likely over-fitting the data. Excellent classification results were obtained on the training set, with slightly less accurate results on a test set which was not used while training the classifier.

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations and symbols

AI	Artificial Intelligence
ASM	Angular Second Moment
ANN	Artificial Neural Network
C	penalty parameter of the error term
CIE	International Commission on Illumination
FHR	Flanders Hydraulic Research
FN	false negative
FP	false positive
GLCM	Grey Level Co-occurrence Matrix
GPS	Global Positioning System
IR	Imbalance Ratio
LUV	(L*, u*, v*) colour space
MSER	Maximal Stable Extremal Regions
No.	Number of
ORB	Oriented FAST and Rotated BRIEF
RGB	Red-Green-Blue
ROSE	Random Over Sampling Examples
SIFT	Scale Invariant Feature transform
SVC	Support Vector Classifier
SVM	Support Vector Machine
T	test set
TIR	Thermal infrared (image)
TN	true negative
TP	true positive
TPR	true positive rate
TR	Training set
UAV	Unmanned Aerial Vehicle
Σ	data set

Chapter 1

Introduction

This thesis combines two worlds. Artificial intelligence (AI) on the one hand and water engineering on the other. The author is aware that many readers will not be familiar with both. It is however impossible to formulate all required information to make the readers with a single background feel comfortable. As this thesis is one that is written to obtain a degree in artificial intelligence, special attention will be paid to explain the water related issues with the necessary detail to the AI reader. For the AI related terminology the level of a master student in artificial intelligence is assumed. An extensive list of references is added to back-up the less familiar with AI reader when and where necessary.

1.1 Problem description

On January 3, 1976 over 2000 people had to be evacuated from the village of Ruisbroek (Belgium) [VHHPM10]. Not fire, nor bomb alarm or war, but water had forced people to leave their homes. On that day, a levee on the river Vliet (a tributary of the river Rupel) failed and flooded the city of Ruisbroek to a depth of three meter. More recently on January 31, 1995 a quarter million people had to be evacuated from central parts of the Netherlands because river levees were thought to fail [Moh08].

From these two examples and because of the geographical location, it is clear that Flanders is prone to flooding. Over the last century levees have been built to protect cities and their houses from being flooded. Levee failure is an important issue because it can affect many persons' life. Therefore, it is important to detect such failure as soon as possible. In this scope a lot of research has been carried out. There are many active actors such as the Catholic University of Leuven, Flanders Hydraulics Research (Flemish Government) but also private companies such as Antea Group.

Currently there is an on-going research project in which Flanders Hydraulics Research and Antea Group try to detect levee failure in a non-destructive way. One of the techniques tested during this research is that of using thermal (infrared) images to detect water flowing out of a dike at the landside. The basic idea is that river water

has a different temperature from that of the dikes surface and that there is a clear contrast between both. It should be mentioned that levees can fail in many ways however [GVH+02]. In this research the fail mechanism of seepage ('*a differential water head that causes water to flow through the permeable layers under/in the dike and can lift soil layers at some distance behind the levee*'[PK00]) is considered.

First results of this research have shown that thermal infrared images can indeed be used to detect leaks. However, too date no automated, fast and objective way to analyse these images exists. Such a tool is essential in applying this kind of leak detection on a large scale. The presented work is such a tool that uses both computer vision techniques and supervised learning to analyse the thermal infrared images.

1.2 Objectives

The objective of this thesis is clear. An algorithm should be designed that can locate leaks (seepage) in thermal infrared images taken on the landside of a river levee.

This algorithm needs to meet the following requirements:

- It should run automatically and with as little user interaction as possible, in order to allow a reproducible and objective output.
- It should be self-learning so that it can easily be applied to new data. This is necessary because the image acquisition technique is currently still under investigation.
- It should be able to cope with the imbalance and lack of variation in the available data. This means it should be able to deal with a lot of examples of what is not a leak (true negatives), but only few of what is truly a leak (true positives).

It should be stressed that the objective of this thesis is not to develop an algorithm that can/will replace human inspectors. Instead, the idea is to aid levee inspectors in identifying leaks in a more efficient way. Or, put differently, the algorithm should screen the data and warn for possible locations that require further human inspection.

1.3 Related work

The use of infrared images in combination with artificial intelligence is not new. In [JT12], an application is presented in the field of an electrical setting. Another example can be found in [Wre06]. Here infrared images are used for fault detection in electrical installations. In this work, interesting regions are found by finding repeated patterns and finding smooth, but steep image gradients. By combining both, a decision support system is obtained that aids (inexperienced) personnel.

There are also applications of thermal infrared images for detecting seepage in levees. An application can be found in the Netherlands as described in [GVH+02]. In this work, thermal infrared imaging is applied on the Nieuwaal in the Netherlands precisely to detect seepage. There were also some early infrared campaigns that were carried out by Rijkswaterstaat (NL) in the period 1993-1995 [FR94]. The objective of taking thermal infrared images (TIR) was to aid decision making in times of critical water levels. In those days the technology had however not yet achieved the possibilities it has nowadays (resolution 4x4 meter, processing of the data, etc.). The technique proved useful but had its limitations. In [FHA95] it is mentioned that objects (buildings) often show up in the same colour as the leaked water. A solution they found was to add topographic images (visual images) to separate those objects from leaked water.

1.4 How to read this document

The structure of this document corresponds with the different steps that were taken to construct the final algorithm to detect leaks from thermal infrared images. This also corresponds with the chronological order in which the different modules were built. The current section formulated the problem at hand and correspondingly listed the objectives.

In a second chapter, the available data (thermal infrared images) are introduced and the ground truth was determined. Different approaches to detect blobs are tested and based on these findings a maximally stable extremal region technique is presented to do the blob detection. In order to improve the blob detection the images are pre-processed.

In a third chapter a set of features is extracted for each of the previously derived blobs. These features describe the blob in a way the classifier (next chapter) can process it and separate the true leaks from other blobs. Three kinds of features are implemented: temperature, shape and texture based features.

In a fourth chapter the actual classification is presented. In a first section, the classifier, here a support vector classifier, is introduced. After that some widely used evaluation metrics are discussed to measure the classification performance. Using the features derived in chapter 3 a first classifier is trained and discussed in section 4.4. The first results introduced the necessity for some improvements. A first paragraph deals with balancing the data set by means of sampling. In a second paragraph, artificial samples are introduced to increase the diversity in the data set. A last improvement made to the classifier is to select the relevant features. Ensemble learning by means of bagging is applied to improve the results as a final step.

In the fifth chapter, the final conclusions are given and the last chapter sketches the future perspectives.

Chapter 2

Blob detection in thermal infrared images

A first step in designing a tool that can detect potentially dangerous points in a levee is to identify suspicious areas. In this chapter a first but computationally demanding approach will be discussed using a moving window. Because this approach is time consuming, and because more intelligent approaches exist, a second and more advanced set of blob detection techniques are tested. The performance of both techniques, in terms of detecting the real leaks and not having too many other suspicious zones, is analysed and one technique is suggested and used as the blob detector in this work.

At the beginning of this chapter it is necessary to emphasize that detecting all potential leaks is really important. Missing out one real leak at this stage needs to be avoided at all cost. Indeed, it is better to identify one thousand suspicious zones that do not correspond to a real leak than to miss out one real leak and having no fake suspicious zones.

2.1 Thermal infrared images

The start point of this thesis is thermal infrared images. The images used here were made during previous research [WVVH+13] and are used with the permission of Flanders Hydraulics Research.

2.1.1 What is measured?

The idea of thermal infrared images used for seepage detection is that there is a temperature difference between the levee body and water leaking through it.

A complete and extended theoretical description of infrared image thermography and how to interpret them can be found in [LKC08, GVH+02, PN83]. For the purposes in this work, it suffices to understand that the camera is able to measure temperature. Using advanced post-processing techniques, it is possible to make a georeferenced

temperature map. Part of such a map used in this work is given in figure 2.1. For more details the reader is referred to [WVVH+13].

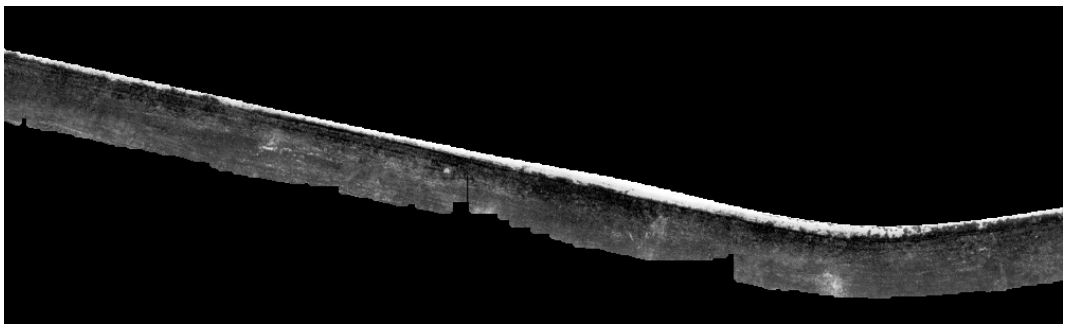


Figure 2.1: Example part of infrared image (Hingene, Belgium)

2.1.2 Data used in this work

Thermal infrared images as described above have been made for two levees along tidal rivers in Belgium. A first project site is a levee in Hingene. There a thermal map was made covering approximately 1.3 kilometre of levee. Only the talus at landside is mapped. This has been done four times. Twice from a camera mounted on a car driving on the road at the toe of the levee and two times more with a similar composition but from the top of the levee. Each time a slightly different camera angle (30° or 40°) was used to do the mapping. Two images will be used that were made at the toe of the levee. The images are made on the same day with 20 minutes in between the start point of image capturing [WVVH+13].

The second project site is a levee in Walem. A thermal map was made there on two locations in total resulting in more than 2 km image material from the top of the levee. Unfortunately, the data obtained from this test site is less useful because there was less thermal contrast. Also, as will be explained later, the situation in the images does not correspond anymore with reality because the levee was repaired, hence making it impossible to assign the ground truth to this image.

In both cases the pixel size is 10 cm by 10 cm[1], which is small enough to detect the leaks[WVVH+13].

[1] Actually a higher resolution is possible. The original data has a pixel size of 2 cm by 2 cm. When georeferencing the image some of the detail was omitted on purpose to limit the total image size. This is however not necessary and as turns out by this work is actually not feasible. Reprocessing the image was unfortunately not an option.

2.2 Ground truth

2.2.1 Acquiring the Ground Truth

In order to label the suspicious zones as true or a false, a field trip to both the project zones was organised. The responsible levee guards were asked to locate the true leaks and the positions of the leaks were captured with a GPS (Global positioning system). Thermal infrared and normal images were made simultaneously from the leaks during the inspection. The camera used was a FLIR T650sc that takes infrared images at a resolution of 640 by 480 pixels[2]. For each of the identified leaks an image was made from the toe and crest under more or less similar camera angles.

A visit to the project zone in Hingene was organised on October 9, 2013. Using the GPS to measure the exact location was sometimes impossible because there were not enough satellites available (tree influence). Therefore, the closest point on the levee crest was measured instead. The images were taken in a time span of 1.25 hours starting approximately one hour after high water. During this survey it was clear again that taking the images from the air is preferable because at some positions on the crest the camera sight was very limited. Therefore, an additional image was made from a point on the slope that allowed a better sight. As mentioned before it was raining (light showers) before and during the survey. The days before had been dry. Therefore, the found wet zones are very likely related to leakage and not accumulated rainwater. The images however turned out to be useless because of a very low thermal contrast, rainy conditions and low resolution.

On December 10 2013, the test zone in Walem was inspected together with the responsible levee guard. The same problem of GPS inaccuracy also occurred here. Therefore, the closest point on the levee crest was measured instead. The images were made between 10.30 and 12.30 and during cold but sunny conditions. The conditions were near optimal (measuring during the night would even be better) because there was a high thermal contrast, the vegetation was mowed and the measurements were made during high water. It is noteworthy that there had been very high water levels some days before due to extreme weather caused by Cyclone Bodil[3]. Unfortunately, it turned out that a lot had changed in the project site since the images had been taken. Large parts of the levee had been rebuilt. This means that the conditions of taking the TIR images and doing the field survey cannot be compared, making this project site less useful for training and validation.

The exact locations will not be listed here but more information can be requested at the appropriate authorities[4]. It should be stressed out that these leaks are known to those responsible, hence the levee guard identified them. It should also be noted that these leaks have been evaluated by the authorities and if necessary have been

[2] Technical specification can be found on: http://www.flir.com/cs/emea/en/view/?id=57921
[3] http://en.wikipedia.org/wiki/Cyclone_Bodil
[4] Flanders Hydraulic Research - waterbouwkundiglabo@vlaanderen.be

dealt with (see Walem, were part of the test site has become useless for annotating the ground truth).

