Background Categorization for Automatic Animal Detection in Aerial Videos Using Neural Networks

Yunfei Fang[1(✉)], Shengzhi Du[2], Rishaad Abdoola[1], and Karim Djouani[1]

[1] Department of Electrical Engineering, French South African Institute of Technology, Tshwane University of Technology, Staatsartillerie Road, Pretoria 0001, South Africa
fangyunfei08@gmail.com

[2] Department of Mechanical Engineering, Mechatronics and Industrial Design, Tshwane University of Technology, Staatsartillerie Road, Pretoria 0001, South Africa
DuS@tut.ac.za

Abstract. This paper addresses the problem of animal detection in natural environment from aerial videos. Since the natural environment is usually composed of several fundamental elements such as trees, grass, streams, etc., it is proposed to distinguish the animal by categorizing the background into several classes. From the manually labeled samples, texture as well as brightness features are extracted to train a feedforward Neural Network. Then the classifier is applied to filter the test frame to locate potential animal regions. Four texture measures calculated from Grey Level Co-occurrence Matrix (GLCM) are used for texture feature description. Instead of obtaining these texture measures from grey level images, it is proposed to carry out calculation for every channel of the RGB image. The implemented results illustrate that this feature extraction method works well and the texture feature is a decisive factor in background categorizing.

Keywords: Image segmentation · Background categorization · Texture analysis · Animal detection · Neural network

1 Introduction

1.1 Animal Detection in a Natural Environment

Applications of Unmanned Aerial Vehicles (UAVs) are of great benefit to the field of nature conservation. Compared to conventional ways of wildlife surveys, it is far more economical to collect information from the region by flying a UAV (mounted with a proper camera) across it. It is especially efficient in some real-time tasks, such as monitoring and anti-poaching. The possibility of applying computer vision techniques to automatically analyse the aerial videos are increasingly being investigated [1, 2].

In general, tasks that wildlife conservation may concern include: (1) Counting the number of animals to monitor the distribution and abundance of animal species.

F. Schwenker et al. (Eds.): ANNPR 2016, LNAI 9896, pp. 220–232, 2016.
DOI: 10.1007/978-3-319-46182-3_19

(2) Identification and investigation of a particular animal. (3) Tracking and monitoring of a herd and risk estimation. Animal detection in the video is to decide whether or not an animal of specific species is present in the scene and where it is located.

Most object detection algorithms are based on machine learning mechanisms. Different views of an object were learned by a set of classifiers using positive and negative examples. By dividing the image into standard-sized sub-windows (patch) and passing them through trained filters, it can be then determined the existence and location of the object. For example, the neural networks [3], support vector machines [4], Bayesian networks [5] and deep learning approaches [6]. Neural networks play very important role in pattern recognition and have a wide range of applications in pedestrian detection [7], speech recognition [8], handwriting recognition [9], fingerprint analysis [10] and so on.

The modern object detection and classification tasks usually cope with ground perspective images with the scale of the object occupying the greatest portion of the image. The problem of animal detection in natural environment from aerial images has not significantly been addressed.

Based on the nature of the problem, it is proposed not merely learn the pattern of the animal but learn the pattern of background alongside by categorizing it into different classes. A feedforward neural network is trained to form a classifier because its architecture is prone to multi-class classification. Texture features are extracted as input for the neural network. By sliding the neural network filter across the testing video frame, the regions where animals exist are successfully highlighted.

1.2 Background Categorization

There are two hypotheses behind this research. Firstly, there are finite visual elements that a general natural environment can be divided into. Secondly, the classes, both the animal and subclasses of the background can be distinguished by Neural Network applying proper feature descriptor. It is proposed to us texture features along with brightness to discriminate different classes in the scene.

Texture describes the content of the object surface. The surface of an object can be considered as a composition of elementary structures, and it is made recognizable due to the balanced presence of some specific elementary structures in it.

Texture feature description is a key step in deciding performance of the Neural Network classifier. There are several texture feature extractors [5] such as Grey-Level Co-occurrence Matrices (GLCM), Local Binary Pattern operator (LBP), Local Phase Quantization (LPQ), and Gabor filters. GLCM measures some statistical characteristics of an image. It has proven to be a very powerful basis for texture feature extraction and has been widely used in areas such as texture classification, remote sensing, medicine, biology and agriculture [11–14].

The GLCM and the four texture measures calculated from it are used as texture feature descriptor to train the Neural Network classifier. It is proven to be effective in background categorization as well as animal detection.

