

Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Rhino and human detection in overlapping RGB and LWIR images

Examensarbete utfört i Bildbehandling
vid Tekniska högskolan vid Linköpings universitet
av


Carl Karlsson Schmidt

LiTH-ISY-EX--15/4837--SE

Linköping 2015



Linköpings universitet
TEKNISKA HÖGSKOLAN

		Avdelning, Institution Division, Department CVL Department of Electrical Engineering SE-581 83 Linköping		Datum Date 2015-06-05	
Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____		Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____		ISBN _____ ISRN LiTH-ISY-EX--15/4837--SE Serietitel och serienummer Title of series, numbering _____	
URL för elektronisk version http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-120323					
Titel Title Författare Author		Noshörnings- och människodetektion i överlappande färg- och LVIR-bilder Rhino and human detection in overlapping RGB and LWIR images Carl Karlsson Schmidt			
Sammanfattning Abstract <p>The poaching of rhinoceros has increased dramatically the last few years and the park rangers are often helpless against the militarised poachers. Linköping University is running several projects with the goal to aid the park rangers in their work.</p> <p>This master thesis was produced at CybAero AB, which builds Remotely Piloted Aircraft System (RPAS). With their helicopters, high end cameras with a range sufficient to cover the whole area can be flown over the parks.</p> <p>The aim of this thesis is to investigate different methods to automatically find rhinos and humans, using airborne cameras. The system uses two cameras, one colour camera and one thermal camera. The latter is used to find interesting objects which are then extracted in the colour image. The object is then classified as either rhino, human or other. Several methods for classification have been evaluated.</p> <p>The results show that classifying solely on the thermal image gives nearly as high accuracy as classifying only in combination with the colour image. This enables the system to be used in dusk and dawn or in bad light conditions. This is an important factor since most poaching occurs at dusk or dawn. As a conclusion a system capable of running on low performance hardware and placeable on board the aircraft is presented.</p>					
Nyckelord Keywords		Machine Learning, Feature Descriptor, Thermal, IR, LWIR, Rhino, Human, Poaching			

Sammanfattning

Tjuvjakten av noshörningar har ökat drastiskt de senaste åren och parkvakterna står ofta handfallna mot militariserade tjuvjägare. Linköpings Universitet arbetar på flera projekt som på olika sätt ska vara ett stöd för parkvakterna i deras arbete.

Examensarbetet genomfördes på CybAero AB som jobbar med att bygga fjärrstyrda helikoptrar, så kallade RPAS (Remotely Piloted Aircraft System). Med deras system kan man bära högkvalitativa kameror och ha stor räckvidd så hela parken kan övervakas.

Det här examensarbetet syftar på att undersöka olika metoder för att från luftburna kameror kunna ge information om vad som pågår i parken. Systemet bygger på att man har två kameror, en vanlig färgkamera och en värmekamera. Värmekameran används för att hitta intressanta objekt som sedan plockas ut ur färgbilden. Objektet klassificeras sedan som antingen noshörningar, människor eller annat. Flertalet metoder har utvärderats utefter deras förmåga att klassificera objekten korrekt.

Det visade sig att man kan få väldigt bra resultat när man klassificerar endast på värmebilden vilket ger systemet möjlighet att operera även när det är skymning eller mörkt ute. Det är en väldigt viktig del då de flesta djuren skjuts vid antingen gryning eller skymning. Som slutsats i rapporten presenteras ett förslag på system som kan köras på lågpresterande hårdvara för att kunna köras direkt i luften.

Abstract

The poaching of rhinoceros has increased dramatically the last few years and the park rangers are often helpless against the militarised poachers. Linköping University is running several projects with the goal to aid the park rangers in their work.

This master thesis was produced at CybAero AB, which builds Remotely Piloted Aircraft System (RPAS). With their helicopters, high end cameras with a range sufficient to cover the whole area can be flown over the parks.

The aim of this thesis is to investigate different methods to automatically find rhinos and humans, using airborne cameras. The system uses two cameras, one colour camera and one thermal camera. The latter is used to find interesting objects which are then extracted in the colour image. The object is then classified as either rhino, human or other. Several methods for classification have been evaluated.

The results show that classifying solely on the thermal image gives nearly as high accuracy as classifying only in combination with the colour image. This enables the system to be used in dusk and dawn or in bad light conditions. This is an important factor since most poaching occurs at dusk or dawn. As a conclusion a system capable of running on low performance hardware and placeable on board the aircraft is presented.

Acknowledgments

I want to thank CybAero for giving the opportunity to make this thesis on this incredibly important subject. Also, a big thanks to my supervisor at LiU, Kristoffer Öfjäll, who have been guiding me throughtout this work.

Finally, I would like to thank my family and friends for supporting me, and Linus for lighting up my way.

Contents

1	Introduction	1
1.1	Background	1
1.2	Problem Description	1
1.3	Goals/Questions	2
1.4	Limitations	3
2	Theory	5
2.1	Infrared Radiation	5
2.2	Image Registration	7
2.3	Image Segmentation	7
2.4	Feature Descriptors	8
2.4.1	Local Binary Pattern	9
2.4.2	Histogram of Oriented Gradients	10
2.5	Machine Learning and Classification	11
2.5.1	Support Vector Machine	12
2.5.2	Decision Trees and Random Forests	15
2.5.3	k -Nearest Neighbours	18
2.5.4	Training and Evaluation	18
3	Method	21
3.1	Data sets	21
3.2	Implementation	23
3.2.1	Programming Language	23
3.2.2	Preprocessing	24
3.2.3	Image Segmentation	25
3.2.4	Image Registration	26
3.2.5	Training and Evaluation Data	26
3.3	Evaluation	27
4	Results	31
4.1	Overview/Presentation	31
4.2	Classifiers	32
4.3	Feature Descriptors	33

4.3.1	LBP and its variants	34
4.3.2	HOG	34
5	Conclusion and Future Work	41
5.1	Conclusion	41
5.2	Future Work	42
5.2.1	Image Registration	42
5.2.2	Image Segmentation	43
5.2.3	Tracking	43
5.2.4	Choosing Sensor	43
A	Appendix	45
A.1	Image segmentation	45
A.2	Classifiers and Feature Descriptors	45
	Bibliography	51

1

Introduction

1.1 Background

The poaching of rhinoceros in Africa has been increasing dramatically the last couple of years which threatens several species of rhinos with extinction. With more and more well armed poachers, often with automatic rifles, night vision and even helicopters, the park rangers needs more advanced technology to be able to protect the hunted animals in the national parks.

There are several projects running worldwide, ranging from educating the public to giving the park rangers the necessary means to stop the poaching while it is happening.

This thesis analyses the possibility to detect and classify animals and humans from a setup with a thermal and a RGB camera. This could be used on board a RPAS (Remotely Piloted Aircraft System), which would give the rangers a fast and accurate method of surveilling a larger area even in the night thanks to the thermal vision. The proposed system is however not limited to usage on the savannah. Any application which needs to automatically find and detect humans or animals could use it. After a catastrophic event such as the tsunami 2003 a RPAS which could automatically find and report the location of humans would have helped the rescue action enormously.

1.2 Problem Description

From two overlapping images, one thermal and one RGB, interesting objects should be located and classified as either humans, rhinos or other. This problem is split into four independent problems, denoted P1-P4 below, which will all

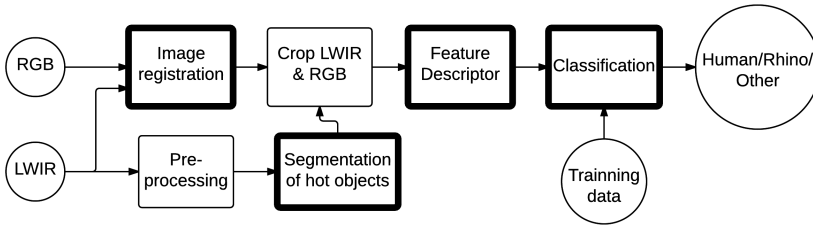


Figure 1.2.1: Overview of the complete system

be described in detail in this thesis. They can be seen in Figure 1.2.1 as the bold squares.

P1: Segmentation of hot Objects The first step of the system is to do a segmentation of cold and hot objects to create a mask for cropping the RGB image. This is done by first using a method of enhancing warm (bright) parts of the image then using an algorithm for calculating an optimal discriminating threshold value.

P2: Image Registration The two cameras neither share the same camera centre nor the same resolution. To be able to use the segmentation mask created in the previous step the two images first needs to be aligned. This is done by approximating a homography between the two cameras.

P3: Feature Descriptors To be able to automatically classify objects in the images is we need to extract interesting features of the object. This is often based on edges, angle of the edges and corners. There are a lot of methods for calculating features, the ones evaluated here have achieved good result in pedestrian detection which makes is plausible that they will also perform well on animals.

P4: Classification The classification task is, from prior knowledge, to determine which discrete category, or class, a new measurement belongs to. In this thesis it is to classify the data from the feature descriptors into *rhino*, *human* or *other*.

1.3 Goals/Questions

- Which combination of feature descriptors and classifiers have the highest performance?
- Does the combination of RGB and thermal images increase the classification performance?
- How well does the system perform with only the thermal camera?
- Consideration for running the system on low performance hardware.

1.4 Limitations

To be able to focus on the relevant parts of the problem, the feature descriptors and classification, some limitations and assumptions have been made.

- **Three classes**

There will only be three classes; *rhino*, *human* and *other*. The *other*-class is a class where all objects which is neither humans nor rhinos are placed, ranging from rocks and trees to zebras and gnus.

- **Constant flight height, fixed cameras**

The cameras should not be able to move independently of each other and the distance to the ground plane should be approximately constant. This limitation enables using a constant image registration over each sequence and it is common for camera gimbals on UAV's to be fixed to each other.

- **Objects hotter than the background**

The available data set is recorded in Sweden in July when the ground was cooler than the animals. This might not be the case in Africa but a lot of poaching is done after sun down which could lower the soil temperature.

2

Theory

In this chapter the four problems, P1-P4, are described in detail with theory and methods considered. The order of the sections corresponds to their logical order in the system. All methods used are described in general without details about the implementation and parameters. Theory behind the thermal imaging and sensors is also presented.

2.1 Infrared Radiation

Infrared radiation (IR) lies in the wavelength span of $0.7 - 1000\mu\text{m}$ which is just above the spectrum of visible light. IR is invisible to the human eye but some animals have evolved the ability to sense parts of the infrared spectrum.

The IR spectrum is often divided into subdivisions, different scientific fields have their own divisions. In this thesis, a division for thermal imaging, found in Table 2.1.1, is used.

The wavelength which an object with a temperature of 300 K (27°C) radiates is $\lambda = 9.7\mu\text{m}$ which lies in the LWIR-spectrum. Images used in this thesis have been captured in this spectrum and will be referred as "LWIR images" in this thesis.

Thermal imaging have one advantage over RGB images in that it works in bad light conditions since the measured energy does not depend on some external light source but instead the radiation of the objects themselves. In Figure 2.1.1 one can hardly see the person due to the combination of dark clothes and a dark background but in the LWIR image the person is clearly visible.