As mentioned above, it was often impossible to obtain the acquired accuracy with the GPS. In that case, the GPS would be placed closer to the top of the levee. The coordinates where then manually moved to correspond to the leak visible in the thermal infrared image. Doing so 18 unique true leaks could be derived in the two thermal infrared images used in this analysis. An example is shown in figure 2.2.

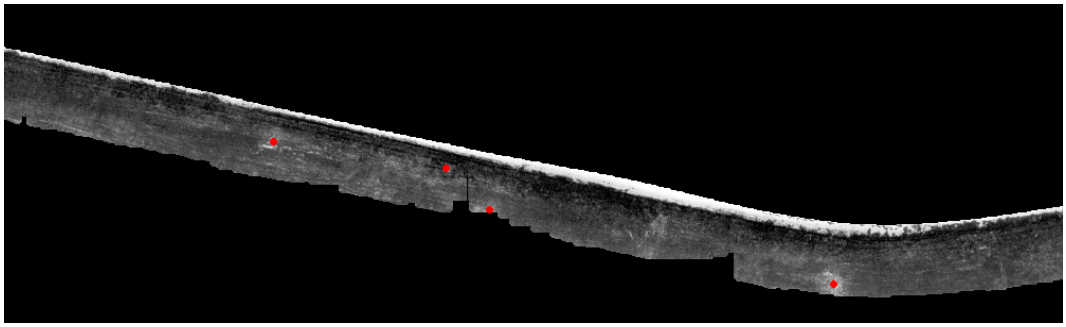


Figure 2.2: Ground Truth knowledge added to the image (red dots correspond to known leaks)

2.2.2 Assigning the ground truth

In the next section, different algorithms to detect blobs will be analysed. The number of blobs that correspond to a known leak will be a scoring metric to evaluate the blob detector. Because these algorithms find many blobs it would not be feasible to perform manual labelling of each of the blobs (leak or not). This would be very time consuming and would also allow errors to sneak in. Manually labelling would also not allow easy and rapid re-usability. Therefore, an algorithm was designed that automatically labels the derived blobs.

The idea is simple. If a known leak is included (or on the boundary) of a derived leak it is labeled as a leak. An algorithm based on [Slo85] was implemented that computes if a point (the position of the known leak) is inside, on the edge of, or outside a polygon (the blob contour). The implementation here also allows computing the distance from the point to the polygon. As such, this distance can be used to assign the true or false label with some tolerance to compensate for inaccuracy (due to GPS localisation inaccuracy, e.g.). The tolerance was set to 0.01 m because GPS inaccuracy had manually been dealt with as explained in paragraph 2.2.1.

2.3 Deciding on a suitable blob detector

2.3.1 Blob detection using a naive approach

The general idea of applying thermal infrared images when detecting leakage is simple. Water coming from the river and leaving the levee might have a completely different temperature compared to the surrounding surface. A first and naive approach could then be to scan the infrared images for pixels with a high temperature and surrounding pixels that are significantly cooler (or vice versa).

A first algorithm was implemented that does exactly that. It scans the image row by row (moving window) and scans for a warm pixel against a cold background. The implementation here considers a pixel and compares its temperature with the mean temperature in four zones away from the pixel (one zone on the left, one on the right, one above and one below). The boxes had a size of 1m by 1 m and were four metres away from the considered pixel (centre to pixel).

On the available data, this naive algorithm found more than 150000 suspicious pixels. This approach is also very computationally demanding and took over three hours[5]. After detecting these pixels, additional computations should take place to derive the contour around that pixel that would correspond with the area that is considered to be a leak. The high computational effort could be reduced by improving the efficiency of this naive approach, but as will be seen next, smarter techniques are available that allow deriving these blobs in a more efficient way.

2.3.2 Using more advanced techniques

All the techniques tested here are actually feature descriptors that look for key-points in the image that characterise the image. This technique is often used to match images, but is only useful here if these key-points correspond to suspicious areas and the known leaks.

OpenCV [Bra00] has a set of feature detectors available that were applied to the available data. In this section, the different detectors will not be introduced separately because it would lead too far and only one of the detectors will be used in the remainder of this work. The selected technique will be introduced in the next section. The results are shown in table 2.1 when using default settings for the different descriptors[6]. In this table, the number of blobs (No. blobs) found by the detector are given first. The number of blobs that correspond to a real leak (No. true blobs) is given in absolute number and in percentage. And the last column indicated how many of the known leaks are found by the detector. From this table it is clear that

[5] Computations took place on a laptop with an Intel i5 processor with four cores running at 2.67 Ghz. Two GB RAM was available and Win7 32bit was installed as operating system.)

[6] Actually, improvement can possibly be obtained by using more advanced settings, but as will be clear from the remainder of the text, there will be a clear preference for one certain technique. Fine-tuning the settings for each of the detectors was hence not done

MSER (on Grid) and ORB (on pyramids) are the only two detectors that detect all eighteen leaks. It is also clear that there are much more blobs found than there are known leaks. This table also shows that there are many blobs that are assigned to the same leak.

Detector		No. blobs	No. true blobs	%	No. leaks
Grid	FAST	1302	42	3.23	6
	STAR	851	11	1.29	3
	SIFT	1222	35	2.86	7
	SURF	1238	114	9.21	15
	ORB	1180	155	13.14	9
	MSER	858	58	6.76	18
	GFTT	1302	41	3.15	6
	HARRIS	1293	24	1.86	5
Basic	FAST	107889	4231	3.92	8
	STAR	2424	43	1.77	8
	ORB	1000	143	14.30	14
	GFTT	2000	42	2.10	7
	HARRIS	2000	38	1.90	7
Pyramid	FAST	127913	2920	2.28	8
	STAR	3074	38	1.24	11
	ORB	2987	1085	36.32	18
	GFTT	5495	60	1.09	8
	HARRIS	2913	20	0.69	3

Table 2.1: Blob detection using feature descriptors. The following abbreviations are used: No. blobs = number of blobs found by the detector; No. true blobs = the number of blobs that correspond to a real leak (given in absolute number and in percentage); No. leaks = how many of the known leaks are found by the detector.

Comparing MSER and ORB it is clear that the number of blobs is lower for MSER. On the other hand, the number of blobs corresponding to known leaks is much higher for ORB compared to MSER. A better insight is obtained by considering figures 2.3 and 2.4. In these figures, the detected key-points with MSER and ORB are shown respectively. Each key-point is given by a centre point and a radius. It is clear that in the case of ORB many key-points more or less have the same centre point but a different radius. This explains why the number of blobs that are close to a known leak is much higher. In both cases there seem to be key-points with a radius that is so large that they can easily cover multiple leaks that are close to each other. For the MSER feature detector this seems not to occur however, while for the ORB feature detector it clearly happens a lot[7].

[7] A possible explanation is that ORB uses Harris corner measure to find a number of best key-points. Since there are not many 'corners' in this image the best key-point are mainly found around the same blobs and thus the same key-points but with a different radius are returned.

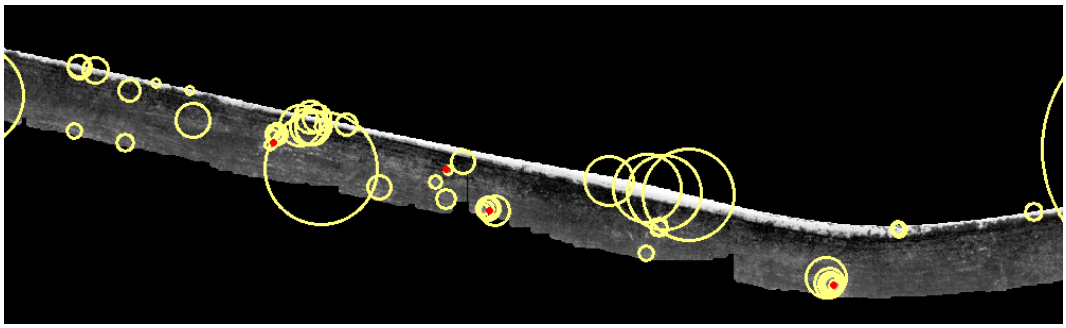


Figure 2.3: Key-points detected with MSER (in grid)

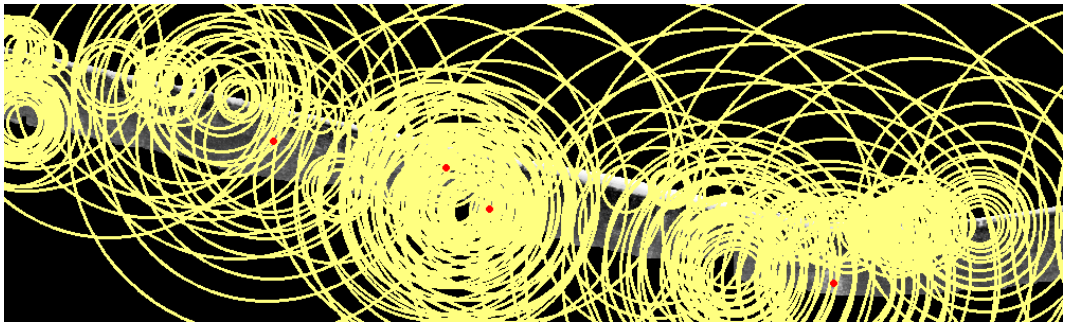


Figure 2.4: Key-points detected using ORB (in pyramids)

Based on the fact that MSER results in fewer blobs, that seem to be corresponding better with the known leaks and the fact that the MSER implementation in openCV allows deriving contours this technique is preferred and will be used further on. Until now, the different key point detectors have been considered as black box models without actually explaining how these work. In the next section, the MSER technique will be introduced as the blob detection algorithm used in the remainder of this work.

2.4 Maximally Stable Extremal Regions

Maximally Stable Extremal Regions (MSER) is a technique originally proposed by Matas et. al [MCMP02]. It is (originally) used to find key-points (correspondences) in images taken from different viewpoints. A MSER is a stable connected component of some level sets of the image as is shown in figure 2.5. The algorithm detects regions which stay the same over a range of thresholds. In the figure two approaches are given. The first (from white to black) is considered here. At initialisation, the threshold is set such that all pixels have an intensity above the threshold. As a result, all pixels are assigned to the foreground (white) in the binary output image. By increasing the threshold, some pixel intensities drop below the threshold level and turn into background (black) pixels in the output image. By continuously increasing the threshold more and more pixels are assigned to the background and

stable background regions start to form. The flower hearts are good examples. In a similar way, the leakages in this application can be detected. More information can be found in [TM08, Wik13, MTS+05, MS04]. It turns out that MSER has some interesting advantages [MTS+05, MS04]:

- it derives stable features of arbitrary shape and scale
- low number of detected regions, repeatability is better compared to other detectors for viewpoint changes
- its efficiency is unbeaten

An obvious extension of this algorithm would be to consider the colour of the image. A slightly modified version of MSER has been used to do colour blob segmentation successfully [DBW06]. In this work, the RGB colours are transformed to the CIE Luv colour space in order to ensure isotropy of the feature space. Another similar approach was suggested in [For07]. Here the MSER algorithm was extended in such a way that it could detect regions based on colours as well. In the scope of this work greyscale images (single spectrum TIR) will be used and such an extension is not desired.

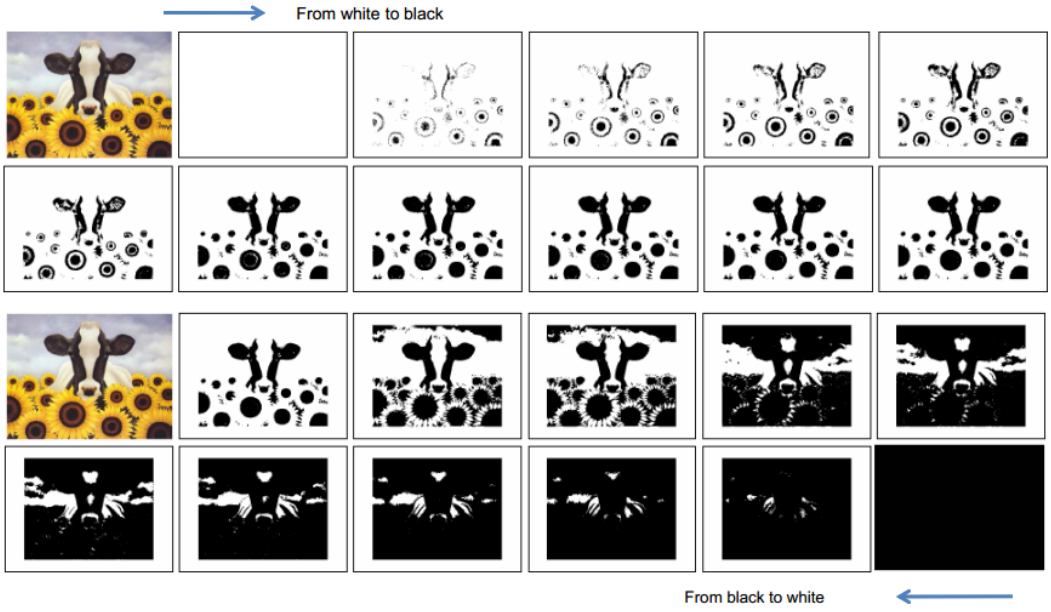


Figure 2.5: MSER: general idea

2.5 Pre-processing the images

The first objective is to have a blob at least at every true (known) location of a leak. In total there are 18 known leaks in the available data in Hingene (also see

4.2). The number of derived blobs is preferably as low as possible from a point of computational effort. The classifier should be able to deal with all these false blobs.

