Active Learning Approaches to Identifying Animals in Camera Trap Data

CoMPLEX MRes Mini-Project 1, March 6, 2014

Author:
Talfan Evans

Supervisors:
Dr. Gabriel Brostow and Prof. Kate Jones

Abstract

The effects on ecology from both anthropogenic and geological sources are being increasingly felt, affecting biodiversity and its dependent industries. Ecological models are crucial to a better understanding of these effects, but must be built upon reliable data. Advances in sensor technology, in particular camera traps, are leading to a role-reversal of the traditional problem of insufficient data, and ecologists are increasingly looking to alternative methods such as crowd sourcing to solve the question of its processing. These approaches rely on the valuable time of human users that are can manually label data instances, which can then train software algorithms for automatic processing. The following is an investigation of the efficacy of active learning for improving the learning efficiency of these software approaches, with the goal of maximising the utility of the human user.

Contents

1 Introduction

Biodiversity today faces pressure both from anthropogenic and geological sources, the consequences of which are easily observed in the increasing number of plant and animal life appearing on the endangered species list, with the effects of climate change, destruction of habitat and poaching all being felt.

Aside from the frank envirnomental detriment, a reduction in biodiversity carries more easily identifiable consequences. Food security is critically dependent on a healthy ecosystem, as is a safe water supply, yet responses to warning signs have so far been scant[19].

Organisations such as the Convention on Biological Diversity[18] and National Ecosystems Assessment[17] are aimed at redressing this balance, committing to issues of conservation and sustainable use of biodiversity, along with the equitable utilisation of genetic resources. However, the understanding that is crucial to combatting these detrimental effects is not easily obtained. In response to urgent necessity, the Intergovernmental Panel on Climate Change is actively developing mechanistic models to better understand the processes shaping our environment, and similar efforts must be undertaken in ecology.

Effective models must be built upon foundations of reliable data and fortunately, advances in sensing and monitoring technology mean that this is now readily achievable at a low cost. Camera trapping is one popular and effective method for obtaining evidence of animal populations, where active tracking is unfeasible. Current technology is inexpensive, long-lived and capable of autonomously communicating to satellite networks without human maintenance, dramatically increasing the potency of researchers working in the field. Accelerometry and RFID allow individuals to be efficiently and discretely tracked, opening new vistas of research and elucidating questions in collective behaviour[26].

This new-found availability of data, whilst undoubtedly beneficial, does however introduce its own problems. Living systems are complex, unsterilised and unbounded, making their analysis computationally expensive at best, and completely intractable at worst. This is in particular the case where camera trap images are concerned, as even the state of the art in computer vision approaches tend not to perform well outside of visually 'clean' environments, unlike those typically found in nature. Foliage, varying lighting conditions and scale variance are all problems that the human visual system has evolved to efficiently overcome, but are problematic to artificial systems which, although the subject of active research, are comparatively primitive in these respects. Thus, researchers in possession of vast image sets are faced by the daunting task of laboriously labelling each instance, a feat that, in the case of most species, is easily achievable by a non-expert.

Citizen science approaches attempt to exploit this fact, employing the general public as inexpensive, labour, freeing the valuable time of trained experts. Zooniverse[22], iSpot[24] and Instant Wild[23] are three such schemes, allowing users to log in and label images, in exchange for an interactive learning experience. The latter two capitalise on the upward trend in smartphone technology and social networking to improve the user experience, and mobilising the labelling process to real-time

local identification.

The time spent labelling by the user is a valuable commodity, and as such must be utilised in as efficient a manner as possible. However, not all of the images presented to the user are necessarily viable, as false triggers caused by non-animal movement are not filtered before being made available for labelling, thus representing a significant drain on useful user time. This investigation is an attempt to remedy that problem by developing a classification system capable of eliminating these unprofitable instances.

To make most efficient use of the user however, the learner should also aim to maximise the utility gained by each label. To do this, the learner should, rather than querying random instances instead have the capability to actively query those instances of maximum utility.

To this end, methods of active learning are investigated for the training of a classification algorithm, which can be divided broadly as either supervised or semi-supervised. Supervised learning algorithms train only on the subset of labelled instances, henceforth denoted L, whilst semi-supervised methods are able to 'see' the unlabelled pool of data U, and exploit its structure to improve its accuracy, effectively 'training itself'[14].

1.1 The Liberia and Missouri Datasets

This investigation concerns two datasets, henceforth denoted as the *Liberia* and *Missouri* datasets. The Liberia dataset is the smaller of the two, comprising of 255 images with a 66:189 animal/non-animal ratio, taken from one camera trap in Liberia's Sapo National Park. The second comprises 1165 images with a 808:357 animal:non-animal ratio, sourced from a number of different camera traps obtained by the University of Missouri[11]. Different traps present different scenes, ranging from dense foliage to open plains, and so present the animals at varying distances and with varying cover, presenting problems to a potential classifier.

Images in the Liberia dataset had a smaller resolution at 1200x1600p, and for reliable comparison, the Missouri images were resized to match this value. It was also necessary in the case of 311/1165 of the Missouri cases to crop the images to adjust their wider aspect ratio.

1.2 Classification Pipeline

The classification of images comprises four main stages. Firstly, information must be extracted from the image in a form that is relevant and useful to the classifier, two methods for which are considered in this investigation, namely the Dense Scale-Invariant Feature Transform (DSIFT)[2], and the Motion History Image (MHI)[6], discussed in further detail in the following section.

Secondly, the features must be encoded in a suitable form for input into the classifier. A 'Visual Bag of Words' model is used to complement the DSIFT descriptor, whereas a custom encoding scheme, decribed later, is used for the MHI. Following encoding, the complete feature descriptors are used to train the classifying algorithm, two classes of which are considered.

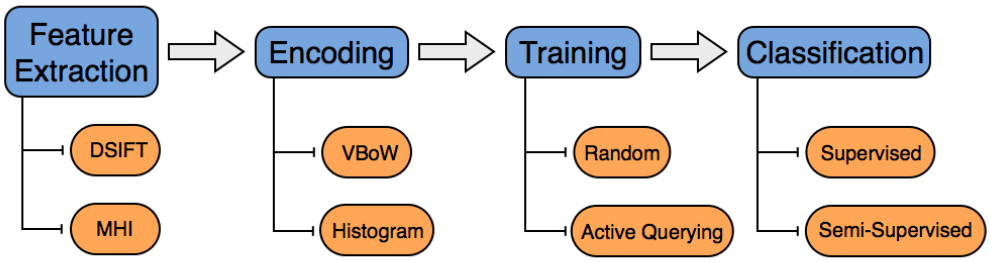


Figure 1: Classification Pipeline

The training process allows the implementation of active querying, where instances are selected based on utility. The full pipeline is illustrated in Fig. 1.