Figure 2.1.1: RGB and IR image of the same scene

There are two types of cameras for capturing thermal images, thermal detectors and quantum detectors.

One type of thermal detector is the microbolometer sensor (MB). It measures the temperature change that the incoming photons create. The MB operates in the spectrum around $8 - 12\mu\text{m}$ which makes it very suitable for finding mammals.

The quantum detectors (QD) measure change of electrons caused by the photons. QD have higher resolution, more responsive and produce less noise but they all need to be cooled to $\sim 170\text{K}$ or lower and they are expensive.

The incident radiant energy is formulated as

$$\text{IncidentEnergy} = \text{EmittedEnergy} + \text{TransmittedEnergy} + \text{ReflectedEnergy}$$

or

$$E = \epsilon E + \tau E + \rho E \Rightarrow 1 = \epsilon + \tau + \rho$$

The *EmittedEnergy* is dependent of the actual temperature of the object and is often intended to be measured. *TransmittedEnergy* is the energy which passed through the object and *ReflectedEnergy* is the energy reflected on the surface of the object.

To be able to calculate the true temperature of an object the knowledge of ϵ , ρ and τ is not sufficient. The transmission medium will also affect the measured energy. In Figure 2.1.2 the transmission in air is shown. Between $5 - 7.5\mu\text{m}$ nearly all the radiation is blocked by the water in the air.

Near infrared (NIR)	$\sim 0.9 - 1.4\mu\text{m}$
Mid-wave infrared (MWIR)	$\sim 2 - 5\mu\text{m}$
Long-wave infrared (LWIR)	$\sim 7.5 - 13.5\mu\text{m}$

Table 2.1.1: Subdivisions of the IR spectrum.

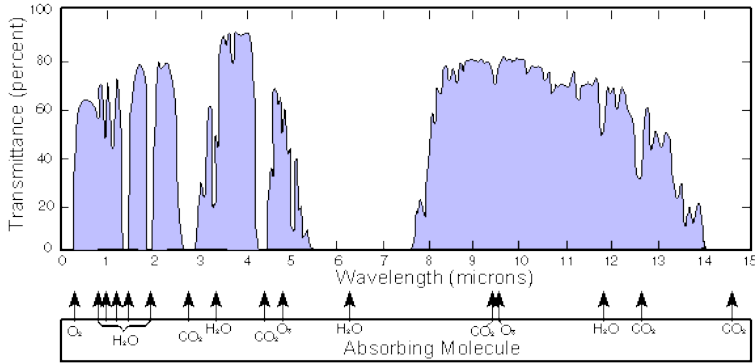


Figure 2.1.2: Transmittance for different wavelengths in air.

2.2 Image Registration

Image registration is a method of transforming an image from one coordinate system into another. In this thesis the registration is made between LWIR and RGB images.

According to the limitations of this thesis the two cameras are mounted together in a way that they cannot move independently of each other and that their images overlap. Together with a approximately constant flight altitude and a flat ground plane the transformation between the two images can be approximated with a constant homography. Transformation between two points, P_1, P_2 , is done by

$$P_1 = H * P_2 \quad (2.2.1)$$

with

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}, P_i = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.2.2)$$

The image registration is important for the whole system and improvements which can be made are discussed in Section 5.2.

2.3 Image Segmentation

In this thesis the image segmentation is used to separating what is considered important parts of the image from the unimportant, in this case hot objects are considered important. Figure 2.3.1 shows, in the top left panel, a grey scale thermal image of a rhino on the savannah and, in the bottom right panel, the binary image of the same scene where the background has been removed.

The segmentation step includes two sub steps, image enhancement and the actual segmentation. They are both described below.

The segmented image often has some level of small pixel noise which can effectively be removed with morphological operations, e.g dilation and erosion. The finished segmented image can look as in the bottom right panel in Figure 2.3.1. All remaining white parts will be used as the mask for the step of cropping the LWIR and RGB image as seen in Figure 1.2.1. This step will include a lot of unwanted regions which is neither subsequent humans nor rhinos but instead just a hot rock or another animal. This is why the classification is a required part of the system.

Thermal Image Enhancement In this thesis an enhancement of the hot (bright) regions is performed since they is considered relevant. One method to do this was proposed by Jadin and Taib in [15]. It emphasises the hot region by calculating

$$I_{imp} = 2 \cdot I - \max(I) \quad (2.3.1)$$

where I is the entire image, $\max(I)$ is the highest pixel value and I_{imp} is the improved image. An example of the difference between I and I_{imp} is shown in Figure 2.3.1.

Segmentation Segmentation of a grey scale image is done by calculating

$$I_{binary}(x, y) = \begin{cases} 1 & \text{if } I(x, y) \geq T(I(x, y)) \\ 0 & \text{if } I(x, y) < T(I(x, y)) \end{cases} \quad (2.3.2)$$

where I is the image and T is a function for calculating the threshold value. There exists several methods of finding an optimal value of $T(x, y)$ based in $I(x, y)$, e.g Otsu's method [23].

Creating a binary mask from the enhanced image is performed to extract regions which are then used in the classification part of the system. Otsu's method has been used in Figure 2.3.1 to take the step between the top right and the bottom left panel.

2.4 Feature Descriptors

The feature descriptors extract information from the input image. The features are also designed to add a layer of robustness against changes in illumination and small translations.

Both methods described below have been used for detection of pedestrians which is why they have been chosen for evaluation.

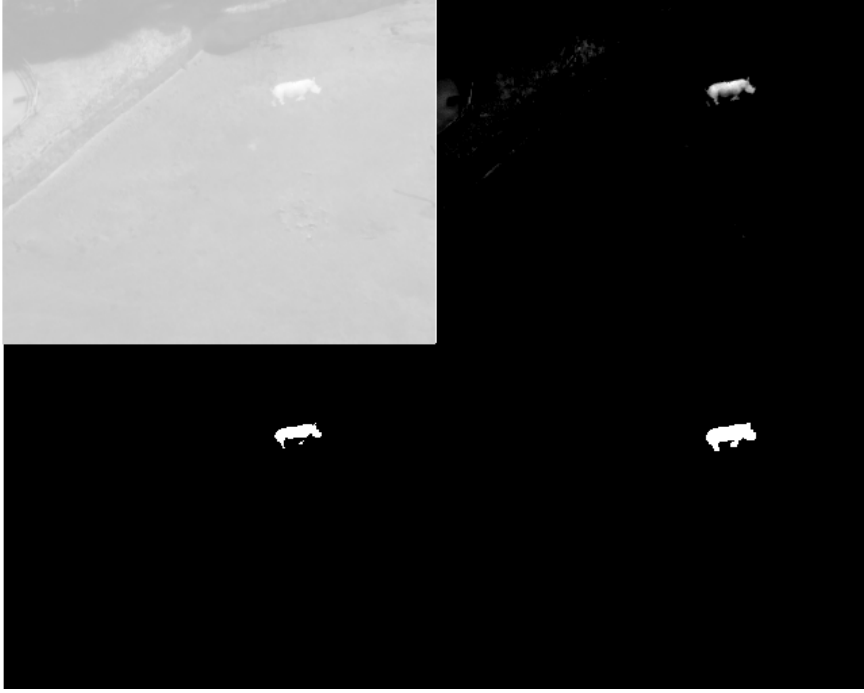


Figure 2.3.1: *Top left: I. Top right: I_{imp}. Bottom left: Segmented I_{imp}. Bottom right: Morphologically opening on segmented I_{imp}*

2.4.1 Local Binary Pattern

Local binary pattern (LBP) was first introduced in 1994 by Ojala et al. as a texture descriptor [20]. To calculate the LBP-value of each pixel

$$LBP(c) = \sum_{p \in \text{neighbours}} f(g_p - g_c) 2^p \quad (2.4.1)$$

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

is used, where c is the centre pixel and g is the grey value of a pixel. An illustration of the algorithm is shown in Figure 2.4.1. The application of the LBP and its cousins has spread from texture to a lot of different fields. It has been used in MRI images [32], in face detection [2] and pedestrian detection [12].

Rotation Invariant LBP LBP was improved by Ojala, Pietikäinen and Mäenpää [21]. They present several improvements to make the LBP grey scale and rotation invariant. This is done by not taking the nearby pixel values but instead creating a symmetric sample pattern as seen in Figure 2.4.2. Points not in the centre of a pixel are interpolated. This is however not rotation invariant since changing where the most significant bit is located changes the LBP value. To accomplish

the rotation invariance the calculated binary LBP-value $LBP_{R,P}$, on radius R and P points, is transformed by

$$LBP_{R,P}^{ri} = \min(ROR(LBP_{R,P}, i)), \quad i = 0, 1, \dots, P-1 \quad (2.4.2)$$

where $ROR(x, i)$ is a circular bit-wise right shift of x i steps. The binary pattern is rotated and the lowest possible value of the pattern is saved.

Uniform Patterns The next step of improvement presented in [21] was to define *uniform* patterns. A binary pattern which has two changes is considered a uniform pattern e. g. does 11110000 have two changes, the change between the last and first bit counts. The authors made an analysis over the distribution of the 36 different LBP^{ri} values possible for $P = 8, R = 1$, close to 90% were uniformed patterns. The authors showed that giving the uniform patterns a higher importance in the descriptor gave a significant increase of the performance.

This measurement is defined by

$$U(LBP_{R,P}) = |f(g_{P-1} - g_c) - f(g_0 - g_c)| + \sum_{p=1}^{P-1} |f(g_p - g_c) - f(g_{p-1} - g_c)| \quad (2.4.3)$$

which gives $U(11110000) = 2$ as stated before. This new measurement is used to calculate a rotation invariant uniform LBP value,

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} f(g_p - g_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{otherwise} \end{cases} \quad (2.4.4)$$

where the superscripted *riu2* describes the use of a rotation invariant uniformed measurement with 2 as the threshold. This gives each uniform LBP-value a unique value and all the others are group together as one big group.

2.4.2 Histogram of Oriented Gradients

Histogram of oriented gradients (HOG) was introduced by Navneet Dalal and Bill Triggs in 2005 as a feature descriptor for detecting of humans [9]. The method has been proven to perform well in detecting and classifying pedestrians [10]. It has also been used in many applications where the shape of the object is an important feature, e.g. finding animals [18], road signs [17] and text recognition [28].

The idea behind the method is that the local shape of an object can be described by local intensity and orientation of the gradient. First, the image is divided into cells, e.g. 8x8 pixels in [9].

For each pixel in a cell the gradient orientation and magnitude are calculated. Dalal and Triggs evaluated several methods and the Sobel filter gave them the

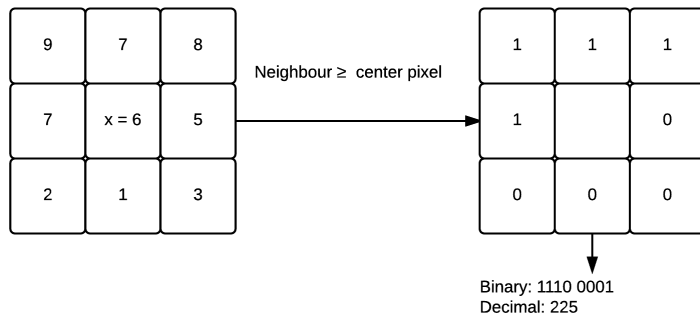


Figure 2.4.1: Example of a LBP calculation of the centre pixel with value $x = 6$. The resulting pattern is uniform

best results. Gaussian filtering before calculating the gradients only reduced the performance.