First analysis using MSER without pre-processing the image resulted in already good performance (see table 2.1). Improving the performance of the blob detection algorithm was obtained by pre-processing the image[8]. OpenCV again has a number of such pre-processing algorithms available. The following were considered: histogram equalisation, morphological filters (erosion and dilation - also: opening operation, closing operation, morphological gradient, top hat and black hat) and blurring (blur, Gaussian blur, Median blur and bilateral filtering). Table 2.2 lists the 10 best combinations that were obtained. The combination also considered using different kernel sizes for the circular kernel used. The kernel size is given in brackets in the table.

Pre-processing	No. derived blobs	No. known leaks found
Dilation (5)	920	18
Dilation (5) + blur (5)	669	18
Blur (5) + dilation (5)	608	18
Blur (5)	385	17
Dilation (3)	944	17
Hist. Eq.	1111	16
Blur (7)	229	16
Gaussian blur (5)	485	16
Bilateral filter (5)	689	16
Dilation (7)	803	16

Table 2.2: Pre-processing influence on blob detection

The best improvements were obtained when the image was blurred with a circular kernel of five pixels. In that case, all leaks were detected and the number of blobs decreased as well. Dilating the image did not affect the performance in terms of number of detected known leaks but further reduced the number of detected blobs. Therefore, in the remaining of the text the MSER will always be applied on an image that was first blurred and then dilated. It should be clear that this pre-processing is only carried out for detecting the blobs. The unmodified images are used for feature extraction.

The results after applying blurring and after applying dilation are given in figures 2.6 and 2.7 respectively. The detected blobs after pre-processing the image are given in figure 2.8. One can fairly argue that this kind of pre-processing may not work very well on unseen data and may even miss out true leaks. More data would indeed

[8]Pre-processing is not strictly necessary here because all true (known) leaks had already been detected. However, this pre-processing was implemented in an attempt to make the blob detection more robust and potentially remove some of the blobs that do not correspond to a known leak.

need to be analysed in order to find out if this is so. This data is unfortunately not available at this moment.

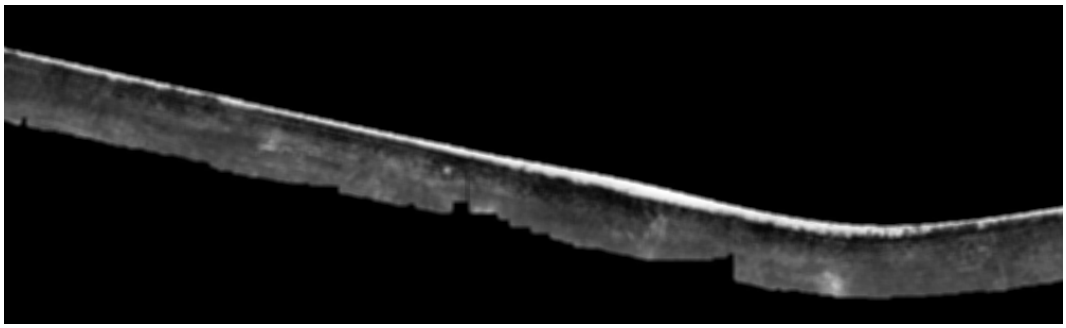


Figure 2.6: Thermal infrared image after applying blur with a circle of five pixels radius

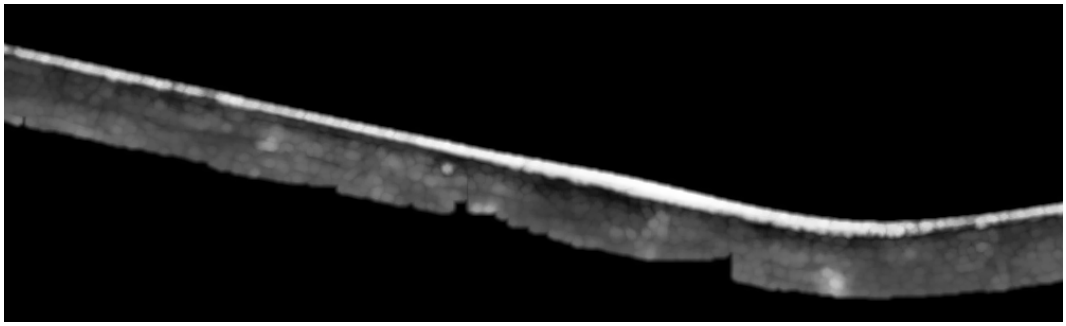


Figure 2.7: Thermal infrared image after applying blur with a circle of five pixels radius and applying dilation again with a circle of five pixel radius

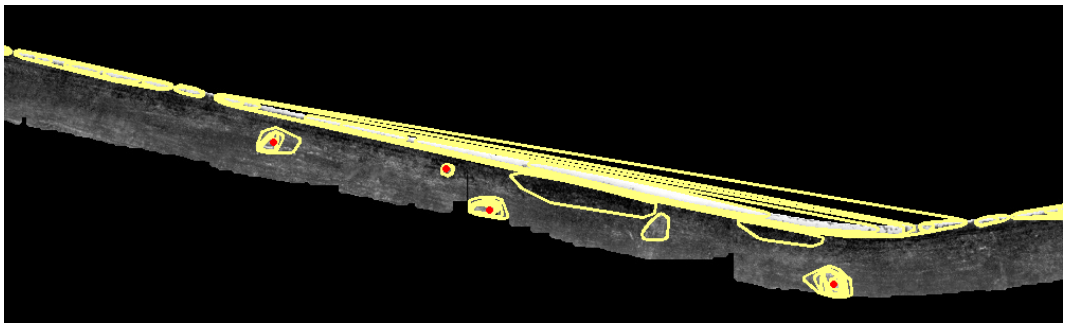


Figure 2.8: Detected blobs after having pre-processed the image

2.6 Post-processing the obtained blobs

2.6.1 Removing blobs that are obviously not leaks

Using the MSER (or any other technique) results in many blobs that can easily be classified as not being a leak. This is because the blobs are too large, too small, etc. These kinds of blobs can easily be detected in a later stage but are removed before being presented to the classifier. This is done because this kind of leaks can cause problems to the true labelling algorithm. An easy to understand example is that of unrealistically large blobs. Such blobs can cover many known leaks and thus cause incorrect true labelling. The results of this step are shown in figure 2.9. Removing this kind of blobs took place in an automated way by thresholding on different features such as the blobs width-height ratio and size.

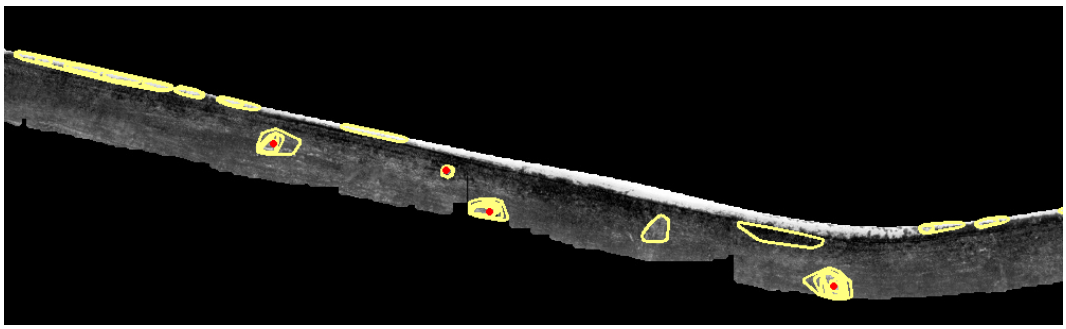


Figure 2.9: Detected blobs after removing obvious erroneous blobs

2.6.2 Removing contours that are not-minimal

In many cases, different blobs, i.e. contours, are obtained corresponding to a known leak as shown in figure 2.8. In this case, the smallest contour usually corresponds better to the area covered by the leak. The larger contours cover the leak, but also a lot of redundant information. This redundant information (part of the image that is not a leak) will influence the features that are derived from the leak. Therefore, in this step of the algorithm the derived leaks that correspond to a known leak are filtered in such a way that only those that are minimal remain. The results are shown in figure 2.10.

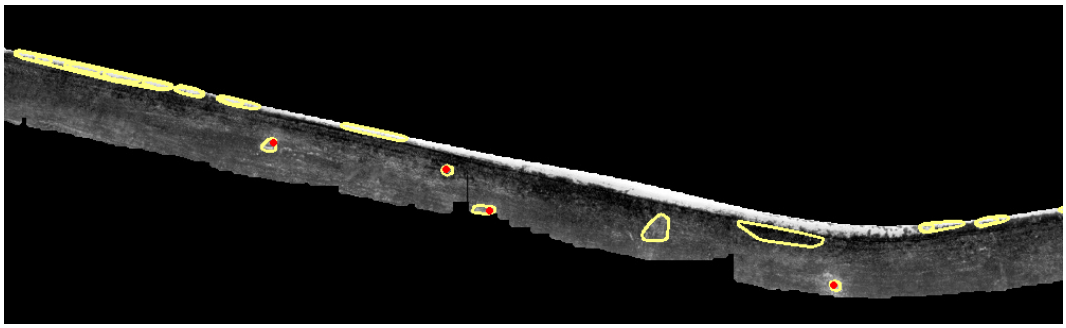


Figure 2.10: Detected blobs after removing non-minimal contours

2.7 Final results

A final step is labelling all the derived blobs as either a true or false leak. The results are shown in figure 2.11. The contours in yellow correspond to blobs that do not belong to a known leak. These in green are associated with a known leak. Using pre- and post-processing it was possible to detect all true leaks and to safely reduce the number of blobs that can never correspond to a true leak.

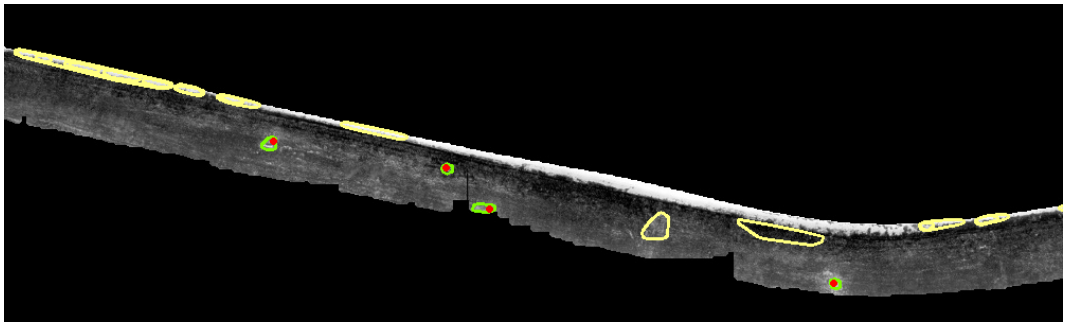


Figure 2.11: Blobs after having assigned a true label

2.8 Conclusions and final thoughts

Two important first steps were made in this chapter. First of all field trips were carried out to establish the ground truth. Based on the experience of the levee guards as well as experience gained during other tests on the same levee (see [WVVH+13]) a number of leak zones were listed. This information serves as the ground truth for all later work.

Different blob detection techniques were tested. MSER turned out to be the most promising. When applying this technique on a pre-processed image it is possible to obtain all the known leaks together with many more suspicious blobs. Applying

some post processing allows reducing the number of blobs without risking to miss out any of the true leaks.

As mentioned before, it is unclear how well the pre-processing will work on unseen data. More data was however not available to further investigate this. This will be a point to take into account when applying the algorithm to new data.

Chapter 3

Feature Extraction

In the previous chapter, a set of suspicious zones was derived from the thermal infrared images. This set was derived in such a way that it was sure to include all real leaks and as few false zones as possible. In a later step, a classifier will be trained that finds the difference between these true and false leaks. To do so a set of input data is needed on which a model can be trained that separates the true from the false leaks. This input data should be representative for the suspicious areas and is called a feature.

Many different features can be derived for different kinds of problems. For the problem at hand, the following three groups of features seemed appropriate:

1. temperature related
2. shape related
3. texture related

The first group is probably the most obvious as the temperature of the blob and how it contrasts to the environment is the initial reason why thermal infrared images were made. The second group is also straightforward because the observed leaks all had a specific shape and size. A third group of features is texture. This is maybe the least obvious group but may turn out to be very useful as well because it is known that there may be a different kind of vegetation at the location or near to leaks. A difference in vegetation might result in a different texture. This last group of features will also allow separating asphalt (roads), concrete (objects), from grass.