2 Feature Extraction

Feature extraction schemes convert information from an image to a form more relevant or useful to the classification algorithm. At their most fundamental, features of an image include edges, corners, 'blobs', or more specific shapes, such as circles or ellipses as in the case of the Hough Transform.

This section considers the selection of two higher-level feature descriptors, the first based on the computation of pixel gradients within a single image, and the second aimed at detecting motion between successive images, thereby encoding a form of temporal information.

2.1 Single-Frame: Scale Invariant Feature Transform (SIFT)

The SIFT descriptor extracts image information in the form of pixel gradients, defined as the change in intensity of a pixel as compared to a number of its neighbours. This particular feature is insensitive to colour, and so is well suited to the datasets considered, which are a mixture of RGB and grayscale. Illustrated in Fig. 2, the SIFT descirptor encodes an image into $K \times 128$-Dimensional (D) histogram vectors, where:

- K is the number of keypoints in the image. In DSIFT, K is a function of the bin-size B and the size of the image, as the key-points are computed in a regular grid.
- The 128-D descriptor vector comes from the fact that every key-point encodes data from a 4x4 grid of bins (each of which is B pixels wide). In each bin, the gradient orientation is computed for each pixel from a selection of 8 quantised orientations. The data for each 16 bins (8-D histogram) is then stacked to give a $16 \times 8 = 128$-D vector histogram. The complete feature representation of one image is then a collection of $K \times 128 - D$ key-points.

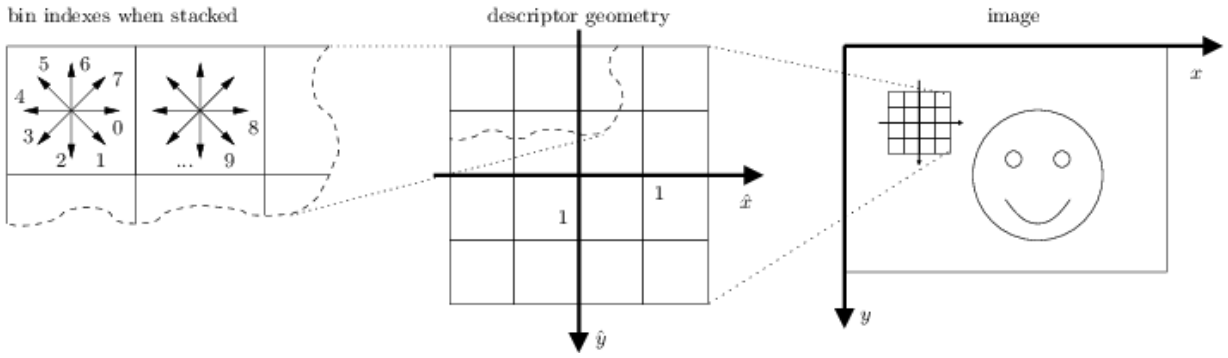


Figure 2: The SIFT Descriptor[1]

This histogram-based encoding scheme is the foundation of the strength of the SIFT descriptor, allowing its invariance to orientation. However, in Lowe's[2] original formulation, the descriptors are computed sparsely across an image, using a Laplacian of Gaussian (LoG) filter to detect points of interest, motivated by the computational difficulty of characterizing entire images. The method has been used extensively for object detection and recognition[3], and a similar scheme, the 'Histogram of Oriented Gradients (HOG)', has proven especially effective in human detection applications[4].

Both HOG and SIFT employ a similar approach based on a histogram of oriented gradients approach for feature extraction, however HOG does not specify interest-points and is typically computed densely on a regular grid throughout the image.

The application considered in this project will not consider classification or number of animal subjects, only binary detection, and as such can be considered a form of scene-classification. Fei-Fei and Perona have shown[5] that in these circumstances, dense-features tend to be more effective, while Yu et al.[12] choose to employ a combination of DSIFT and cLBP[13] for object recognition. It is also likely that interest-point based feature extraction may not perform effectively in noisy scenes such as those encountered in the camera trap data.

The VLFeat[1] toolbox offers a variation on the original method, DSIFT, computing the SIFT vector densely over the image in a manner similar to HOG, and is adopted as the single-frame feature descriptor for this investigation.

2.1.1 Visual Bag of Words (VBoW)

The Visual Bag of Words model is analogous to the BoW model in text classification, where the text words equate to 'visual signatures'. Since in the case of SIFT descriptors there are a large number of possible histogram binning combinations, the descriptor set can be quantised into a reduced 'dictionary', thereby reducing its size.

This is done by firstly clustering the descriptor data into its w largest clusters, where w is the number of words that make up the dictionary W. Descriptor data from new images can then be represented by this dictionary, producing a feature vector of constant dimension.

One possible issue is that the dictionary is assembled in the training phase, and when testing new images, it may not be sufficiently representative. Ideally, the dictionary would be updated with each new instance, however this is computationally unfeasible in most cases, and would require re-clustering the data and reassigning the labelled set with each iteration. It is however assumed reasonably assumed that the training data is sufficiently representative of all the data that the classifier might operate on.

2.2 Multi-Frame: Motion History Image

Often, identification of animals in the camera-trap data is difficult even for the human 'oracle', however this is made easier in sequences of images, where it is possible to trace movement across the scene, allowing a human to make inferences in ambiguous situation using temporal data, prompting the investigation of a similar method for feature extraction, computed over multiple subsequent frames.

A Motion History Image (MHI)[6] is a technique for representing a moving subject as a single image. The MHI is generated by computing the difference in temporally successive images and applying a threshold filter, such that only those pixels whose intensity has changed by a prescripted amount are captured, and all others are discarded. To incorporate temporal information into a sequence of images, the differenced pixels with each comparison of two successive images are encoded with a particular value, discarding again those pixels where no movement is detected.

When the algorithm compares the next two images, the values of those pixels that changed in the previous time-step are updated to reflect their 'age'. MHI is useful for detecting moving subjects against static backgrounds, however the algorithm is susceptible to slight changes, such as the swaying of leaves in a forest scene. Filters can be applied to reduce these effects, of which the most effective was found to be the median filter, illustrated in Fig.3.

2.2.1 Optical Character Recognition (OCR)

The MHIs must be constructed using relevant image sequences, where available, and thus the images were clustered according to their $date : time$ information. Unfortunately, this information was only encoded in the EXIF tags of 14% of the Missouri dataset, whereas the smaller Liberia dataset was completely labelled. However, this information is encoded visually into each image in the Missouri dataset, and a simple optical character recognition algorithm was devised to automatically capture and encode it into the images, proceeding as follows:

1. The *date:time* header bar is cropped from the image, and converted to binary black-and-white format, shown in Figure 4a

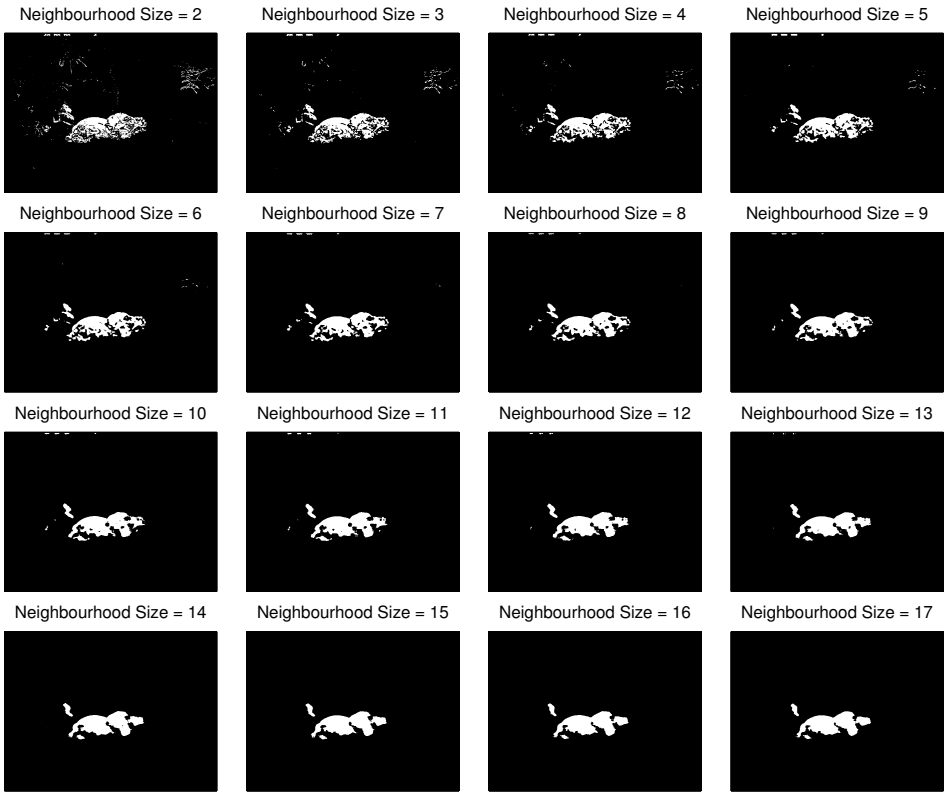


Figure 3: Median Filter

(a) Timestamp Header Bar

(b) Timestamp Read

Figure 4: Reading the Timestamp Data

2. The columns of the header bars are analysed sequentially from left to right, with the algorithm recording only those columns with non-zero entries. A new character is recorded each time white-space is encountered, producing output as illustrated in Figure 4b. Each character is resized to a standard dimension for comparison.

3. A character template vector is constructed using manually from a susbset of images. Each template is superimposed on the new character read, and the sum of their differences calculated. The minimum of these differences is used to determine the character.

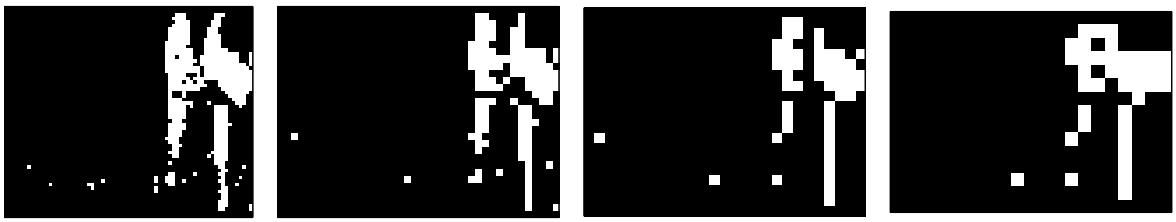
4. Each time-stamp is corrected to display 24-hour format, corrected using the AM/PM notation in the image, and encoded in the EXIF tag. Finally, the images are clustered to within 30 seconds of each other if they originate from the same camera trap.

The number of pixels encoding movement are relatively few, and the focus of movement is likely to change with every image, both in position and scale. Feeding the MHI into the SVM is not therefore particularly useful in this case, and an alternative encoding method is used. The MHI window is three frames long, resulting in an image with pixel values of either 0, 1 or 2 (corresponding to a window size $W = 3$), depending on whether there is no movement, movement between the first and second frame, or movement between the second and last frame. A histogram of the latter two valued pixels is recorded as a proportion of the total number of pixels in the image. To improve the encoding, these proportions are extracted on multiple scale levels, where the sampling windows represent the mode pixel value. Figure 5b shows the visual representation of these features on four scales, with the original image being condensed to 60x80 pixels.

3 Active Learning Methods for Query Selection

Active learning, a subset of machine learning, seeks to improve the efficiency of learning algorithms through judicious 'active' querying of the most useful data instances. Typically, machine learning employs passive sampling, where new instances are queried at random or according to supply, and labelled by an 'oracle', a classifier of 100% accuracy, typically human. Sheng et al.[15] investigate

(a) The Original Image



(b) Feature Representation on Four Scales

Figure 5: Pyramid Scaling

the possibility of inaccuracies in oracle labelling, as is often the case in reality where fatigued users are concerned.

AL occurs predominantly in one of three scenarios; stream-based selective sampling, query-synthesis, and pool-based sampling. Stream-based sampling considers a scenario where new instances are continuously becoming available, and the learner must decide whether or not to query those instances, a decision that must weigh the utility gained vs. the cost of querying.

In the query-synthesis scenario, the learner is able generate queries *de novo*, assuming that it has a definition of the inputs space, defined by the feature dimensions and range.

The pool-based sampling scenario considers the problem of querying the most useful point from a pool of instances, representing the user-based scenario that is the focus of this investigation.

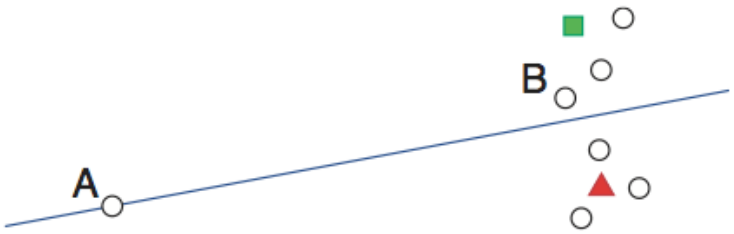


Figure 6: Uncertainty Sampling: Point B is likely to be more informative than point A, though it lies further from the decision boundary (reproduced from [14]).

3.1 Uncertainty Sampling

Uncertainty sampling aims to select the most informative query based on the uncertainty of its prior classification. In the case of an SVM, this reduces simply to instances closest to the separating decision boundary, or in general binary classification, the unlabelled instance classified with a probability closest to 50%.

3.2 Exploring Data in Structure

Uncertainty sampling in its fundamental form has one crucial setback. Whereas the least certain instance will always lie on the classification boundary, this instance may not always improve the accuracy of the classifier as a whole. In fact, these may be outliers precisely because they are uncertain or mislabelled. It is therefore important to consider how representative the instance is of the data a a whole, illustrated by instance B in Fig. 6.

Density-weighted schemes aim to take into account this representativeness, measurable by a number of distance metrics, and implemented using Eq. 1:

$$x^*_{1D} = \underset{x}{argmax} \phi_A(x) \times \left(\frac{1}{U} \sum_{x' \epsilon U} sim(x, x') \right)^{\beta} \tag{1}$$

Where $x*_{1D}$ is the queried instance, $\phi_A(x)$ the base utility metric (in this case uncertainty), U the set of unlabelled points and β a coefficient controlling the weight of the similarity term. For this investigation, three common distance metrics will be utilised, namely the Euclidean distance, Cosine similarity, calculated from the dot products of two feature vectors, and the Spearman's rank correlation.

Density-weighted heuristics are particularly suited to pool-based scenarios, as the additional computation required can be cached in advance of a training session, thus not representing any cost to the oracle.

3.2.1 Active+Semi-Supervised Learning

Semi-supervised learning, like active learning, aims to exploit the vast amounts of unlabelled data that may be available in the structure of the dataset. They are complementary approaches in this respect, however they differ in how they this information is extracted. Active learning focuses on maximising the utility from each query, minimising the cost to the oracle, whereas semi-supervised learning attempts to effectively train itself in absence of labelling.

3.3 Maximising Disagreement

A third active learning scheme compares the outcomes of each possible query, and selects the instance that produces the most disagreement in classification, were it labelled positive or negative. This method is computationally expensive, requiring $O\left(U^2\right)$ computations, none of which can be cached in advance. This scheme is thus unfavourable in the scenario under consideration.

4 Classification Schemes

4.1 Supervised

4.1.1 Supervised Learning: Support Vector Machines (SVMs)

The fundamental implementation of an SVM[7] is a $(K-1)$-dimensional separating hyperplane in a k-dimensional feature space. Instances in feature space can then be determined by their relative partition, achieved in multi class-separation by multiple separating hyperplanes, the parameters of which are typically determined by maximising the distance to the nearest training point. This approach is known as $max-margin$ separation, and intuitively produces a good separation and reduces the generalization error, defined as the ability of the algorithm to extrapolate learned information to untrained points.

The simplest case of an SVM may be visualised as a 1-D line separating points in 2-D cartesian space, however problems may arise where the points are not linearly separable into two distinct regions. Here, so called 'kernel methods' have been introduced, mapping the data into a higher dimensional space in which it is separable, giving SVMs non-linear separating capability. The $libSVM$ toolbox[8] offers a kernel-based SVM solution employing the Radial Basis Function[9], and is used as the default SVM method for this investigation.

4.2 Semi-Supervised Learning: Gaussian Fields with Harmonic Label Propagation

The semi-supervised algorithm used in this investigation, formulated by Zhu et al.[10] exploits structure present in the data to aid with unlabelled classification. The dataset can be considered a connected graph, in which edges connect nodes (data instances) in a multi-dimensional feature

space. Harmonic information propagation is then used to classify the unlabelled nodes, which is influenced by the connectivity of each point within the graph, represented by the weights of the connecting edges. These weights are computed according to a metric based on an exponentially decaying euclidean distance, illustrated in Eq. 2:

$$w_{ij} = exp\left(-\sum_{d=1}^{n} \frac{(x_{id} - x_{jd})^2}{\sigma_d^2}\right) \tag{2}$$

Each weight w_{ij} is stored in an $n \times n$ weight matrix, where n is the the number of data instances. The parameter sigma is a normalising factor with a characteristic size determined from the entire dataset.

The method by which the harmonic function propagates information can be treated as analogous[27] to electrical networks, the edges connecting the nodes equating to resistors of varying resistance corresponding to the weights w_{ij}. In a binary classification situation, the positive labels are connected to a positive voltage, and the negatives to ground. The allocated labels are then obtained from rounding the voltages resultant in the network, which represents the minimal energy dissipation. Whilst this scheme is effective for well separated data, in real datasets, it can lead to an imbalance in classification. To overcome this, Class Mass Normalization[10] is employed, which appropriately manipulates the threshold boundary from the default of 0.5 to ensure that the ratio of unlabelled classifications is equal to the ratio of labelled nodes in the network.

Graph based methods such as this extend naturally to combination with active learning. In this algorithm, instances are queried greedily based on risk-minimisation criteria, where in this instance risk is defined as the generalisation error of a Bayes classifier.

5 Results

In each test, accuracy was defined as the proportion of the total dataset classified correctly, compared to the ground-truth data. Whereas this data was available for the Missouri Set, a simple user interface was devised to obtain labels for the Liberia dataset, where the mode vote of five participants was used, and instances of significant disagreement independently verified.

In each test, the classifier is initially trained on one positive and one negative instance, selected at random. Instances are added successively to the training set, chosen according to either random or active sampling criteria, until the training and testing sets are equivalent. Each test is an average of five independent runs.

In the following, C denotes the number of codewords used, B the bin size, K the key-step and *Euclidean*, *Cosine* and *Spearman* denote weighted uncertainty-sampling schemes.

5.1 Feature Extraction

5.1.1 Single Frame

C = 255					
B\K	*128*	*64*	*32*	*16*	*8*
128	83.85%	85.94%	84.31%	81.80%	88.12%
64	81.66%	88.64%	84.08%	80.71%	90.35%
32	80.61%	89.60%	85.08%	80.13%	89.69%
16	80.42%	88.79%	85.99%	81.67%	88.75%
8	79.84%	89.13%	85.61%	82.05%	86.56%
C = 64					
B\K	*128*	*64*	*32*	*16*	*8*
128	82.23%	88.53%	85.61%	81.92%	89.90%
64	81.13%	88.61%	88.00%	84.58%	89.58%
32	78.94%	89.44%	87.84%	82.66%	86.84%
16	80.88%	88.76%	87.62%	84.96%	62.32%
8	81.06%	85.80%	88.95%	87.05%	87.13%

Table 1: Effect of Bin, Key-Step and Codebook Size on Mean Classification Accuracy

Before proceeding further, a feature extraction parameter comparison was undertaken, the results of which are illustrated in Table 1. Tests were undertaken using the baseline RBF-SVM classifier on the Liberia dataset, found to outperform *MatLab*'s tree-bagger algorithm at a significantly reduced copmutational cost on the Liberia dataset (results not shown). A comparison of the best five parameter sets is illustrated in Fig. 7.

Firstly, it is noted that in all cases, the SVM is able to achieve 100% separation on the testing set when trained exhaustively, as would be expected when implementing non-linear kernel-based methods. Secondly, it should be noted that the scores represent averages over all training sets, and so small increases in accuracy in reality represent significant improvements in learning rate.

There is seen to be a general improvement from denser SIFTing (smaller key-step). This is as expected, as a denser sampling grid would improve the probability of superimposing a characteristic visual codeword exactly where such a codeword may be present in the image, contrasting the case where they are improperly superimposed, and may not signify a match.

There does not seem to be a consensus as to the effects of bin-size, suggesting that whilst smaller characteristic signatures are able to sufficiently represent the larger, the smaller are not reliably detectable enough to offer significant improvements over the larger, which, though fewer, are more clearly defined.

Likewise, the improvements gained from an increased codebook size are not drastic. This could also be expected, as the majority of the features in the images are likely to represent either animals or

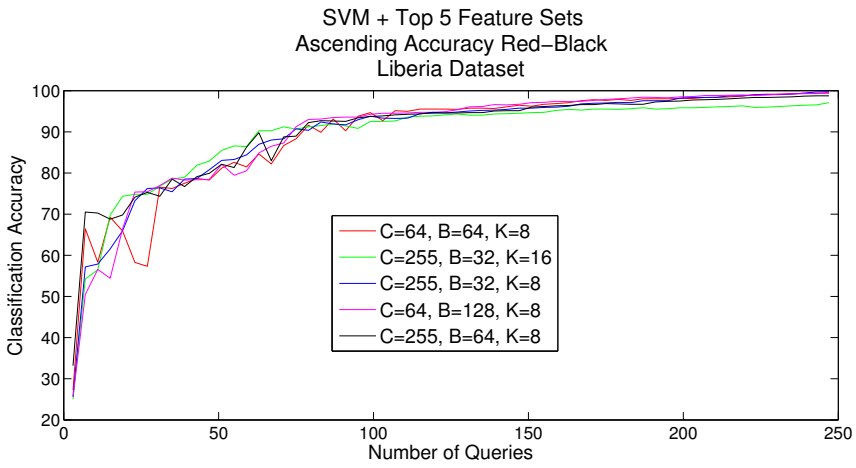


Figure 7: Copmarison of Feature Parameters

ambiguous noise, that is to say, from the perspective of the classifier, one patch of foliage is likely to be indistinguishable from another.

With these consideration in mind, the parameters $C = 255$, $B = 64$ and $K = 8$ were chosen, representing the greatest overall accuracy, and it is assumed for the purpose of the investigation that this feature set performs comparably on the Missouri dataset.

5.1.2 Motion History Image

Similarly, Fig. 8 shows no significant performance with additional pyramind scaling. However, since the encoding of these features is far less data intensive than DSIFT, it was possible to proceed using $N = 10$ feature scales without significant computational cost.

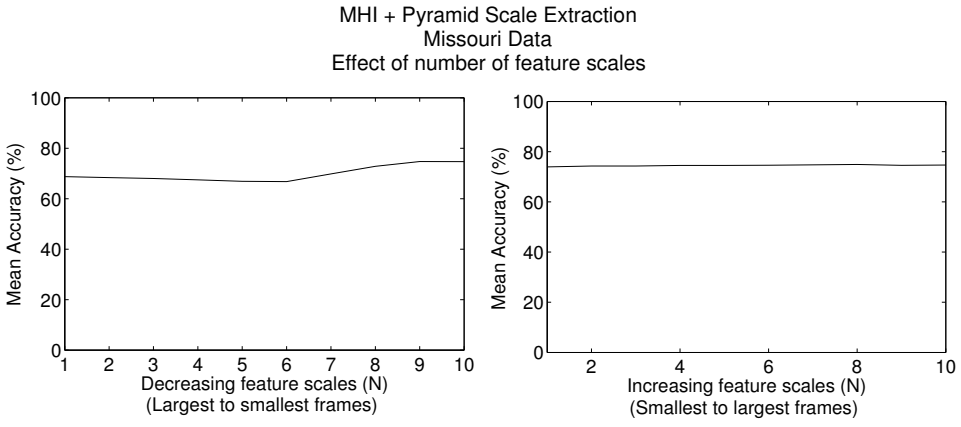
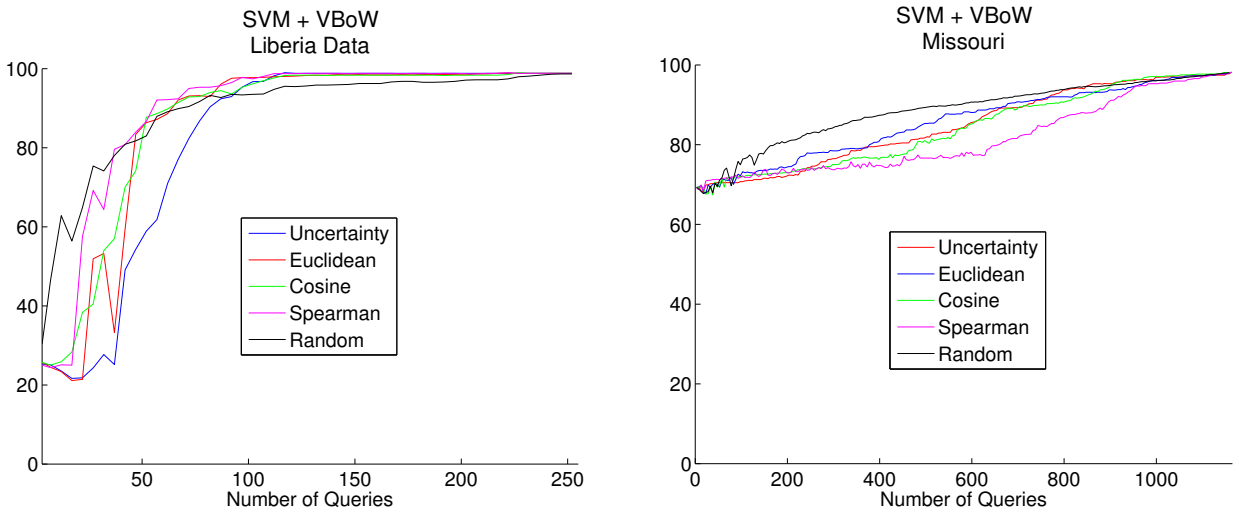


Figure 8: Effect of Number of Pyramid Scales

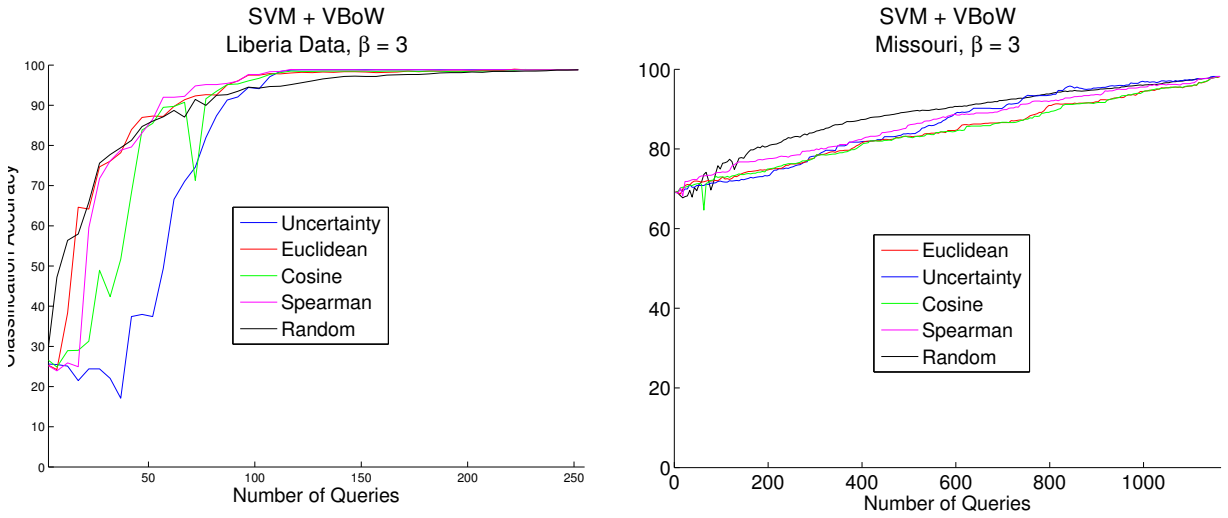
5.2 Classification

5.2.1 Single Frame SVM



(a) Liberia (b) Missouri

Figure 9: VBoW + SVM



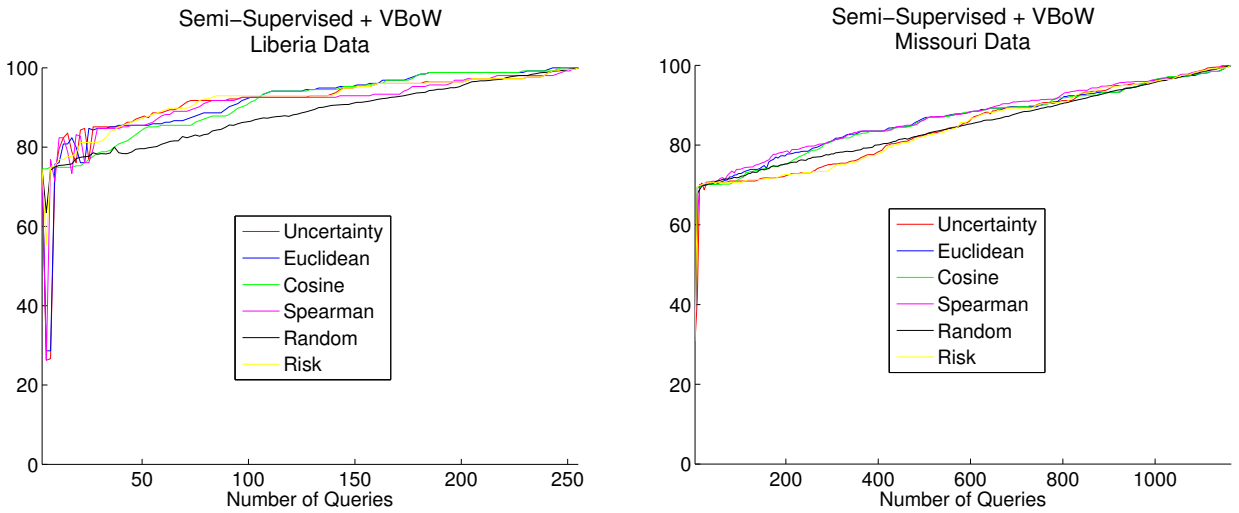
(a) Liberia (b) Missouri

Figure 10: MHI + Semi-Supervised, $\beta = 3$

Fig. 9 shows that initially, active querying of points leads to a poorer learning curve than in random querying, although the AL criterions' maximum accuracy is achieved sooner, and is superior to random in the Liberia dataset. No such performance advantages are observed in the Missouri set, and this somewhat disappointing result suggests that the most ambiguous queries are misrepresentative of the data, causing erroneous misclassification.

This is shown in the Liberia dataset, where increasing the effect of the weighting parameter remedies the initial error, leading to performance that, whilst initially not superior, outperforms random querying in the long term. This is not the case in the Missouri dataset, where the uncertainty-sampling is still detrimental to the classifier's performance.

5.2.2 Single Frame Gaussian Fields



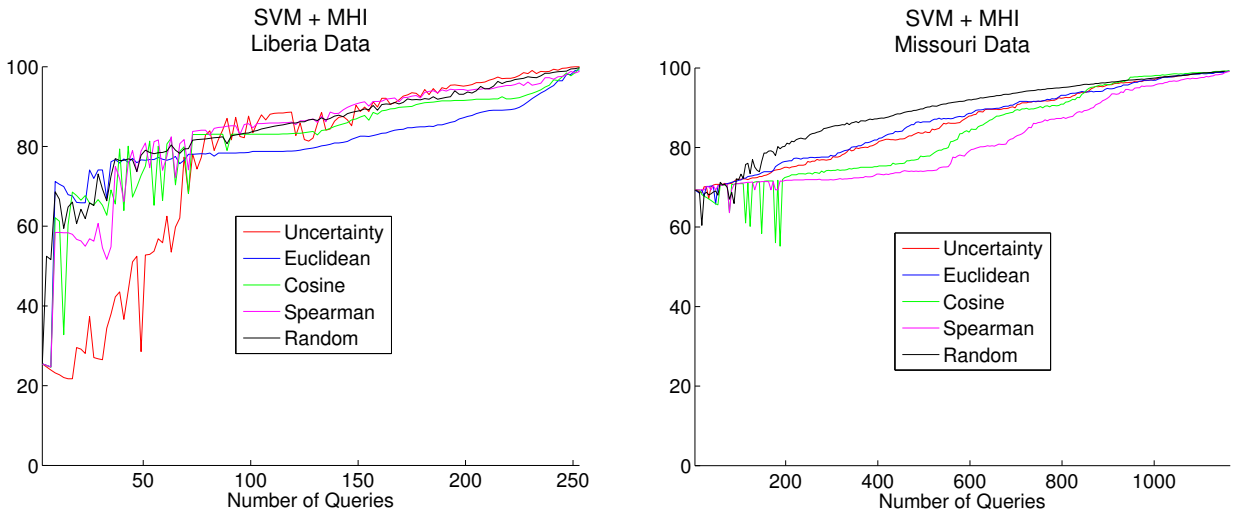
(a) Liberia

(b) Missouri

Figure 11: VBoW + Semi-Supervised

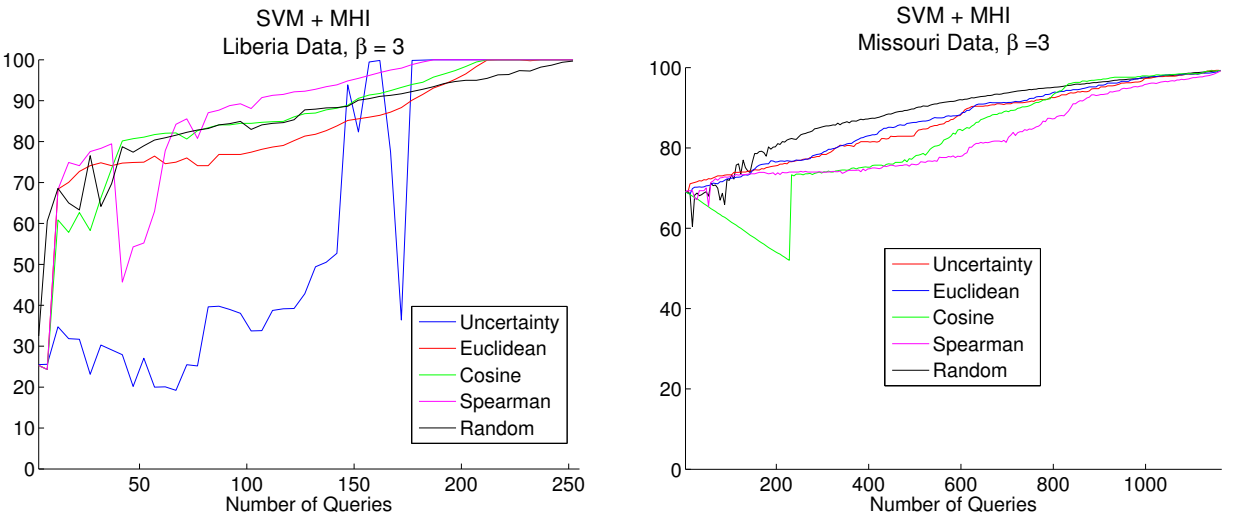
Contrastingly, the semi-supervised algorithm, whilst also notably superior to the SVM, makes effective use of the AL criteria, with each one significantly outperforming the random querying. The effects of density-weighting in the Liberia dataset are not as prominent as in the SVM case, however the Missouri dataset weighted uncertainty-sampling shows a marked improvement on random querying,

5.2.3 MHI SVM



(a) Liberia (b) Missouri

Figure 12: MHI + SVM



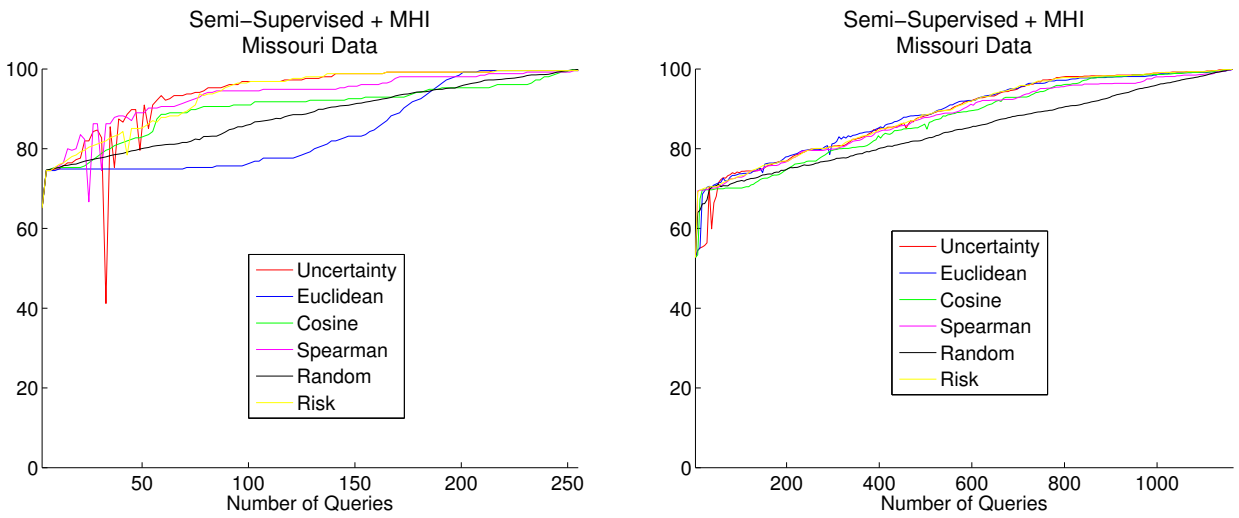
(a) Liberia (b) Missouri

Figure 13: MHI+SVM, $\beta = 3$

As seen in the DSIFT case, the base uncertainty-sampling mthod is seen to be detrimental to the initial performance, though this effect is not as pronounced in the Liberia dataset, where density-

The problem is again effectively remedied in the Liberia dataset by increasing the weigting parameter β, however the correction does not extend to the missouri dataset, with the Cosine weighting in particular leading the classifier astray.