The next step is orientation binning, where all the gradients in a cell create a histogram over the different orientations. The orientations are placed in a number of equally spaced bins, 9 was used by the authors. Each pixel votes with its orientation and the magnitudes are used for weighting of the votes.

According to Dalal & Triggs the strength of the gradients can vary between neighbouring cells which reduces the performance of the descriptor. They propose a normalising step where several cells are grouped together in so called blocks. In these blocks, a normalisation of the contrast is done. They also found out that using overlapping blocks gave a performance boost, i. e. each cell contributes to several blocks. Figure 2.4.3 shows how the relationship between pixels, cells and blocks.

In [9] they used windows with size 128x64 to exploit the fact that humans are standing upright and are higher than wide. This cannot be used in this thesis since all other animals have the inverse proportions.

2.5 Machine Learning and Classification

Machine learning (ML) is often considered as a branch of artificial intelligence (AI) [19]. The idea of ML is to generate intelligence from empirical data instead of programming.

There are three main paradigms of ML; supervised, unsupervised and reinforced learning [14]. The latter two will not be considered further in this thesis and the interested reader is instead referred to [19]. Supervised learning is applied when pairs of inputs and desired outputs are given and a model is fitted.

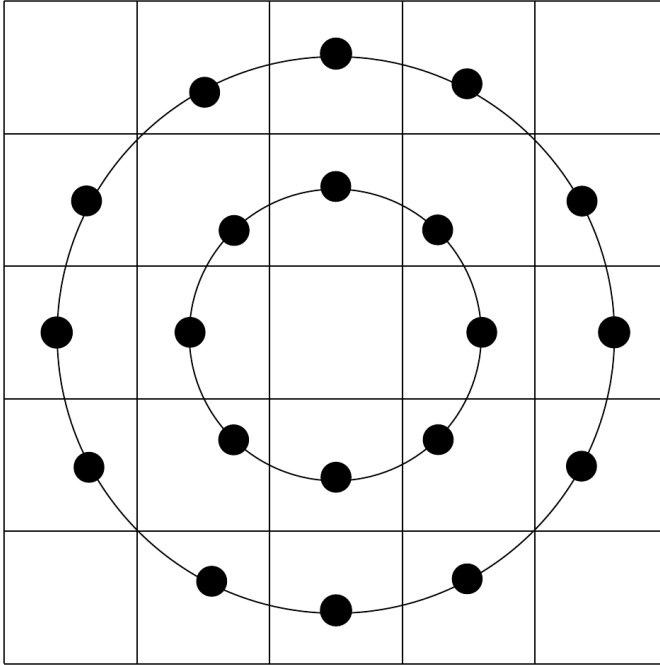


Figure 2.4.2: Circular LBP with $LBP_{R=1, P=8}$ (inner) and $LBP_{R=2, P=16}$ (outer)

The classification problem is to determine the discrete category, or class, of a new measurement based on prior knowledge. An example is to classify a person either as child or adult based on height and weight. In this thesis there will be three classes; rhino, human and other. Everything which is neither a rhino nor human is classified as other. This makes the problem a *multi-class* problem.

2.5.1 Support Vector Machine

A support vector machine (SVM) is one model used for supervised learning. The linear SVM was proposed by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963 [31]. It was then improved by Corinna Cortes and Vladimir Vapnik in 1995 by making it non-linear [8].

Linear SVM

The linear SVM calculates a hyperplane which separates data points of two classes i.e. it gives the lowest misclassification error and the largest possible margin to both of the classes. Figure 2.5.1 shows an example where any line separating the two classes (filled and not filled) would have zero error but only the proposed plane has the largest possible margin. The three samples which lie on the dot-

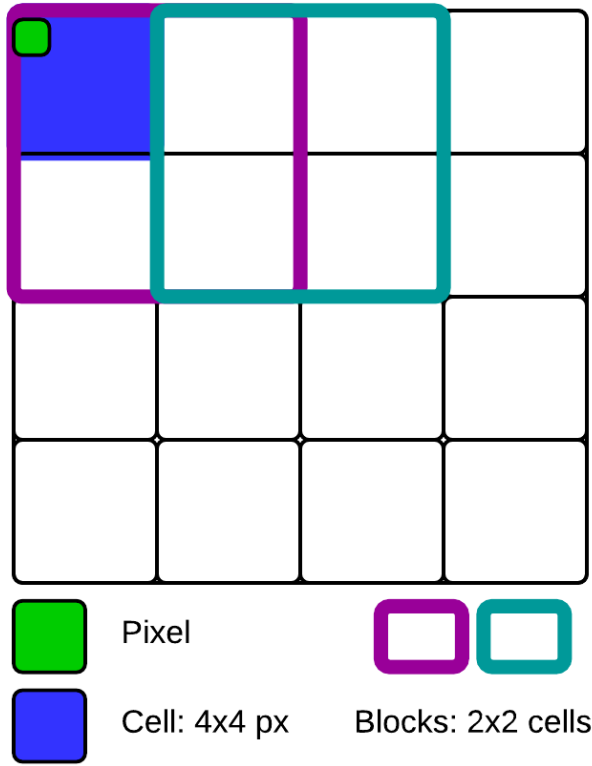


Figure 2.4.3: Example of HOG with cell size = 4x4 and block size = 2x2 with 50% overlap.

ted lines are the *support vectors*. For classifying a new sample only the support vectors are needed.

If the two classes are linearly separable the SVM can be described as

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{x}_i - b) &\geq 1 \\ y_i &= \pm 1 \end{aligned} \quad (2.5.1)$$

where \mathbf{x}_i are the sample vectors, y_i corresponding classes, \mathbf{w} is the normal of the wanted hyper plane and b the bias.

Finding \mathbf{w} and b in 2.5.1 leads to the optimisation problem

$$\arg \min_{\mathbf{w}, b} \max_{\alpha \geq 0} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1] \right\} \quad (2.5.2)$$

which is solved by [31]

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad (2.5.3)$$

$$b = \mathbf{w} \cdot \mathbf{x}_i - y_i \quad (2.5.4)$$

Only a few α_i will be non-zero and the corresponding \mathbf{x}_i will be the support vectors of the set.

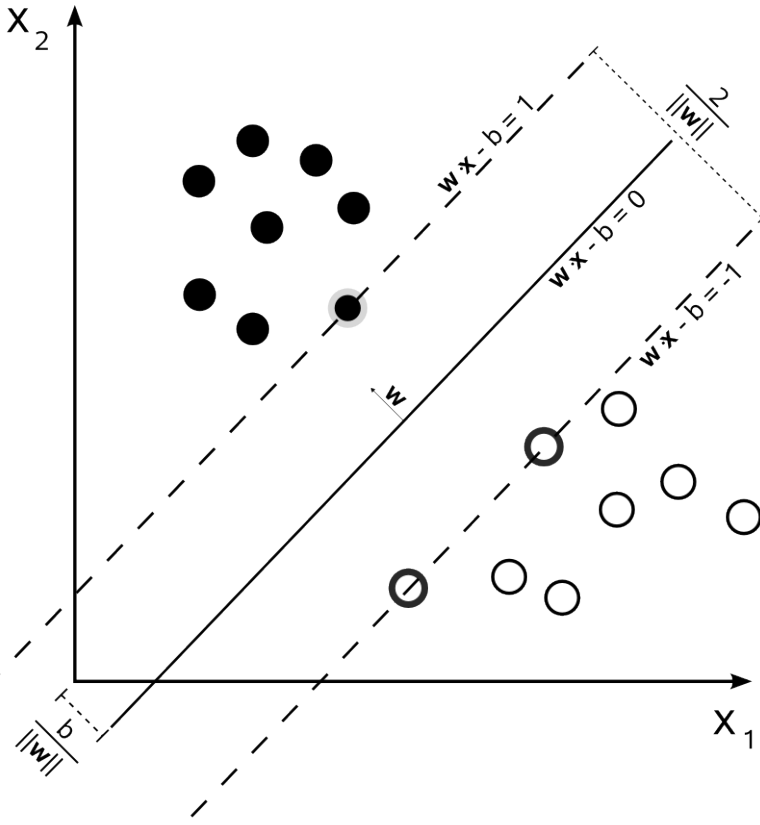


Figure 2.5.1: Resulting hyperplane and support vectors which lie on the dotted line.

Soft Margin SVM If the two classes are not linearly separable, which is very common in computer vision applications, Equation 2.5.1 will not hold true $\forall i$. This is solved by introducing a slack variable ξ_i [8]. This so called *soft margin* SVM allows samples being on the wrong side of the hyper plane \mathbf{w} . The corresponding criteria is describes as

$$\arg \min_{\mathbf{w}, \xi, b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \right\} \quad (2.5.5)$$

subject to

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (2.5.6)$$

where C is a constant set by the user.

Nonlinear SVM

There are cases where the linear SVM will perform badly due to the structure of the data, as seen in Figure 2.5.2, where the linear SVM will not perform better than 50% correct classifications. In 1992 Boser et. al introduced a way of using nonlinear separation by applying the *kernel trick* which in short transforms the data into some high dimensional feature space [5]. In the new space the classifier will be a hyperplane which allows the algorithm to fit a classifier in the same way as in the linear case.

Two common kernels used for the *kernel trick* are the radial basis function (RBF)

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \quad \gamma > 0 \quad (2.5.7)$$

and the polynomial kernels

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d, \quad d \in \mathbb{N} \quad (2.5.8)$$

where γ and d is set by the user. Both, together with the linear SVM, will be evaluated in this thesis. In Figure 2.5.2 the RBF kernel is used. A deeper explanation of the usage of kernels can be found in [14, Chapter 6].

2.5.2 Decision Trees and Random Forests

In this thesis random forests, a generalisation of decision trees, will be considered as a method of classification. The general theory behind decision trees will be presented before an analysis of random forests.

Decision Trees

A decision tree is created by arranging the features as shown in the toy example in Figure 2.5.3, where the features are "will be raining" and "will be windy". The decision is whether to take a walk is a good idea or not. The idea behind decision trees has been around for a long time, Carl von Linné used it for classifying plants in the 18th century [33].

The main question when creating the decision tree is in which order the features should appear in the tree, since the higher up in the tree a feature exists the more important it should be. There are several methods for finding these trees, e.g. ID3 [25] and CART [22].