3.1 Introduction

A lot of work has been carried out in the last years to derived meaningful features from images that can be used to classify or match images. In what follows, a short enumeration of available work will be given. Many approaches can be followed but

in this work the recent work of Wang [WLYC14] will be considered as a start point. In his work Support Vector Machines (SVM) are used to classify images. They used colour, texture and shape based features to feed the classifier.

In [MS02] and [MCMP02], the problem of analysing images under arbitrary viewing conditions is considered. These papers mention a *scale space* of an image. This is important when one want to combine different sources (having a different scale) and combine in the hope to have a better leakage detection.

In [WSL+13] computer vision techniques are used to detect animals in the Mojave desert. This is of course not the objective of this work, but nevertheless it addresses some of the pitfalls that need to be addressed. The images should be scale invariant (for classification, different zoom factors) and need to be noise tolerant (camera vibration, orthorectification errors, etc.).

In [LSP03] a method is given that is invariant to geometric transformations and can be used in texture analysis. Obviously, temperature is an important feature for the work at hand, but water, however, also has a different texture from dry grass. Hence, texture might be a discriminating feature in classifying leaks.

Tuytelaars [TG00] detects affine invariant regions based on image intensities. The number of regions is however limited and depends on the content. The interesting thing about this is that image intensities are used. Colour intensity and temperature are linked, so this technique might be useful.

In [Phi07] a suggestion is made to make a comparison between the temperature (pixel or area of pixels) and the colour of the visual pictures. One of the advantages of this approach is that a leak does not always show up in the visual picture. Therefore, when this visual picture shows no variation but the infrared does this might be indicative of a leakage. Of course, separate checking for objects should take place.

3.2 Temperature based features

Temperature based features are straightforward and can be computed easily. The following features were implemented: mean ($\bar{T}$), maximum (T_{max}), median (T_{med}) and minimum (T_{min}) temperature calculated over the region inside the previously determined blob contour. In addition, the temperature difference with the area around the leak, T_{Δ}, and the standard deviation of the temperature in the leak, σ_T, are computed. All but the temperature difference are straightforward to implement.

The temperature difference between the maximum observed temperature in the blob contour and the area around the blob the following calculated as follows. From the blob contour, a minimum radius is computed from the centre point of the contour such that the contour is completely inside the circle described by this minimal radius. After that, a second radius is computed from the same centre point. This radius is constant and a little bit larger than the first. In the current implementation the

radius is 0.75m larger. The temperature inside the area between the two circles is averaged and stored away as the average temperature around the leak.

3.3 Shape based features

Next to temperature, a second set of features that is straightforward to use is shape related. The following features were implemented: area (A), height (h), width (w), eccentricity (ϵ), rotation (Θ), eigenvalues of the intensity-weighted moment matrix (λ) and height difference between the river water level and the height of the blob (z)[1]. Data from the hydrological information centre (part of Flanders Hydraulic Research institute) was used for the river water level at the time of acquiring the thermal infrared images.

Based on the image moment (a good tutorial can be found online at [CV13]), a different set of features can be derived. First features that can be computed are the central moments. For images, the central moments, μ, can be derived as:

$$\mu_{pq} = \sum_{m}^{p} \sum_{n}^{q} \binom{p}{m} \binom{q}{n} (-\bar{x})^{(p-m)} (-\bar{y})^{(q-n)} M_{mn} \tag{3.1}$$

In which $p, q = 0, 1, 2, ...$ and $\bar{x}, \bar{y}$ the image centroid:

$$M_{mn} = \sum_{x} \sum_{y} x^m y^n I(x, y) \tag{3.2}$$

and $I(x, y)$ is the pixel intensity matrix. M is called a (raw) moment of order $(p+q)$. From this, the eigenvalues can be computed as:

$$\lambda_i = \frac{\mu'_{20} + \mu'_{02}}{2} \pm \frac{\sqrt{4\mu'^2_{11} + (\mu'_{20} - \mu'_{02})^2}}{2} \tag{3.3}$$

In which:

$$\mu'_{20} = \mu_{20}/\mu_{00} \tag{3.4}$$

$$\mu'_{02} = \mu_{02}/\mu_{00} \tag{3.5}$$

$$\mu'_{11} = \mu_{11}/\mu_{00} \tag{3.6}$$

the orientation, Θ, can be computed as:

$$\Theta = \frac{1}{2} \arctan\left(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}}\right) \tag{3.7}$$

[1] z is actually not related to shape but is listed here because it fits best in this group of features.

And finally, the eccentricity (how elongated it is) as:

$$\epsilon = \sqrt{1 - \frac{\lambda_2}{\lambda_1}} \tag{3.8}$$

3.4 Texture based features

Texture can be a potentially important feature to aid classification. It is a priori known that the vegetation near leaks is different [WVVH+13], which may be visible even in infrared images. Furthermore, the texture of grass on the levee is significantly different from that of an asphalt road or a concrete object. For this reason, texture may lead to discriminating features. Expressing texture is however less straightforward compared to temperature or shape. Nevertheless, frameworks exist to set up such features.

In this work Grey-Level Co-Occurrence Matrix (GLCM) texture features are implemented. This framework was proposed in 1970 by Haralick [HSD73] and has since then been widely used in classification (for example: [Bra13, Pea00, CR04]). An introduction to the use and implementation of GLCM is given in [HB07]. Based on this work a number of features were implemented.

The basic idea is simple. The relation between two pixels is considered: a first which is the reference pixel and a second called the neighbouring pixel (here the pixel to the right of the reference pixel). Each pixel in the image becomes a reference pixel in turn[2] so that all pixels have been reference pixel exactly once. A square matrix M of as many rows and columns as there are levels[3] is constructed that keeps track how many times a reference pixel on a level, l_1, has a neighbouring pixel of another level, l_2. Thus $M(l_1, l_2)$ is incremented with one each time such a pixel combination occurs. After constructing M (by considering all reference pixels in turn) the matrix is made symmetrical by adding the transposed matrix to itself ($V = M + M^T$). The matrix is then normalised which results in equation 3.9. This matrix is the probability distribution of finding a pixel on level i that has a neighbouring pixel that is on level j.

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{levels-1} V_{i,j}} \tag{3.9}$$

A first feature derived from this co-occurrence matrix is the contrast, which is computed as in equation 3.10. The contrast is a positive (or zero) number. When

[2] To be precise not all pixels are considered. The last column of the image has no adjacent cell to its right and it is therefore impossible to use this pixel as reference pixel.

[3] The number of levels corresponds to the different values each pixel can take. In this work eight bit unsigned images are used. This means that the pixel value is an integer from 0 to $2^8 = 256$.

there is very low contrast in the image a low number will be obtained for this feature. When it is zero (or close to zero) it means there is no contrast[4].

$$contrast = \sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2 \tag{3.10}$$

A second feature implemented is dissimilarity. It is closely related to contrast but the weights $(i-j)$ are not increased exponentially but linearly. Dissimilarity is thus computed as:

$$dissimilarity = \sum_{i,j=0}^{levels-1} P_{i,j}|i-j| \tag{3.11}$$

A third feature that is implemented is homogeneity, which is also related to contrast but here the weights decrease exponentially away from the diagonal:

$$homogeneity = \sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2} \tag{3.12}$$

The three above-mentioned equation are often described as the contrast group. A second group is the orderliness group which includes four more features that were implemented. A first is the angular second moment (ASM) which is computed as in 3.13. The energy feature is derived by taking the square root as shown in eq. 3.14. High values occur when the image is very orderly (uniform). When all pixels are exactly the same, both energy and ASM equal to unity.

$$ASM = \sum_{i,j=0}^{levels-1} P_{i,j}^2 \tag{3.13}$$

$$energy = \sqrt{ASM} \tag{3.14}$$

Entropy, S, is a measure to quantify the disorder in the image. Higher values correspond to higher disorder. The maximum value is 0.5:

$$S = \sum_{i,j=0}^{levels-1} P_{i,j} \cdot (-lnP_{i,j}) \tag{3.15}$$

A last feature that was implemented is correlation, ρ. This feature expresses the linear dependency of the grey levels and that of the neighbouring pixels. This feature

[4]To be accurate, it means that there is no difference between the reference pixel and its neighbour. This not necessarily means that all the pixels are the same, but rather that there are *stripes* of equal grey level value.

is independent of the above mentioned and has an intuitive meaning. Zero means uncorrelated, and the maximal value is unity which means perfect correlation.

$$\rho = \sum_{i,j=0}^{levels-1} P_{i,j} \left[\frac{(i - \mu_i) \cdot (j - \mu_j)}{\sqrt{\sigma_i^2 \cdot \sigma_j^2}} \right] \tag{3.16}$$

The mean and variance in this equation is computed as:

$$\mu_i = \sum_{i,j=0}^{levels-1} i \cdot P_{i,j} \tag{3.17}$$

$$\mu_j = \sum_{i,j=0}^{levels-1} j \cdot P_{i,j} \tag{3.18}$$

$$\sigma_i^2 = \sum_{i,j=0}^{levels-1} P_{i,j} \cdot (i - \mu_i)^2 \tag{3.19}$$

$$\sigma_j^2 = \sum_{i,j=0}^{levels-1} P_{i,j} \cdot (j - \mu_j)^2 \tag{3.20}$$

It is of importance to stress out that not all pixels were considered when computing these features. only those pixels which are within the blobs contour were considered.

3.5 Conclusions and final thoughts

Many features have been introduced, and all have been implemented. However, it is not necessarily so that all need to be used for the best classification. Many features may turn out to be less relevant or they may already be included in others. In the next chapter, a framework is suggested and implemented to reduce the number of features based on their relevance.

Chapter 4

Classification

So far, all preparative work has been carried out (blob detection and feature extraction) to perform the actual classification. In this chapter, the real engine is presented. A model is designed that separates false from true leaks. It is a model that takes complex features as input and returns a very simple output: true or false. Leak or not a leak.

In this work a support vector classifier will be used. Support vector classifiers and their proven use in this kind of problems are discussed in 4.1. Having derived such a classifier model it is also necessary to evaluate it. Especially in the case of unbalanced data, a suitable evaluation metric is indispensable. The evaluation metrics implemented in the scope of this work are discussed in 4.3. Knowing how to evaluate the performance of a classifier the first results are given in section 4.4. Improvements are described and given in section 4.5. The results, and why and where classification fails is covered in section 4.7. The final section of this chapter draws some conclusions and sketches some final thoughts.

4.1 Support Vector Classifiers

Nowadays many people are acquainted with the term artificial neural network (ANN). An ANN is a universal approximator. This means that every continuous nonlinear function can be approximated arbitrarily well over a compact interval. This allows approximating very complex problems even without fully understanding it. Such ANN's find relations between a set of input and output parameters and if trained well can generalise very well even on unseen data. This is exactly the kind of tool that could proof useful in this work. Given a set of features as described in chapter 3 a mapping could be found that has a simple output: *leak* or *not a leak*.

In this work, however, ANN's will not be used. Instead, a less familiar technique will be used to do the classification. Support vector machines, or when used for classification very often called support vector classifiers (SVC), will be used instead. The reason why SVC is preferred above ANN is twofold. First of all, setting up

the architecture of an ANN can be tricky as pointed out in for example [SGB+02]. Selecting the number of hidden layers and the number of neurons on each of these layers can be time consuming and in our case would easily result in over-fitting (memorising) because of the limited number of samples. SVM on the other hand does not require this as the number of support vectors follows from solving the optimisation problem. A second drawback of ANN compared to SVM is that there may exist many local minima. Multiple random initialisations of the weights are often required to obtain the best result and even then the best result might not be found. On the other hand SVM has only one unique solution because a convex, compared to a non-convex for ANN, optimisation problem is solved.

Choosing support vector classifiers for this kind of problem has shown to be a very fruitful approach over the last years [SGD02, QTS14, AHP14]. A lot of research has also been carried out by the Catholic University of Leuven [SGB+02]. The less AI-familiar reader is referred to this book, as it is an accessible introduction to SVM.

In choosing an appropriate classifier, SVC is also very suitable because it easily allows using different kernels. In this work the scikit-learn library [PVG+11] is used as a base to do classification. This library supports complex kernel usage, but in this scope a linear SVC will be used to separate the different classes. Using a simple kernel allows less complex decision boundaries but prevents over-fitting on the data and thus ensures better generalisation on unseen data. However choosing a support vector classifier easily allows plugging in a more complex (radial basis function kernel, polynomial kernel, etc.) when more data may become available in the future.

4.2 Training and test set

When training the classifier it is important to have two disjoint data sets. A first to train the model and a second to evaluate the models performance. The data presented under 2.1.2 will therefore be split up in two disjoint sets as follows:

1. In a first, Σ_1, one of the images will be used as training data and the other as test data. This will allow seeing how robust the algorithm is for different camera angles when acquiring the images. The training set contains ten leaks, while the test set only contains eight. This is because the image used for the test set is incomplete at some parts.