5.2.4 MHI Gaussian Fields



(a) Liberia

(b) Missouri

Figure 14: MHI + Semi-Supervised

Again, as in the case of the DSIFT descriptor, the semi-supervised outperforms the SVM, and makes significantly better use of the uncertainty-sampling methods implemented. The effect of the density-weighting is here more salient, in all cases but the Euclidean clearly reducing the erratic behaviour of the base uncertainty-sampling method at the beginning of the training, although in this case coming at a slight cost in final accuracy.

5.3 Discussion

The results show clearly that the semi-supervised agorithm is superior in all cases to the SVM, not only displaying faster learning, but also making efficient use of the uncertainty-sampling heuristics. However, both schemes achieved a success exceeding 70% classification accuracy with relatively few training instances, with this accuracy increasing further improving with additional training. It is however noted that, whilst 100% accuracy is achieved with the complete training set, this would be expected from two such non-linear classifiers, and this accuracy would not be expected to extend to such a degree to further testing instances. No noticeable benefits were observed in the MHI feature extraction over DSIFT, both performing comparably.

In general, the base uncertainty-weighted method was detrimental to the peformance of the SVM, suggesting that the most uncertain instances in this case were in fact unrepresentative of the data as a whole, leading to classification error. This was shown to be the case, as increasing β remedied the problem in most cases, rendering the uncertainty sampling method far less erratic, suggesting that unrepresentative points were no longer being queried. The density-weighted methods, in most cases led to a superior long-term classification accuracy, even when intitial learning rates were inferior to random sampling.

However, density-weighting was unable to improve upon random querying in the Missouri dataset. It is possible that in this case, the feature extraction does not extend to the same extent from the Liberia dataset.

Of the density-weighting metrics investigated, the Spearman's rank correlation was shown to be most consistent, performing well in conjunction with both SVM, with $\beta = 3$ and the semi-supervised algorithm, where $\beta = 1$, on both datasets. Furthermore, the Spearman's weighting was found in some case to outperform the risk-based querying built complementarily into the S-S algorithm, as in the case of VBoW applied to the Missouri dataset 11b.

5.4 Conclusion and Further Work

The current work investigated the efficacy of applying software methods to binary scene classification. The methods devloped, though only encompassing a small subset of potential approaches, proved moderately succesful in achieving classification.

It is noted that the full effects of increasing the effect of density-weighting were not investigated fully, and it is possible that uncertainty-sampling methods may yield further improvements in the semi-supervised case, and should be a topic of future investigation.

Whilst to an extent, the feature extraction methods proved effective enough for achieving modest results, especially on the Liberia dataset, it is possible that they are not sufficient for effective use on the Missouri dataset, demonstrated by the inability of the scheme to match the random querying baseline.

Further research should be conducted into the MHI representation, where a possible source of error may be in the misclassification of intial images in each sequence, compared only to the subsequent image as no previous exists. This problem may be remedied by establishing a 'background' scene for each sequence and using this for image differencing, though regrettably there was not time to investigate this possibility fully.

Appendix

Software

VLFeat[1] provide open-source implentations in MatLab and C of popular algorithms for computer vision applications. These include feature extraction methods such as the DSIFT algorithm used in this investigation, encoding tools and classifiers.

For classification, *libSVM*[8] provide a comprehensive MatLab toolbox, from which the SVM used was adopted.

References

[1] www.vlfeat.org

[2] Lowe, D. G. (2004), 'Distinctive Image Features from Scale-Invariant Keypoints'.

[3] Mikolajczyk, K., Schmid, C. (2005), 'A performance evaluation of local descriptors'.

[4] Dalal, N., Triggs, B. (2005), 'Histograms of Oriented Gradients for Human Detection'.

[5] Fei-Fei, L., Perona., P. (2005), 'A Bayesian hierarchical model for learning natural scene categories'.

[6] Davis, J. (2001), 'Hierarchical Motion History Images for Recognizing Human Motion '.

[7] VAPNIK, V. (1979), 'Estimation of Dependences Based on Empirical Data'.

[8] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[9] Smola, A. J., Scholkopf, B. (1997), 'The connection between regularization operators and support vector kernels'.

[10] Zhu, X., Ghahramani, Z., Lafferty, J. (2003), 'Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions'.

[11] Ren, X., Tony X. Han, T., He, Z. (2013), 'Ensemble Video Object Cut in Highly Dynamic Scenes'.

[12] AYu, X., Wang, J., Kays, R., Jansen, P., Wang, T. (2013), 'Automated identification of animal species in camera trap images'.

[13] Zhang, J., Huang, K., Yu, Y., Tan, T. (2008), 'Boosted Local Structured HOG-LBP for Object Localization'.

[14] Settles, B., 'Active Learning', Morgan and Claypool.

[15] Sheng, V., Provost, F., Ipeirotis, P. (2008), 'Get another label? improving data quality and data mining using multiple, noisy labelers'.

[16] https://worldwildlife.org/

[17] http://uknea.unep-wcmc.org/

[18] https://www.cbd.int/cop/

[19] Butchart et al. (2010), 'Global Biodiversity: Indicators of Recent Declines'.

[20] http://www.ipcc.ch/

[21] Burghardt, T., Calic, J. (2002), 'Analysing Animal Behaviour in Wildlife Videos Using Face Detection and Tracking'.

[22] www.Zooniverse.org/

[23] http://www.edgeofexistence.org/instantwild/

[24] http://www.ispotnature.org/

[25] McNeill, S., Barton, K., Lyver, L., Pairma, (2011), 'Semi-automated penguin counting from digital aerial photographs'.

[26] Sumner, S., Lucas, E., Barker, J. & Isaac, N. (2007), 'Radio-tagging technology reveals extreme nest-drifting behavior in a eusocial insects'.

[27] Doyle, P., Snell, J. (1984), 'Random walks and electric networks'.