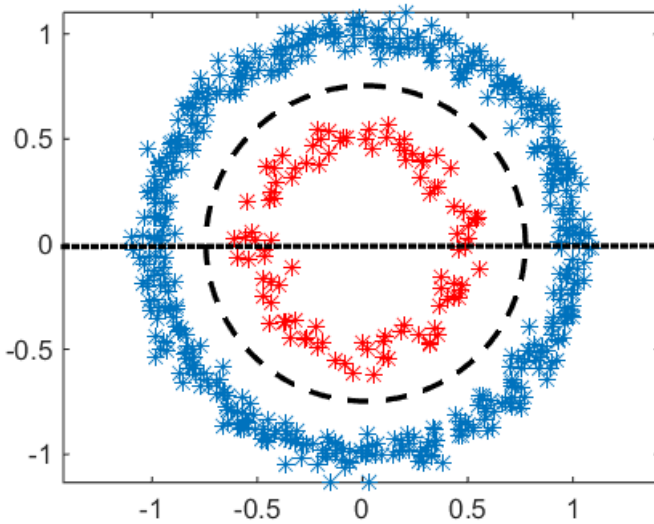


Figure 2.5.2: The dotted line is the linear SVM classification boundary and the dashed circle is the corresponding boundary for the nonlinear SVM with an RBF kernel

One drawback with decision trees are their high variance. If a division high up in the tree is erroneous it will spread down to all splits below which makes the tree result unstable [14, p. 312]. They also need pruning to avoid even higher variance, where parts of the tree which do not contribute significantly to the overall performance, are removed.

Random Forests

The random forest approach to improving the performance of decision trees was introduced by Breiman in 2001 [6]. The outline of the algorithm is found in Algorithm 1. When creating a random forest, each tree will be given a subset of the samples at random. At each branch a comparison between a random subset of the remaining features is done. The feature which gives the best split is chosen. How the split is evaluated is described below. This is done until the leaf is either pure¹ or contains fewer samples than a threshold. Then the whole procedure is repeated to create a forest of trees. When classifying an unseen sample all the trees cast a vote and the class with most votes wins. [6]

¹All the samples in the node are from the same class

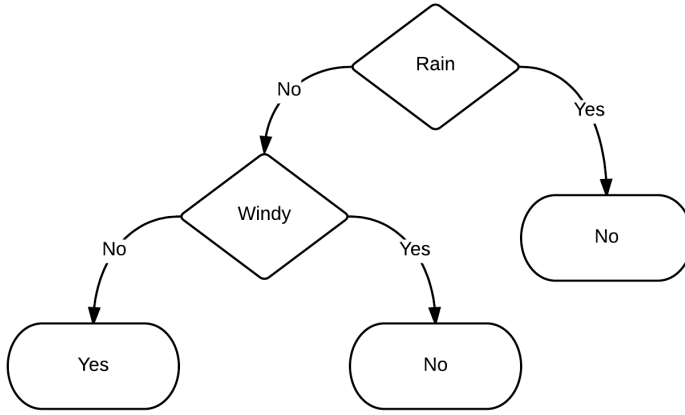


Figure 2.5.3: Decision tree over whether a walk outside is a good idea or not

Algorithm 1 Random Forest

Training set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ with corresponding classes in $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}$ and T trees

For $t = 1, \dots, T$

1. Pick $m, m < n$, random samples from \mathbf{X}, \mathbf{Y} and create a new set of $\mathbf{X}_t, \mathbf{Y}_t$, with replacement²
2. Grow a random tree, T_b , from $\mathbf{X}_t, \mathbf{Y}_t$ by recursively perform the following steps until the $nodesize = 1$ or the node is pure
 - (a) Select k features at random
 - (b) Choose the best feature, from the k chosen, to make the split

Classify an unseen sample \mathbf{x}' by taking the majority vote from the T trees.

Choosing the split In both decision trees and random forests one important part is the evaluation of each division. There are several of measurements for evaluating a division, *Gini impurity* (GI) [22] and *information gain* (IG) [25]. They are defined as

$$I_{Gini}(f) = 1 - \sum_{i=1}^n f_i^2 \quad (2.5.9)$$

and

$$I_{IG}(f) = - \sum_{i=1}^n f_i \log_2 f_i \quad (2.5.10)$$

where f_i is the fraction of which samples in the set f belongs to class i .

²The sample can be picked more than one time

2.5.3 k -Nearest Neighbours

The kNN algorithm for classification is easy to use and understand. For classifying a new sample the algorithm calculates the distances in the feature space to all neighbours and out of the k nearest neighbours the most common class is given to the new sample. As distance measure the Euclidean or Manhattan distances can be used. One difference between kNN and the other classification methods described here is that there is no training involved in kNN as it only measures the distance from a new sample to the older samples with known labels.

There are many improvements for the kNN. In 1976 Dudani [11] introduced the idea where the distances to the neighbours is included in the vote and not only which neighbours were closest. With a large amount of samples in the dataset calculating the distances to every neighbour can be very computational expensive. It can be avoided by e.g. using binary search tree [3] or using the nearest neighbour search [35]. Another problem with the original algorithm is that it is sensitive to the *curse of dimensionality* [4]. This can be suppressed by dimension reduction through e.g. *principal component analysis* [16], *Fisher's linear discriminant* [13] or *independent component analysis* [7].

2.5.4 Training and Evaluation

A main goal for this thesis is to compare the performance between different features and machine learning algorithms. To be able to draw any conclusion, the performance measurements needs to be statistically meaningful. Problems with evaluation and training sets are discussed by Salzberg in [26]. One problem which he brings up is *repeated tuning*, or parameter tuning on the data set. Each new parameter value should be considered a new experiment, to be able to accurately say anything about the performance. Salzbergs solution to this problem is to use cross validation, which is described below.

Evaluation

Evaluating the classification is an important part of this thesis, since a parameter change in both the classifier or the feature descriptor is reflected only in the classification performance. There exist several evaluation measurements; a lot of them are described in detail by Sokolova and Lapalme in [27]. The measurements which are going to be used here are presented in Table 2.5.3.

Another interesting measurement is the confusion matrix. It contains information about the classes which the classifier often mixes up. Table 2.5.2 shows an example; note that the data is made up and not a part of the result.

True \ Predicted	Pos	Neg
Pos	True Positive (TP)	False Negative (FN)
Neg	False Positive (FP)	True Negative (TN)

Table 2.5.1: Confusion matrix for a binary classification problem. Only elements on the diagonal are correctly classified

True \ Predicted	Rhino	Human	Other
Rhino	40	1	9
Human	2	57	1
Other	12	2	46

Table 2.5.2: Example of a confusion matrix (made up data and not a part of the result)

Measure	Equation	Description
Average Accuracy	$\frac{1}{N} \sum_{n=1}^N \frac{TP_n + TN_n}{TP_n + TN_n + FN_n + FP_n}$	The average per-class effectiveness of a classifier
Precision	$\frac{\sum_{n=1}^N TP_n}{\sum_{n=1}^N TP_n + FP_n}$	Agreement of the data class labels with those of a classifiers if calculated from sums of per-text decisions

Table 2.5.3: Performance measurements used in this thesis. Parts of Table 3 in [27]. N is the number of classes and TP, TN, FP and FN as in Figure 2.5.1

Overfitting

A big problem in machine learning and classification is overfitting, where a model with too many degrees of freedom is fitted to the training data. In Figure 2.5.4, two polynomials (blue lines) are fitted to the given noisy data points sampled from a true function, (green line). In the right panel the mean approximation error in the sample points is lower than in the left panel, but visually the difference between the true function and the model is larger than in the left panel.

To avoid overfitting the data sets need to be split up into a part to train on and a part to validate the model on.

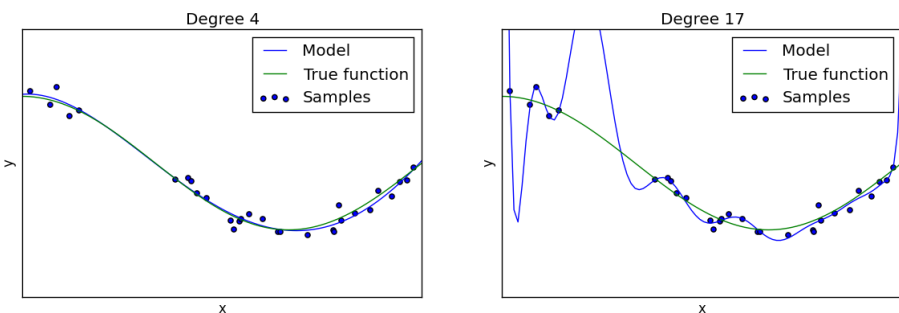


Figure 2.5.4: Example of overfitting

Cross validation

For training, a subset of the data is chosen and used to train the model, minimising the classification error. Then the remaining part of the data is used to validate the performance of the model.

If this splitting is done as in Figure 2.5.5 with several splits on the same data set, it is called *cross validation*. It is a good way for avoiding that the model is overfitted to the training data.

One advantage over a simple split of the data is that cross validation is more robust against overfitting and it assesses the generalisation of the model better. A high variance in the individual errors $error_i$ in Figure 2.5.5 is an indication of overfitting.

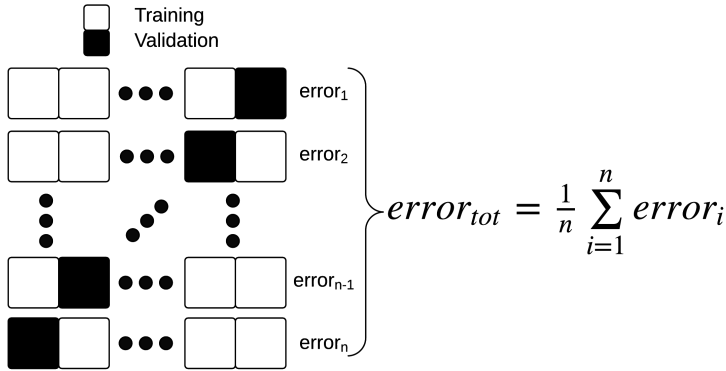


Figure 2.5.5: Cross validation scheme

3

Method

The implementation and the data evaluation is described in this chapter. This includes details about programming language and libraries.

3.1 Data sets

The data set mainly used in this thesis is provided by *Department of Electrical Engineering, ISY, at Linköping University*. It contains videos recorded at Kolmården Zoo from an octacopter, shown in Figure 3.1.4. It carried a Gopro Hero 3 and a Flir A35, which is a microbolometer thermal camera. A comparison between the two cameras can be found in Table 3.1.1 and some specification for the A35 in Table 3.1.2. Note that the thermal images are stored as 16 bits but the actual bit depth is only 14 bits. The two cameras were mounted so they could not move independently of each other.

The sequences include rhinos, elephants, gnus, zebras and several antelope species and humans. However, the humans in the sequence are very small in the images from the thermal camera, often 10x5 pixels, which makes them too small to be detected. Since this thesis is about comparing method for classifying humans, another data set was needed. Data sets with overlapping IR and RGB images are not very common but the Ohio State University offers one which can be used for research purposes [1]. The two data sets will be abbreviated as ISY and OSU for simplicity. Figures 3.1.2 and 3.1.3 show samples from the two sets, both RGB and thermal images.