2. In a second, Σ_2, the test set will be based on a geographical split. Data on the left side of a point is in the training set, while data on the right of the point is in the test set. As such, it is possible to examine the performance of the classifier on truly unseen data. This point has been chosen such that the training data contains $\frac{2}{3}$ of the positives and the test set contains the other $\frac{1}{3}$ of the available positives.

In what follows results will not always be given for both sets. In those cases, it will be explicitly mentioned for which data set the displayed results are.

4.3 Evaluation metrics

An important part in training a classifier is to keep track of its performance on the training and test set. Different evaluation metrics exist and those used in this work will be introduced here. There are however many more evaluation metrics than given here. A more complete overview can be found in [SH00], [MT12] and more recently in [LFG+13].

Not listed here as well are cost-sensitive evaluation metrics. These are not considered because it is unknown what cost should be given to detecting a true leak and not detecting a true leak. This is a different kind of discussion that requires putting value on levee failure and associated lost (economical, human) and is not part of this research[1].

The most well-known evaluation metric when classifying is accuracy. This is the ratio of the number of correctly classified samples over the total number of samples. The higher the accuracy, *Acc*, the more samples are correctly classified. *Acc* is a value between 0 and 1. An evaluation metric that allows more insight is the confusion matrix. Such a matrix is given in table 4.1. This kind of matrix gives an idea of how well the classifier predicted corresponding with the ground truth.

	Predicted as negative	Predicted as Positive
Actual negative	True Negative (TN)	False Positive (FP)
Actual positive	False Negative (FN)	True Positive (TP)

Table 4.1: Confusion matrix

From this table the accuracy can be computed as:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

Having a skewed class distribution (here data imbalance) not only affects the estimation of the classification model, it also affects the evaluation of the classifier [MT12, Cha05, LFG+13]. This is because the rareness of one class may lead to a poor accuracy estimate. Thus, (overall) accuracy might be a bad way of evaluating a classifier. For this reason, some other metrics are implemented. A first is the recall, also called true positive rate or sensitivity:

$$recall = \frac{TP}{TP + FN} \tag{4.2}$$

A second is the false positives rate (FPR):

[1] Assigning a cost to such types of loss (flooding, drought, ...) is however currently a topic of research within Flanders Hydraulic Research. When the outcome is known, it may allow such an implementation.

$$FPR = \frac{FP}{TN + FP} \tag{4.3}$$

A third used in this work is precision which is computed as:

$$precision = \frac{TP}{TP + FP} \tag{4.4}$$

When training a classifier, and especially so for unbalanced data sets, the idea is to improve the recall without lowering the precision. There is however a trade-off between both and therefore the $F - value$ was introduced by [BG94]. This can be seen as a goodness factor and is computed as:

$$F - value = \frac{(1 + \beta^2) \cdot recall \cdot precision}{\beta^2 \cdot recall + precision} \tag{4.5}$$

In this equation, β allows to play with the relative importance of precision versus recall. It is often set to 1. In this the metric is often called the balanced $F - value$. The best results are obtained when the $F - value$ equals 1. The worst when equal to 0.

4.4 First results

Tables 4.2 and 4.3 show the performance of the classifier on both Σ_1 and Σ_2. In this table, the results are presented for different penalty values of the error term, C (this is the only parameter that can be set using a linear kernel). Considering Σ_1, the performance in terms of accuracy seems to be excellent on both the training and test set. However when the number of false negatives (these blobs are actually leaks!) are considered one can see that the results are less good. At best, a recall of 5/8 is obtained. This is slightly better than a random guess. In terms of usability this is unacceptable. These findings again point out that accuracy can be a misleading evaluation metric.

A similar conclusion can be made when looking at the performance on Σ_2. Again, the training accuracy is very high and recall is not much better than a random guess. The accuracy on the test set is even lower compared to that of Σ_1. This is because the training and test data of the latter is actually very similar (only a slight time difference and different camera angle) compared to Σ_2 where a geographical split was made between the training and test data.

Σ_1	No. TP		No. FP		No. TN		No. FN		*Acc*	
C	Tr	T	Tr	T	Tr	T	Tr	T	Tr	T
0.1	3	1	0	4	244	280	7	7	0.972	0.962
1.0	6	4	1	8	243	276	4	4	0.980	0.959
10.0	6	4	1	5	243	279	4	4	0.980	0.969
100.0	6	5	1	7	243	277	4	3	0.980	0.966

Table 4.2: Classification results on the unbalanced data for Σ_1. In this table, C is the penalty value of the error term and TR and T represent the Training and test set respectively.

Σ_2	No. TP		No. FP		No. TN		No. FN		*Acc*	
C	Tr	T	Tr	T	Tr	T	Tr	T	Tr	T
0.1	8	1	0	29	302	197	4	5	0.987	0.853
1.0	11	1	0	80	302	146	1	5	0.997	0.634
10.0	11	3	0	58	302	168	1	3	0.997	0.737
100.0	11	3	0	58	302	168	1	3	0.997	0.737

Table 4.3: Classification results on the unbalanced data for Σ_2. In this table, C is the penalty value of the error term and TR and T represent the Training and test set respectively.

The trained classifier clearly over-fitted the training data and does not generalise well on the test data. Plausible causes are:

- The large number of features used allow over-fitting. It also explains the short training time, as well as the perfect classification results.
- Because of the low number of real positive samples (true leaks), the diversity in the training examples is limited.

In the following section these issues are tackled.

4.5 Improving classification results

Fortunately for those living near the monitored levees, but unfortunately for the purpose of this thesis, there are only few known leaks on the monitored levees. This means that the majority of detected blobs correspond to false leaks and only few to true leaks. This is called an unbalanced data set. If not handled with caution this kind of unbalance may result in an inaccurate classification (on unseen data). This kind of problem however is not new and has been tackled in other research domains such as fraud detection with credit cards [PDK+07] and medical image analysis [BKT14].

In this section, different techniques to counter data set unbalance and to create data diversity will be presented and evaluated. Next to that, it is also important to have a good and representative set of features to train the classifier. Therefore the features used in this work will be examined to see their importance in the classification. In a first step, the three feature groups will be evaluated and in a second step a technique to automatically derive the best set of features will be discussed. A third improvement that will be made is obtained by adding artificial samples to generate more diversity in the data set. And a final measure examined is that of ensemble learning.

4.5.1 Balancing by sampling

Only few of the previously derived blobs correspond to true leaks which may lead to bad classification. The severity of imbalance is usually given by the imbalance ratio, IR, which is defined as the number of samples in the majority class divided by the number of examples in the minority class.

Different approaches have been suggested to deal with unbalanced data sets [ZBRV04, Jap00, DPCB11, LFG+13, Cha05, PDK+07, BKT14]. The first and earliest approaches simply re-sized the data set. This approach is both simple and intuitive: the imbalance is countered by either under-sampling the majority class or over-sampling the minority class. Different variants of this approach have been reported as well, including random sampling, focused sampling and combinations of these techniques [CJK04].

In general when under-sampling the majority class it is important to keep track if removing an example affects the decision boundary. If it is far away from the decision boundary it can be removed safely. However, the first mentioned approach of under- or over-sampling the data set has some drawbacks as pointed out in [MZW05]. Under-sampling has the drawback that it may potentially remove useful data from the set. Over-sampling, on the other hand, may potentially lead to over-fitting and may thus impede generalisation on unseen data.

In [MT12] a unified and systematic framework is given to deal with these shortcomings. The technique is based on smoothed bootstrap re-sampling and is called rose-sampling (Random Over Sampling Examples). The same article also brings up the idea of creating artificial data that will be presented in paragraph 4.5.3. In [DK01] it is shown that the best results are not always obtained for sampling so that the data set is completely balanced (50-50). This is thus a point that should be taken into consideration.

The three above-mentioned techniques were implemented in this work. Over-sampling is implemented as a recursive method that selects samples from the minority class until a certain set size is reached. Under-represented samples in the new set will be assigned a higher weight and as such they will have a higher probability of being selected compared to examples that are better represented in the new set. In a similar

way under-sampling was implemented. The majority class is reduced to a certain set size by removing samples. Again a proportional probability is assigned for being selected. Rose-sampling was also implemented with this proportional probability of being selected. The selection procedure continues until a predefined number of samples, s, is reached.

In what follows, the different sampling techniques that were implemented are analysed on Σ_2. The displayed results are the average results out of ten trials. Also included in this analysis is the choice of the imbalance ratio and the penalty parameter of the error term, C, of the SVC. In the following tables, TR is used to indicate the training set and T the test set.

Under-sampling (table 4.4) clearly over-fits the training set and generalises badly on the test set because the accuracy on the training set is perfect while that of the test set is much lower. This is because only few data is used to train the classifier. The classification on unseen data seems to improve for higher IR, which is probably because more negative samples are allowed which results in more training data and as a consequence better generalisation. From these experiments it is recommended not to use under-sampling.

Σ_2		No. TP		No. FP		No. TN		No. FN		*Acc*	
C	IR	Tr	T	Tr	T	Tr	T	Tr	T	Tr	T
0.1	0.8	12.0	5.7	0.0	77.8	9.0	148.2	0.0	0.3	1.000	0.663
0.1	1.0	12.0	5.4	0.0	76.0	12.0	150.0	0.0	0.6	1.000	0.670
0.1	1.2	12.0	6.0	0.0	71.4	14.0	154.6	0.0	0.0	1.000	0.692
1	0.8	12.0	5.1	0.0	80.8	9.0	145.2	0.0	0.9	1.000	0.648
1	1.0	12.0	4.7	0.0	68.8	12.0	157.2	0.0	1.3	1.000	0.698
1	1.2	12.0	4.0	0.0	66.0	14.0	160.0	0.0	2.0	1.000	0.707
10	0.8	12.0	5.9	0.0	86.5	9.0	139.5	0.0	0.1	1.000	0.627
10	1.0	12.0	5.1	0.0	82.0	12.0	144.0	0.0	0.9	1.000	0.643
10	1.2	12.0	5.4	0.0	65.9	14.0	160.1	0.0	0.6	1.000	0.713

Table 4.4: Classification results when applying under-sampling. In this table, C is the penalty value of the error term and TR and T represent the Training and test set respectively. IR is the imbalance ratio.

Over-sampling (table 4.5) adds examples that seems to lower the accuracy on the training set but performs better on the test set. It is not possible to make sound conclusions concerning the IR factor because over-sampling inserts the same positive samples in the training set, which does not increase the diversity. Applying over-sampling however is also not recommended because it has a bad recall.

Σ_2		No. TP		No. FP		No. TN		No. FN		*Acc*	
C	IR	Tr	T	Tr	T	Tr	T	Tr	T	Tr	T
0.1	0.8	378.0	4.0	9.0	44.0	293.0	182.0	0.0	2.0	0.987	0.802
0.1	1.0	302.0	4.0	9.0	44.0	293.0	182.0	0.0	2.0	0.985	0.802
0.1	1.2	252.0	4.0	9.0	44.0	293.0	182.0	0.0	2.0	0.984	0.802
1	0.8	378.0	4.0	8.0	46.0	294.0	180.0	0.0	2.0	0.988	0.793
1	1.0	302.0	4.0	8.0	46.0	294.0	180.0	0.0	2.0	0.987	0.793
1	1.2	252.0	4.0	8.0	46.0	294.0	180.0	0.0	2.0	0.986	0.793
10	0.8	349.0	2.0	3.0	35.0	299.0	191.0	29.0	4.0	0.953	0.832
10	1.0	276.6	2.0	3.0	35.0	299.0	191.0	25.4	4.0	0.953	0.832
10	1.2	232.3	2.0	3.0	35.0	299.0	191.0	19.7	4.0	0.959	0.832

Table 4.5: Classification results when applying over-sampling. In this table, C is the penalty value of the error term and TR and T represent the Training and test set respectively. IR is the imbalance ratio.

Rose-sampling (4.6) seems to combine the best of both previously sampled techniques: a large number of positive and negative samples and a higher recall compared to over-sampling. Therefore, in the remainder of this text rose-sampling will be used. The preliminary results for rose-sampling do not reveal a preference towards IR to be applied, nor the number of samples to use for training. However it is clear that the penalty parameter is preferred to be rather low (0.1 or 1.0) but not higher.

Σ_2			No. TP		No. FP		No. TN		No. FN		*Acc*	
C	IR	s	TR	T	TR	T	TR	T	TR	T	TR	T
0.1	1.2	1000	456.1	4.8	14.1	45.3	529.8	180.7	0.0	1.2	0.986	0.800
0.1	0.8	250	141.2	4.6	2.7	47.5	106.1	178.5	0.0	1.4	0.989	0.789
0.1	1.0	1000	499.6	4.5	13.5	45.3	486.9	180.7	0.0	1.5	0.987	0.798
0.1	1.0	500	253.1	4.4	6.0	45.0	240.9	181.0	0.0	1.6	0.988	0.799
0.1	0.8	500	279.0	4.4	5.9	44.0	215.1	182.0	0.0	1.6	0.988	0.803
0.1	0.8	1000	561.4	4.0	12.9	45.3	425.7	180.7	0.0	2.0	0.987	0.796
0.1	1.2	500	225.2	3.9	6.3	41.4	268.5	184.6	0.0	2.1	0.987	0.813
1	1.0	500	244.5	3.8	3.9	48.9	251.6	177.1	0.0	2.2	0.992	0.780
1	1.0	250	123.4	3.7	0.8	46.7	125.8	179.3	0.0	2.3	0.997	0.789
1	1.0	1000	504.3	3.6	8.3	48.2	487.4	177.8	0.0	2.4	0.992	0.782

Table 4.6: Classification results when applying rose-sampling. In this table, C is the penalty value of the error term and TR and T represent the Training and test set respectively. IR is the imbalance ratio.

