Improving automatic object detection performance for wildlife conservation drones

Dion Oosterman

Improving automatic object detection performance for wildlife conservation drones

Dion Oosterman
10306048

Bachelor thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
dhr. dr. A. Visser

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

24th June 2016

Abstract

In this thesis, a method for improving automatic object detection performance for wildlife conservation is proposed. Automatic object detection methods designed for human-scale images are not directly applicable to aerial imagery, because objects on aerial images appear smaller than objects on human-scale images. Improving performance is attempted by reducing the number of object detection proposals by half, while maintaining recall. This reduction is achieved by re-ranking and filtering detection proposals based on their bounding box metadata and corresponding flight information. Re-ranking is performed using a logistic regression model after which the lower ranking proposals are discarded so that the number of proposals can be reduced while recall is maintained. The results in this thesis show that with this method, the number of proposals can be reduced by half while only having minor impact on recall.

Acknowledgements

I would like to thank my supervisor Arnoud Visser for his guidance and support. Without his effort, this thesis would not have been the way it is.

I would also like to thank *Dutch UAS* for their data, in particular Anouk Visser for her invaluable advice and Nick Noordam for drawing my attention to *Pix4D*, which has proven to be an extremely useful piece of software for this thesis.

Additionally, I am very grateful to *Magic Group Media* for providing me with an amazing custom-made cover image.

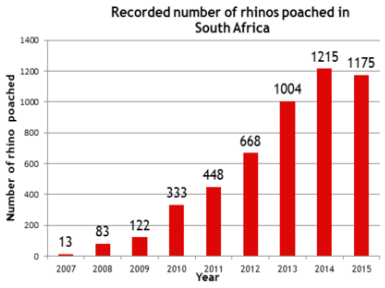
Last but not least, I am thankful to my family for their patience, support and encouragement.

Contents

1 Introduction

Extinction is a natural process that has been going on since the origin of life on earth. Human activities such as poaching and overfishing are considered to be the main cause of the increasing rate of extinction, accounting for the extinction of numerous animal species in recent history (Vitousek, Mooney, Lubchenco & Melillo, 1997). An example of these activities is the recent increase in rhino poaching in South Africa. Figure 1.1a shows the white rhino, which has become more endangered since the last few years as a result of this increase. As illustrated in figure 1.1b, the recorded number of rhinos poached between 2007 and 2014 increased by more than 9346 per cent. In 2015 the recorded number of poached rhinos decreased, yet it is still significantly higher than in the years prior to 2014.

(a)



(b)

Figure 1.1: On the left the white rhinoceros (*Ceratotherium simum*). The graph on the right depicts the increase in rhino poaching in South Africa, using data published by the South African Department of Environmental Affairs in 2016. Image from Holger Langmaier, graph from Save the Rhino International.

For the reduction or prevention of future extinctions and conservation of wildlife, it is crucial that the distribution and abundance of animal species are accurately monitored (Buckland et al. 2001; 2004, as cited in Van Gemert et al., 2014; Verschoor, 2016).

Conventional methods of wildlife conservation such as on-foot monitoring are costly and time-consuming, as large areas must be covered for proper analysis of species abundance and distribution (Van Gemert et al., 2014). Although aerial surveys methods could be utilised to cover large areas, they are expensive and involve high risk of crashing (Wich, Dellatore, Houghton, Ardi & Koh, 2015). Unmanned aerial vehicles (UAVs), also known as drones, offer the benefits of aerial surveys, without the cost and risk of manned aerial surveys. Research by Van Andel et al. (2015) shows that drones offer great possibilities as a swift assessment tool for detecting animal presence.

Van Gemert et al. (2014), state that automating object detection methods is a necessity for the advancement in wildlife conservation, as the amount of data recorded by drones "quickly grows to thousands of photos and hundreds of hours of video" (p. 2) . Most object detection methods are, however, designed

for human-scale images rather than aerial imagery, and are thus evaluated on images taken from a height of 1-2 metres instead of 10-100 metres, the height drone imagery is usually taken from (Verschoor, 2016). Therefore, objects on drone imagery appear relatively small and have a skewed vantage point when compared to objects on human-scale images (Van Gemert et al., 2014; Verschoor, 2016). For these reasons, current automatic object detection methods are unlikely to be applicable to aerial images and thus to wildlife conservation drones (Van Gemert et al., 2014).

One possible solution to this problem is to make object detection methods suitable for automatic wildlife conservation by locating object detection proposals through a conventional detection proposal method and re-ranking the obtained detection proposals based on their corresponding bounding box metadata and flight information. To make object detection more efficient, detection proposals can be filtered so that the number of proposals that have to be analysed is reduced by a significant amount. The bounding boxes' areas and aspect ratios will serve as features on which the re-ranking and filtering of detection proposals is performed. Impact on recall should be limited in order to maintain detection performance. In this thesis, it is evaluated if this approach is effective and makes automatic object detection more efficient by reducing the amount of detection proposals required. The research question is as follows:

Is it possible to reduce the amount of detection proposals by half, using an algorithm that re-ranks and filters proposals based on their bounding box metadata and flight information, while maintaining a recall of at least 0.75?

The research for this thesis is performed in cooperation with the startup *Dutch Unmanned Aerial Solutions* (Dutch UAS), which has received a considerable amount of attention from the media for their efforts in rhino conservation and has been nominated for numerous 'innovative startup'-awards. Their method involves making use of wildlife conservation drones in combination with artificial intelligence algorithms in order to analyse aerial imagery. In this thesis, a dataset from Dutch UAS is used which exists of images of a crash of 5 rhinos, shot by a conservation drone near the Marakele National Park in Limpopo, South-Africa (see figure 1.2).

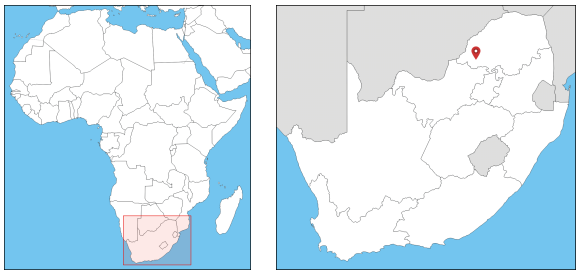


Figure 1.2: Left: states map of Africa with the outline of South-Africa. Right: province map of South-Africa with the location of where the images for the dataset were taken.

The drone used for creating the dataset is a customised *Twinstar*, a glider type drone, which is shown in figure 1.3. The images are taken with a *GoPro Hero3-Black Edition* camera, which is attached to the bottom of the drone. Additionally, a GPS recorder has logged the attitude and altitude of the drone during the flight. This dataset will be used to train a logistic regression model, with which detection proposals will be ranked on the likelihood of their bounding boxes enclosing a rhino. In consultation with Dutch UAS, the algorithm will be considered successful if a decrease in number of proposals by half is achieved, given that the recall does not drop below 0.75

In the next chapter, the required theory for the algorithm will be assessed. In Chapter 3, related research will be reviewed on the topics of object detection proposals and re-ranking detection proposals in order to improve detection performance. The adopted methodology and the algorithm pipeline will be outlined in Chapter 4. Finally, the results will be presented and discussed.

Figure 1.3: Left: the Twinstar glider drone which was used for taking the images for the dataset. Right: an aerial image of a white rhino killed by poachers.

2 Theory

2.1 Object detection

Object detection is a technology in computer vision and image processing, which employs classifier algorithms in order to detect objects of a certain class in digital images. This can be done for numerous purposes, examples including face detection, video surveillance and animal detection and classification (see figure 2.1). Detection and classification of animals is useful for automatic wildlife monitoring.

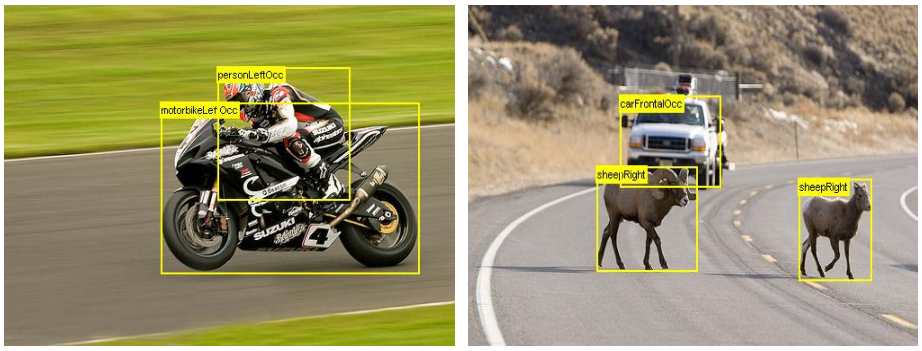


Figure 2.1: Two examples of images with detected and classified objects. In this example, aside from the object's class, the object's orientation is determined. Taken from the Pascal Visual Object Classes Challenge 2012 dataset.

2.1.1 Detection proposals

Detection proposal methods are object detection methods that operate under the premise that objects share visual attributes which differentiate them from the background (Hosang, Benenson & Schiele, 2014). In the traditional 'sliding window' approach, images are thoroughly evaluated on a large variety of scales, positions and and aspect ratios (Hosang et al., 2014). Contrary to the sliding window paradigm, in object detection proposals a method is designed and trained such that for an image used as input, a collection of detection proposal windows which are likely to contain objects of interest is returned. Detection proposals improve upon the sliding window paradigm, as they require a 'mere' 10^4 - 10^5 windows per image instead of 10^6 - 10^7 evaluations (Hosang, Benenson & Dollár, 2016). They therefore offer an increase in detection speed as well as an improvement in detection quality, since they reduce the likelihood of acquiring false positives (Hosang et al., 2014).

2.1.2 Edge Boxes

In their paper, Zitnick and Dollár (2014) introduce the Edge Boxes detection proposal method. The method seeks for detection proposals by evaluating the number of edges wholly enclosed in a bounding box, which is indicative of the likelihood that the box contains an object of interest. Then, an edge map is initialised with the help of a Structured Edge detector, which is used for calculating a proposal score for the corresponding bounding box. This score correlates with the likelihood of the bounding box containing an object, and is calculated "by summing the edge strength of all edge groups within the box, minus the strength of edge groups that are part of a contour that straddles the box's boundary" (Zitnick and Dollár, 2014, p. 3). Furthermore, they observe that edges near the boundary of the box are of less value than edges in the centre. A comparison is made between the Edge Boxes method and other state-of-the-art methods, including Selective Search, and results show that Edge Boxes is both faster and more accurate in detecting objects than other methods (Zitnick & Dollár, 2014; Hosang et al., 2014, 2016).

2.2 Perspective transformation

The drone imagery for this thesis is taken at a certain height and at a certain angle and rotation. These are different for each image, since they depend on the drone's attitude and altitude at the time the image is taken. The attitude of a drone (or any aircraft in general) is determined by the angles of rotation in three dimensions around the drone's centre of gravity. The roll corresponds with the rotation around the x-axis, whereas the pitch and yaw correspond with the y- and z- axes respectively (see figure 2.2).

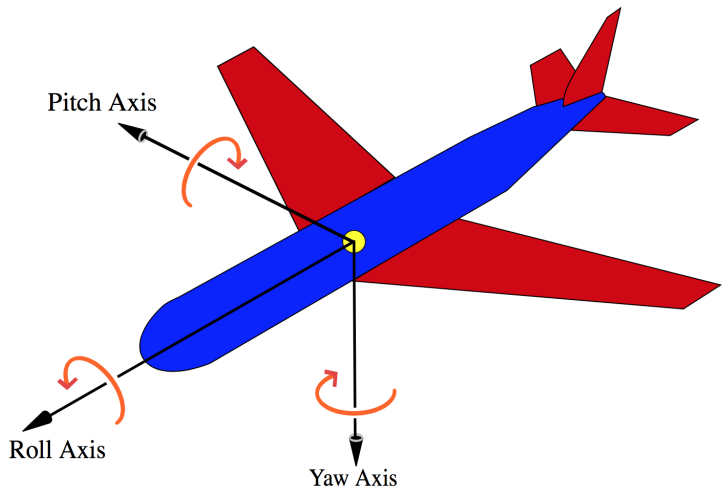


Figure 2.2: Illustration of the axes of rotation of an aircraft. Image from Wikimedia Commons.

Consequently, the object detection proposals bounding boxes are distorted as they have been assigned on the perspective distorted images. In order for one object to have the same bounding box size, regardless of the attitude and altitude of different images on which it is displayed, it is necessary that the images are normalised so that they are represented as being observed from the same angle and height. It has yet to be determined if this actually improves comparison of bounding boxes. Normalisation is achieved by the multiplication of the homogeneous coordinates of bounding boxes with the inverse matrix of the perspective transformation matrix of the corresponding image (Hartley and Zisserman, 2003, p. 34).

The perspective transformation matrix is calculated with the drone's attitude and altitude, in the following matrix:

$$P = K * T * R * D \tag{1}$$

Where K is the internal camera matrix (also known as camera calibration matrix) with skew factor a, focal length f in pixels and w and h as width and height of the image in pixels, given by:

$$K = \begin{pmatrix} f & a & \frac{w}{2} & 0 \\ 0 & f & \frac{h}{2} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2}$$

Focal length f is calculated with the following formula, where *fov* is the horizontal field of view of the camera in degrees:

$$f = \frac{w}{2} * \frac{fov}{2} * \frac{\pi}{180} \tag{3}$$

T is the matrix describing the translation on the z-axis that normalises the height d, of which the unit is irrelevant as long as the heights retain their proportions, given by:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

R is the rotation matrix calculated by multiplying the homogeneous 3-dimensional Givens rotations in the $x-$, $y-$ and $z-$direction (Hartley and Zisserman, 2003, p. 579), given by:

$$R = \begin{pmatrix} R_x & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} R_y & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} R_z & 0 \\ 0 & 1 \end{pmatrix} \tag{5}$$

For which:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{6}$$

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where θ is equal to the corresponding angle, determined by the drone's attitude.

D is the matrix that projects 2D-coordinates to 3D, with w and h as width and height of the image in pixels, given by:

$$D = \begin{pmatrix} 1 & 0 & -\frac{w}{2} \\ 0 & 1 & -\frac{h}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

Finally, it must be noted that many cameras have some form of radial distortion in the images taken by them. Although this can be corrected, just as perspective distortion, this will not be done in this thesis for the reason that radial distortion has minimal impact on the overlap of two rectangles (which appear as bounding boxes in this study).

2.3 Logistic regression

Predicting whether a detection proposal has correctly detected a rhino is a dichotomous classification problem, since the class labels are either 0 (no rhino) or 1 (rhino). A logistic regression model serves this purpose excellently, as it is designed to predict probabilities which always have a value between 0 and 1 (Kleinbaum & Klein, 2010). In order to re-rank the initial detection proposals, the probability of each proposal belonging to class 1 will be calculated by a logistic regression model.

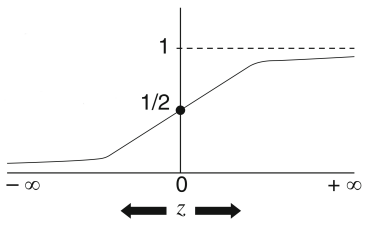


Figure 2.3: Plot of the logistic function, as in equation 8, as the value of z varies from -∞ to ∞. Image from Kleinbaum and Klein (2010).

The model is based on the logistic function or the sigmoid function, as in equation 8, which is plotted and shown in figure 2.3:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{8}$$

The variable z of the logistic function $f(z)$ is written as the linear sum of features X_i and corresponding parameters θ_i:

$$z = \theta_0 + \theta_1 X_1 + ... + \theta_\mathrm{n} X_\mathrm{n} \tag{9}$$

The parameters $\theta_0, ..., \theta_\mathrm{n}$ are determined through the maximum log likelihood. The log likelihood is maximised by gradient ascent, and looks as follows, with C_p being the predicted class of the proposals and C being the actual class:

$$\ell(\theta) = \log L(\theta) = \sum C \log(C_\mathrm{p}) + (1 - C) \log(1 - C_\mathrm{p}) \tag{10}$$

After the model parameters have been fitted, the probability of a detection proposal, with set of features X, belonging to class 1 ($C = 1$) is calculated as follows:

$$P(X) = P(C = 1 | X_1, ..., X_\mathrm{n}) = \frac{1}{1 + e^{-(\theta_0 + \sum \theta_\mathrm{i} X_\mathrm{i})}} \tag{11}$$

3 Related work

3.1 Object detection proposals

In recently conducted research on determining the effectiveness of detection proposals for object detection by Hosang et al. (2014), several detection proposal methods are evaluated based on their repeatability and recall. Repeatability of detection proposal methods is suggested to be relevant, because a consistent detection of wildlife is essential for the algorithm to be reliable. Recall is important, since 'missed' objects are not recoverable and would lead the object detection method to fail. Hosang et al. (2014) conclude that the Selective Search and Edge Boxes methods are yielding the best results, and that the latter is the most promising for speed versus quality. Hence, the Edge Boxes method seems most interesting for conservation drones.

In their other research, Hosang et al. (2016) revisit several detection proposal methods and assess various methods of evaluation. The average recall (AR), a novel metric that simultaneously measures proposal recall and localisation accuracy, is introduced and they find that AR correlates well with detection performance. Furthermore, Edge Boxes is found to be the best method in terms of performance with a low number of proposals (Hosang et al., 2016) and is thus very suitable for conservation drones, that often have limited hardware. Hosang et al. (2016) finalise their paper with the expectation that proposal methods will still improve and evolve quickly in the near future.

3.2 Re-ranking detection proposals

Kuo, Hariharan and Malik (2015) attempt to improve detection proposal methods by re-ranking the obtained proposals based on the likelihood of them enclosing an object of interest. Convolutional neural networks (CNNs) are utilised to re-rank detection proposals based on high-level structures "such as the limbs of animals or robots" ((Kuo et al., 2015), p. 1), which they believe to be excellent indicators of 'objectness'. In their pipeline, an initial set of proposals is generated which is then re-ranked using scores which are calculated by their DeepBox network. Training of their CNN is done by feeding their network ground truth samples as well as hard negatives, both obtained from conventional detection proposal methods such as Edge Boxes. Results show that DeepBox improves upon Edge Boxes in each regime, achieving same recall with a lower number of proposals (Kuo et al., 2015).

The approach employed in this thesis is similar to the DeepBox approach as Edge Boxes is also utilised to obtain the initial detection proposals whereafter these proposals are re-ranked and sorted. The method on which re-ranking is performed, differs however. Where DeepBox re-ranks on high-level structures, detection proposals in this thesis are re-ranked based on their bounding box metadata and flight information. The contents of the bounding boxes are thus not further evaluated, aside from the initial evaluation with Edge Boxes.

4 Methodology

4.1 Dataset

As mentioned in the introduction, the dataset used for the algorithm is obtained from Dutch UAS and exists of drone imagery and a GPS log that holds the messages sent by the drone during the flight, such as attitude and altitude. There are 383 different images in the dataset, which each include approximately 10,000 object detection proposals generated by Edge Boxes. A representative subset of images is shown in appendix A for illustrative purposes. Additionally, there is a ground truth for each image which contains the bounding boxes for the objects of interest. This ground truth is created manually, and also describes the class of the bounding box. In the ground truth there are 901 rhinos, 361 vehicles, 253 persons and 4 elephants. The GPS log contains information about the attitude, altitude and geographic coordinates of the drone. However, in the first two minutes of the log the GPS signal was very weak, resulting in poor and unreliable messages. It is therefore unknown at what timestamp in the GPS log the drone lifted off.

4.2 Pipeline

The algorithm pipeline implemented for reducing the number of detection proposals can be separated into four distinct components.

- **Dataset preparation:** the process of assigning each image its corresponding attitude and altitude.
- **Perspective correction:** the process of transforming each image and its related detection proposals, so that they are normalised.
- **Calculating features:** the process of calculating bounding box features for each detection proposal. Aside from features, the best Intersection-over-Union (IoU) with the bounding boxes from the ground truth is calculated and its corresponding identity (which rhino on which image) is stored.
- **Machine learning:** the process of training a supervised machine learning model in order to predict which detection proposals contain a rhino.

4.2.1 Dataset preparation

The initial plan for assigning the attitude and altitude to each corresponding image, was to synchronise the images and the GPS log by time. However, I experienced that there is an (unknown) time offset between the internal clock of the camera and the GPS clock. I have attempted to approach the offset by comparing the images with the latitude and longitude of the GPS log, however this method proved too unreliable to use. As an alternative approach, I used the *Pix4D* drone mapping software to calculate the attitude and altitude for each

image. Pix4D takes the aerial images as input, after which it calculates both the orientation and the relative position of the drone at the time the images are taken. This is achieved through georeferencing. The graph of the Z-positions, which stands for height in metres, obtained from Pix4D looks similar to the partial graph of the altitude from the GPS log, as can be seen in 4.1. Hence, I sought to approach both lines by fitting a polynomial function which can also be seen in figure 4.1. However, this method resulted in inaccurate perspective transformations. For this reason I utilised the attitude and altitude calculated by Pix4D, which yielded the best transformations.

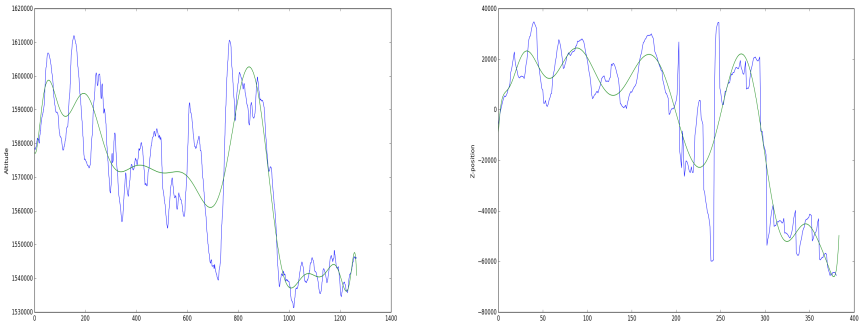
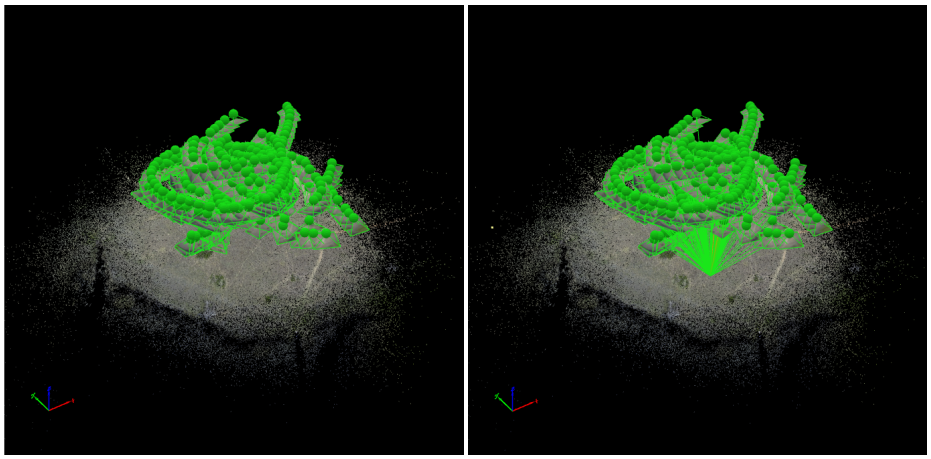


Figure 4.1: Left: the absolute altitude from the GPS log. Right: the relative Z-position from Pix4D. The blue lines illustrates the altitude and the Z-position, respectively. The green line is the fitted polynomial function.



(a) (b)

Figure 4.2: Screenshots from Pix4D. The green dots represent the positions of the drone, the planes are the images represented with their respective orientation. The green lines in (b) from the orthomosaic (point cloud) are pointing to the corresponding reference point on each image.

4.2.2 Perspective correction

For all images, their perspective transformation matrix as in equation 1 is calculated by using their corresponding attitude and altitude as found in Pix4D, which are filled in as parameters d and θ in equations 4 and 6 respectively. The *GoPro Hero3-Black Edition* camera has a horizontal field of view of 120°, which is used to fill in the parameter f from equation 3 for each image. The detection proposals bounding boxes are then multiplied with the inverse transformation matrices, resulting in normalised bounding boxes. To verify that the perspective transformation was done correctly, the transformation method was applied to a calibration image. The result can be observed in figure 4.3.

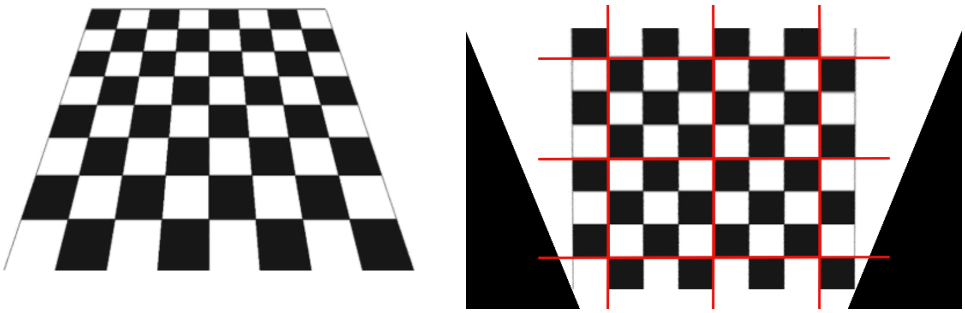


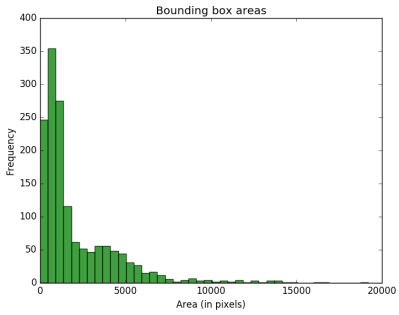
Figure 4.3: Left: image of a checkerboard at a pitch of 25.5°. Right: the transformed image. The horizontal and vertical red lines are parallel to the other horizontal and vertical lines respectively, and are added to demonstrate the correctness of the transformation.

4.2.3 Calculating features

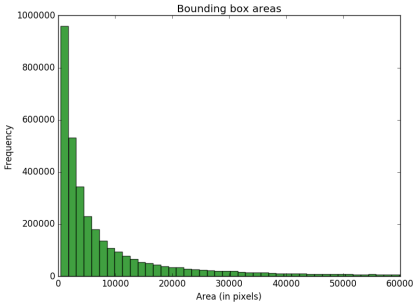
After transformation of the detection proposals' bounding boxes, the features for each bounding box are calculated. Features are important for distinguishing bounding boxes that contain rhinos from bounding boxes that do not contain rhinos. In figure 4.4, the bounding boxes' areas are plotted in a histogram for both the ground truth and all detection proposals. As can be seen in the figure, the bounding box areas for all proposals have a much wider spread than the areas for just the ground truth. It is thus expected that the area of a bounding box is a strong indicator of the probability that the box contains a rhino or not, and will therefore be used as a feature.

All features used are:

1. The area of the bounding box.
2. The aspect ratio of the bounding box's diagonals.
3. The aspect ratio of the angles between the bounding box's diagonals.
4. The position of the bounding box's centre.
5. The detection proposal's confidence score as calculated by Edge Boxes.



(a) Ground truth bounding box areas.



(b) All proposals bounding box areas.

Figure 4.4: Histogram plot of detection proposal bounding box areas.

The aspect ratio of the bounding box's diagonals and the aspect ratio of the angles between these diagonals are used as features, because they resemble the bounding box's form, which is also expected to be useful for distinguishing rhinos from non-rhinos. The reason that the aspect ratio of the diagonals and the angles between the diagonals are utilised as features instead of the ordinary aspect ratio, is that the bounding boxes are transformed from rectangles to quadrilaterals, meaning that their sides are not necessarily of equal length.

The bounding box's position and detection proposal's confidence score as calculated by Edge Boxes are also used as features, since it is expected that there are hidden patterns in these variables. However, it is yet unknown if this is actually the case.

Additionally, the highest IoU with the bounding boxes from the ground truth is calculated for each detection proposal. Together with the IoU, the 'identity' of the bounding box it best overlaps with, is stored to be able to keep track of which ground truth bounding box from which image is already retrieved in order to monitor recall.

4.2.4 Machine learning

The dataset for the machine learning part consists of the detection proposals, their features and their highest IoU. The detection proposals' IoU values are compared with the IoU threshold. If they are greater than or equal to this threshold their class label is set to 1. Else, it is set to 0. The dataset is divided into a training set and a test set with a ratio of 80 to 20 respectively, such that all detection proposals of 80 per cent of the images are in the training set and all detection proposals of the remaining 20 per cent of the images is in the test set. To reduce bias, stratified K-fold cross-validation is used with a number of two folds. The model is then fitted on the training set, making use of weighted classes since there are significant less matching samples (1) than samples that do not match (0). Class weights are calculated as the inverse of the proportion they occupy in the training set, so that the class '1' weighs heavier than the class '0'. Hereafter, the model makes predictions on the probabilities that the detection proposals in the test set belong to the class '1'.

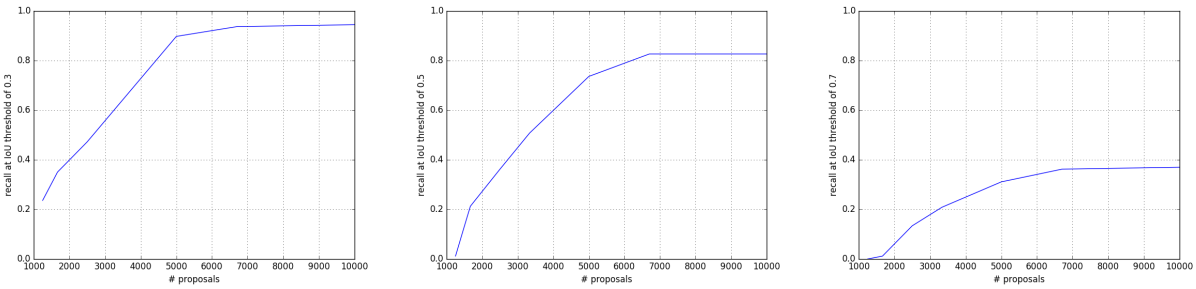
4.3 Evaluation

To evaluate the performance of the algorithm, the recall versus the amount of used proposals is assessed. As stated in Chapter 3.1, a high recall is essential and a low amount of retrieved proposals is desired to make automatic object detection more efficient. The recall is assessed for the IoU thresholds of 0.3, 0.5 and 0.7, as it is expected that these thresholds provide balanced trade-offs between recall and number of proposals, since the values are not too strict or too loose. Furthermore, the AR of IoU values between 0.3 and 0.7 will be evaluated as it is suggested in Hosang et al. (2016) that AR correlates well with detection performance. The recall is calculated by dividing the number of different retrieved rhinos by the total amount of different rhinos in the test set. This is carried out for different random shuffle states, after which the average is taken to reduce bias. In order to assess the added value of the sorting procedure, recall will also be calculated for an unsorted test set.

As mentioned in Chapter 1, the algorithm will be considered successful if the number of detection proposals is reduced by half, while maintaining a recall of at least 0.75.

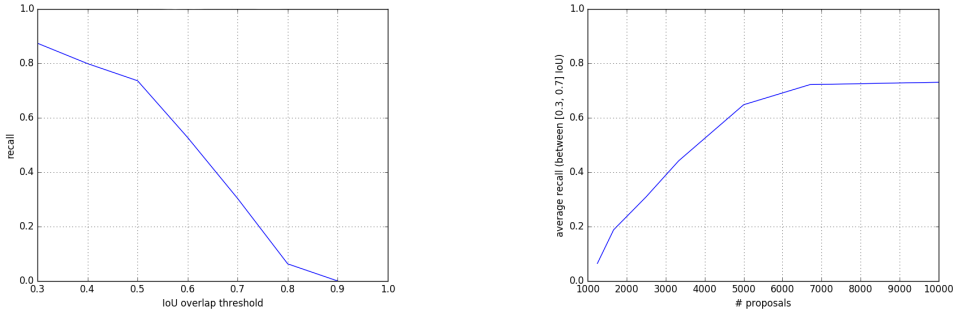
5 Results

The recall of the algorithm is plotted against the number of proposals in figure 5.1. Recall for the initial number of proposals $(10,000)$ is 0.94 for an IoU value of 0.3, 0.83 for IoU value 0.5 and 0.38 for IoU value 0.7. For an IoU of 0.7, recall is below the lowest acceptable value even before re-ranking and sorting detection proposals, making this IoU value inadequate for this thesis. The figure shows that the reduction of the number of proposals by a third (from $10,000$ to $6,667$) has a negligible effect on recall for IoU values of 0.3, 0.5 and 0.7 If the number of proposals is further reduced to $5,000$ proposals, recall decreases to 0.90 for IoU value 0.3. The number of proposals at IoU value 0.3 could even be reduced to as little as $\pm 4,200$ befor recall drops below 0.75. For IoU value 0.5, if the number of proposals is reduced to $5,000$, recall drops to 0.73 which is just below the aforementioned lowest acceptable value. If a recall of 0.75 is to be achieved with an IoU of 0.5, the number of proposals should be slightly higher than $5,000$. Alternatively, if the priority would be to reduce the number of proposals exactly by half while keeping the highest possible overlap, the IoU threshold could be reduced to 0.487, as can be deducted from figure 5.2a.



(a) Recall at 0.3 IoU. (b) Recall at 0.5 IoU. (c) Recall at 0.7 IoU.

Figure 5.1: Recall versus number of proposals for different IoU values.



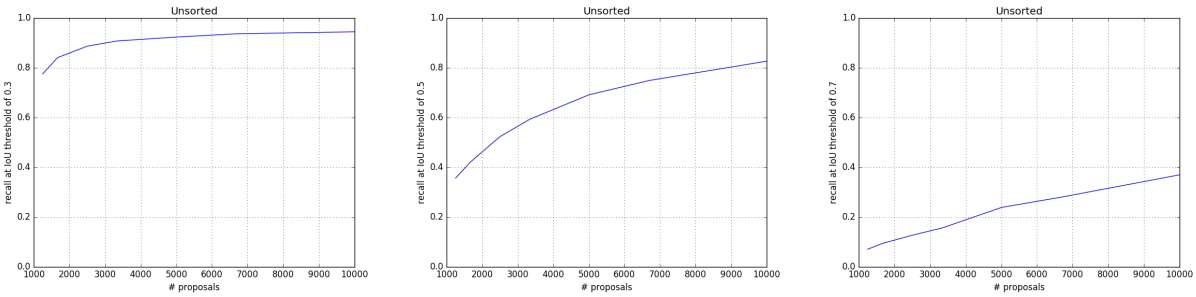
(a) Recall at 5,000 proposals per image. (b) Average recall (between [0.5, 1.] IoU).

Figure 5.2: Left: recall versus IoU. Right: recall versus number of proposals.

Figure 5.2a shows the recall at half of the initial number of proposals plotted against the IoU overlap threshold. The recall of 0.75 is exceeded between the IoU values of 0.4 and 0.5, but closer to 0.5. The AR is plotted against recall in figure 5.2b and shows that the recall decreases significanly when the number of proposals is reduced by more than half for IoU values between 0.5 and 1.0.

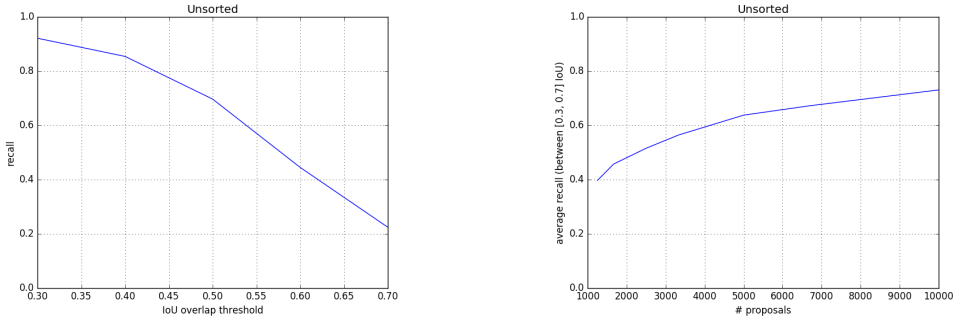
Recall for the test set which has not been sorted on its probabilities of samples belonging to class 1, is plotted in figures 5.3 and 5.4. The lines in figure 5.3 decrease more rapidly than the lines in figure 5.1, when the number of proposals are reduced by half to $5,000$. At the IoU threshold of 0.5, the recall boundary of 0.75 is crossed at a higher number of proposals when they have not been sorted on their probabilities (figure 5.3b) than when these proposals have been sorted (figure 5.1b). Thus, the number of proposals can be reduced further when sorting is performed than when sorting is not performed at an IoU of 0.5.

A very remarkable observation is that the lines in figure 5.3 decrease significantly less than the lines in figure 5.1 when the number of proposals are further reduced than by half.



(a) Recall at 0.3 IoU. (b) Recall at 0.5 IoU. (c) Recall at 0.7 IoU.

Figure 5.3: Recall versus number of unsorted proposals for different IoU values.



(a) Recall at 5,000 unsorted proposals per image. (b) Average recall (between [0.5, 1.] IoU).

Figure 5.4: Left: recall versus IoU. Right: recall versus number of proposals.

6 Conclusion & Discussion

As discussed in Chapter 1, conservation drones have great potential for automatic monitoring of wildlife. However, conventional object detection methods can not be directly applied to drone imagery. The purpose of this thesis is to make automatic object detection more suitable for wildlife conservation. This is attempted by generating detection proposals with the Edge Boxes method, which are transformed and normalised, so that bounding boxes that contain a rhino are approximately the same size. Features are then calculated and a logistic regression model is trained to predict the probability of a detection proposal enclosing a rhino.

The aim of this thesis is to reduce the amount of detection proposals while maintaining recall. If the amount of detection proposals could be reduced by half while maintaining a recall of at least 0.75, the thesis goal would have been achieved. This goal is achieved for the IoU value of 0.3, having a recall of 0.90 at half the initial amount of proposals. For an IoU value of 0.5, recall is just below minimum acceptable level at 0.73. To comply to the aim of maintaining a recall of at least 0.75, the IoU could be reduced to a value slightly lower than 0.5. Recall for the IoU value of 0.7 is far below the threshold of 0.75 even before the amount of proposals are reduced, making this IoU value inadequate for this thesis. Since a higher IoU threshold means a higher overlap and thus more accurate detection proposals, it is suggested to take the highest possible IoU value that has a recall of 0.75 of higher, which is 0.487.

The observation that recall for unsorted test sets declines less than for sorted test sets when the amount of proposals is reduced by more than half, is remarkable and requires further investigation. My suggestion is that the logistic regression model generates numerous anomalies with a probability close to 1, while they actually belong to class 0. This is only a hypothesis and should be confirmed in future research.

As mentioned in Chapter 2.2, it is unclear whether bounding box normalisation through perspective transformation improves the comparison of bounding boxes in a better way than a simpler normalisation approach. Such an approach could be where the bounding box area is the only feature which is normalised, by dividing its value by its corresponding height. If this does not decrease performance of bounding box comparison, it could be used to make automatic object detection for wildlife conservation drones more efficient. Yet, the impact such an approach would have on the algorithm should be evaluated in future research.

In future work, the performance of this algorithm could be compared for different machine learning models. In this thesis, logistic regression is used to predict bounding box probabilities, yet there are numerous classification models, for example Gaussian Naive Bayes. Evaluating different machine learning models might also improve the steep decline in recall when the amount of proposals is reduced by more than half, given that my hypothesis regarding this issue is proven to be correct.

To conclude, this thesis has demonstrated that it is possible to increase automatic object detection efficiency by reducing the amount of object detection proposals by half, while having minor impact on recall. Yet, this is a young field of science and to make automatic object detection fully functional for wildlife conservation, so that animals can be monitored with less effort, more research is required to further improve efficiency and address the issues encountered in this thesis.

References

Hartley, R. & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Hosang, J., Benenson, R. & Dollár, P. (2016). What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, *38*(4), 814–830.

Hosang, J., Benenson, R. & Schiele, B. (2014). How good are detection proposals, really? *arXiv preprint arXiv:1406.6962*.

Kleinbaum, D. G. & Klein, M. (2010). *Logistic regression: a self-learning text* (Third Edition). Springer Science & Business Media.

Kuo, W., Hariharan, B. & Malik, J. (2015). Deepbox: learning objectness with convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2479–2487.

Van Andel, A., Wich, S., Boesch, C., Koh, L., Robbins, M., Kelly, J. & Kuehl, H. (2015). Locating chimpanzee nests and identifying fruiting trees with an unmanned aerial vehicle. *American Journal of Primatology*, *77*(10), 1122–1134.

Van Gemert, J. C., Verschoor, C. R., Mettes, P., Epema, K., Koh, L. P. & Wich, S. (2014). Nature conservation drones for automatic localization and counting of animals. In *Computer vision-eccv 2014 workshops* (pp. 255–270). Springer.

Verschoor, C. (2016). *Conservation drones for animal monitoring* (Master's thesis, University of Amsterdam).

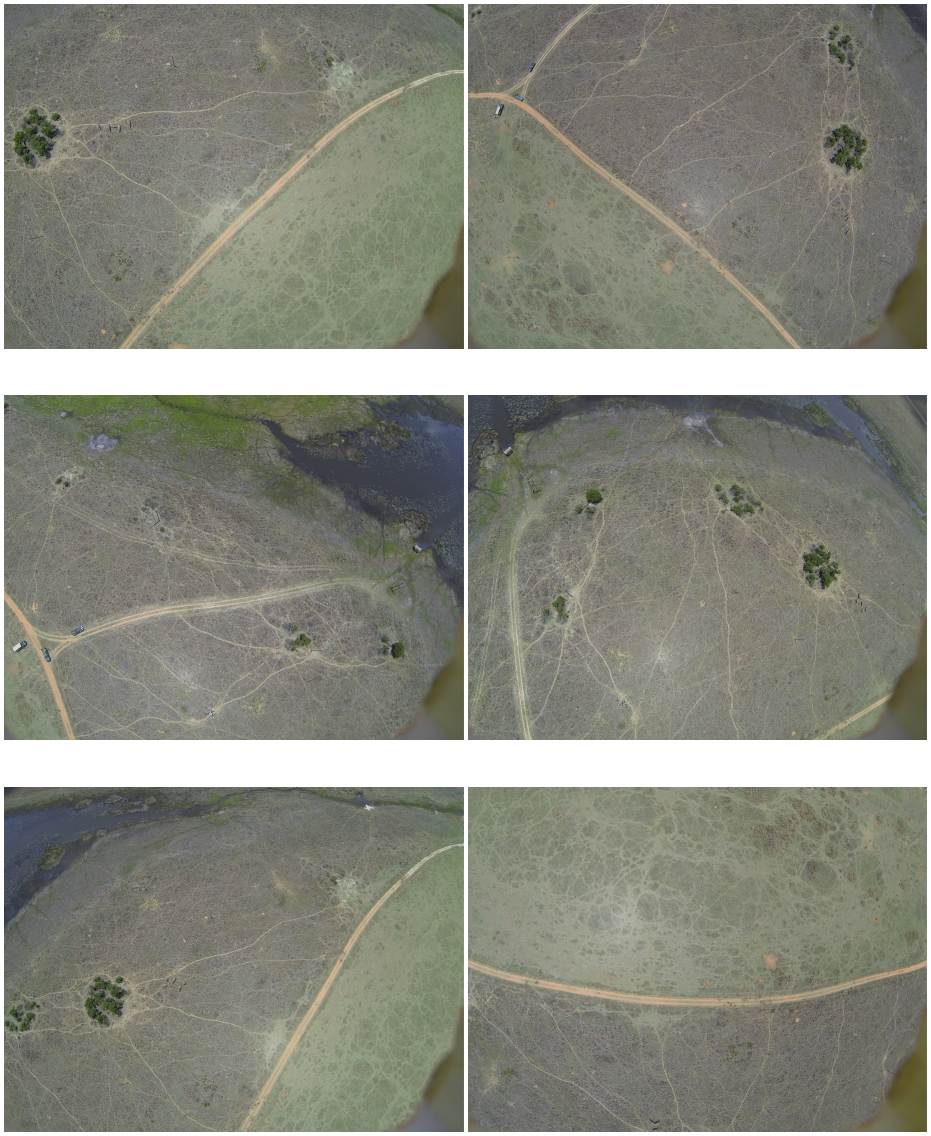
Vitousek, P. M., Mooney, H. A., Lubchenco, J. & Melillo, J. M. (1997). Human domination of earth's ecosystems. *Science*, *277*(5325), 494–499.

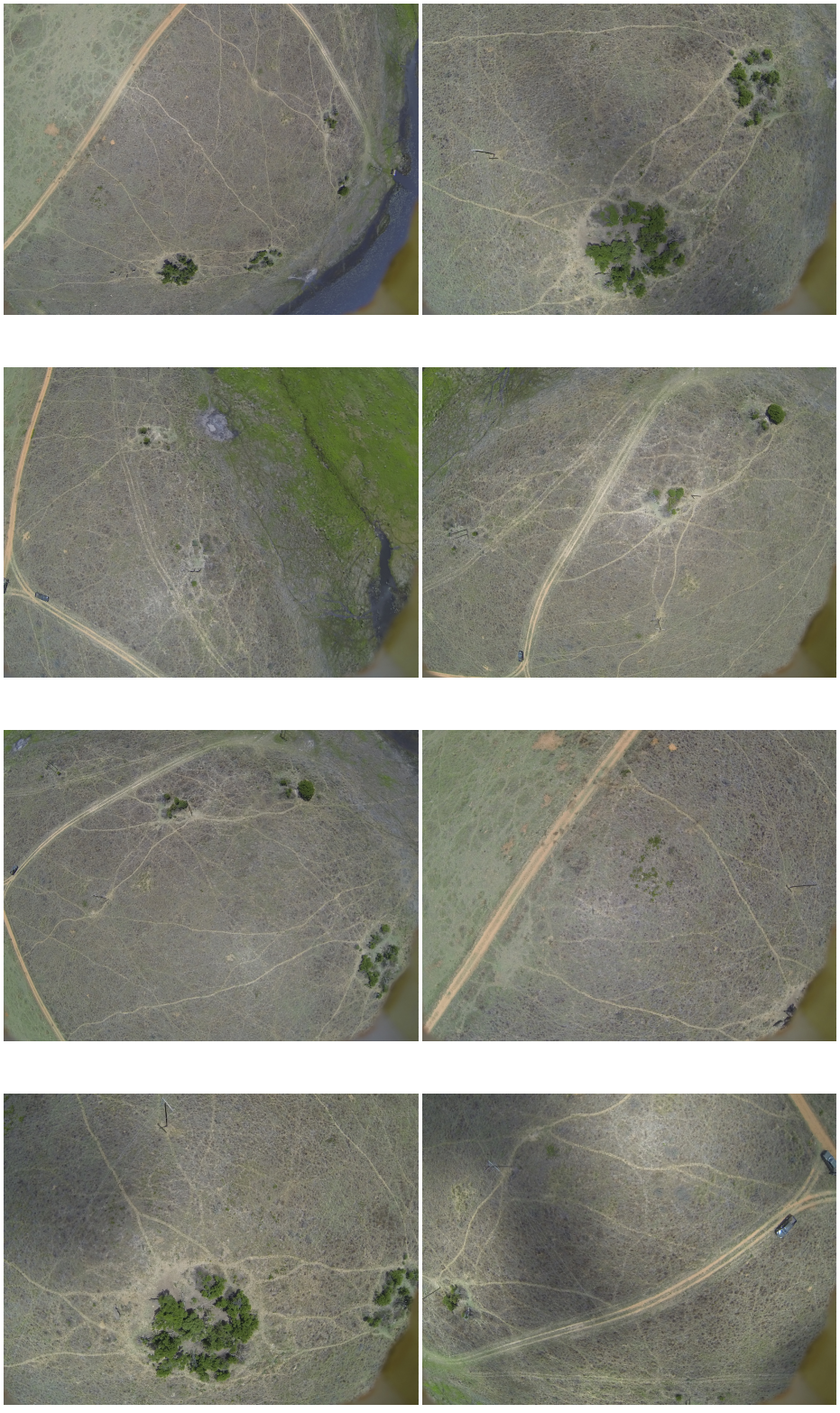
Wich, S., Dellatore, D., Houghton, M., Ardi, R. & Koh, L. P. (2015). A preliminary assessment of using conservation drones for sumatran orang-utan (pongo abelii) distribution and density. *Journal Unmanned Vehicle Systems 4: 45–52 (2016) dx.doi.org/10.1139/juvs-2015-0015*.

Zitnick, C. & Dollár, P. (2014). Edge boxes: locating object proposals from edges. *Computer Vision–ECCV 2014*, 391–405.

Appendices

A Aerial image examples





B Code

Pseudocode describing how machine learning is applied and how recall is calculated (actual code is in Python):

```
X = array of feature values
y = array of iou values

for all iou in y:
    if iou >= iou_threshold:
        assign to class '1'
    else:
        assign to class '0'

split X,y in training set and test set
shuffle rows of training set and test set

fit model on training set
predict probabilities of test set

sort predicted probabilities in descending order
delete lower ranked proposals from test set

total = 0
detected = 0
id = dictionary of rhino identities
for all identities in id:
    if identity is in test set:
        total += 1
        if identity's predicted class == '1':
            detected += 1

recall = detected / total
```