The OSU has the same setup with a thermal and RGB camera as the ISY but it uses static cameras. However, the same algorithm is used for segmentation on both sets, no background subtraction method was used. OSU's images have near

perfect overlap. The data set from ISY does not and therefore requires image registration. Figure 3.1.1 shows how the LWIR image overlaps the RGB image.

The thermal cameras measure the emitted energy from the objects. This is then transformed to a grey scale. This scaling is however not known in either of the two data sets which makes it impossible to calculate the true temperature of the objects.



Figure 3.1.1: The black box in the upper image shows how the lower, thermal, image fits the RGB image. Note that the thermal (lower) image is scaled by a factor 2 compared to the RGB image

	Resolution	Framerate	Bit depth
Gopro	1920 x 1080	25	3x8 RGB
A35	320 x 256	60	14 Grey scale

Table 3.1.1: Comparison between the two cameras used at ISY on the octa-copter

Resolution	320 x 256
Detector pitch	50 micrometer
Frame rate	60 Hz
Temperature accuracy	$\pm 5^{\circ}\text{C}$ or $\pm 5\%$ of reading

Table 3.1.2: Specifications of FLIR A35 Thermal imaging camera



Figure 3.1.2: Three images from the ISY data set. Note that the thermal (lower) images are scaled by a factor 3 compared to the RGB image

3.2 Implementation

In this part details of the implementation will be presented. It will include programming language, libraries used and parameters for preprocessing. Information about the parameters which will be evaluated in the next chapter will also be presented.

3.2.1 Programming Language

The entire project was implemented in Python 2.7 using OpenCV 2.4.10 for almost all image processing. The segmentation functions in OpenCV does only



Figure 3.1.3: Four images from the OSU data set.

support 8 bits images and the thermal images are 14 bits (stored in 16 bits containers). To solve this problem *scikit-image* [30], which is an open source image processing library for Python, was used. It contains a lot of low level processing with comparable execution times as OpenCV but it is written in Python, where OpenCV is a C++ wrapper. Both libraries use Numpy [29] matrices to store images which made using both together seamless.

For classification and evaluation the *scikit-learn* [24], machine learning for Python, library was used. It contains a lot of different machine learning methods but also an extensive section with evaluation tools and also methods for creating a grid of parameters which should be tested.

3.2.2 Preprocessing

In the preprocessing step two methods were used for image enhancement, with the goal to highlight the warm parts of the image. This depends on the fact that the ground has a lower temperature than the humans and animals.

The first step is to use *Contrast Limited Adaptive Histogram Equalization*, CLAHE, from *scikit-image* with $n_bins = 100$, $ntiles_x = ntiles_y = 16$, $clip_limt = 0.01$. The resulting image is found in Figure 3.2.1. To further enhance the warm parts the *Hot Region* method described in Chapter 2.3 is applied on the resulting image which is also shown in Figure 3.2.1 too.

Another way of creating the segmentation mask is discussed in Section 5.2.



Figure 3.1.4: The octacopter used to record the data from ISY



Figure 3.2.1: *Left: Original, Middle: CLAHE, Right: CLAHE + Hot Region*

3.2.3 Image Segmentation

After the preprocessing step the image is segmented into a binary image. This is done by first using Otsu's method which is parameter free and it returns a threshold value for the original image. Morphological operations are used to remove most of the remaining noise.

Bounding boxes are then formed from the binary image with *findContours* in OpenCV. Boxes which are too small or too large are removed. See Figure 3.2.2 for an example where the small remaining noise is not marked with a box and neither is the large road. These bounding boxes are then used by the feature descriptors. In this image there was only one object but it is common for hot rocks and other animals to be included. The area restriction was $50 \text{ px} \leq \text{Area} \leq 2000 \text{ px}$, as reference, the box in Figure 3.2.2 is 330 px and the road object is $\sim 4000 \text{ px}$. This setting is highly dependent on resolution and distance to the objects but the

same restrictions were used both in ISY and OSU without missing any important objects.



Figure 3.2.2: *Left: Original, Middle: Original after morphological operations, Right: Same as the middle with bounding boxes with area constraints*

3.2.4 Image Registration

The registration between the thermal and the RGB camera is performed by manually finding 4 corresponding points in the first frame, calculating a transformation with OpenCV's *getPerspectiveTransform* and using it for the whole sequence. Objects close to the edge are sometimes missed due to a bad registration but otherwise it performs well. In Section 5.2 an improvement of this is discussed. Figure 3.1.1 shows an example of how the images overlap in the ISY data set. In the OSU data set the registration is already performed.

3.2.5 Training and Evaluation Data

Combing image registration and segmentation on the recorded videos gives a huge amount of data. Each frame contains a couple of boxes and there are about 30 000 frames. From these small boxes around 200 from each class have been picked to be used as training and evaluation data. Boxes from consequential frames are often very similar due to the small time between them. Figure 3.2.3 shows a couple of images used in the evaluation.



Figure 3.2.3: Examples from the training data with images from the three classes on corresponding RGB and LWIR images

3.3 Evaluation

For the evaluation five classifiers and five feature descriptors were chosen. Table 3.3.1 & 3.3.2 list the methods and their abbreviations which will be used in the results chapter. The unintuitive way of naming non-rotation-invariant uniform LBP LBP-NRI-UNI is to keep it in line with the implementation in sci-kit image where the method is called *nri_uniform*.

Cross Validation Throughout the evaluation a five-fold cross validation were used. The folds were chosen to keep the same class distribution of each split as in the whole set. From those five results the mean and standard deviation were extracted and are presented in the result chapter.

Measurements All tests have been evaluated by mean accuracy and some by precision, both are described in Section 2.5.4. To show which classes the classifiers often mix up, confusions matrices will be used.

Parameter testing Several of the classifiers have more than one parameter which are important to get a good result. They are often also dependent of each other. To find the best combination an exhaustive search over these parameters was made. This search method tries all possible combinations for the input given by the user. In the polynomial SVM there are three important parameters (C, gamma and d). Together with the 5 folds in the cross validation it makes the number of classifiers needed to be calculated very large. Manually adjusting the size and resolution of the search parameters is necessary to keep the computational time down.

The parameters which are included in the evaluation are listed in Table 3.3.3 & 3.3.4 for the respective methods. Each parameter is described in the respective part in Chapter 2.

Dimension reduction for kNN The kNN method is sensitive to high dimensionality of the input data. The feature descriptors used here have > 1000 dimensions which needs to be reduced to 10-50. There are several methods for this and Principal Component Analysis (PCA) was chosen.

T-test The results for most of the methods are very similar to each other and therefore a statistical measure is needed to find the results which are significant different. This was done by hypothesis testing with a t-test with the null hypothesis that the two compared results are from the same distribution. The interested reader can find a good explanation of the t-test and hypothesis testing in [34]. Only a few important results have been analysed with the t-test. Here the double sided 5% (or 95%) t-test was used.

Classifiers	Abbreviation
Linear SVM	SVM-L
RBF SVM	SVM-RBF
Polynomial SVM	SVM-P
k-Nearest Neighbours	kNN
Random Forest	RF

Table 3.3.1: Abbreviations used for the classifiers

Feature Descriptor	Abbreviation
Histogram of Oriented Gradients	HOG
Local Binary Pattern	LBP
Rotation invariant LBP	LBP-ROR
Uniform LBP	LBP-UNI
Non rotation-invariant uniform LBP	LBP-NRI-UNI

Table 3.3.2: Abbreviations used for the feature descriptors

Classifier	Parameters
SVM-L	C
SVM-RBF	C, gamma
SVM-P	C, gamma, d (degree)
kNN	k (n_neighbors), pca_dims
RF	n_trees

Table 3.3.3: Parameters which will be evaluated for the different classifiers. Each parameter is described in Chapter 2

Feature Descriptor	Parameters
HOG	blockStride, blockSize, cellSize, nbins
LBP	R,P
LBP-ROR	R,P
LBP-UNI	R,P
LBP-NRI-UNI	R,P

Table 3.3.4: Parameters which will be evaluated for the different feature descriptors. R is the radius and P the number of points samples, as seen in Figure 2.4.2

4

Results

In this chapter the result of the evaluation of the different feature descriptors and classifiers are presented.

Many of these methods include a large amount of parameters, most of them will not be presented at all, instead the default values in the implementation will be used. Some are however important to get a good result and those will be presented, showing how the performance depend on them. All parameters specified in the function calls are listed in Appendix A.

For all tests cross validation with a 5 split was used with the average error as result. This corresponds to around 130 validation samples for each split (40-50 for each class). Figure 4.0.1 shows a confusion matrix with images from the data sets.

4.1 Overview/Presentation

To make the presentation of all the different combinations pedagogical a new notation is used. It is defined as

$\langle \textit{FeatureDescriptor}, \textit{ImageType} \rangle$

and e.g. the HOG descriptor on the LWIR image would be denoted $\langle \textit{HOG}, \textit{LWIR} \rangle$, similarly, where both LBP and HOG have been used on RGB, is marked as $\langle \textit{HOG} + \textit{LBP}, \textit{RGB} \rangle$.

The presentation of the results is divided into two parts. First is the result of the different classifiers presented and then the results for the feature descriptors. Since the performance of some methods seems equal, a selection of t-test



Figure 4.0.1: Examples of images with true and predicted labels. Examples along the diagonal are correctly classified.

is performed. These tests indicate if there is a statistically significant difference between the methods. Table 4.3.9 shows the results of the t-tests.

4.2 Classifiers

Here the result for the five different classifiers are presented. They have all been tuned with regard to the parameters presented earlier. They were trained and evaluated with the same feature descriptors and images. The combinations evaluated were:

$\langle HOG, RGB \rangle$, $\langle HOG, LWIR \rangle$, $\langle HOG, RGB + LWIR \rangle$, $\langle LBP, RGB \rangle$, $\langle LBP, LWIR \rangle$, $\langle LBP, LWIR + RGB \rangle$ and $\langle HOG + LBP, LWIR + RGB \rangle$. For the feature descriptors the following parameters were used:

- **HOG:** $winSize = (32, 32)$, $blockSize = (8, 8)$, $BlockStride = (4, 4)$, $cellSize = (4, 4)$, $nbins = 12$
- **LBP:** $Method = 'default'$, $radius (R) = 1$, $n_points (P) = 8$

Table 4.2.1 to 4.2.5 show the accuracy and precision results for the different classifiers on several combinations of feature descriptors and image types. These

results are unfortunately very inconclusive. No method outperform the others in a significant way.

They do however differ on other aspects, e.g. computational performance which one should take into account. If this system is implemented on a low performing hardware, e. g. on ARM SoC, the evaluation speed and memory usage are important factors. The classification error of all the three SVM versions is very similar across the board. However, SVM-L with its linearity is much cheaper computationally and it scales with a large amount of samples or/and features. Table 4.3.10 shows a comparison of the computational time for the different SVM classifiers. SVM-L is 13% faster than SVM-RBF and 32% faster than SVM-P.

KNN have two drawbacks; the need of storing all the training samples and the need of dimensionality reduction. The complexity of PCA is $\mathcal{O}(n_{features}^2 \cdot n_{samples} + n_{features}^3)$ [16]. The performance of KNN depending on the number of dimensions is shown in figure 4.3.1

Figure 4.2.1 to 4.2.5 shows confusion matrices for the different classifiers on <HOG+LBP,RGB+LWIR>. As seen they perform very similar but the accuracy on rhinos is high for all methods.

HOG (mean STD)	RGB		LWIR		RGB+LWIR	
SVM-L	0.97	0.01	0.94	0.02	0.98	0.01
SVM-RBF	0.97	0.01	0.97	0.02	0.98	0.01
SVM-P	0.97	0.01	0.97	0.02	0.98	0.01
kNN	0.97	0.01	0.97	0.01	0.98	0.02
RF	0.975	0.002	0.930	0.003	0.982	0.002

Table 4.2.1: Accuracy result (mean & STD) for the different classifiers on HOG features

LBP (mean STD)	RGB		LWIR		RGB+LWIR	
SVM-L	0.96	0.02	0.90	0.03	0.98	0.01
SVM-RBF	0.97	0.02	0.89	0.05	0.98	0.02
SVM-P	0.96	0.02	0.89	0.04	0.98	0.02
kNN	0.98	0.01	0.94	0.01	0.98	0.01
RF	0.960	0.003	0.906	0.004	0.965	0.003

Table 4.2.2: Accuracy result (mean & STD) for the different classifiers on LBP features

4.3 Feature Descriptors

Here the result for the different feature descriptors (FD) are presented. The parameters for the classifiers have been tuned for each FD to maximise the accuracy. Tables 4.3.1 to 4.3.6 contain the result for the different LBP implementations.

HOG+LBP (mean STD)	RGB+LWIR	
SVM-L	0.99	0.01
SVM-RBF	0.99	0.01
SVM-P	0.99	0.01
kNN	0.98	0.01
RF	0.985	0.002

Table 4.2.3: Accuracy result (mean & STD) for the different classifiers on HOG+LBP features

HOG (mean STD)	RGB		LWIR		RGB+LWIR	
SVM-L	0.97	0.01	0.94	0.02	0.98	0.01
SVM-RBF	0.97	0.01	0.97	0.02	0.98	0.01
SVM-P	0.97	0.01	0.97	0.02	0.99	0.01
kNN	0.97	0.01	0.97	0.01	0.98	0.02
RF	0.973	0.003	0.926	0.007	0.973	0.003

Table 4.2.4: Precision result (mean & STD) for the different classifiers on HOG features

In Table 4.3.9 the result of several t-tests between different FDs are presented.

4.3.1 LBP and its variants

For most of the LBP-methods, increasing the radius of the sample points is lowering the accuracy in the RGB images. This is seen in Tables 4.3.3, 4.3.1 and 4.3.5 for $R = 2, P = 16$. However, this seems to not be the case in the LWIR images. Tables 4.3.6 and 4.3.2 show small difference between $R = 1, P = 8$ and $R = 2, P = 16$. For some classifiers the increased radius improves the performance and for some it decreases it. The standard deviation is also too high to draw any conclusions for the LWIR images.

Another difference worth noting between the methods is the rotation invariance, or the lack of it. In Table 4.3.1 the $\langle \text{LBP}, \text{RGB} \rangle$ and $\langle \text{LBP-ROR}, \text{RGB} \rangle$ are compared. With the same radius the LBP-ROR perform worse than the LBP. Comparing the $\langle \text{LBP-UNI}, \text{RGB} \rangle$ in Table 4.3.3 and $\langle \text{LBP-NRI-UNI}, \text{RGB} \rangle$ in Table 4.3.5 gives the same results, that the rotation invariance only decreases the performance. In the LWIR case the deviation of the result is to large for any conclusion to be made about the rotation invariance.

4.3.2 HOG

The HOG method include a lot of parameters which are dependent of each other. All parameters (except *nbins*) must be 2^n -multiple and *blockSize* must be a multiple of *blockStride*. All these requirements makes it difficult to perform a grid search over the parameters. To simplify the search the *winSize* were set to (32,32) because nearly all image patches were around 32x32 pixels. From these con-

LBP (mean STD)	RGB		LWIR		RGB+LWIR	
SVM-L	0.96	0.02	0.90	0.03	0.98	0.01
SVM-RBF	0.97	0.02	0.89	0.05	0.98	0.02
SVM-P	0.96	0.02	0.89	0.03	0.98	0.02
kNN	0.98	0.01	0.94	0.01	0.98	0.01
RF	0.956	0.003	0.899	0.005	0.961	0.004

Table 4.2.5: Precision result (mean & STD) for the different classifiers on LBP features

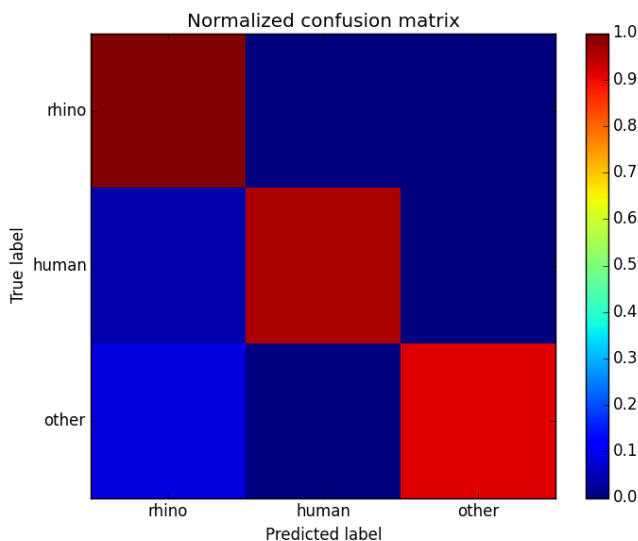


Figure 4.2.1: Confusion matrix from the SVM-L classifier on LBP+HOG features

straints the improved HOG, denoted HOG_imp, was found as:

- **HOG_imp:** $winSize = (32, 32)$, $blockSize = (8, 8)$, $BlockStride = (4, 4)$, $cellSize = (8, 8)$, $nbins = 29$

The difference between HOG and HOG_imp are the $nbins$ and $cellSize$. In Table 4.3.7 and 4.3.8 the results for HOG and HOG_imp for both RGB and LWIR.

There were no significant difference between HOG and HOG_imp on LWIR. However, as seen in Table 4.3.9 there is a significant difference between the two methods on RGB images.

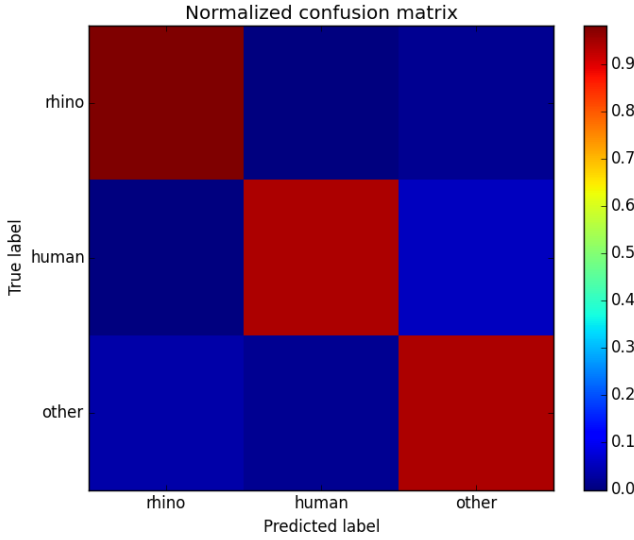


Figure 4.2.2: Confusion matrix from the SVM-RBF classifier on LBP+HOG features

RGB (mean STD)	default,1,8		ror,1,8		ror,2,16	
SVM-L	0.96	0.02	0.94	0.01	0.87	0.03
SVM-RBF	0.97	0.02	0.95	0.01	0.88	0.03
SVM-P	0.96	0.02	0.93	0.01	0.83	0.02
kNN	0.98	0.01	0.95	0.01	0.91	0.01
RF	0.960	0.003	0.941	0.003	0.952	0.003

Table 4.3.1: Accuracy result (mean & STD) for the uniform LBP-ROR on RGB with ($R = 1, P = 8$) & ($R = 2, P = 16$) with the default LBP as comparison

LWIR (mean STD)	default,1,8		ror,1,8		ror,2,16	
SVM-L	0.90	0.03	0.84	0.03	0.79	0.05
SVM-RBF	0.89	0.05	0.88	0.02	0.84	0.01
SVM-P	0.89	0.04	0.84	0.04	0.78	0.03
kNN	0.94	0.01	0.91	0.02	0.87	0.05
RF	0.906	0.004	0.862	0.006	0.892	0.005

Table 4.3.2: Accuracy result (mean & STD) for LBP-ROR on LWIR with ($R = 1, P = 8$) & ($R = 2, P = 16$) with the default LBP as comparison

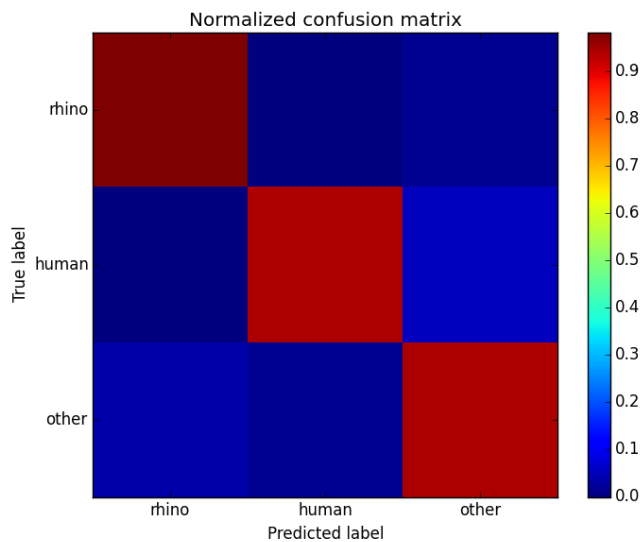


Figure 4.2.3: Confusion matrix from the SVM-P classifier on LBP+HOG features

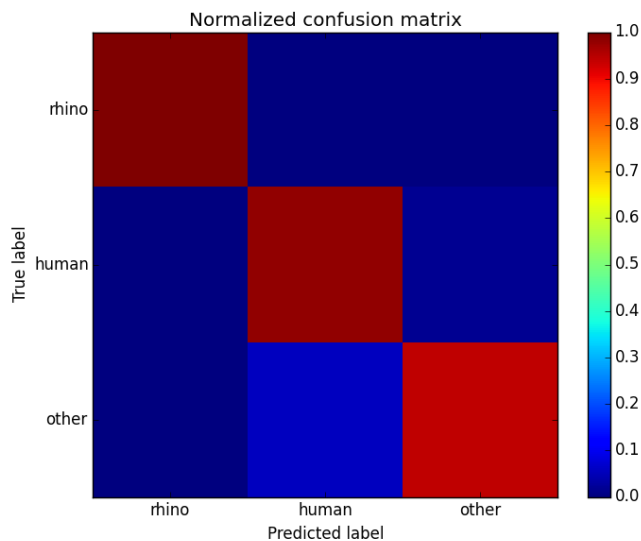


Figure 4.2.4: Confusion matrix from the kNN classifier on LBP+HOG features

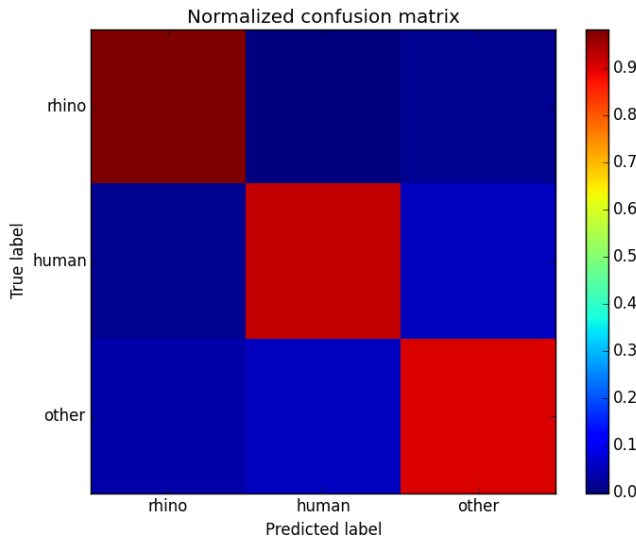


Figure 4.2.5: Confusion matrix from the RF classifier on LBP+HOG features

RGB (mean STD)	default,1,8		uniform,1,8		uniform,2,16	
SVM-L	0.96	0.02	0.94	0.01	0.88	0.01
SVM-RBF	0.97	0.02	0.96	0.01	0.95	0.01
SVM-P	0.96	0.02	0.93	0.01	0.88	0.01
kNN	0.98	0.01	0.97	0.01	0.94	0.01
RF	0.960	0.003	0.944	0.003	0.954	0.003

Table 4.3.3: Accuracy result (mean & STD) for the LBP-UNI on RGB with ($R = 1, P = 8$) & ($R = 2, P = 16$) with the default LBP as comparison

LWIR (mean STD)	default,1,8		nri_uniform,1,8		nri_uniform,2,16	
SVM-L	0.90	0.03	0.84	0.03	0.86	0.01
SVM-RBF	0.89	0.05	0.85	0.04	0.89	0.03
SVM-P	0.89	0.04	0.82	0.02	0.86	0.03
kNN	0.94	0.01	0.92	0.01	0.92	0.01
RF	0.906	0.004	0.857	0.008	0.890	0.006

Table 4.3.4: Accuracy result (mean & STD) for LBP-UNI on LWIR with ($R = 1, P = 8$) & ($R = 2, P = 16$) with the default LBP as comparison

RGB (mean STD)	default,1,8		nri_uniform,1,8		nri_uniform,2,16	
SVM-L	0.96	0.02	0.95	0.01	0.90	0.01
SVM-RBF	0.97	0.02	0.97	0.01	0.95	0.01
SVM-P	0.96	0.02	0.95	0.01	0.89	0.01
kNN	0.98	0.01	0.973	0.004	0.93	0.02
RF	0.960	0.003	0.950	0.002	0.956	0.003

Table 4.3.5: Accuracy result (mean & STD) for LBP-NRI-UNI on RGB with ($R = 1, P = 8$) & ($R = 2, P = 16$) with the default LBP as comparison

LWIR (mean STD)	default,1,8		nri_uniform,1,8		nri_uniform,2,16	
SVM-L	0.90	0.03	0.86	0.03	0.88	0.01
SVM-RBF	0.89	0.05	0.86	0.03	0.90	0.02
SVM-P	0.89	0.04	0.83	0.03	0.87	0.02
kNN	0.94	0.01	0.930	0.029	0.92	0.02
RF	0.906	0.004	0.882	0.005	0.895	0.006

Table 4.3.6: Accuracy result (mean & STD) for LBP-NRI-UNI on LWIR with ($R = 1, P = 8$) & ($R = 2, P = 16$) with the default LBP as comparison

RGB (mean STD)	HOG		HOG_imp	
SVM-L	0.97	0.01	0.988	0.004
SVM-RBF	0.97	0.01	0.99	0.01
SVM-P	0.97	0.01	0.99	0.01
kNN	0.97	0.01	0.98	0.01
RF	0.975	0.002	0.977	0.002

Table 4.3.7: Accuracy result (mean & STD) for HOG and HOG_imp on RGB

LWIR (mean STD)	HOG		HOG_imp	
SVM-L	0.95	0.02	0.94	0.02
SVM-RBF	0.97	0.02	0.97	0.02
SVM-P	0.97	0.02	0.97	0.02
kNN	0.98	0.01	0.97	0.01
RF	0.936	0.004	0.930	0.003

Table 4.3.8: Accuracy result (mean & STD) for HOG and HOG_imp on LWIR

Classifier	FD1	FD2	Significant difference?
SVM-L	<HOG,RGB>	<LBP,RGB>	No
SVM-L	<HOG,RGB>	<HOG_imp,RGB>	Yes, FD2 better
SVM-L	<HOG_imp,RGB>	<LBP,RGB>	Yes, FD1 better
SVM-L	<HOG_imp,RGB>	<HOG+LBP,RGB+LWIR>	No
SVM-L	<HOG_imp,LWIR>	<HOG_imp,RGB+LWIR>	Yes, FD2 better
KNN	<HOG,LWIR>	<LBP,LWIR>	Yes, FD1 better

Table 4.3.9: T-test on 95% level between 5 different combinations of classifiers, feature descriptors and image types.

Classifier	Time [s]
SVM-L	0.78 ± 0.01
SVM-RBF	0.92 ± 0.01
SVM-P	1.07 ± 0.01

Table 4.3.10: Computational time for the different SVM classifiers. All classifiers classified 165 samples 10 times, average time is presented.

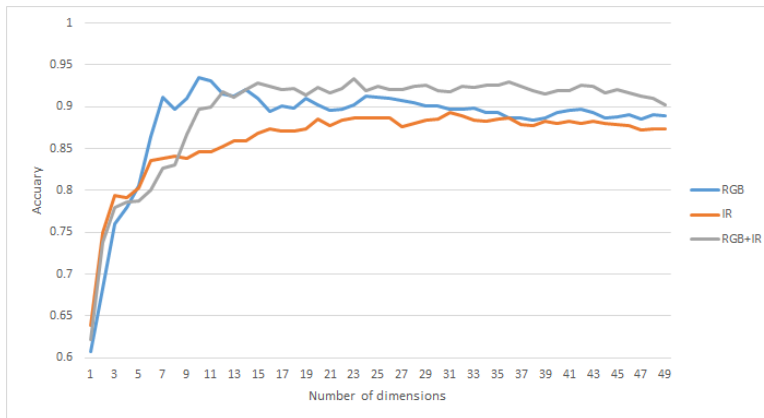


Figure 4.3.1: Graph of the accuracy of k -NN with $k = 5$ and PCA for dimensionality reduction

5

Conclusion and Future Work

In this chapter the conclusion of the thesis will be presented together with suggestions of where future work should be focused.

5.1 Conclusion

One should note that the available data sets are limited in some areas. This has been discussed earlier in the report but it is worth a reminder. To be able to separate the results, a more diverse data set is needed. Videos recorded in dusk or dawn, animals hiding behind foliage. More data on humans in the savannah environment is also very important.

The first question about which combination of classifier and feature descriptor to be used is difficult to answer. There was no clear winner of the different classifiers. However, the HOG_imp descriptor performed significantly better on both RGB images than the other methods evaluated, see Table 4.3.9. One could also combine HOG_imp with LBP if the system can handle the increased dimensionality.

One difference between this thesis and others which evaluate common classification systems is the focus on the LWIR images; not just as a method of detecting candidates in the RGB image but for classification. This is an important part since much of the poaching takes place in dusk or dawn where the RGB camera will be less useful. It has been shown in this thesis that classification on only the LWIR images is close to the RGB in accuracy. It has also shown that the common feature descriptors used on RGB images also work very well on LWIR images.

The results of the classifiers do not separate them significantly, however other

properties such computational costs do. The scenario is that the camera system is mounted on some airborne vehicle. Sending image data live has several implications on the system. Firstly, it is very expensive to sustain the needed data link to be able to send images live. Secondly, being dependent on the data link greatly reduces the operation distance of the vehicle. However, if the system is implemented on hardware in the air, only the classification result is needed to be sent to the ground station.

With low performing hardware the SVM-L stands out with its linear scaling and very low memory usage. Table 4.3.10 shows the difference in computation time between the different SVM classifiers. There exists a great amount of C/C++ implementations which all compete for the lowest computation time.

The conclusion after this thesis is to use <HOG_imp,RGB+LWIR> with the SVM-L to get one of the best performance with low computationally cost. This results in a system with a high accuracy which works both during the day and during the night.

However, to solve the problem of poaching focus cannot only lie on the supply but also on the demand. Nearly all elephant tusks and rhino horns are sold to Asia where it is used among other things as medicine. This part of the puzzle needs to be solved by politicians and law makers.

5.2 Future Work

This thesis gives a base for building a system for classifying animals and humans, from RGB and thermal images. However, there are some question marks which could not be solved in the scope the thesis. Here some possible improvements will be discussed.

5.2.1 Image Registration

For this report a constant projective transformation between the two cameras was used, which worked good sufficiently well since the two cameras were fixed together. This is almost always the case in modern camera gimbals used on aerial vehicle but they often have a zoom lens. A more sophisticated method of finding the image registration is needed.

There are some articles which use straight lines and the hough transform to find the transformation. This is not possible in this case since the straight edges are always from some human made object like buildings and pavements which are quite sparse in the savannah. One possibility would be to match the contours in both images as they look similar in both images. Most methods of matching two images, SIFT, SURF, MSER and so forth use corners to find corresponding points. Unfortunately, this does not work well in the combination of thermal and RGB images. One problem is that something visible in the RGB image could be invisible in the thermal, e.g. a rhino horn cannot be seen in the thermal image.

Another problem is the low resolution and contrast in the thermal image which gives very blurry edges and corners.

5.2.2 Image Segmentation

The thermal images from Kolmården had to be extracted from Flir's proprietary file format .seq which causes information loss. All information about how the pixel values correspond to the measured energy level are lost which makes it impossible to calculate the true temperature. With the true temperature of the objects known a smarter image segmentation could have been implemented. Instead of trying to enhance the bright parts, a threshold function

$$I_{binary}(x, y) = \begin{cases} 1 & \text{if } 39 \geq I_{celsius}(x, y) \geq 35 \\ 0 & \text{otherwise} \end{cases} \quad (5.2.1)$$

could be used, depending on the temperature of rhinos and humans.

5.2.3 Tracking

The proposed system can identify rhinos and humans in each frame with a high accuracy but it is oblivious to the fact that the rhino in the next frame is the same as in the current. This can be solved by introducing a tracker which tracks the bounding box between frames. For this to work well the camera movement relative to the world needs to be small or known. The information about the camera movements can be used as an input to the tracker to make it work when the camera e.g. is rotating.

5.2.4 Choosing Sensor

Both data sets used are captured with the cheaper and common microbolometer sensor. It would be interesting to evaluate which part of the IR spectrum gives the best performance. The sensors able to capture shorter wavelength than the bolometer are very expensive which makes images from them hard to come by, especially on rhinos.

Not using an action camera with fish-eye lens would also be preferable to reduce lens distortion.

A

Appendix

Here all the different tuned parameters for all methods are presented. All classifiers are from *sci-kit learn*, HOG from *OpenCV* and LBP from *sci-kit image*. If a parameter is not listed here the default value was used.

A.1 Image segmentation

Besides using the *Hot Region* described in Chapter 2, CLAHE was also used with the following parameters:

$ntiles_x = 16$, $ntiles_y = 16$, $nbins = 100$, $clip_limit = 0.01$

A.2 Classifiers and Feature Descriptors

All the parameters used for the different classifiers and feature descriptors are presented here.

Classifier	Parameters	<HOG,RGB>	<HOG,LWIR>	<HOG,RGB+LWIR>
SVM-L	C	0.1	0.1	0.014
SVM-RBF	C, Gamma	1.3,0.016	3,0.023	2.1,0.001
SVM-P	C, Gamma, Poly	0.14,0.015,3	0.31,0.02,3	1,0.4,3
kNN	k,weighth,pca_dims	3,'dist',40	3,'dist',36	3,'dist',53
RF	trees	800	600	500

Table A.2.1: Parameters for the different classifiers on HOG features

Classifier	Parameters	<LBP,RGB>	<LBP,LWIR>	<LBP,RGB+LWIR>
SVM-L	C	0.004	0.004	0.004
SVM-RBF	C, Gamma	0.73,0.0042	0.65,0.004	0.65,0.003
SVM-P	C, Gamma, Poly	0.07,0.01,3	0.07,0.007,3	0.02,0.006,3
kNN	k,weighth,pca_dims	3,'dist',15	3,'dist',18	3,'dist',17
RF	trees	400	600	500

Table A.2.2: Parameters for the different classifiers on LBP features

Classifier	Parameters	<HOG+LBP,RGB+LWIR>
SVM-L	C	0,1
SVM-RBF	C, Gamma	2,0.001
SVM-P	C, Gamma, Poly	0.01,0.01,3
kNN	k,weighth,pca_dims	3,'dist',17
RF	trees	800

Table A.2.3: Parameters for the different classifiers on HOG+LBP features

Classifier	Parameters	<LBP_ROR(1,8),RGB>	<LBP_ROR(2,16),RGB>
SVM-L	C	0.01	0.012
SVM-RBF	C, Gamma	2,9,0.004	1.85,0.008
SVM-P	C, Gamma, d	0.15,0.031,2	0.23,0.056,2
kNN	k,weighth,pca	3,'dist',13	3,'dist',10
RF	trees, max_feat	800	800

Table A.2.4: Parameters for the different classifiers on LBP_ROR features

Classifier	Parameters	<LBP_ROR(1,8),LWIR>	<LBP_ROR(2,16),LWIR>
SVM-L	C	0.007	0.0074
SVM-RBF	C, Gamma	1.7,0.0085	3.4,0.0095
SVM-P	C, Gamma, d	0.07,0.025,2	0.165,0.0058,2
kNN	k,weighth,pca	3,'dist',15	5,'dist',8
RF	trees, max_feat	1000	1100

Table A.2.5: Parameters for the different classifiers on LPB_ROR features

Classifier	Parameters	<LBP_UNI(1,8),LWIR>	<LBP_UNI(2,16),LWIR>
SVM-L	C	0.017	0.011
SVM-RBF	C, Gamma	2.46,0.0074	2.4,0.0071
SVM-P	C, Gamma, d	0.019,0.008,4	0.023,0.0081,4
kNN	k,weighth,pca	5,d,9	3,d,12
RF	trees, max_feat	700	700

Table A.2.6: Parameters for the different classifiers on LBP_UNI features

Classifier	Parameters	<LBP_UNI(1,8),RGB>	<LBP_UNI(2,16),RGB>
SVM-L	C	0.01	0.0054
SVM-RBF	C, Gamma	2.2,0.0055	2.4,0.0065
SVM-P	C, Gamma, d	0.14,0.0063,3	0.01,0.0058,3
kNN	k,weighth,pca	3,5	3,12
RF	trees, max_feat	700	800

Classifier	Parameters	<LBP_NRI_UNI(1,8),RGB>	<LBP_NRI_UNI(2,16),RGB>
SVM-L	C	0.009	0.0042
SVM-RBF	C, Gamma	1.7,0.006	1.6,0.006
SVM-P	C, Gamma, d	0.2,0.005,3	0.1,0.005,3
kNN	k,weighth,pca	5,11	3,17
RF	trees, max_feat	800	600

Table A.2.7: Parameters for the different classifiers on LBP_NRI_UNI features

Classifier	Parameters	<LBP_NRI_UNI(1,8),LWIR>	<LBP_NRI_UNI(2,16),LWIR>
SVM-L	C	0.013	0.012
SVM-RBF	C, Gamma	2.1,0.008	1.8,0.008
SVM-P	C, Gamma, d	0.01,0.0078,4	0.01,0.008,4
kNN	k,weighth,pca	3,d,8	3,d,14
RF	trees, max_feat	900	500

Table A.2.8: Parameters for the different classifiers on LBP_NRI_UNI features

Bibliography

- [1] Dataset 03: Osu color and thermal database, <http://www.vcipl.okstate.edu/otcbvs/bench/data/03/download.html>. URL <http://www.vcipl.okstate.edu/otcbvs/bench/Data/03/download.html>. Cited on page 21.
- [2] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, Dec 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.244. Cited on page 9.
- [3] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975. ISSN 0001-0782. doi: 10.1145/361002.361007. URL <http://doi.acm.org/10.1145/361002.361007>. Cited on page 18.
- [4] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is “nearest neighbor” meaningful? In *Database Theory—ICDT’99*, pages 217–235. Springer, 1999. Cited on page 18.
- [5] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. Cited on page 15.
- [6] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. Cited on page 16.
- [7] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994. Cited on page 18.
- [8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 09/01 1995. URL <http://dx.doi.org/10.1007/BF00994018>. J2: Mach Learn. Cited on pages 12 and 14.
- [9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE*

- Computer Society Conference on*, volume 1, pages 886–893 vol. 1, 2005. ISBN 1063-6919. ID: 1. Cited on pages 10 and 11.
- [10] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009. Cited on page 10.
 - [11] Sahibsingh A. Dudani. The distance-weighted k-nearest-neighbor rule. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-6(4):325–327, April 1976. ISSN 0018-9472. doi: 10.1109/TSMC.1976.5408784. Cited on page 18.
 - [12] M. Enzweiler and D.M. Gavrila. A multilevel mixture-of-experts framework for pedestrian classification. *Image Processing, IEEE Transactions on*, 20(10):2967–2979, Oct 2011. ISSN 1057-7149. doi: 10.1109/TIP.2011.2142006. Cited on page 9.
 - [13] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936. Cited on page 18.
 - [14] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009. Cited on pages 11, 15, and 16.
 - [15] MS Jadin and S Taib. Infrared image enhancement and segmentation for extracting the thermal anomalies in electrical equipment. *Elektronika ir Elektrotechnika*, 120(4):107–112, 2012. Cited on page 8.
 - [16] Iain M Johnstone and Arthur Yu Lu. Sparse principal components analysis. *arXiv preprint arXiv:0901.4392*, 2009. Cited on pages 18 and 33.
 - [17] A. Mammeri, E. H Khiari, and A. Boukerche. Road-sign text recognition architecture for intelligent transportation systems. In *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, pages 1–5, 2014. ID: 1. Cited on page 10.
 - [18] A. Mammeri, Depu Zhou, A. Boukerche, and M. Almula. An efficient animal detection system for smart cars using cascaded classifiers. In *Communications (ICC), 2014 IEEE International Conference on*, pages 1854–1859, 2014. ID: 1. Cited on page 10.
 - [19] Stephen Marsland. *Machine learning : an algorithmic perspective*. Chapman & Hall/CRC Machine learning & pattern recognition series. Boca Raton, FL : Chapman & Hall/CRC, cop. 2009, 2009. ISBN 9781420067187. URL <https://login.e.bibl.liu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat00115a&AN=1kp.666193&site=eds-live>. Cited on page 11.
 - [20] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of dis-

- tributions. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision and Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582–585 vol.1, Oct 1994. doi: 10.1109/ICPR.1994.576366. Cited on page 9.
- [21] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, Jul 2002. ISSN 0162-8828. doi: 10.1109/TPAMI.2002.1017623. Cited on pages 9 and 10.
- [22] LBJFR Olshen, Charles J Stone, et al. Classification and regression trees. *Wadsworth International Group*, 93(99):101, 1984. Cited on pages 15 and 17.
- [23] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975. Cited on page 8.
- [24] 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. Cited on page 24.
- [25] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986. Cited on pages 15 and 17.
- [26] Steven L Salzberg. On comparing classifiers: A critique of current research and methods. *Data mining and knowledge discovery*, 1(1), 1999. Cited on page 18.
- [27] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427 – 437, 2009. ISSN 0306-4573. doi: <http://dx.doi.org/10.1016/j.ipm.2009.03.002>. URL <http://www.sciencedirect.com/science/article/pii/S0306457309000259>. Cited on pages 18 and 19.
- [28] Shangxuan Tian, Shijian Lu, Bolan Su, and Chew Lim Tan. Scene text recognition using co-occurrence of histogram of oriented gradients. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 912–916, 2013. ISBN 1520-5363. ID: 1. Cited on page 10.
- [29] S. van der Walt, S.C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30, March 2011. ISSN 1521-9615. doi: 10.1109/MCSE.2011.37. Cited on page 24.
- [30] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python.

- PeerJ*, 2:e453, 6 2014. ISSN 2167-8359. doi: 10.7717/peerj.453. URL <http://dx.doi.org/10.7717/peerj.453>. Cited on page 24.
- [31] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. 1963. ID: 2042609. Cited on pages 12 and 14.
- [32] A. Varghese, R.R. Varghese, K. Balakrishnan, and J.S. Paul. Level identification of brain mr images using histogram of a lbp variant. In *Computational Intelligence Computing Research (ICCIC), 2012 IEEE International Conference on*, pages 1–4, Dec 2012. doi: 10.1109/ICCIC.2012.6510233. Cited on page 9.
- [33] Carl von Linné. *Classes Plantarum*. Leiden, 1738. Cited on page 15.
- [34] Wikipedia. Statistical hypothesis testing — wikipedia, the free encyclopedia, 2015. URL http://en.wikipedia.org/w/index.php?title=Statistical_hypothesis_testing&oldid=656221883. [Online; accessed 9-May-2015]. Cited on page 28.
- [35] Peter N Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, volume 93, pages 311–321, 1993. Cited on page 18.

Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: <http://www.ep.liu.se/>