4.5.2 Relevant feature selection

It would be interesting to select only those features that are relevant to the problem. Therefore relevant feature selection is implemented here. This technique allows to

select features in an automated way. Before discussing this, an analysis is made of how the classifier performs when using data from only one of the three feature groups.

Relevant feature group

The idea in this paragraph is to investigate the importance of the three feature groups from chapter 3. Classification will be carried out for each of the feature groups (using all features in the group) and as usual, the performance on the test and training set will be evaluated. Next to that, the importance (weight) of each of the features will be studied. To do so the assigned feature weights will be analysed. These weights are a measure for the significance of the feature when classifying and have a sign; a positive weight pushes samples towards the positive class (a true leak), while a negative weight pushes the sample towards the negative class (not a true leak).

The obtained classification results for Σ_2 are given in table 4.7. The displayed numbers are again the average of ten computations when applying rose-sampling with $IR = 0.8$ and $s = 1000$. An interesting finding is that the training performance is very good when using the shape and texture based features, but not when using temperature features. This is an interesting conclusion because it again points out the necessity for a 'smart algorithm' using more than temperature. Also interesting to note is that there is perfect recall when using shape or texture features. However the accuracy on the test set is below that obtained when using all features (see for example table 4.6).

No. TP		No. FP		No. TN		No. FN		*Acc*	
Temperature									
373.6	1.7	158.3	80.6	387.4	145.4	80.7	4.3	0.761	0.634
19.4	1.5	19.2	2.1	18.2	2.1	13.9	1.5	0.0136	0.0090
Shape									
453.1	6.0	44.3	77.2	498.3	148.8	4.3	0.0	0.951	0.667
17.1	0.0	4.6	3.4	20.1	3.4	1.6	0.0	0.0045	0.0148
Texture									
423.2	6.0	56.4	77.2	488.3	148.8	32.1	0.0	0.912	0.667
14.0	0.0	5.6	3.4	13.4	3.4	4.7	0.0	0.0065	0.0148

Table 4.7: Feature group influence on classification results. The first row are the mean values while the second contains the standard deviation. These results are obtained for Σ_2 when applying rose-sampling with $IR = 0.8$ and $s = 1000$.

In table 4.8 the feature weights for the trained SVC are listed. These are the weights corresponding to the computations as presented in table 4.7. When only classifying with the temperature group the weights are rather small. This corresponds with the lower training performance. For the shape and texture group there are larger

weights, which is related to the better training accuracy. The largest weights are found for h, w and z. These results can be interpreted as follows: w and z have a negative weight, indicating that when the blob is wider or located higher on the terrain it is less likely to be a true leak. The positive sign for h indicates that it is unlikely that very narrow leaks are true leaks. Looking at the texture features the most significant weights are correlation, dissimilarity, homogeneity and energy. Homogeneity has a negative sign indicating that when the image is more homogenous it is less likely to be a true leak. Such examples could be roads or concrete objects. The same applies for the correlation. Dissimilarity has a positive sign, which is also expected because dissimilarity strongly and negatively correlated with homogeneity. Energy has a positive weight, which is somewhat surprising because it is a measure of orderly (uniformity). The weight is however much smaller compared to the other before mentioned features.

Temperature							
	$\bar{T}$	T_Δ	T_{max}	T_{med}	T_{min}	σ_T	
	-0.012	0.021	0.017	-0.018	0.025	-0.041	
Shape							
A	h	w	ϵ	Θ	λ_1	λ_2	z
0.178	2.916	-3.984	-0.272	-0.311	0.002	-0.073	-0.894
Texture							
ASM	contrast	ρ	dissimilarity	energy	S	homogeneity	
0.117	-0.017	-3.998	2.288	0.982	0.096	-1.662	

Table 4.8: Feature weights per feature group. These results are obtained for Σ_2. For the used abbreviations see chapter 3. These results are obtained when applying rose-sampling with $IR = 0.8$ and $s = 1000$.

Based on the findings in table 4.7 and 4.8 all features will be considered as a starting point for classification in the remainder of this work. In what follows a technique called relevant feature selection will be introduced that tries to obtain the best set of features (from all three groups) in an automated way.

Relevant feature selection

Relevant feature selection is done by means of feature ranking with recursive feature elimination and cross-validation as guideline [GWBV02]. In the mentioned reference, the technique is applied on a very high dimensional data set (genes) for cancer classification but this applies to the problem considered in this work as well. The relevant features are detected by recursively eliminating features and validating the use of the reduced features by means of cross-validation. Different evaluation metrics were tested to guide the feature selection.

The results for different feature sets selected using this technique with different scoring metrics are given in table 4.9. Rose-sampling was performed to balance the data set. The results given in this table are the results on the test set. This set was not used during the feature selection. The results in the table are the best out of ten runs for each configuration. Twofold cross validation was used as guidance. This means that the training set was folded in a first set that is used for fitting the classifier, while the remainder of the set is used to evaluate the performance using the selected features. The results in this table are computed on Σ_1.

score metric	TP	FP	TN	FN
accuracy	5.3 ± 0.67	32.0 ± 6.29	258 ± 6.29	2.7 ± 0.67
adjusted random score	5.4 ± 1.17	35.0 ± 10.09	255 ± 10.09	2.6 ± 1.17
average precision	6.5 ± 0.97	38.4 ± 9.95	251.6 ± 9.95	1.5 ± 0.97
f1	5.2 ± 1.14	36.3 ± 10.13	253.7 ± 10.13	2.8 ± 1.14
mse	5.9 ± 0.74	33.9 ± 9.29	256.1 ± 9.29	2.1 ± 0.74
precision	5.3 ± 0.82	33.4 ± 11.25	256.6 ± 11.25	2.7 ± 0.82
R^2	5.5 ± 0.71	34.4 ± 9.47	255.6 ± 9.47	2.5 ± 0.71
recall	6.2 ± 1.23	38.4 ± 10.47	251.6 ± 10.47	1.8 ± 1.23
roc auc	5.7 ± 1.06	39.1 ± 9.41	250.9 ± 9.41	2.3 ± 1.06

Table 4.9: relevant feature selection: classification results on the test set for different scoring metrics (mean value $\pm$ standard deviation). Results derived from Σ_1.

Looking at the number of true positives it seems that there are two significant scoring metrics that perform well on the test set. The best results are obtained when using the average precision and recall to guide the feature selection. In both cases, seven out of eight positive examples were often correctly classified. In none of the cases a perfect classification of the true positives was obtained. When on the other hand the number of true negatives, false positives and false negatives considered the results are very similar.

Based on these findings only *recall* and *average precision* will be used as scoring metrics in the remainder of this work. The best results for both metrics were obtained with the following features:

- *average precision*: h, ASM, homogeneity, w and Θ
- *recall*: h, ASM, homogeneity, w, A, ρ and energy

Many of the selected features correspond with the findings of 4.9. Homogeneity, ρ, h, w and energy were found there as well. The relevant features for Σ_2 will not be discussed here but in the next paragraph that deals with adding artificial samples.

4.5.3 Artificial samples

Sampling as previously discussed allows to balance the data set, but does not add additional diversity to it. To increase the diversity new data is required. In the scope of this project it was however not possible to go on the field and collect additional images. Such attempts were made using a hand held camera, but the weather conditions and the quality of the camera did not allow making useful new samples (see 2.2.1). However, to have more samples one possibility is to create artificial examples based on the original samples as suggested in [MT12, Lee99]. The following was implemented: the detected blobs, together with the assigned true label, were modified in a random manner according to a predefined probability. This probability should not be too high because it could otherwise change the blob too much. The original blobs of course remained unmodified. Four modifications were implemented to occur conditionally independent of each other:

1. A pixel would be swapped with another pixel in the image (thus modifying both pixels)
2. A pixel would be changed to another pixel in the image (without modifying that pixel)[2]
3. The blob image would be scaled ($\pm 10\%$)
4. The blob image would be rotated ($\pm 5°$)

The performance on the data set using rose-sampling is given table 4.10. The results (again the average of 10 computations) are in favour of adding artificial samples and also suggest to use a penalty parameter set to unity, as this seems to perform better on the test set. The most important conclusion from this table is that the accuracy is already acceptable and that this is not at the cost of tolerating false negatives in the test set.

Σ_2		No. TP		No. FP		No. TN		No. FN		*Acc*	
C	IR	Tr	T	Tr	T	Tr	T	Tr	T	Tr	T
0.1	0.8	552.4	6.0	46.7	65.3	391.3	160.7	9.6	0.0	0.944	0.719
0.1	1.0	494.8	6.0	45.9	59.8	446.5	166.2	12.8	0.0	0.941	0.742
0.1	1.2	428.4	6.0	46.8	54.8	503.6	171.2	21.2	0.0	0.932	0.764
1	0.8	536.4	6.0	40.6	56.0	403.0	170.0	20.0	0.0	0.939	0.759
1	1.0	477.2	5.9	44.2	51.1	454.2	174.9	24.4	0.1	0.931	0.779
1	1.2	435.1	6.0	39.5	53.2	502.3	172.8	23.1	0.0	0.937	0.771

Table 4.10: Performance of adding artificial samples

[2] These first two modifications are actually based on operations used in genetic algorithms [Wil08].

The use of artificial samples applied to this problem is examined while also including automatic feature selection. Two score metrics were selected to guide the feature selection: *average precision* and *recall*. Based on the originally derived blobs 250 artificial positive samples were generated. For the negative class, for which there are much more examples, 1000 artificial samples were created. The probability in image that a pixel would be modified was 25%. In this way the original training set was extended by almost 200%. After adding the artificial samples, rose-sampling was applied to generate a training set that has an imbalance ratio of about 0.6 and keeping 500 samples. The performance on the test set is given in table 4.11.

score metric	TP	FP	TN	FN
recall	6.1±1.23	36.2±10.91	253.8±10.91	1.9±1.23
ave. prec.	6.3±1.03	32.5±7.18	257.5±7.18	1.7±1.03

Table 4.11: relevant feature selection: classification results on the test set for different scoring metrics after adding artificial samples (mean value ± standard deviation). These results were obtained for Σ_1. ave. prec. is the abbreviation for average precision.

Comparing these findings with the results when not using artificial samples the following becomes clear. The number of true positives has decreased a little bit with almost similar spread (standard deviation). The number of false positives on the other hand seems to have decreased in average by two samples. The number of true negatives has changed up to six samples and the number of false negatives has (consequently) decreased a little bit. Considering the positive samples not much improvement seems to be have been obtained. Considering the negative samples a decent improvement of in average 5 samples can obtained.

4.5.4 Ensemble learner

An ensemble learner is a system that uses different models when predicting the output class [BK99, Die00]. In this thesis this means that more than one classifier will be used to predict if a blob is a leak or not. The general idea is that many weak(er) classes together will classify better than one strong classifier alone. Many implementations of ensemble learning exist, but here bagging [SW10] will be considered.

In bagging, a number of bags of training samples are made. The samples are selected from all available training data according to a uniform distribution. For each of the bags a classifier is trained. These classifiers are then combined to give one final prediction. Majority voting is a very common way to make the final prediction. This is implemented here as well.

The following test has been carried out for Σ_1. Analogous to the previous sections a training set of size 500 is created using rose-sampling with an imbalance ratio of 0.6. For each of these training sets relevant feature selection (with *recall* as guidance)

is carried out and a classifier is trained. This is done fifteen times. These fifteen classifiers are saved and used to produce a majority vote. This means that the most frequent output of the individual classifiers is suggested as the ensemble prediction. The results on the training and test set are given below in table 4.12.

TP	FP	TN	FN
7	33	257	1

Table 4.12: Ensemble learning with relevant feature selection: classification results on the test and training set with *recall* as scoring metric and after adding artificial samples. These results were obtained for Σ_1.

The accuracy on the test set indeed improved up to 88.60% but this was at cost of tolerating one false negative. Using an ensemble learner did not manage to keep the number of true positives maximal and decreasing the number of false positives.