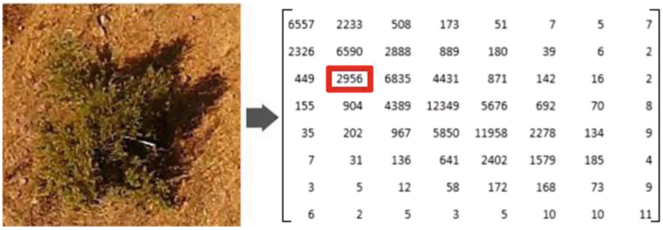
2 Theoretical Background

2.1 GLCM

The Grey Level Co-occurrence Matrix (GLCM) is often used for a series of “second order” texture calculation, which considers pixel neighbouring relationships. It measures how often different combinations of pixel brightness values or grey levels occur in an image. GLCM is based on the relationship between two pixels. The spatial relationship between a reference pixel and its neighbour pixel defines the offset.

To reduce the size of the GLCM and increase the occupancy level of the matrix, the quantization level is set to 16.

To demonstrate the GLCM data structure, the image (300 × 300 pixels) shown in Fig. 1 is used as an example. The corresponding GLCM data is depicted in Fig. 1, where the grey level is set to 8 for convenient display. The highlighted cell at row 3, column 2 means the combination 3, 2 occurs 2956 times in the image, where the grey level of the reference pixel is 3 and its immediate neighbour on the left (defined by the offset) is 2.



$$\begin{bmatrix} 6557 & 2233 & 508 & 173 & 51 & 7 & 5 & 7 \\ 2326 & 6590 & 2888 & 889 & 180 & 39 & 6 & 2 \\ 449 & \boxed{2956} & 6835 & 4431 & 871 & 142 & 16 & 2 \\ 155 & 904 & 4389 & 12349 & 5676 & 692 & 70 & 8 \\ 35 & 202 & 967 & 5850 & 11958 & 2278 & 134 & 9 \\ 7 & 31 & 136 & 641 & 2402 & 1579 & 185 & 4 \\ 3 & 5 & 12 & 58 & 172 & 168 & 73 & 9 \\ 6 & 2 & 5 & 3 & 5 & 10 & 10 & 11 \end{bmatrix}$$

Fig. 1. An example image and its corresponding GLCM

2.2 Texture Measures from the GLCM

Texture calculations are weighted averages of the normalized GLCM cell contents. Certain measures are based on the contrast information while others are based on the orderliness and the descriptive statistics of the GLCM texture measures. The following describes only the texture measures from GLCM which are used in this paper

$$Cn = \sum_i \sum_j |i - j|^2 P(i,j) \tag{1}$$

where Cn is the measure of contrast, $P\,(i, j)$ is the content of the normalized GLCM at row i and column j, illustrating the occurrence probability of gray level i and j.

$$Co = \sum_i \sum_j \frac{(i - \mu_i)(j - \mu_j)P(i,j)}{\sigma_i \sigma_j} \tag{2}$$

where Co is the measure of correlation, $\mu_{i,}$ $\mu_{j,}$ σ_i, σ_j are the means and standard deviations respectively. Correlation is a measure of how correlated a pixel is to its neighbour over the whole image. A value of 0 implies that the pattern is uncorrelated, 1 implies perfect correlation and −1 implies that the spatial set exhibits a dissimilar, deterministic structure.

$$E = \sum_i \sum_j P(i,j)^2 \tag{3}$$

where E is the measure of energy. Energy will be equal to 1 for a constant image.

$$H = \sum_i \sum_j \frac{P(i,j)}{1 + |i - j|} \tag{4}$$

where H is the measure of homogeneity.the closeness of the distribution of elements in the GLCM to the GLCM diagonal. In the homogeneity measure, the weight values are the inverses of the contrast weight values. As we move further away from the diagonal, the weights decrease quadratically.

3 Proposed Method

Figure 2 shows the flow chart of the proposed method. With the aerial video captured from a natural environment, it is at the first place to investigate the visual elements existing in the scene so that the animal and elements of the background can be assigned a class name. Training data of the Neural Network classifier is obtained by taking some frames from the video. These video frames will be partitioned into sub-windows of standard size, which are also called “patches”. A set of patches for each class will be achieved through manual labelling. The classes will include the animal and subclasses of the background. Features extracted from the patch are used as input for the Neural Network classifier.

In the testing procedure, for each frame of the video clip, a window of proper size will slide end-to-end across the whole image. The window region from the image will be taken and GLCM and texture measures will be calculated. The Neural Network classifier will assign each patch a class name. Patches classified as animal will be

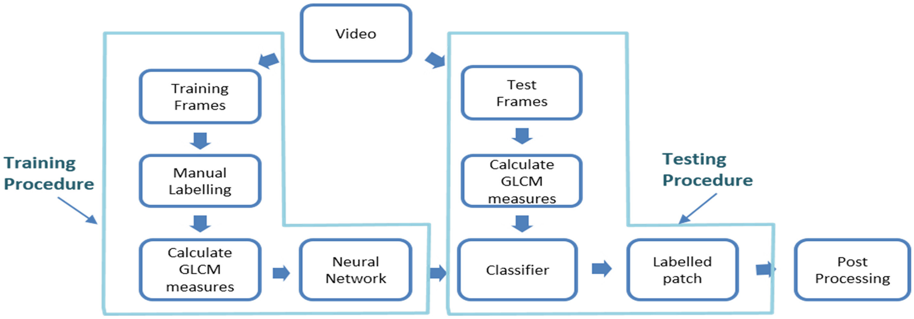


Fig. 2. Flow chart of the proposed method

highlighted to form the detection. Further processing such as morphological operations can be applied to reduce errors and improve the results.

The testing video is captured from a typical natural environment in South Africa. The original frame size is 1080 × 1920 pixels. Animals existing in the video are mostly blesboks. The method is implemented on this video for a detailed explanation.

3.1 Selection of Patch Size

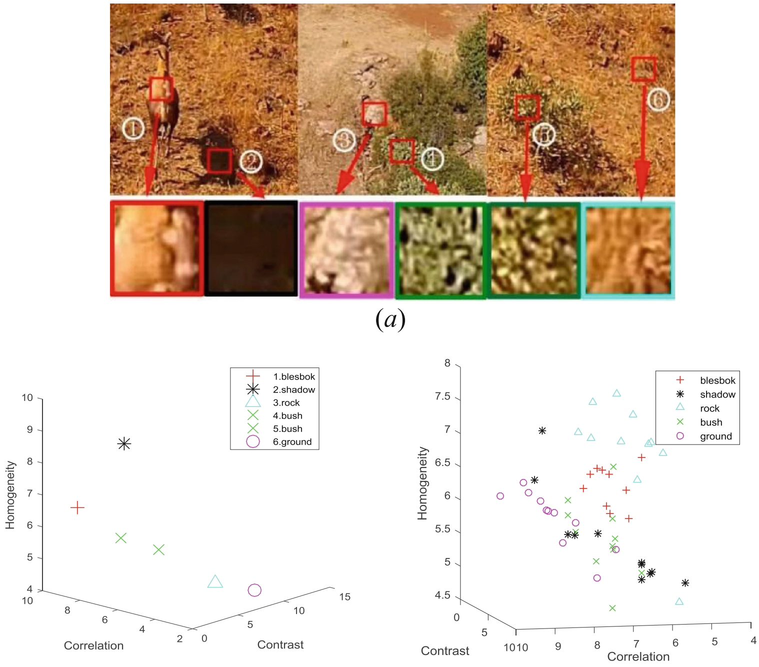
The resolution and scale of the patch make the texture of objects' surfaces different, especially for an animal, which has a clear silhouette. Empirically, the training and testing patch size chosen here is 40-by-40 pixels. One reason is that in the testing video, texture in this size is discernible and can be described properly. Another reason is that, under such setup, animals shall contain hundreds of pixels, and an animal in this size will consist of several patches, it is more stable than forming a detection using a single patch. For comparison, results of 100-by-100-pixel patch size are illustrated in the next section.

It is left to the future work to train a classifier for animals that are viewed from a distance by supplying enough relevant training examples, or using adaptive patch sizes according to the estimated distance from the camera to the animals.

3.2 Background Categorization

The required number of classes needed for the Neural Network classifier depends on the classification performance. It will be sufficient to train a two-class (animal and background) classifier as long as the performance is good enough. Categorizing the background into different classes will help form a background model, so that the ani-mal is distinguished if it does not match any subclasses of the background. Figure 3(a) shows some example textures of different elements captured from different scenes. Potential classes are placed in bounding boxes of different colours. Notice that the 4th and 5th patches exhibit similar texture patterns, though the environment looks different. In Fig. 3(b), 3-D points representing texture features of each patch are plotted. The point value $P = [x, y, z]^T$ is derived by calculating three texture measures (Contrast, Correlation, Homogeneity) of the GLCM of each patch. It can be seen that the distance between different elements, and the measures of the 4th and 5th patch are quite close.

In our testing video, five classes are finally selected which are blesbok, bush, ground, gravel and shadow respectively. It is worth mentioning that the class "shadow" (as the second patch in Fig. 3(a)) is not a real object in the background, but an "element" that shares a common texture pattern. This categorization may not include all the elements in the scene, but is sensible and easy for training and testing. Figure 3(c) takes 10 manually assigned patches of each class and visualizes the distance between the texture measures of the classes. The point values are derived in the same way as in Fig. 3(b).



(*a*)

(*b*) (*c*)

Fig. 3. (*a*) Texture of different elements and potential categorization (*b*) Texture measures of the six patches (*c*) Example texture measures from every class

3.3 Feature Extraction

Feature extraction is essential to the classification performance. As described before, the GLCM is a 16-by-16 matrix derived from the specific patch. The texture measures calculated from the GLCM can be used to describe the texture feature:

$$F1 = [Cn, Co, E, H]^T \tag{5}$$

where Cn, Co, E, H are Contrast, Correlation, Energy and Homogeneity respectively.

As the GLCM is calculated on grey-scale level, it might neglect some useful information in the colour domain. In this case, it is proposed to calculate the GLCM and texture measures for every single channel of the RGB image. The descriptor vector is:

$$F2 = \left[Cn_r, Cn_g, Cn_b, Co_r, Co_g, Co_b, E_r, E_g, E_b, H_r, H_g, H_b\right]^T \tag{6}$$

where the subscripts r, g, b represents corresponding measures calculated from the 3 channels respectively. This feature extractor for 3 colour channels will be proven to have a higher distinguishing capacity compared to the traditional GLCM.

As shown in Fig. 2(c), the black asterisks representing the class shadow seem to not be distinguishable enough from other classes. Herein, brightness feature is introduced by calculating the average intensity of the grey level image:

$$F3 = \left[I, Cn_r, Cn_g, Cn_{b,}Co_r, Co_g, Co_b, E_r, E_g, E_b, H_r, H_g, H_b\right]^T \tag{7}$$

where I is the average intensity of the grey level image. In the next section, the feature descriptor mentioned will be compared, as well as describing the brightness feature by calculating the average intensity of 3 channels of the colour image. The feature descriptor for the Neural Network will be a 15-by-1vector, denoted as:

$$F4 = \left[I_r, I_g, I_b, Cn_r, Cn_g, Cn_{b,}Co_r, Co_g, Co_b, E_r, E_g, E_b, H_r, H_g, H_b\right]^T \tag{8}$$

where I_r, I_g, I_b are the average intensity of the red, green and blue channel of the colour image respectively.

3.4 Neural Network

In this work, a tow-layer feedforward neural network with a sigmoid transfer function in the hidden layer and a softmax transfer function in the output layer classified the vectors well. The diagram of the neural network structure is shown in Fig. 4. Vectors of size 15-by-1 are inputs for the network. The 5 output neurons correspond to the 5 target classes. Experiments showed that 10 neurons in the hidden layer generates promising results. Most of the training algorithms do not have significant influence on estimation performance. And the training time and number of iterations do not matter much with such small data. The results illustrated in the next section are trained using the learning algorithm: Scaled Conjugate Gradient (SCG), as described in [15].

In the test, 25 frames are selected from the video sequence, after manually labellingm the counts of the input data for each class are 322, 684, 1244, 524, 244 (blesbok, bush, ground, gravel, shadow). 70 % of the data was divided for training, 15 % for validation, and the last 15 % for testing.

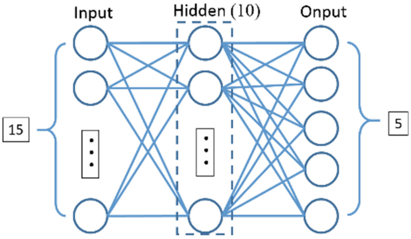
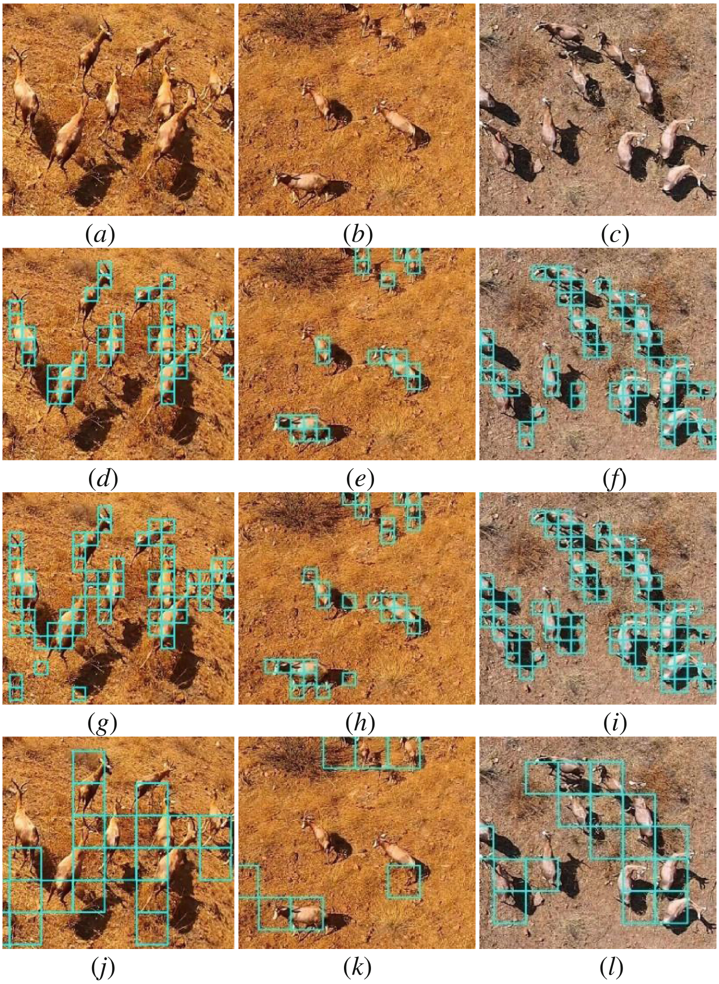


Fig. 4. Neural network structure diagram

4 Results and Discussion

For illustration purposes, Fig. 5(a), (b) and (c) are chosen from the testing video to demonstrate the results. The animals are from the same herd in subsequent frames of the testing video, but postures and sizes of the blesboks in Fig. 5(b) are different from (a), and the saturation in (c) is different from (a) and (b). These 3 scenarios are chosen to depict the effects of animal postures and the size on the proposed method as well as changes in lighting conditions. Inputs for the Neural Networks related in Fig. 5 are the same, which are texture measures calculated from R, G, B channels of the patch ($F2$ in (6)).



(*a*) (*b*) (*c*)

(*d*) (*e*) (*f*)

(*g*) (*h*) (*i*)

(*j*) (*k*) (*l*)

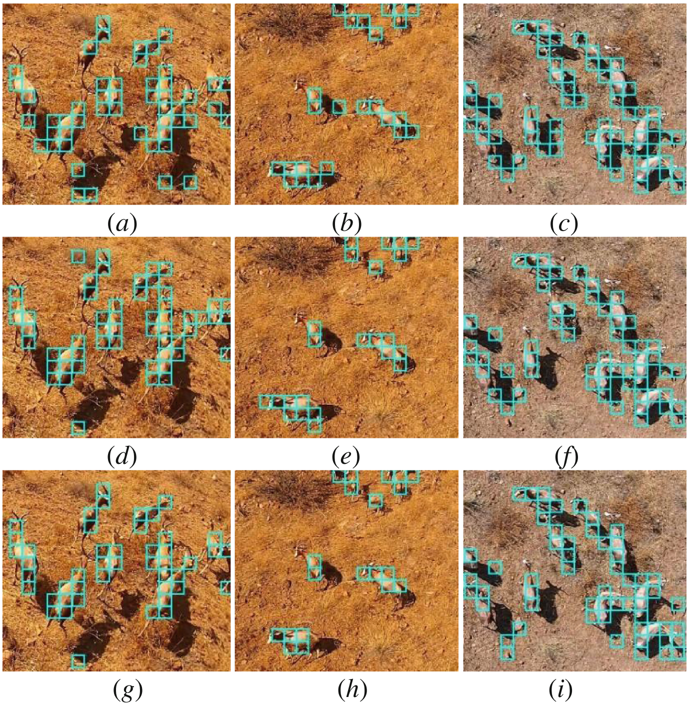
Fig. 5. Detection results when only take the texture measures from 3 colour channels as descriptor ($F2$ in (6)). (*a*), (*b*), (*c*) original images (*d*), (*e*), (*f*) detection results in 5 classes (*g*), (*h*), (*i*) detection results in 2 classes (*j*), (*k*), (*l*) detection results of 100-by-100 patch size and in 3 classes

Figure 5(d), (e) and (f) show the detection results in 5 classes (blesbok, bush, ground, gravel, shadow) and the 40-by-40 patch size as described in the previous section. The results look promising with nearly no part of the animal excluded from detection. Most parts of the animals' body are highlighted and due to the selection of the patch size, most of the animals are covered by at least 2 bounding boxes. This will make it easy to filter out false detections by requiring a minimum number of hits within region. Different postures of the animals have little effect on the detection performance. Variation in saturation also doesn't influence the detection of the animals much but the portion of background classified as animals does increase, as shown in (f).

Figure 5(g), (h) and (i) are the detection results of categorizing the output to 2 classes (blesbok, background) with the selected 40-by-40 patch size. With all the elements besides blesbok treated as "background", the results are obviously not as neat as in 5 classes, with more of the background being classified as part of the animal class.

Figure 5(j), (k) and (l) are the detection results in 3 classes (blesbok, background and shadow) using a 100-by-100 patch size. Using this patch size, a single detection of an animal will not contain more than 1 bounding box making it difficult to erase the false positive detection and the animal is easily lost, making the result noisy.

Figure 6 shows detection results of different feature extraction methods. The original images are the same as in Fig. 4, and output of the Neural Network classifier



(*a*) (*b*) (*c*)

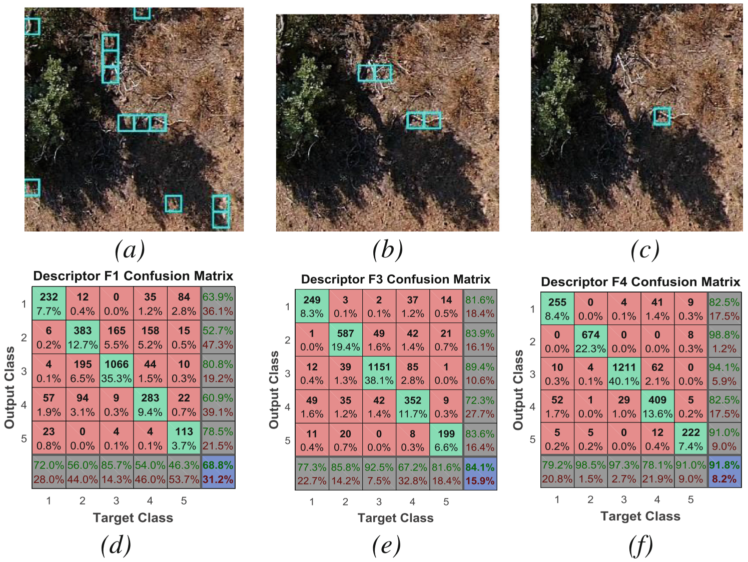
(*d*) (*e*) (*f*)

(*g*) (*h*) (*i*)

Fig. 6. Detection results all in 5 classes with different feature extraction methods. (*a*), (*b*), (*c*) 4 texture measures calculated from GLCM ($F1in(5)$) (*d*), (*e*), (*f*) 12 texture measures with average intensity of grey level image ($F3in(7)$) (*g*), (*h*), (*i*) 12 texture measures with average intensity of each channel ($F4in(8)$).

consists of 5 classes. Figure 6(a), (b) and (c) take the 4 texture measures of the GLCM as input descriptor, as $F1$ in (5). Input descriptor vector for Fig. 6(d), (e) and (f) is $F3$ in (7), and $F4$ in (8) for Fig. 6(g), (h) and (i). Results generated by using Neural Network input vectors $F1, F2, F3, F4$ will be referred to as method I, II, III and IV respectively.

Introduction of brightness feature makes the class “shadow” more distinguishable from other classes. Difference is obvious in scenes without animals. Figure 7(a), (b) and (c) shows false positive detections using method I, III, IV separately. In Fig. 7(a), shadows of the bush are totally not distinguished from the class “blesbok”. Results are promoted by introducing the average pixel intensity as input descriptor as in Fig. 7(b). Figure 7(c) achieves the best performance, where average intensity values of R, G, B



Descriptor F1 Confusion Matrix

	1	2	3	4	5	
1	232 7.7%	12 0.4%	0 0.0%	35 1.2%	84 2.8%	63.9% 36.1%
2	6 0.2%	383 12.7%	165 5.5%	158 5.2%	15 0.5%	52.7% 47.3%
3	4 0.1%	195 6.5%	1066 35.3%	44 1.5%	10 0.3%	80.8% 19.2%
4	57 1.9%	94 3.1%	9 0.3%	283 9.4%	22 0.7%	60.9% 39.1%
5	23 0.8%	0 0.0%	4 0.1%	4 0.1%	113 3.7%	78.5% 21.5%
	72.0% 28.0%	56.0% 44.0%	85.7% 14.3%	54.0% 46.0%	46.3% 53.7%	68.8% 31.2%

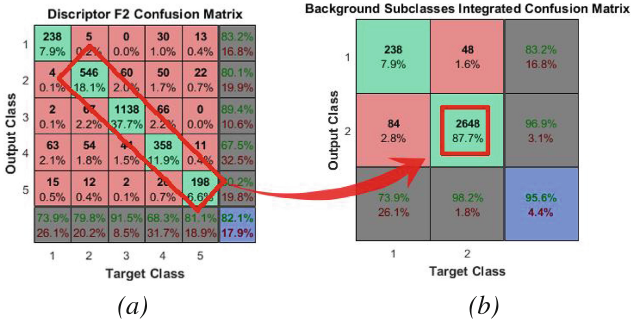
Descriptor F3 Confusion Matrix

	1	2	3	4	5	
1	249 8.3%	3 0.1%	2 0.1%	37 1.2%	14 0.5%	81.6% 18.4%
2	1 0.0%	587 19.4%	49 1.6%	42 1.4%	21 0.7%	83.9% 16.1%
3	12 0.4%	39 1.3%	1151 38.1%	85 2.8%	1 0.0%	89.4% 10.6%
4	49 1.6%	35 1.2%	42 1.4%	352 11.7%	9 0.3%	72.3% 27.7%
5	11 0.4%	20 0.7%	0 0.0%	8 0.3%	199 6.6%	83.6% 16.4%
	77.3% 22.7%	85.8% 14.2%	92.5% 7.5%	67.2% 32.8%	81.6% 18.4%	84.1% 15.9%

Descriptor F4 Confusion Matrix

	1	2	3	4	5	
1	255 8.4%	0 0.0%	4 0.1%	41 1.4%	9 0.3%	82.5% 17.5%
2	0 0.0%	674 22.3%	0 0.0%	0 0.0%	8 0.3%	98.8% 1.2%
3	10 0.3%	4 0.1%	1211 40.1%	62 2.1%	0 0.0%	94.1% 5.9%
4	52 1.7%	1 0.0%	29 1.0%	409 13.6%	5 0.2%	82.5% 17.5%
5	5 0.2%	5 0.2%	0 0.0%	12 0.4%	222 7.4%	91.0% 9.0%
	79.2% 20.8%	98.5% 1.5%	97.3% 2.7%	78.1% 21.9%	91.0% 9.0%	91.8% 8.2%

Fig. 7. (*a*), (*b*), (*c*) False positive detections of discriptor $F1, F3, F4$ seperately (*d*), (*e*), (*f*) Classification confusion matrix of discriptor $F1, F3, F4$ seperately.



Discriptor F2 Confusion Matrix

	1	2	3	4	5	
1	238 7.9%	5 0.2%	0 0.0%	30 1.0%	13 0.4%	83.2% 16.8%
2	4 0.1%	546 18.1%	60 2.0%	50 1.7%	22 0.7%	80.1% 19.9%
3	2 0.1%	67 2.2%	1138 37.7%	66 2.2%	0 0.0%	89.4% 10.6%
4	63 2.1%	54 1.8%	44 1.5%	358 11.9%	11 0.4%	67.5% 32.5%
5	15 0.5%	12 0.4%	2 0.1%	20 0.7%	198 6.6%	80.2% 19.8%
	73.9% 26.1%	79.8% 20.2%	91.5% 8.5%	68.3% 31.7%	81.1% 18.9%	82.1% 17.9%

Background Subclasses Integrated Confusion Matrix

	1	2	
1	238 7.9%	48 1.6%	83.2% 16.8%
2	84 2.8%	2648 87.7%	96.9% 3.1%
	73.9% 26.1%	98.2% 1.8%	95.6% 4.4%

Fig. 8. Neural network classification confusion matrix *(a)* confusion matrix of descriptor *F2 (b)* confusion matrix of 4 background subclasses taken as background.

channels are calculated separately. Method IV also achieves the best classification accuracy as demonstrated by the confusion matrix in Fig. 7(d), (e) and (f).

The performance of method I using traditional GLCM for intensity image is far from satisfactory as depicted in the confusion matrix in Fig. 7(a). But performance of applying GLCM and corresponding texture measures in every channel of the colour image as proposed is promising. Comparison can be made between Figs. 7(a) and 8(a), which shows the classification confusion matrix of method II, where $F2$ is taken as the descriptor.

It is worthy to mention that the other 4 classes besides "blesbok" shall be taken as one single class "background" as shown in Fig. 8(b). Because the misclassification between the background elements doesn't affect detection of the animals. In Fig. 8(b), the false negative rate of "16.8 %" means that 16.8 percent of the "blesbok" patches are classified as "background". This will not affect the performance significantly if the constraint that each animal is supposed to consist of several patches of the selected size. The true positive rate "83.2 %" is fine enough to catch major parts of the animal's body and make a high rate of true positive in the animal scale. Unequal number of training examples from the 2 classes makes the false discovery rate "26.1 %" quite large. Generally, most of the false positive errors are isolated patches as shown in Fig. 9 that are easily reduced by restricting the number of hits in the region.

As seen in Fig. 8(a), there seems to be more misclassifications between the class "gravel" and "blesbok". To improve the performance, additional features should be included to make the two classes more discriminative, or target classes shall be reinvestigated.

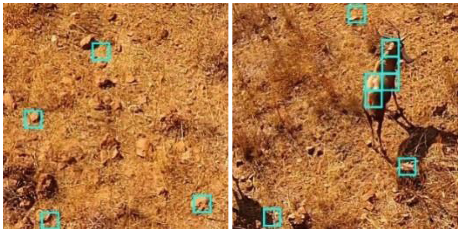


Fig. 9. False positive detections

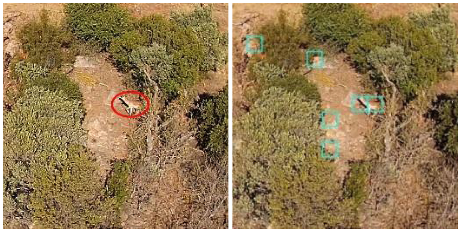


Fig. 10. A successfully detected animal that is camouflaged and partially occluded

Figure 10 shows an interesting fact during the testing procedure. A blesbok in a cluttered background is detected though it is barely noticeable intuitively. This shows that the proposed feature extraction and background categorization method makes the animal distinguishable enough to be detected.

5 Conclusion and Further Work

A specific natural environment is believed to be composed of finite visual elements. The presented method shows that it is possible to categorize these elements into several classes, and the proposed feature descriptor combined with a Neural Network form a robust classifier to distinguish the animals from subclasses of background. Future work will involve testing in various environments with different background features. The animal species shall be extended and whether the method is suitable for classification between the animals shall be investigated. Once an animal is detected it will be tracked to eliminate the need for further detections in the subsequent frames. Real-time capabilities will be investigated.

Acknowledgements. The authors wish to thank Glen Afric Country Lodge for their welcoming hospitality and the opportunity offered to collect data.

References

1. Sirmacek, B., Wegmann, M., Cross, A., Hopcraft, J., Reinartz, P., Dech, S.: Automatic population counts for improved wildlife management using aerial photography. In: iEMSs (2012)
2. van Gemert, J.C., Verschoor, C.R., Mettes, P., Epema, K., Koh, L.P., Wich, S.: Nature conservation drones for automatic localization and counting of animals. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014 Workshops. LNCS, vol. 8925, pp. 255–270. Springer, Heidelberg (2015)
3. Henry, A., Shumee, B., Takeo, K.: Neural network-based face detection. IEEE Trans. Pattern Anal. Mach. Intell. **20**(1), 23–38 (1998)
4. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
5. Park, S., Deriche, R.: A hierarchical Bayesian network for event recognition of human actions and interactions. Multimed. Syst. **10**(2), 164–179 (2004)
6. Zhao, L., Thorpe, C.E.: Stereo-and neural network-based pedestrian detection. IEEE Trans. Intell. Transp. Syst. **1**(3), 148–154 (2000)
7. Lang, K.J., Waibel, A.H., Hinton, G.E.: A time-delay neural network architecture for isolated word recognition. Neural Netw. **3**(1), 23–43 (1990)
8. Lee, S.B.: Neural-network classifiers for recognizing totally unconstrained handwritten numerals. IEEE Trans. Neural Netw. **8**(1), 43–53 (1997)
9. Wilson, C.L., Candela, G.T.: Neural network fingerprint classification. Artif. Neural Netw. **1**, 2 (1993)
10. Zhang, J., Tan, T.: Brief review of invariant texture analysis methods. Pattern Recogn. **35**(3), 735–747 (2002)

11. Guo, Y., Zhao, G., Pietikinen, M.: Discriminative features for texture description. Pattern Recogn. **45**(10), 3834–3843 (2012)
12. Guo, B., Damper, R.I., Gunn, S.R.: A fast separability-based-feature-selection method for high-dimensional remotely sensed image classification. Pattern Recogn. **41**, 1653–1882 (2008)
13. Guang-ming, X.: An identification method of malignant and benign liver tumors from ultrasonography based on GLCM texture features and fuzzy SVM. Expert Syst. Appl. **37**, 6737–6741 (2010)
14. Huang, K.: Application of artificial neural network for detecting Phalaenopsis seedling diseases using color and texture features. Comput. Electron. Agric. **57**, 3–11 (2007)
15. Martin, F.M.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw. **6**(4), 525–533 (1993)