4.6 Best results obtained

The best results obtained on Σ_2 are given in table 4.13. It is clear that the obtained results are much better compared to those obtained in section 4.4. The recall is much better and at times even perfect on the training set. Best results were systematically obtained when using $C = 1.0$ and $IR = 1.2$. Interestingly the presented results were obtained using all features. Thus automatic feature selection did not remove any of the features. There were however classifiers that only used a couple of features but the accuracy was always worse than listed here. Training an ensemble classifier with these data did not improve the classification. This is because the classifiers here are not weak learners anymore and they systematically misclassified the same samples (see next paragraph). The best results obtained on Σ_2 are listed in the previous sections.

Σ_2		No. TP		No. FP		No. TN		No. FN		*Acc*	
C	IR	Tr	T	Tr	T	Tr	T	Tr	T	Tr	T
1	1.2	199	5	23	41	256	185	22	1	0.910	0.819
1	1.2	223	5	35	43	218	183	24	1	0.882	0.810
1	1.2	193	5	20	45	268	181	19	1	0.922	0.802
1	1.2	204	6	31	46	251	180	14	0	0.910	0.802
10	1.0	262	6	21	46	210	180	7	0	0.944	0.802

Table 4.13: Top five results obtained on Σ_2.

4.7 Analysis of the classification result

In this section a short analysis will be made of the classification results. A first interesting question is which of the blobs in the test set is detected as a leak. Is it

systematically the same blob that is not found and if so, why is it not recognised while the other leaks are correctly classified. A second question of interest is why some innocent blobs are correctly classified as not being a leak, while other are recognised as such. Based on this analysis further actions may be considered. It might for example be recommended to add an additional feature. In any way special attention should be paid to the problem of over-fitting. Adding an additional feature may help to improve the classification on both the training and test set, but it may be too specialised on unseen data.

When looking at the results of the ensemble classifier the following conclusion could be drawn. Each of the fifteen models in the ensemble learners systematically missed out one of the known leaks. This leak (centred in the image) is shown in figure 4.1. It is clear from this figure that that the thermal infrared image is incomplete at the location of the image (see red circle where part of the thermal infrared map was not made). The pattern of the leak if very likely not complete and as a result misclassification takes place.

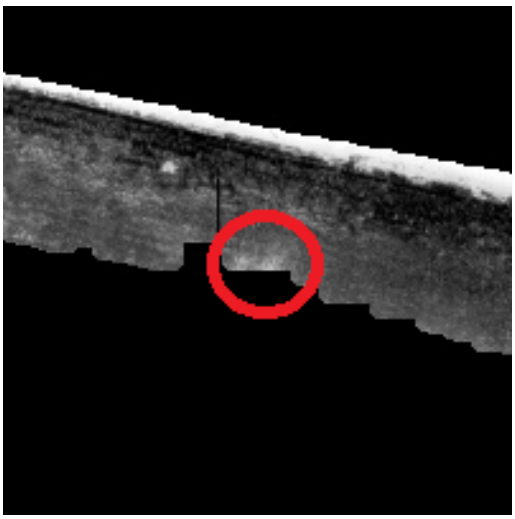
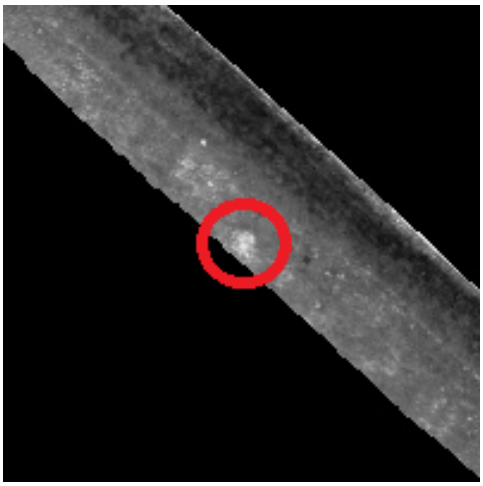


Figure 4.1: Blob systematically and incorrectly classified as negative while actually positive. Misclassification probably takes place because the image is incomplete

When looking at the false positives it was remarkable that each of the models in the ensemble classifier also systematically misclassified some of the blobs. Two examples of such false positives are given in figure 4.2. For the blob in 4.2a it is difficult to tell if the small blob in the centre corresponds to a leak. There indeed seems to be an intensity contrast with the levee body around it but it seems to be very small and most likely it is incorrectly classified as a leak. The second leak shown in 4.2b on the other hand clearly seems to be a leak. The zone is much larger and there also seems to be a very distinct difference with the surrounding pixels. Here he question might very well be if the true label is not incorrect.

(a) Blob classified as a leak, while not corresponding to an observed leak (centre of the picture, not to be mistaken with the blob more to the upper right.



(b) Blob classified as a leak, while not corresponding to an observed leak. Possibly a true true leak.

Figure 4.2: Examples of misclassified blobs as false positives.

4.8 Conclusions and final thoughts

From these results, it is clear that a decent classification system can be learned using support vector machines. A very good training accuracy could be obtained (94.6%) even with a linear kernel. The performance on the test set was a little less but still high and perfect recall could be obtained in most cases. This indicates that the set of features implemented is representative for the difference between leaks and area that does not correspond to leaks. The set of features even seems to be too large, as many of them are not used when relevance feature selection is applied.

It should be mentioned that a linear kernel was chosen here not to over-fit the data. When more complex kernels are used (such as rbf or polynomial; see [SS01, CST00] for an extended overview) more complex decision boundaries can be derived that allow a better classification. When using these kinds of kernels one should be very careful not to over-fit the data.

The analysis of the misclassified blobs (false positives and false negatives) revealed two interesting findings. First of all, perfect recall was not obtained for Σ_1. This misclassification always belonged (in ensemble learning all classifiers would be misled at all times) to the same blob, while all other images would be classified correctly. The reason for misclassification might be that the full pattern is not visible because the image is incomplete at that location, rather than the classifier not being able to derive the leak if it would have been completely visible.

A second finding is that some of the leaks treated as false positives are very likely true positives. However because they were not indicated by the levee guard as such and never noticed during any of the field trips, these were not labelled as true leaks. Again, the cost of having levee failure because of having missed out one leak is nothing compared to having investigated many that were false alarms.

As a final thought to conclude this chapter. Because the images in Walem could not be used (changed conditions and different set-up (car on top of the levee)) only images in Hingene could be used. Two different training and testing sets were created which allows examining generalisation on unseen data (Σ_2) and on data from the same area at a slightly different time and camera angle. However, the amount of data is still very limited and more data is required to validate how well this technique allows generalisation on unseen data.

Chapter 5

Conclusion

A three-step approach is suggested and implemented to solve the problem of detecting seepage (leakages) in river levees based on thermal infrared images. In a first step, blobs are detected using MSER in the pre-processed thermal infrared images. Using some pre- and post-processing techniques a number of blobs can easily be removed without risking to throw away a real leak. All known leaks could be detected.

In the second step a set of features are derived for each of the blobs. These features are used in the third step, which is training a classifier to distinguish between real leaks and other areas that are not indicated as leakages. From the final results it is clear that the set of implemented features is able to describe the variation in the images and allows a good separation.

In the third step the features of step two are used to train a support vector classifier with a linear kernel. A linear kernel was selected rather than a more complex kernel in order not to over-fit the available data.

Different measures were implemented to deal with the lack of available data and especially the imbalance between the number of blobs that correspond with a leak and those that do not . First of all a set of sampling techniques was implemented to balance the data set. In order to increase the diversity in the data set artificial samples were created based on the derived leaks. Both techniques seem to improve the classification results on the data set. A third point of improvement was to implement automatic feature selection based on cross-fold validation with recall as a guideline, rather than accuracy. A last improvement made to the classification result was to build an ensemble classifier that uses a number of support vector machines to predict the outcome of a blob. Majority voting was used to make the final prediction.

Experimental results have shown that in the best cases a training accuracy of almost 95% can be obtained and almost 88% on the test set for σ_1. Analysis of the classification results of Σ_1 revealed that a perfect recall was very difficult to obtain because one blob corresponding to a known leak is systematically misclassified. Presumable this is because the image is incomplete at the location of that leak. Nevertheless, a classifier was obtained with perfect recall. The increase in recall was

however at the cost of tolerating extra false positives. Analysis of the classification results also revealed that some of the blobs classified as false positives, might actually correspond to true leaks that were not labelled as such.

The analysis performed in this report is done based on data obtained by a technique that is yet to be perfected. The tool reported here was created in such a way that it can easily be used again on future and improved data. It should be mentioned explicitly that the possibility of using a nonlinear kernel is readily available. Using such a non-linear kernel allows much more complex classifiers. However, using more complex kernels, one should at all times be aware of over-fitting.

Considering again the original objectives as presented in 1.2 the following can be said:

- *It should run automatically and with as little user interaction as possible, in order to allow a reproducible and objective output*: The algorithm does not require any user input. The only input from the user is the thermal infrared image data (currently as a text file). From that point on everything happens automatically. Once a suitable support vector classifier is found it can be used on unseen data. The output will be the same at all time and is thus reproducible and objective because the user does not need to make any decisions. It should be noted that during the training phase of the classifier supervision is still required (labelling of the training and test data).
- *It should be self-learning so that it can easily be applied on new data. This is necessary because the image acquisition technique is currently still a point of research*: The library that was created exactly allows to re-train the support vector classifier without actually requiring the user to do anything (only providing a text file). When more data becomes available from future measurements this data can thus very easily be used. Not only can it be classified by the previously trained classifier, it can also be used to re-train the classifier and make it more robust and improve its generalisation on unseen data.
- It should be able to cope with the imbalance and lack of variation in the available data (a lot of examples of what is not a leak, but only few of what is truly a leak): a number of strategies were implemented to cope with the imbalance and lack of variation (sampling, adding artificial samples and ensemble learning). Results have shown that using these strategies the imbalance can be countered and diversity can be increased.

It is thus clear that the objectives were met for the available data. It is however important to realise that the performance on unseen data that differs even only slightly might cause the classifier to become unreliable. More data will be required to see how the blob detection algorithm and the classifier perform on this kind of data. Therefore, the choice for a self-learning system was perfect.

Chapter 6

Future perspectives

The results presented in this work indicate that this tool has potential to be used in many settings. There are still improvements that can and need to be made. Both to the image acquisition technique and the algorithms presented in this work. These will be discussed here.

In [WVVH+13] it was mentioned that the individual components for making much better thermal infrared images already exist nowadays. The challenge is to combine the most suitable camera, with the most suitable drone, while having the best data acquisition, and so on at an acceptable economic cost. However, it seems very likely that it will not take much longer before all this will be possible at an acceptable cost. One other issue, especially when using drones, is the lack of legislation. This is without any doubt a milestone in the full-scale application of this technique.

While waiting for the perfect technical match of individual components and the legislative framework, the following is already possible today: the data used now had a spatial resolution of 10 by 10 cm. This resolution was chosen to reduce the data mass. The spatial resolution could at the time of writing already have been much higher. It is technically possible to produce images with a resolution of 2 x 2 cm. This means that one pixel now can be refined into 25 pixels. This increase in detail will very likely allow an even better classification, but at the cost of processing time. Already now for the smaller data set a lot of computational problems had to be dealt with. There were some problems with memory that could not be allocated. Therefore the image had to be split, taking in consideration a certain overlap to be certain not to miss out any blobs. This issue can at first be solved by extending the systems memory, but with increasing data sets there inevitable will be a moment where this cannot be achieved anymore at an acceptable cost. One possible approach could then be to write a wrapper that can deal with these large images and split them up automatically. This extension should be easy to achieve with openCV because this library even supports movies that are then cut into frames.

Coupling with visual images could aid and further improve the removal of false positives. This could be based, for example, on Google maps images or on visual

images taken together with the TIR images. The first has the advantage of always being available and the disadvantage of not corresponding completely with the actual situation on the moment the TIR were made (for example, animals will not be on the same spot, garbage might be removed/added, grass might be longer, ...). The latter has the advantage of completely corresponding to the TIR but cannot be made during dark conditions when precisely the TIR are most useful (no influence of the sun). It might also be interesting to investigate the information gain when using other or multiple spectral bands next to the infrared thermal image [MK11].

The algorithm in this thesis is designed as a support tool for the levee guards. As could be seen from the examples used to train the classifier in this work, there are multiple kinds of leaks. A first kind are spread over a large distance and is sometimes more than 4 or 5 meter wide. Others are more concentrated on a smaller distance. It might therefore be interesting to introduce different classes of suspicious blobs and train a multiclass classifier. These classes could then be used to prioritise which leaks should be investigated first. Anyway, this is an interesting exercise that should be made together with the responsible government officials. Also not discussed before is the possibility to assign probabilities to the classification. This would give the responsible authorities a less strict classification (true/false) but rather a probability that leaks belong to a class. Such framework can easily be implemented for example by computing the distance to the decision hyper plane. Examples of such classification can be found in [SGB+02].

Next to the possibilities that future (and present) developments can bring to this detection and classification technique there are different environments in which the algorithm could be applied. Obvious implementations would be to train the algorithm on data gathered for concrete levees or other hydraulic structures such as dams or concrete embankments. These have the advantage of not being covered by vegetation as is the case for most of the Flemish river levees. When considering the example of a dam it has the further advantage of having a (more) steady water level that makes the time scope in which images can be made much wider compared to the tidal rivers in this thesis. Also, this technique lends itself very much in warm countries where the thermal contrast can be significantly larger and visible for a longer period of time.

To end this list of recommendations and possibilities a last and maybe very futuristic and optimistic thought is given. In the Netherlands, the idea of using infrared images has already been considered and tested in the mid-nineties and even more recently in 2010. Infrared images were made from an F16 flight plane that was equipped with an infrared camera to detect land mines. The Dutch air force flew during a very critical moment (very high water levels) and took images of more than 200 kilometre levee. They thus saw thermal infrared imagery as a tool to get a very fast indication of problems in critical situations. Very fast in the sense that it would be impossible to cover a large distance along a levee with only few levee guards. In hazardous situations one cannot wait for someone to go on site. It would therefore be ideal

that in critical situations a tool could access the data on-line and process it as the plane/drone/satellite as it moves. This would give an instantaneous indication of (new) problems such that immediate action can be undertaken. An example in given in [RDB08] where robust online object learning and recognition by MSER tracking is implemented on a standard computer.

Bibliography

[AHP14] Julio Cesar L. Alves, Claudete B. Henriques, and Ronei J. Poppi. Classification of diesel pool refinery streams through near infrared spectroscopy and support vector machines using c-svc and v-svc. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 117(0):389 – 396, 2014.

[BG94] Michael Buckland and Fredric Gey. The relationship between recall and precision. *J. Am. Soc. Inf. Sci.*, 45(1):12–19, January 1994.

[BK99] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.*, 36(1-2):105–139, July 1999.

[BKT14] A. Bria, N. Karssemeijer, and F. Tortorella. Learning from unbalanced data: A cascade-based approach for detecting clustered microcalcifications. *Medical Image Analysis*, 18(2):241 – 252, 2014.

[Bra00] G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[Bra13] BethanyA. Bradley. Remote detection of invasive plants: a review of spectral, textural and phenological approaches. *Biological Invasions*, pages 1–15, 2013.

[Cha05] NiteshV. Chawla. Data mining for imbalanced datasets: An overview. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 853–867. Springer US, 2005.

[CJK04] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6(1):1–6, June 2004.

[CR04] C. A. Coburn and A. C. B. Roberts. A multiscale texture analysis procedure for improved forest stand classification. *International Journal of Remote Sensing*, 25(20):4287–4308, 2004.

[CST00] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 1 edition, 2000.

[CV13] Open CV. Image moments, 2013. [Online; accessed 23-Dec-2013].

[DBW06] Michael Donoser, Horst Bischof, and Mario Wiltsche. Color blob segmentation by mser analysis. In *In Proceedings of IEEE International Conference on Image Processing*, pages 757–760, 2006.

[Die00] ThomasG. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[DK01] Georges Dupret and Masato Koda. Bootstrap re-sampling for unbalanced data in supervised learning. *European Journal of Operational Research*, 134(1):141–156, 2001.

[DPCB11] Andrea Dal Pozzolo, Olivier Caelen, and Gianluca Bontempi. Comparison of balancing techniques for unbalanced datasets, 2011.

[FHA95] S.J. Fraikin, R. Hartman, and Reijneveld A. De inzet van remote sensing en fotogrammetrie bij wateroverlast, 1995.

[For07] Per-Erik Forssen. Maximally stable colour regions for recognition and matching. In *CVPR*. IEEE Computer Society, 2007.

[FR94] S.J. Fraikin and A. Reijneveld. *De inzet van Remote Sensing en Fotogrammetrie bij Wateroverlast: hoe verder? : verslag van de workshop gehouden op 9 februari 1994 te Delft*. Rijkswaterstaat, 1994.

[GVH+02] M. Givehchi, J.K. Vrijling, A. Hartmann, P.H.A.J.M. van Gelder, and S. van Baars. Application of remotely sensed data for detection of seepage in dikes. http://wwwde.uni.lu/media/files/pdf33, 2002.

[GWBV02] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[HB07] Mryka Hall-Beyer. The glcm tutorial home page. http://http://www.fp.ucalgary.ca/mhallbey/tutorial.htm, February 2007.

[HSD73] Robert M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6):610–621, November 1973.

[Jap00] Nathalie Japkowicz. Learning from imbalanced data sets: A comparison of various strategies, 2000.

[JT12] Mohd Shawal Jadin and Soib Taib. Recent progress in diagnosing the reliability of electrical equipment by using infrared thermography. *Infrared Physics and Technology*, 55(4):236 – 245, 2012.

[Lee99] Sauchi Stephen Lee. Regularization in skewed binary classification. *Computational Statistics*, 14(2):277–292, 1999.

[LFG+13] Victoria Lopez, Alberto Fernandez, Salvador Garcia, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250(0):113 – 141, 2013.

[LKC08] T.M. Lillesand, R.W. Kiefer, and J.W. Chipman. *Remote sensing and image interpretation*. John Wiley & Sons, 2008.

[LSP03] S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Computer Vision Proceedings. Ninth IEEE International Conference on*, volume 1, pages 649–655, 2003.

[MCMP02] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393, London, 2002.

[MK11] P. Mather and M. Koch. *Computer Processing of Remotely-Sensed Images: An Introduction*. Wiley, 2011.

[Moh08] Ralf Mohnen. Massa-evacuaties bij overstromingen in nederland. hoe goed is nederland voorbereid? Master's thesis, Nederlands instituut fysieke veiligheid, 2008.

[MS02] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, pages 128–142, London, UK, UK, 2002. Springer-Verlag.

[MS04] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[MT12] Giovanna Menardi and Nicola Torelli. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, pages 1–31, 2012.

[MTS+05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[MZW05] Kate McCarthy, Bibi Zabar, and Gary Weiss. Does cost-sensitive learning beat sampling for classifying rare classes? In *Proceedings of*

the 1st International Workshop on Utility-based Data Mining, UBDM '05, pages 69–77, New York, NY, USA, 2005. ACM.

[PDK+07] T.Maruthi Padmaja, Narendra Dhulipalla, P.Radha Krishna, RajuS. Bapi, and A. Laha. An unbalanced data classification model using hybrid sampling technique for fraud detection. In Ashish Ghosh, RajatK. De, and SankarK. Pal, editors, *Pattern Recognition and Machine Intelligence*, volume 4815 of *Lecture Notes in Computer Science*, pages 341–348. Springer Berlin Heidelberg, 2007.

[Pea00] L. G. Pearlstine. Discrimination of an invasive plant, schinus terebinthifolius, from aerial digital imagery. Master's thesis, University of Florida, 2000.

[Phi07] Susan Philip. Active fire detection using remote sensing based polor-orbiting and geostationary observations: an approach towards near real-time fire monitoring. Master's thesis, International Institute for Geo-Information Science and Earth Observation, 2007.

[PK00] D. Pluym and W.D. Keij. Organisation of sar-based deformation measurements on river dikes, 2000.

[PN83] L.F. Pau and M.Y. El Nahas. An introduction to infrared image acquisition and classification systems. *Optics and Laser Technology*, 16(5):274 – 275, 1983.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[QTS14] Zhiquan Qi, Yingjie Tian, and Yong Shi. A nonparallel support vector machine for a classification problem with universum learning. *Journal of Computational and Applied Mathematics*, 263(0):288 – 298, 2014.

[RDB08] H. Riemenschneider, M. Donoser, and H. Bischof. Online object recognition by mser trajectories. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, 2008.

[SGB+02] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[SGD02] Shoaib Sehgal, Iqbal Gondal, and Laurence Dooley. Statistical neural networks and support vector machine for the classification of genetic mutations in ovarian cancer. In *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB?04)*, November 2002. ISBN: 0-7803-8728-7.

[SH00] Rosa A. Schiavo and David J. Hand. Ten more years of error rate research. *International Statistical Review*, 68(3):295–310, 2000.

[Slo85] S.W. Sloan. A point-in-polygon program. *Advances in Engineering Software (1978)*, 7(1):45 – 47, 1985.

[SS01] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001.

[SW10] Claude Sammut and GeoffreyI. Webb. Bagging. In Claude Sammut and GeoffreyI. Webb, editors, *Encyclopedia of Machine Learning*, pages 73–73. Springer US, 2010.

[TG00] Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *In Proc. BMVC*, pages 412–425, 2000.

[TM08] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008.

[VHHPM10] Thomas Van Hoestenberghe, Marc Huygens, Peeters Patrik, and Frank Mostaert. Opstellen bresgroeiparameters vlaamse rivierdijken: Deelopdracht 1. literatuurstudie bresgroeiproces. Technical Report 706 08c, Waterbouwkundig Laboratorium en Soresma, Berchemlei 115, 2140 Antwerpen, Belgium, 2010.

[Wik13] Wikipedia. Maximally stable extremal regions, 2013. [Online; accessed 3-Jan-2014].

[Wil08] K. Wildemeersch. Combining genetic algorithms and boundary elements to optimize coastal aquifers' management using sheet pile walls. Master's thesis, University of Gent, 2008.

[WLYC14] Xiang-Yang Wang, Yong-Wei Li, Hong-Ying Yang, and Jing-Wei Chen. An image retrieval scheme with relevance feedback using feature reconstruction and {SVM} reclassification. *Neurocomputing*, 127(0):214 – 230, 2014. ¡ce:title¿Advances in Intelligent Systems¡/ce:title¿ ¡ce:subtitle¿Selected papers from the 2012 Brazilian Symposium on Neural Networks (SBRN 2012)¡/ce:subtitle¿.

[Wre06] David Wretman. Finding regions of interest in a decision support system for analysis of infrared images. Master's thesis, Royal Institute of Technology, 2006. ISSN-1653-5715.

[WSL+13] Michael J. Wilber, Walter J. Scheirer, Phil Leitner, Brian Heflin, James Zott, Daniel Reinke, David K. Delaney, and Terrance E. Boult.

Animal recognition in the mojave desert: Vision tools for field biologists. *Applications of Computer Vision, IEEE Workshop on*, 0:206–213, 2013.

[WVVH+13] K. Wildemeersch, K. P. Visser, T. Van Hoestenberghe, P. Peeters, and F. Mostaert. Evaluatie (potentiele) dijkmonitoringstechnieken. detectie kwel/lekkage onder en/of doorheen dijken. Technical report, Waterbouwkundig Laboratorium, 2013.

[ZBRV04] Jianping Zhang, E. Bloedorn, L. Rosen, and D. Venese. Learning rules from highly unbalanced data sets. In *Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on*, pages 571–574, 2004.

Master thesis filing card

Student: Ir. Koen Wildemeersch

Title: Smart detection of seepage in river dikes based on thermal infrared images

Dutch title: 'Slimme detectie van kwel doorheen rivierdijken op basis van infrarood beelden'

UDC: 621.3

Abstract:
In this thesis an algorithm is developed to detect seepage in river levees based on thermal infrared images. From previous research it is known that thermal contrast (a cold levee body and warmer water leaking through the levee for example) can be a good, fast and non destructive indicator of seepage which may eventually lead to levee failure. Early detection is essential in good levee management and this algorithm is developed to be a practical tool that can simplify and help the levee guards in keeping levees safe. A supervised learning approach is chosen which allows this tool to continuously improve on new training and testing data. Such an approach is feasible because the image acquisition itself still needs to be improved and only few training and testing data was available at the time of research. In a first step, suspicious areas (blobs) are detected in the thermal infrared image. To do so the image is preprocessed and the contours of suspicious areas are derived using maximally stable extremal regions. As a result of this step a collection of blobs becomes available. Using simple post processing some of the blobs can be removed without much computational effort. In a second step, a set of features is derived that represent each of these blobs. Three kinds of features are implemented. A first kind is based on temperature, a second kind is based on shape and a third is based on texture. These features are used to feed a classifier that separates the leaks from objects, image imperfections and other blobs that do not correspond to a leak. To decide on which features are best to be used, a recursive feature elimination algorithm guided by cross validation is implemented. It is inherent to this problem that there is an imbalance between the number of blobs that are actually a leak and those that are not. This impedes the classification task at hand. To deal with this, different sampling techniques are implemented. Next to the imbalance in the data set, there is not much diversity in the set of true leaks because only few examples are available. Therefore artificial samples are generated based on the derived blobs in step 1. In a third step a support vector classifier with linear kernel is implemented to perform the actual classification. Using a linear kernel has the disadvantage of only allowing relatively simple decision boundaries but the advantage of less likely over-fitting the data. Excellent classification results were obtained on the training set, with slightly less accurate results on a test set which was not used while training the classifier.

Thesis submitted for the degree of Master of Science in Artificial Intelligence, option Engineering and Computer Science

Thesis supervisor: Prof. Dr. Ir. Dirk Vandermeulen and Prof. Dr. Ir. Tinne Tuytelaars
Assessor: Prof. Dr. Ir. Jaak Monbaliu and Dr. Ir. Marco Pedersoli
Mentor: