

## An Experiment on Animal Re-Identification from Video

Kuncheva, Ludmila; Garrido-Labrador, Jose; Ramos-Perez, Ismael; Hennessey, Samuel; Rodriguez, Juan

### Ecological Informatics

DOI:

[10.1016/j.ecoinf.2023.101994](https://doi.org/10.1016/j.ecoinf.2023.101994)

Published: 01/05/2023

Peer reviewed version

[Cyswllt i'r cyhoeddiad / Link to publication](#)

*Dyfyniad o'r fersiwn a gyhoeddwyd / Citation for published version (APA):*

Kuncheva, L., Garrido-Labrador, J., Ramos-Perez, I., Hennessey, S., & Rodriguez, J. (2023). An Experiment on Animal Re-Identification from Video. *Ecological Informatics*.  
<https://doi.org/10.1016/j.ecoinf.2023.101994>

#### Hawliau Cyffredinol / General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# An Experiment on Animal Re-Identification from Video

Ludmila I. Kuncheva<sup>a</sup>, José Luis Garrido-Labrador<sup>b</sup>, Ismael Ramos-Pérez<sup>b</sup>, Samuel L. Hennessey<sup>a</sup>, Juan J. Rodríguez<sup>b</sup>

<sup>a</sup>*Bangor University, Bangor, UK*

<sup>b</sup>*Universidad de Burgos, Burgos, Spain*

---

## Abstract

In the face of the global concern about climate change and endangered ecosystems, monitoring individual animals is of paramount importance. Computer vision methods for animal recognition and re-identification from video or image collections are a modern alternative to more traditional but intrusive methods such as tagging or branding. While there are many studies reporting results on various animal re-identification databases, there is a notable lack of comparative studies between different classification methods. In this paper we offer a comparison of 25 classification methods including linear, non-linear and ensemble models, as well as deep learning networks. Since the animal databases are vastly different in characteristics and difficulty, we propose an experimental protocol that can be applied to a chosen data collections. We use a publicly available database of five video clips, each containing multiple identities (9 to 27), where the animals are typically present as a group in each video frame. Our experiment involves five data representations: colour, shape, texture, and two feature spaces extracted by deep learning. In our experiments, simpler models (linear classifiers) and just colour feature space gave the best classification accuracy, demonstrating the importance of running a comparative study before resorting to complex, time-consuming, and potentially less robust methods.

*Keywords:* Animal re-identification, Computer vision, Classification, Convolutional networks, Comparative study

---

## 1. Introduction

According to predictions, climate change, global pollution, and uncontrollable growth of plastic waste are among the factors heralding an ecological catastrophe. Multidis-

---

*Email addresses:* l.kuncheva@bangor.ac.uk (Ludmila I. Kuncheva), jlgarrido@ubu.es (José Luis Garrido-Labrador), ismaelrp@ubu.es (Ismael Ramos-Pérez), sml18vly@bangor.ac.uk (Samuel L. Hennessey), jjrodriguez@ubu.es (Juan J. Rodríguez)

ciplinary effort in monitoring and managing of animal populations and ecosystems can reduce the risk of losing animal species and destroying natural habitats [1], [2] [3], [4].

Scientists have monitored animals for a long time using a variety of methods. Individual animal recognition has been in place for a long time, but the methods have been predominantly intrusive, and often invasive, including branding, tattooing, notching, and tagging. Ethical issues aside, this may affect the behaviour of the animal and also compromise the demographic study [2, 5, 6, 7]. Geared predominantly towards human re-identification [8], computer vision is currently also making large strides towards aiding or replacing the outdated physical identification methods in animal re-identification [9]. Distinctive individual patterns allow for re-identification of animals in many species such as zebras, giraffes [10], whale shark [7], African penguins [11], ringed seals [12, 13], common dolphins [14], giraffes [15], giant panda [16], honey bees [17], yaks [18], and many more. To succeed in this quest, large databases of animals need to be collected, annotated, and made available to researchers. Additionally, interdisciplinary teams should be involved, bringing together the state-of-the-art in animal studies and machine learning.

Animal re-identification is the task of recognizing the animal’s identity from an image or a collection of images. In a video, an animal may be present in different frames; it may come in and out of camera view multiple times. Re-identification means that the animal’s identity is recognised correctly each time it is in camera view. Sometimes, (human) re-identification is understood as identifying the same individual from different camera views, at the same time moment. Practically, this is the same task, only differently phrased. In this context, we will stipulate that animal re-identification is a standard classification task. While there is a large body of literature on animal re-identification, the classification part rarely explores more than one designated classifier or different feature representations.

The prospective real-life scenario, which our experiments are targeted at, is as follows. A long video footage is available containing nearly the same individuals (a few may join in or drop off). A small part of the video is annotated with bounding boxes and class labels (identities). Classifiers are trained and tested on the labelled part, and the most successful classifier is identified. The unlabelled part is subsequently processed by detecting bounding boxes and labelling each one by the chosen classifier.

In this study, we consider only the first part of this scenario, where we train and test

classifier models across different feature representations. We view a two-fold contribution of this study. Our methodological contribution is an experimental protocol that can be applied when choosing animal re-identification method for a given, partly annotated dataset. Second, we offer insights and general recommendations for the type of data similar to that used in the experiment. Our results could be used as a benchmark for further re-identification experiments for the chosen dataset.

## 2. Related work

Camera trap, bespoke fixed camera setting, and unconstrained video footage are useful sources of imaging data for tracking, species identification, and individual animal recognition [3]. Fixed camera setting has been used primarily in managing livestock, for example, Holstein cows [19, 20, 21, 22] and pigs [23], [24], where the animals are kept into an enclosure or herded through a gate. Drone videos (unconstrained) have been used for re-identification of livestock as well [25]. Camera traps are mostly suitable for monitoring the type of species in a given location [26, 27]. They are rarely used for animal re-identification. On the other hand, unconstrained video footage is the common source of data for tracking and recognising animals in the wild [10, 11, 28, 29]. However, this makes the task of bounding box identification and subsequent classification a lot more challenging.

A notable work on animal re-identification from video are the idTracker models [30, 31], reporting excellent identification accuracy on a group of simultaneously moving animals. Their experiments, as well as several related studies [32, 33], include ants, mice, fruit fly, and zebra fish, none of which presents clear biometric markers. The videos are taken in a non-cluttered lab environment and the individual recognition is solely based on the trajectories. While acknowledging the strong information potential of the animals' movement, our study is focused on the appearance alone. By ignoring the frame continuity, we gauge the potential of the classification methods chosen here to work for image collections obtained through different means, such as crowd-sourcing and time-lapse video footage. Schneider et al. [3] recommend using video as the richest source of images for animal re-identification and encourage researchers to publish curated and annotated animal image datasets. The database we chose for this study is available at: <https://github.com/LucyKuncheva/Animal-Identification-from-Video> [34].

Schneider et al. [3] present a timeline of the development of the area of animal re-identification, highlighting the paradigm shift from what they call ‘feature engineering’ towards ‘machine learning’. The feature engineering era was rooted in standard image descriptors of colour, texture, and shape of the objects to be recognised. Similarity-type features such as SIFT were also included there. Bespoke image processing approaches were adapted to suit the task of animal re-identification. An example comes from the realisation that primate faces bear resemblance to human faces, hence the well-developed human face recognition can lend a hand [29, 35, 36]. In fact, face recognition has been attempted for cows [37, 38], goats [39] and dogs [40] as well. The shortcoming of the feature engineering approach was lack of generality. The methods were also deemed impractical because, in addition to domain knowledge about the species being studied, these methods required computing expertise [3].

The machine learning stage defined by Schneider et al. [3] is exclusively associated with deep learning [3, 41]. The earliest deep learning methods were based on convolutional neural networks (CNN). Older studies applied standard or adapted classification methods, the most intuitive of which is the nearest neighbour. Moskvayak et al. [42] proposed a system architecture to re-identify manta rays by generating an embedding of a target image using a CNN and subsequently applying a K-nearest neighbour classifier. Miele et al. [43] propose two CNN-based methods for animal re-identification which include deep metric learning and a pipeline where the CNN is followed by matching through SIFT features. The standard CNN networks are progressively being replaced by similarity-based networks such as Siamese networks [44], [45], [46], [47], [48]. This was dictated by the observation that the training data would often be insufficient for learning a multitude of classes compared to learning the two-class problem of ‘same/different’. In essence, similarity-based networks are trained to extract a metric (features most relevant to pairwise comparison). This is followed by a nearest neighbour (1-nn) or k-nn classification in order to return the animal identity.

Deep learning models can seamlessly integrate the three stages of the re-identification process: detection of the bounding box with the animal in the image, feature extraction, and finally classification. For example, reinforcement learning networks such as Faster RCNN [49] have proven useful as part of the pipeline responsible for bounding box detection. Once bounding boxes were identified, a CNN version is used for feature extraction.

Ravoor et al. [41] consider a final stage of the pipeline called Identity association. At this stage, either the trained CNN predictions are taken forward through softmax (the feature extraction and classification task are fused and performed entirely by the CNN) or another classifier is used on the features extracted by the CNN. The typical classifiers, as reported by Ravoor et al., are the support vector machine (SVM) classifier, k-nn, Euclidean distance, and cosine similarity. A two-step process decoupling feature extraction and classification is also considered by Bodesheim et al. [50], with SVM and Gaussian processes used at the classification step. In our study, we are interested in testing a large collection of different classifier models at the Identity association stage. To the best of our knowledge, very little experimental research has been done to explore the success of the plethora of state-of-the-art classifier models with either engineered features or deep learning features. Following the previous research, in this study, we decouple the two stages of animal recognition into feature extraction and classification. The novelty of our study can be summarised as follows:

(1). Most previous studies focus on feature extraction, typically using a deep neural network model. We suggest that, for complicated tasks with relatively small number of instances per class, simpler, more basic feature extraction methods may be useful. Our experiment demonstrates this argument.

(2). While most previous studies apply a basic classifier model at the classification stage, here we test a variety of state-of-the-art classifiers.

(3). We propose a generic method and a testing protocol for animal re-identification from video or an image collection, where part of the data is annotated with bounding boxes and class labels.

### **3. The proposed experimental protocol**

We start with a labelled data set of images (instances), each containing one animal. We assume that these images are obtained by applying object detection or tracking in the video, and extracting bounding boxes with one animal in each. The images are therefore of different sizes. An example of the type of images (resized to identical squares), taken from our case study, is shown in Figure 1.

For the experiment here, we do not need to keep the location of the bounding box within the frame, but these locations are available in the database.

Dwayne

---



Humphrey

---



Siobhan

---



Figure 1: An example of the type of images in our case study. These are three of the nine classes in the Koi fish video.

The proposed protocol consists of two steps: feature extraction and classification, as illustrated in the diagram in Figure 2. The feature extraction is carried out over the whole collection of images, ignoring the class label. A two-fold cross-validation classification is applied to test various state-of-the-art classification models, where the video is split into halves. The reason for keeping the video halves intact is to avoid near-identical instances coming from time-adjacent frames to be split between training and testing. Randomised cross-validation will not guard against such splits, which will lead to deceptively high accuracy rates. Bypassing the feature extraction branch in the diagram, deep learning models can be applied directly to the original image data. This is the currently preferred method for animal recognition and re-identification. We argue here (and demonstrate through our case study) that deep learning is not necessarily the best approach.

#### 4. Experimental study

Following the proposed protocol, in this section we detail our data and design choices.

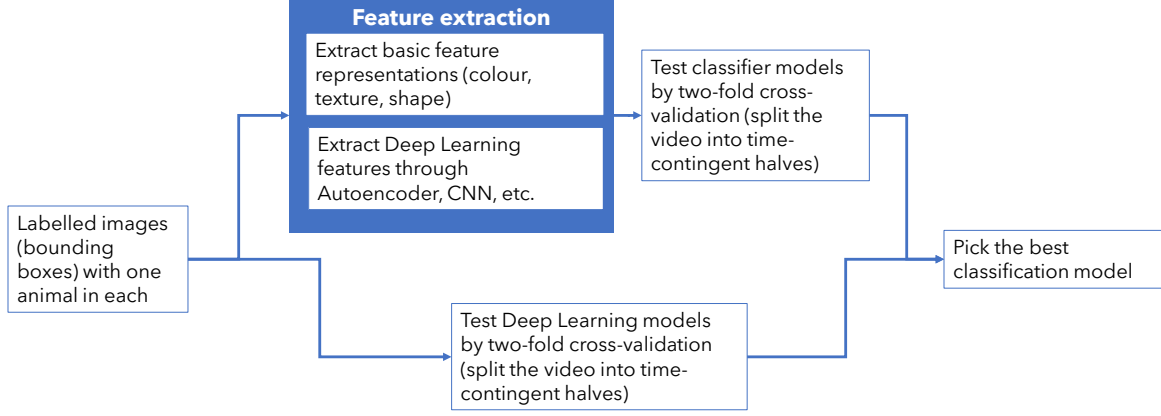


Figure 2: Diagram of the proposed experimental protocol for animal re-identification.

#### 4.1. Data

The dataset chosen for our case study consists of five video clips sourced from Pixabay <https://pixabay.com/> under the Pixabay license. The unconstrained videos capture the movement of groups of animals within 9-24 seconds. The animals in each video are of the same species: Koi fish<sup>1</sup>, pigeons (square)<sup>2</sup>, pigeons (pavement)<sup>3</sup>, pigeons (curb)<sup>4</sup>, and pigs<sup>5</sup>, available from [www.pixabay.com](http://www.pixabay.com). Each video has been manually annotated by creating bounding boxes (BB) with one animal in each BB. The BBs have been labelled with the respective animal identities. Examples of annotated frames are shown in Figure 3.

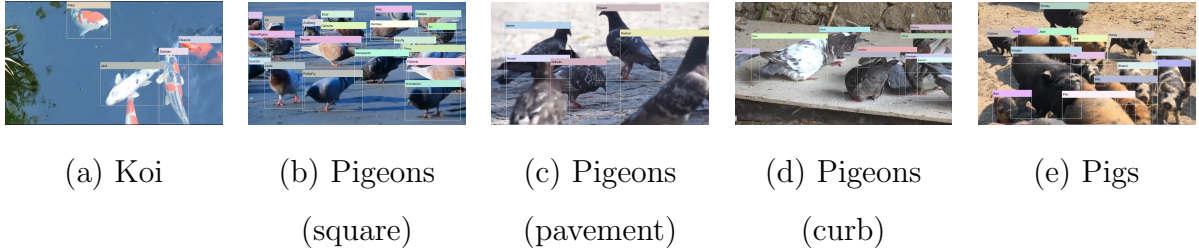


Figure 3: Examples of annotated frames from the animal re-identification database used as our case-study.

The full database is available at <https://doi.org/10.5281/zenodo.7322820> [51]. It contains the annotations, individual images and datasets with different feature repre-

<sup>1</sup>[www.pixabay.com/videos/koi-carp-fishes-ornamental-fish-5652/](https://www.pixabay.com/videos/koi-carp-fishes-ornamental-fish-5652/)

<sup>2</sup>[www.pixabay.com/videos/birds-street-pigeon-29033/](https://www.pixabay.com/videos/birds-street-pigeon-29033/)

<sup>3</sup>[www.pixabay.com/videos/pigeons-doves-and-pigeons-bird-city-4927/](https://www.pixabay.com/videos/pigeons-doves-and-pigeons-bird-city-4927/)

<sup>4</sup>[www.pixabay.com/videos/pigeons-eating-nature-birds-food-8234/](https://www.pixabay.com/videos/pigeons-eating-nature-birds-food-8234/)

<sup>5</sup>[www.pixabay.com/videos/pigs-farm-animals-livestock-49651/](https://www.pixabay.com/videos/pigs-farm-animals-livestock-49651/)



sentations. The characteristics of the five videos are summarised in Table 1. We also display an imbalance metric for each video, which is calculated as the size of the largest class divided by the size of the smallest class.

Table 1: Characteristics of the videos

| Video              | $k$ | $l$ | $N$  | $c$ | Min p/f | Max p/f | Avr p/f | Imbalance |
|--------------------|-----|-----|------|-----|---------|---------|---------|-----------|
| Koi fish           | 536 | 22  | 1635 | 9   | 1       | 6       | 3.1     | 2.8       |
| Pigs               | 500 | 16  | 6184 | 26  | 4       | 20      | 12.4    | 10.5      |
| Pigeons (square)   | 300 | 9   | 4892 | 27  | 1       | 23      | 16.3    | 24.8      |
| Pigeons (pavement) | 600 | 24  | 3079 | 17  | 3       | 8       | 5.1     | 19.3      |
| Pigeons (curb)     | 443 | 17  | 4700 | 14  | 8       | 13      | 10.6    | 3.1       |

Table notes:  $k$  is the number of frames;  $l$  is the video length in seconds;  $N$  is the number of objects (individual animal images);  $c$  is the number of classes (animal identities); Min p/f is the minimum number of animals per frame (image); Max p/f and Avr p/f are respectively the maximum and the average numbers.

In order to visualise the complexity of our problem, we used the feature reduction algorithm UMAP [52] to reduce the dimensionality to two. We applied UMAP to the colour feature representation (RGB, detailed in the next section) of the Koi fish video. Figure 4 (a) shows the overall scatterplot of the nine classes plotted with different markers and colours. Instances in consecutive frames are joined by lines. Figure 4 (b) is a close-up of the three classes illustrated in Figure 1. The instances in the second half of the video are additionally marked with circles.

The figure demonstrates that the classes are heavily intertwined. More importantly, Figure 4 (b) shows that there could be large differences in the representations of the animal identities in the two halves of the video. This implies that classification models which may learn very well in the training data, e.g., deep neural networks, may be inadequate for the testing half of the video.

#### 4.2. Feature extraction

- *Colour-related (RGB)*. RGB moments: The image with the animal was divided into

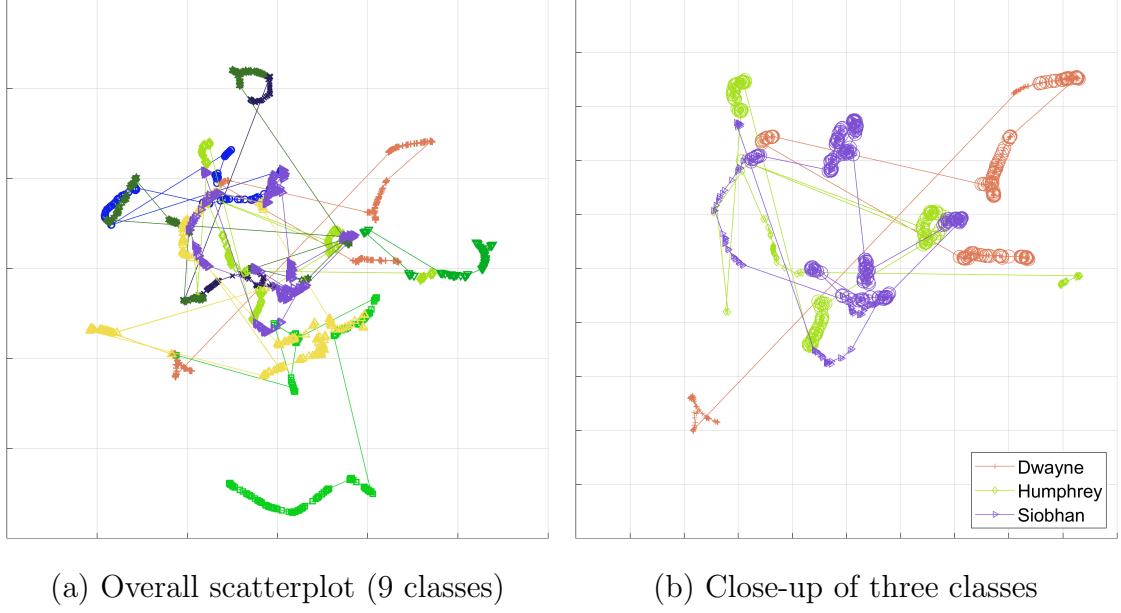


Figure 4: Two-dimensional representation of the RGB dataset from the Koi fish video after applying UMAP

3-by-3 blocks. For each block, we calculate and store the mean and the standard deviations of the red, the green, and the blue panel, which results in a total of 54 RGB features. A MATLAB function `get_rgb_features` is provided in the Github repository <https://github.com/admirable-ubu/animal-recognition/>. We store all the code there.

- *Shape-related (HOG)*. We resized all individual images to a square with side  $a$  (we used  $a = 40$ ) and extracted a Histogram of Oriented Gradients (HOG) from the colour image. MATLAB function `extractHOGfeatures` was used with default parameters, resulting in 576 HOG features.
- *Texture-related (LBP)*. Local Binary Patterns (LBP) features were extracted using from the grey-scale, resized image using MATLAB function `extractLBPfeatures`. We used the default parameters, apart from setting ‘Upright’ to false, in order to allow for rotation-invariant features. The function returned 10 LBP features.
- *Autoencoder (AE)*. Autoencoders are deep learning neural networks which are trained to reconstruct the input. There is a “code” layer, which contains the so called latent representation of the input. The outputs of the code layer are the features extracted by the AE network. MATLAB function `trainAutoencoder` was used with default

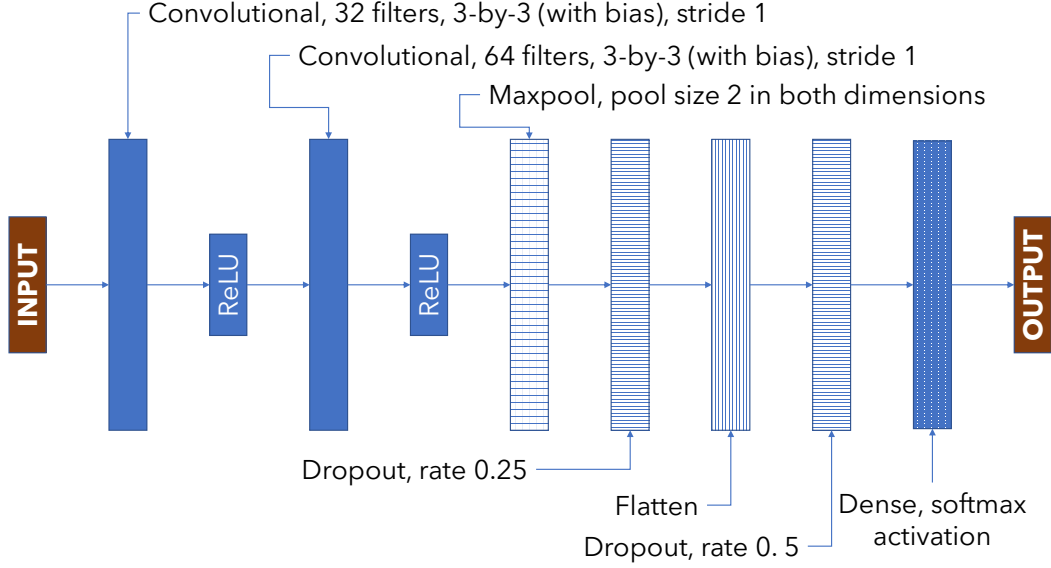


Figure 5: The CNN configuration

parameters. The latent representation is of size 10, which gives 10 AE features. The network was trained on the whole dataset while ignoring any class labels.

- *MobileNetV2 (MN2)*. We used the Keras MobileNetV2 model pre-trained on Imagenet. The last layer was cut off, and replaced with a GlobalAveragePooling layer, which yielded 1280 features. Python code for this part of the experiment is provided using function `extractMobilNetfeatures` from `functions.py`.

#### 4.3. Classifiers




We included 23 classifiers from the Python library *lazypredict* [53], based on *scikit-learn* [54]. These were all the classifiers in this library that could be applied to our data. We grouped the classifiers into: baseline, linear, non-linear, and ensembles, as shown in Table 2. Details of these methods can be found in the scikit-learn documentation and the books by Géron [55] and Raschka et al. [56]. These classifiers were applied to the five data representations detailed in Section 4.2. The Largest Prior classifier (Classifier 1 in the Table; also known as Majority or ZeroR classifier) was chosen as a baseline.

We also trained a bespoke Convolutional Neural Network (CNN) using a standard structure as detailed in Figure 5.

The CNN was trained using the Adam optimiser and binary cross-entropy loss.

To explore further the potential of deep learning, we used the Keras MobileNetV2 model pre-trained on Imagenet. The last layer was cut off, and replaced with: GlobalAv-

Table 2: Classifiers used in this study. The colour boxes correspond to the colours in the figures, with results 7 – 11. In the electronic version of the document, classifier names include a hyperlink to the classifier implementation documentation.

| <b>Baseline</b>  |  |
|--|--|
| 1. Largest Prior classifier (ZeroR/ Majority)  |  |
|  <b>Linear</b>          |  <b>Non-Linear</b>  |
| 2. Bernoulli (Naïve Bayes)   | 14. DecisionTree (C45)   |
| 3. Calibrated CV   | 15. Extra Tree   |
| 4. Gaussian Naïve Bayes  | 16. K-nn   |
| 5. Linear Discriminant Analysis  | 17. Quadratic Discriminant Analysis  |
| 6. Linear SVM  | 18. SVM  |
| 7. Logistic Regression   |  |
| 8. Nearest Centroid  |  <b>Ensembles</b> |
| 9. Passive Aggressive Classifier   |  |
| 10. Perceptron   | 19. AdaBoost   |
| 11. Ridge Regression   | 20. Bagging  |
| 12. Ridge Regression CV  | 21. Extra Tree Ensemble  |
| 13. SGD  | 22. LGBM   |
|  | 23. Random Forest  |
|  <b>Deep Learning</b> |  |
| 24. Convolutional Neural Network (CNN)   |  |
| 25. Transfer learning using MobileNetV2 (MNV2)   |  |

eragePooling layer, followed by a Dropout layer with rate 0.2. Finally, we added a Dense layer with softmax activation. The network was trained with the same training options as the bespoke CNN network.

For both deep learning models, we used data augmentation. During training, we modified each image according to a random augmentation selected among: random rotation at up to 30 degrees, zoom with a range of 0.2, random horizontal shift up 0.1 of the image width, random vertical shift up 0.1 of the image height, and horizontal flip.

Python code is available at <https://github.com/admirable-ubu/animal-recognition/>. All experiments were carried out on Rocky Linux 8.5 with two Intel Xeon Platinum 8358 CPU @ 2.60GHz and Nvidia RTX A6000 with 48 GiB of VRAM.

## 5. Results

### 5.1. Ranking of the classifier models

According to the proposed protocol, two-fold cross-validation was carried out where the videos were split into halves so that the frame continuity is preserved. Each half has been used once for training and once for testing. The classification accuracies were averaged across the two folds. These experiments were run separately for each of the five feature representations described in Section 4.2.

The obtained classification accuracies for all classifiers and all feature representations can be found in Tables 6 – 5 in the Appendix.

The first set of results we show are average ranking box plots. As the classification accuracies are not commensurable from one video to another, we used ranks. Each video together with a feature representation are considered a separate data set, so the total number of datasets is 25. For each dataset, the classifiers are assigned a rank between 1 and 25, as there are also 25 classifiers. The most accurate classifier receives rank 1 and the least accurate classifier, rank 25. Tied ranks are shared so that the sum of ranks is preserved. Thus, each classifier receives 25 rank values. Figure 6 shows a boxplot of the classifier ranks. The classifiers are arranged from the best (leftmost, the lowest rank), to the worst, which, as expected, is the Largest Prior classifier (the baseline). The best classifier in our experiment was the LDA.

Admittedly, while some classification models are quite robust (e.g., LDA), others depend substantially on how they are tuned for the application task. In this experiment

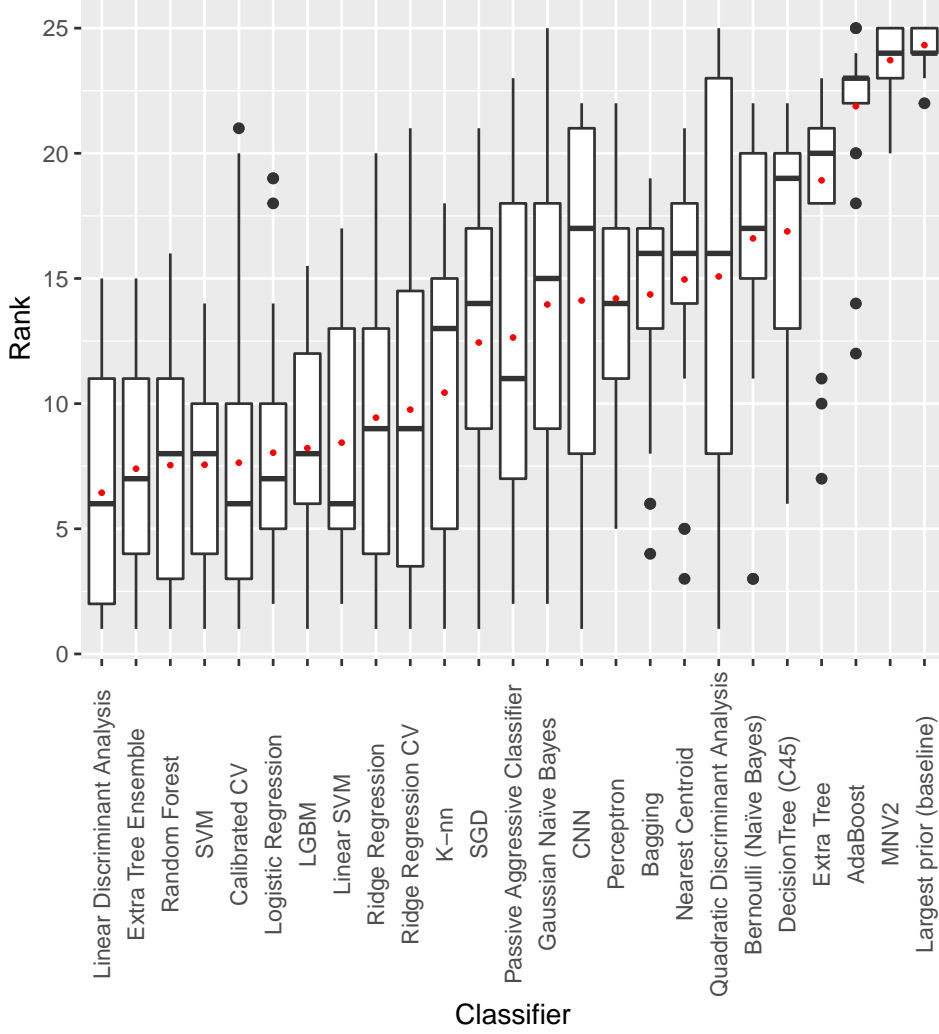


Figure 6: Box plot of the 25 classifier ranks across the 25 datasets. The classifiers are arranged from best (left) to worst (right) according to the mean (red dot).

we used the default parameters of all classification methods and their training options, apart from the CNN, which was designed ad hoc. We ran a small additional study to check whether tuning of some of the more susceptible models will lead to a great difference in the classification accuracy.

## 5.2. Feature representation results

Next, we look at the feature representations. Figures 7 – 11 show glyph plots of the classification accuracies. Each video has a separate figure. The five plots in each figure correspond to feature representations. The classification accuracies are represented by the spoke sizes. The subplots in each figure are scaled so that the largest spoke corresponds to the maximum accuracy across all feature representations for the respective

video. This spoke is shown in red. The numbers in parentheses within the subplot titles are the average classification accuracy for the respective feature representation across the 25 classifiers. The classifier groups detailed in Table 2 are shown with different shading. It can be seen that sometimes the best accuracy does not come from the feature representation with the best overall accuracy.

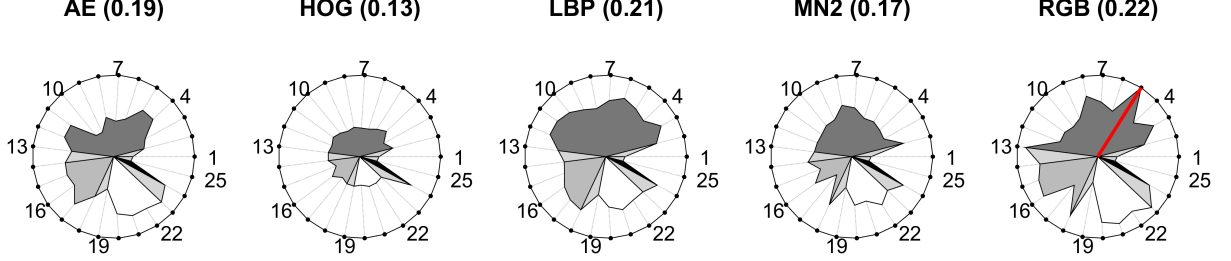


Figure 7: Classification accuracy of the 25 classifiers for the five feature representations for the Koi Fish video. Best accuracy of 34.13% was achieved with RGB feature representation and the LDA classifier.

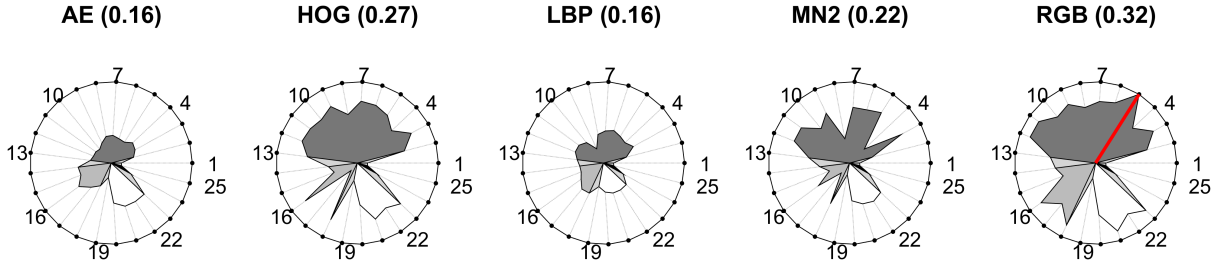


Figure 8: Classification accuracy of the 25 classifiers for the five feature representations for the Pigeons (square) video. Best accuracy of 49.13% was achieved with RGB feature representation and the LDA classifier.

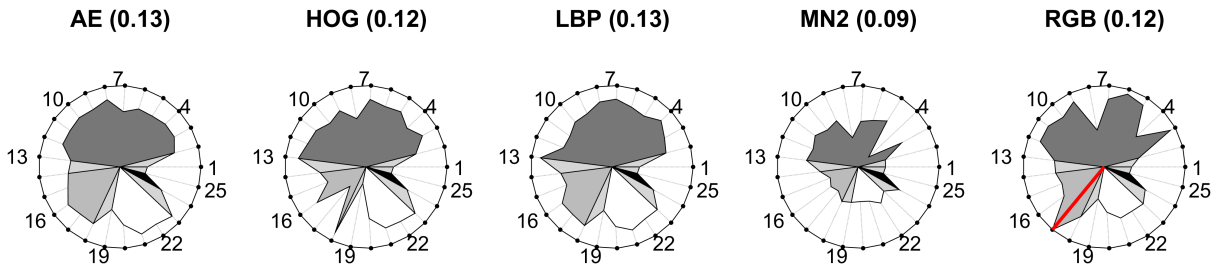


Figure 9: Classification accuracy of the 25 classifiers for the five feature representations for the Pigeons (pavement) video. Best accuracy of 18.41% was achieved with RGB feature representation and the QDA classifier.

Figure 12 shows the ranks for the feature representations, considering each pair of classifier and video as an item. Hence, for each feature representation there are  $23 \times 5 = 115$  ranks. The figure shows that the RGB representation markedly outperforms the rest.

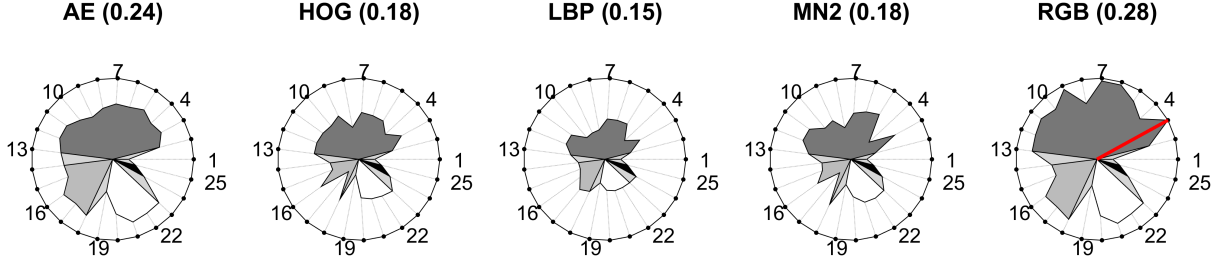


Figure 10: Classification accuracy of the 25 classifiers for the five feature representations for the Pigeons (curb) video. Best accuracy of 38.53% was achieved with RGB feature representation and the Calibrated CV classifier.

As we are interested in the combination of feature representation and classifier, the best accuracies are shown in Table 3.

### 5.3. Tuning parameters

The previous results were obtained with default parameter values. Results can usually be improved tuning parameters, but with a higher computational cost. In order to get a first idea of the effect of this tuning, we conducted an affordable experiment with a subset of the classifiers (i.e., Linear Discriminant Analysis, Quadratic Discriminant Analysis, SVM, Extra Tree Ensemble and Random Forest) and the RGB representation. They were selected because they were the best classifiers and representation in the previous experiment.

Parameter tuning was performed using an AutoML (automated machine learning) [57] tool, *auto-sklearn* [58]. The parameters to tune and their possible values were those predefined in *auto-sklearn*.

Table 4 shows the best accuracies for each video from the classifiers with tuning. Comparing with the results without tuning in Table 3, the results with tuning are only better for the Pigs video and worse for the others. The reason could be that parameter

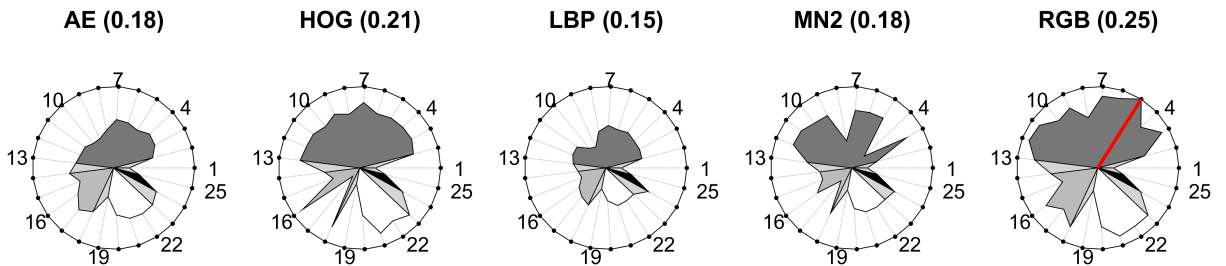


Figure 11: Classification accuracy of the 25 classifiers for the five feature representations for the Pigs video. Best accuracy of 34.51% was achieved with RGB feature representation and the LDA classifier.



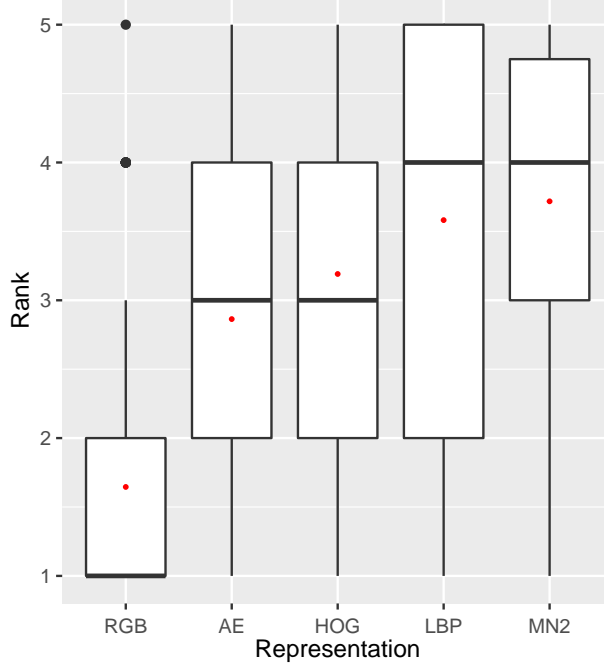


Figure 12: Box plot of the ranking of the 5 feature representations arranged from best (left) to worst (right) according to the mean (red dot).

tuning in these data sets increases over-fitting.

The CNN parameters were also tuned in another experiment, using *keras-tuner* [59]. For each fold, the size of the validation data was 25%. The search space is defined by the following parameters and values. Number of filters for the first convolutional layer: 16, 32 and 64. Number of filters for the second convolutional layer: 32, 64 and 128. Dropout rate for both dropout layers: between 0 and 0.5 in steps of 0.1. Dimensions of the output space in the intermediate dense layer: 64, 128 and 256. Learning rates: 0.01, 0.001 and 0.0001. Epochs: between 3 and 200.

The results of this experiment are shown in Table 5. The results with tuning are close

Table 3: Accuracy for the best combination of classifier and feature representation for each video.

|              | Koi   | Pigeons (square) | Pigeons (pavement) | Pigeons (curb) | Pigs  |
|--------------|-------|------------------|--------------------|----------------|-------|
| Classifier   | LDA   | LDA              | QDA                | Calibrated CV  | LDA   |
| Features set | RGB   | RGB              | RGB                | RGB            | RGB   |
| Accuracy     | 34.13 | 49.13            | 18.41              | 38.53          | 34.57 |

to the results without tuning and they are worse for two of the five videos.

#### 5.4. Discussion

The overwhelming opinion, backed by numerous studies, is that the use of deep learning methods for animal re-identification improves accuracy [3, 41]. In contrast, our results show that, for our data, LDA is the best classifier. It stands at the top of the overall average ranking, obtaining the best accuracy on three of the five datasets. The overall and individual best performance for all datasets is obtained using the colour feature representation, surpassing the texture and shape representations, which resonates with findings by other authors [28]. Neither the ensemble methods nor the deep learning models, both of which were expected to work well, produced reasonable accuracy. Moreover, parameter tuning also does not produce more promising results, for the classifiers considered. We have already discussed the possible reasons in Section 4.1.

In real-life scenarios, the data will likely be fairly different from one part of the video to the next. Classifiers which are capable of capturing intricate classification structures will suffer more heavily when distribution changes, compared to simpler classifiers which rely on the most generic representation of the data. This is why LDA, and the linear models in general, outperformed their competitors in our case study, using the RGB features. If, however, the distributions are likely to be relatively static along the video (e.g., if a fixed camera is used), more complex models may dominate. This reinforces the importance on carrying out an experimental study to compare different classifier models, keeping the cross-validation folds contingent.

We observe that, compared to similar studies on animal re-identification, our classification accuracy is low. There are different reasons for this result. First, in many studies, a typical measure of accuracy is based on whether the correct identity is among

Table 4: Accuracy for the best classifiers with tuning for each video. The complete set of results for these classifiers are in Table 11.

|            | Koi   | Pigeons (square) | Pigeons (pavement) | Pigeons (curb) | Pigs  |
|------------|-------|------------------|--------------------|----------------|-------|
| Classifier | LDA   | LDA              | LDA                | QDA            | LDA   |
| Accuracy   | 29.30 | 45.86            | 15.73              | 32.01          | 35.10 |

Table 5: Cross-validation classification accuracy [in %] for the deep learning models using raw images and data augmentation.

|                 | Koi   | Pigeons (square) | Pigeons (pavement) | Pigeons (curb) | Pigs  |
|-----------------|-------|------------------|--------------------|----------------|-------|
| MNV2            | 7.53  | 6.44             | 5.93               | 11.15          | 11.21 |
| CNN             | 24.74 | 13.64            | 10.63              | 16.95          | 20.68 |
| CNN with tuning | 26.20 | 13.32            | 12.51              | 17.52          | 17.47 |

the top five results retrieved from the database (top-5 accuracy) [50, 45, 12]. This metric assumes that the classifier approach is a version of the nearest-neighbour classifier. We experimented with standard classifier models which return only one class label. Variants of the standard classification models can be devised, especially for the most successful methods, which return a ranked list of similar instances from a chosen reference database.

One issue that we faced, even in these short video clips, which accounts for the low accuracy, is the so-called *open set* recognition problem [50]. This refers to the introduction of new classes (identities) in the testing part of the video. As the classifier has not seen these classes in the training, it will mistakenly label the instances as belonging to some of the existing classes. This will inevitably introduce extra classification error compared to the closed-set case. One possible approach to address this problem is to use classifiers which are confidence-conscious. Such classifiers will refuse to assign a label if the certainty of the classification decision is low. They can be tuned to achieve acceptable accuracy at the expense of declining to label a proportion of instances from known classes.

Finally, in addition to the concept change between training and testing data, the low accuracy can be attributed to the large intra-class variability illustrated in Figure 4. Besides, some of the classes were too small for the classification algorithm to learn the pattern properly.

While we do not offer answers to all the questions raised here, the proposed protocol is meant to give general guidelines to the practitioners in the field.

## 6. Conclusion

In this paper we argue that classification experiments to determine the best model for animal re-identification for a given dataset are paramount. We propose a general protocol and carry out a case study on a difficult animal re-identification database. We examined 25 classification models and five data representations. Two of the classification models and two of the data representations were based on the currently acclaimed deep learning. Our results favoured simple linear models (LDA) and basic feature representation (colour through RGB moments). We attribute this result to the complexity of the data distribution, multitude of overlapping classes, and the difference of the distributions between the cross-validation folds. Our findings highlight a somewhat overlooked message that deep learning is not the answer to all tasks, and many times simple classifier models work better for animal re-identification, e.g., [12].

There are several limitations to our study.

First, the data set for each animal group is only one video. The appearance of the animals in a video taken at a different time or under different illumination conditions may not match the current model. If multiple videos are available, annotation should be carried out on a small portion of each video. The annotated data can be pooled together and the protocol can be applied thereafter.

At the start, we decided to ignore time contingency of the video frames in order to be able to apply the protocol to a collection of images that does not come necessarily from video. However, if the source is video footage, we can use consecutive frames to establish links between the objects that are being classified. Also, the instances coming from the same frame must have different identities. Thus, we can impose Must Link (ML) constraints and Cannot Link (CL) constraints on the *testing* data, without the need of any further supervision or annotation. Our next study is on incorporating this information in the classification process. Future research may also explore combinations of feature representations, as well as various methods for dimensionality reduction.

The use of deep learning can be explored further. As advocated by recent studies, similarity networks may learn an informative feature representation from small numbers of instance per class.

The future in animal re-identification is likely to involve active learning (a human in the loop) due to the changing environments, concept drift, and inevitable problem

variability [50]. Classes will appear, disappear, or reappear; the class description may change, i.e., the appearance of the same individual may vary with time. Involving a human in the loop will ensure that the open-set classification process is steered in the correct vein. Adaptive, semi-supervised classification methods are likely to be the most suitable choice, contributing to end-to-end automation for tasks such as tracking.

## Acknowledgement

This work is supported by the UKRI Centre for Doctoral Training in Artificial Intelligence, Machine Learning and Advanced Computing (AIMLAC), funded by grant EP/S023992/1. This work is also supported by the Junta de Castilla León under project BU055P20 (JCyL/FEDER, UE), and the Ministry of Science and Innovation under project PID2020-119894GB-I00 co-financed through European Union FEDER funds. J.L. Garrido-Labrador is supported through Consejería de Educación of the Junta de Castilla y León and the European Social Fund through a pre-doctoral grant (EDU/875/2021). I. Ramos-Perez is supported by the predoctoral grant (BDNS 510149) awarded by the Universidad de Burgos, Spain. J.J. Rodríguez was supported by mobility grant PRX21/00638 of the Spanish Ministry of Universities.

## References

- [1] S. Kumar, S. K. Singh, Visual animal biometrics: survey, *IET Biometrics* 6 (3) (2017) 139–156. doi:10.1049/iet-bmt.2016.0017.
- [2] H. S. Kühl, T. Burghardt, Animal biometrics: quantifying and detecting phenotypic appearance, *Trends in ecology & evolution* 28 (7) (2013) 432–441.
- [3] S. Schneider, G. W. Taylor, S. Linquist, S. C. Kremer, Past, present and future approaches using computer vision for animal re-identification from camera trap data, *Methods in Ecology and Evolution* 10 (4) (2019) 461–470. doi:10.1111/2041-210x.13133.
- [4] R. Kays, M. C. Crofoot, W. Jetz, M. Wikelski, Terrestrial animal tracking as an eye on life and planet, *Science* 348 (6240) (2015). arXiv:<https://science.sciencemag.org/content/348/6240/aaa2478.full.pdf>, doi:10.1126/science.aaa2478. URL <https://science.sciencemag.org/content/348/6240/aaa2478>
- [5] W. J. Eradus, M. B. Jansen, Animal identification and monitoring, *Computers and Electronics in Agriculture* 24 (1-2) (1999) 91–98.
- [6] A. I. Awad, From classical methods to animal biometrics: A review on cattle identification and tracking, *Computers and Electronics in Agriculture* 123 (2016) 423–435. doi:10.1016/j.compag.2016.03.014.

- [7] C. W. Speed, M. G. Meekan, C. J. A. Bradshaw, Spot the match—wildlife photo-identification using information theory, *Frontiers in zoology* 4 (1) (2007) 1–11.
- [8] N. K. S. Behera, P. K. Sa, S. Bakshi, R. P. Padhy, Person re-identification: A taxonomic survey and the path ahead, *Image and Vision Computing* 122 (2022) 104432.
- [9] B. G. Weinstein, A computer vision for animal ecology, *Journal of Animal Ecology* 87 (3) (2018) 533–545.
- [10] D. Ramanan, D. A. Forsyth, K. Barnard, Building models of animals from video, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (8) (2006) 1319–1334.
- [11] T. Burghardt, N. Campbell, Individual animal identification using visual biometrics on deformable coat patterns, in: *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)*, 2007.
- [12] E. Nepovinskykh, T. Eerola, V. Biard, P. Mutka, M. Niemi, M. Kunnasranta, H. Kalviainen, SealID: Saimaa ringed seal re-identification dataset, *SENSORS* 22 (19) (OCT 2022). doi:10.3390/s22197602.
- [13] K. Nadolin, Cnn-based ringed seal pelage pattern extraction, Ph.D. thesis, University of Technology.
- [14] S. Bouma, M. D. Pawley, K. Hupman, A. Gilman, Individual common dolphin identification via metric embedding learning, in: *2018 international conference on image and vision computing New Zealand (IVCNZ)*, IEEE, 2018, pp. 1–6.
- [15] V. Miele, G. Dussert, B. Spataro, S. Chamaille-Jammes, D. Allaine, C. Bonenfant, Revisiting animal photo-identification using deep metric learning and network analysis (2021).
- [16] L. Wang, R. Ding, Y. Zhai, Q. Zhang, W. Tang, N. Zheng, G. Hua, Giant panda identification, *IEEE Transactions on Image Processing* 30 (2021) 2837–2849.
- [17] J. Chan, H. Carrion, R. Megret, J. L. Rivera Rivera, T. Giray, Honeybee re-identification in video: New datasets and impact of self-supervision, in: G. Farinella, P. Radeva, K. Bouatouch (Eds.), *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, Vol. 5 of VISIGRAPP, 2022, pp. 517–525. doi:10.5220/0010843100003124.
- [18] T. Zhang, Q. Zhao, C. Da, L. Zhou, L. Li, S. Jiancuo, Yakreid-103: A benchmark for yak re-identification, in: *2021 IEEE International Joint Conference on Biometrics (IJCB)*, 2021, pp. 1–8. doi:IEEE.
- [19] F. Okura, S. Ikuma, Y. Makihara, D. Muramatsu, K. Nakada, Y. Yagi, RGB-d video-based individual identification of dairy cows using gait and texture analyses, *Computers and Electronics in Agriculture* 165 (2019) 104944. doi:10.1016/j.compag.2019.104944.
- [20] K. Zhao, X. Jin, J. Ji, J. Wang, H. Ma, X. Zhu, Individual identification of holstein dairy cows based on detecting and matching feature points in body images, *Biosystems Engineering* 181 (2019) 128–139. doi:10.1016/j.biosystemseng.2019.03.004.
- [21] O. Guzhva, H. Ardö, M. Nilsson, A. Herlin, L. Tufvesson, Now you see me: Convolutional neural network based tracker for dairy cows, *Frontiers in Robotics and AI* 5 (2019). doi:10.3389/frobt.2018.00107.

- [22] T. T. Zin, C. N. Phyto, P. Tin, H. Hama, I. Kobayashi, Image technology based cow identification system using deep learning, in: *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 1, 2018, pp. 236–247.
- [23] M. Kashiha, C. Bahr, S. Ott, C. P. Moons, T. A. Niewold, F. Ödberg, D. Berckmans, Automatic identification of marked pigs in a pen using image pattern recognition, *Computers and Electronics in Agriculture* 93 (2013) 111–120. doi:10.1016/j.compag.2013.01.013.
- [24] M. Wang, M. L. Larsen, D. Liu, J. F. Winters, J.-L. Rault, T. Norton, Towards re-identification for long-term tracking of group housed pigs, *Biosystems Engineering* 222 (2022) 71–81.
- [25] W. Andrew, C. Greatwood, T. Burghardt, Visual localisation and individual identification of Holstein Friesian cattle via deep learning, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, 2017, pp. 2850–2859.
- [26] M. Willi, R. T. Pitman, A. W. Cardoso, C. Locke, A. Swanson, A. Boyer, M. Veldhuis, L. Fortson, Identifying animal species in camera trap images using deep learning and citizen science, *Methods in Ecology and Evolution* 10 (1) (2018) 80–91. doi:10.1111/2041-210x.13099.
- [27] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, J. Clune, Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning, *Proceedings of the National Academy of Sciences* 115 (25) (2018) E5716–E5725. doi:10.1073/pnas.1719367115.
- [28] M. Zeppelzauer, Automated detection of elephants in wildlife video, *EURASIP Journal on Image and Video Processing* 2013 (1) (2013) 1–23. doi:10.1186/1687-5281-2013-46.
- [29] D. Schofield, A. Nagrani, A. Zisserman, M. Hayashi, T. Matsuzawa, D. Biro, S. Carvalho, Chimpanzee face recognition from videos in the wild using deep learning, *Science Advances* 5 (9) (2019) eaaw0736. doi:10.1126/sciadv.aaw0736.
- [30] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, G. G. de Polavieja, idTracker: tracking individuals in a group by automatic identification of unmarked animals, *Nature Methods* 11 (7) (2014) 743–748. doi:10.1038/nmeth.2994.
- [31] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. H. Heras, G. G. de Polavieja, idtracker.ai: tracking all individuals in small or large collectives of unmarked animals, *Nature Methods* 16 (2) (2019) 179–182. doi:10.1038/s41592-018-0295-5.
- [32] Z. XU, X. E. Cheng, Zebrafish tracking using convolutional neural networks, *Scientific Reports* 7 (1) (2017). doi:10.1038/srep42815.
- [33] F. Naiser, M. Šmíd, J. Matas, Tracking and re-identification system for multiple laboratory animals, in: *International Conference on Pattern Recognition (ICPR)*, 2018, workshop: Visual observation and analysis of vertebrate and insect behavior.
- [34] L. I. Kuncheva, F. Williams, S. L. Hennessey, J. J. Rodríguez, A benchmark database for animal re-identification and tracking, in: *Proc. of the Fifth IEEE International Conference on Image Processing, Applications and Systems (IPAS 2022)*, 2022.
- [35] D. Deb, S. Wiper, S. Gong, Y. Shi, C. Tymoszek, A. Fletcher, A. K. Jain, Face recognition: Primates in the wild, in: *2018 IEEE 9th International Conference on Biometrics Theory, Applications and*

- Systems (BTAS), IEEE, 2018. doi:10.1109/btas.2018.8698538.
- [36] D. Crouse, R. L. Jacobs, Z. Richardson, S. Klum, A. Jain, A. L. Baden, S. R. Tecot, LemurFaceID: A face recognition system to facilitate individual identification of lemurs, *Bmc Zoology* 2 (1) (2017) 1–14.
  - [37] L. Bergamini, A. Porrello, A. C. Dondona, E. D. Negro, M. Mattioli, N. D’Alterio, S. Calderara, Multi-views embedding for cattle re-identification, *arXiv* (2019). arXiv:1902.04886v1.
  - [38] S. Kumar, S. Tiwari, S. K. Singh, Face recognition for cattle, in: *Processing of the Third International Conference on Image Information Processing (ICIIP)*, IEEE, 2015, pp. 65–72.
  - [39] M. Billah, X. Wang, J. Yu, Y. Jiang, Real-time goat face recognition using convolutional neural network, *Computers and Electronics in Agriculture* 194 (2022) 106730.
  - [40] S. Kumar, S. K. Singh, Monitoring of pet animal in smart cities using animal biometrics, *Future Generation Computer Systems* 83 (2018) 553–563. doi:10.1016/j.future.2016.12.006.
  - [41] P. C. Ravor, T. Sudarshan, Deep learning methods for multi-species animal re-identification and tracking—a survey, *Computer Science Review* 38 (2020) 100289.
  - [42] O. Moskvayak, F. Maire, F. Dayoub, A. O. Armstrong, M. Baktashmotlagh, Robust re-identification of manta rays from natural markings by learning pose invariant embeddings, in: *2021 Digital Image Computing: Techniques and Applications (DICTA)*, IEEE, 2021, pp. 1–8.
  - [43] V. Miele, G. Dussert, B. Spataro, S. Chamailé-Jammes, D. Allainé, C. Bonenfant, Revisiting giraffe photo-identification using deep learning and network analysis (2020).
  - [44] S. Schneider, G. W. Taylor, S. C. Kremer, Similarity learning networks for animal individual re-identification: an ecological perspective, *Mammalian Biology* doi:10.1007/s42991-021-00215-1.
  - [45] N. Dlamini, T. L. van Zyl, Comparing class-aware and pairwise loss functions for deep metric learning in wildlife re-identification, *Sensors* 21 (18) (2021) 6109.
  - [46] E. Nepovinskykh, T. Eerola, H. Kalviainen, Siamese network based pelage pattern matching for ringed seal re-identification, in: *Proceedings of the IEEE/CVF winter conference on applications of computer vision workshops*, 2020, pp. 25–34.
  - [47] T. L. Van Zyl, M. Woolway, B. Engelbrecht, Unique animal identification using deep transfer learning for data fusion in siamese networks, in: *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, IEEE, 2020, pp. 1–6.
  - [48] J. P. Crall, Identifying individual animals using ranking, verification, and connectivity, *Rensselaer Polytechnic Institute*, 2017.
  - [49] S. Ren, K. He, R. B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, *CoRR* abs/1506.01497 (2015). arXiv:1506.01497.  
URL <http://arxiv.org/abs/1506.01497>
  - [50] P. Bodesheim, J. Blunk, M. Körschens, C.-A. Brust, C. Käding, J. Denzler, Pre-trained models are not enough: active and lifelong learning is important for long-term visual monitoring of mammals in biodiversity research—individual identification and attribute prediction with image features from deep neural networks and decoupled decision models applied to elephants and great apes, *Mammalian Biology* (2022) 1–23.



- [51] L. I. Kuncheva, J. L. Garrido-Labrador, I. Ramos-Pérez, S. L. Hennessey, J. J. Rodriguez, Animal re-identification from video [data set], Zenodo (Nov. 2022). doi:10.5281/zenodo.7322821. URL <https://doi.org/10.5281/zenodo.7322821>
- [52] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018).
- [53] S. R. Pandala, B. B. da Silva, Lazy predict, github repository, <https://github.com/shankarpandala/lazypredict> (2022).
- [54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [55] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O’Reilly Media, Inc., 2022.
- [56] S. Raschka, Y. H. Liu, V. Mirjalili, D. Dzhulgakov, Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python, Packt Publishing Ltd, 2022.
- [57] X. He, K. Zhao, X. Chu, AutoML: A survey of the state-of-the-art, *Knowledge-Based Systems* 212 (2021) 106622.
- [58] M. Feurer, A. Klein, K. Eggenberger, J. Springenberg, M. Blum, F. Hutter, Efficient and robust automated machine learning, in: *Advances in Neural Information Processing Systems* 28 (2015), 2015, pp. 2962–2970.
- [59] T. O’Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, et al., Keras Tuner, <https://github.com/keras-team/keras-tuner> (2019).

## Appendix

Tables 6–11 show the complete set of results.

Table 6: Classification accuracy of the classifiers for the feature representations for the Koi fish video.

|    | Classifier                      | AE    | HOG   | LBP   | MN2   | RGB   | Average |
|----|---------------------------------|-------|-------|-------|-------|-------|---------|
| 1  | Largest prior (baseline)        | 8.19  | 8.19  | 8.19  | 8.19  | 8.19  | 8.19    |
| 2  | Bernoulli (Naïve Bayes)         | 13.22 | 14.41 | 22.04 | 22.37 | 20.12 | 18.43   |
| 3  | Calibrated CV                   | 15.14 | 11.57 | 26.63 | 15.38 | 26.77 | 19.10   |
| 4  | Gaussian Naïve Bayes            | 22.77 | 15.46 | 21.84 | 16.14 | 20.57 | 19.36   |
| 5  | Linear Discriminant Analysis    | 22.93 | 13.56 | 23.63 | 17.23 | 34.13 | 22.30   |
| 6  | Linear SVM                      | 16.51 | 12.09 | 25.59 | 16.82 | 19.96 | 18.19   |
| 7  | Logistic Regression             | 14.85 | 11.96 | 23.46 | 20.41 | 22.88 | 18.71   |
| 8  | Nearest Centroid                | 16.68 | 12.46 | 20.41 | 21.80 | 25.68 | 19.41   |
| 9  | Passive Aggressive Classifier   | 11.17 | 11.71 | 21.65 | 16.94 | 17.70 | 15.83   |
| 10 | Perceptron                      | 11.47 | 12.72 | 23.89 | 16.77 | 16.13 | 16.20   |
| 11 | Ridge Regression                | 22.23 | 13.72 | 25.15 | 15.58 | 17.20 | 18.78   |
| 12 | Ridge Regression CV             | 22.18 | 12.50 | 25.15 | 15.58 | 16.98 | 18.48   |
| 13 | SGD                             | 17.72 | 13.27 | 20.08 | 15.82 | 30.60 | 19.50   |
| 14 | DecisionTree (C45)              | 20.47 | 13.28 | 17.57 | 18.52 | 24.50 | 18.87   |
| 15 | Extra Tree                      | 20.99 | 10.56 | 18.62 | 14.13 | 26.70 | 18.20   |
| 16 | K-nn                            | 20.29 | 14.10 | 21.43 | 18.81 | 28.10 | 20.55   |
| 17 | Quadratic Discriminant Analysis | 25.67 | 11.31 | 23.41 | 9.86  | 15.53 | 17.16   |
| 18 | SVM                             | 18.27 | 12.52 | 24.54 | 22.10 | 27.07 | 20.90   |
| 19 | AdaBoost                        | 13.73 | 12.57 | 16.17 | 13.13 | 10.87 | 13.29   |
| 20 | Bagging                         | 23.95 | 11.95 | 21.09 | 20.35 | 27.51 | 20.97   |
| 21 | Extra Tree Ensemble             | 25.91 | 13.07 | 23.27 | 19.52 | 29.39 | 22.23   |
| 22 | LGBM                            | 25.69 | 13.02 | 21.54 | 23.79 | 27.00 | 22.21   |
| 23 | Random Forest                   | 27.66 | 12.56 | 21.54 | 20.64 | 31.05 | 22.69   |
|    | Average                         | 19.03 | 12.55 | 21.60 | 17.39 | 22.81 | 18.68   |

Table 7: Classification accuracy of the classifiers for the feature representations for the Pigeons (square) video.

|    | Classifier                      | AE    | HOG   | LBP   | MN2   | RGB   | Average |
|----|---------------------------------|-------|-------|-------|-------|-------|---------|
| 1  | Largest prior (baseline)        | 6.18  | 6.18  | 6.18  | 6.18  | 6.18  | 6.18    |
| 2  | Bernoulli (Naïve Bayes)         | 14.02 | 29.41 | 14.34 | 14.11 | 32.03 | 20.78   |
| 3  | Calibrated CV                   | 16.51 | 37.26 | 20.67 | 35.66 | 39.83 | 29.99   |
| 4  | Gaussian Naïve Bayes            | 17.84 | 28.51 | 17.93 | 12.33 | 30.70 | 21.46   |
| 5  | Linear Discriminant Analysis    | 14.92 | 34.05 | 19.04 | 35.99 | 49.13 | 30.62   |
| 6  | Linear SVM                      | 15.42 | 36.59 | 20.60 | 33.77 | 38.23 | 28.92   |
| 7  | Logistic Regression             | 16.72 | 37.37 | 19.41 | 33.67 | 37.19 | 28.87   |
| 8  | Nearest Centroid                | 16.35 | 29.81 | 17.55 | 14.94 | 34.13 | 22.55   |
| 9  | Passive Aggressive Classifier   | 13.24 | 37.68 | 9.48  | 33.18 | 39.08 | 26.53   |
| 10 | Perceptron                      | 10.81 | 34.01 | 16.01 | 26.97 | 36.15 | 24.79   |
| 11 | Ridge Regression                | 11.47 | 35.16 | 18.28 | 36.12 | 42.54 | 28.71   |
| 12 | Ridge Regression CV             | 11.45 | 35.93 | 18.28 | 35.98 | 42.62 | 28.85   |
| 13 | SGD                             | 12.61 | 30.41 | 15.97 | 24.69 | 27.59 | 22.26   |
| 14 | DecisionTree (C45)              | 20.10 | 17.61 | 14.42 | 18.07 | 24.52 | 18.94   |
| 15 | Extra Tree                      | 19.67 | 15.02 | 14.47 | 13.08 | 21.12 | 16.67   |
| 16 | K-nn                            | 24.17 | 38.91 | 16.69 | 23.08 | 40.80 | 28.73   |
| 17 | Quadratic Discriminant Analysis | 18.97 | 5.92  | 20.40 | 7.82  | 31.88 | 17.00   |
| 18 | SVM                             | 15.19 | 37.84 | 20.07 | 27.10 | 41.36 | 28.31   |
| 19 | AdaBoost                        | 8.91  | 10.77 | 14.54 | 7.08  | 9.33  | 10.13   |
| 20 | Bagging                         | 24.21 | 25.25 | 17.59 | 22.86 | 31.38 | 24.26   |
| 21 | Extra Tree Ensemble             | 27.41 | 35.58 | 19.04 | 25.88 | 43.16 | 30.22   |
| 22 | LGBM                            | 27.46 | 32.09 | 19.25 | 27.31 | 36.36 | 28.50   |
| 23 | Random Forest                   | 27.46 | 35.95 | 19.00 | 25.88 | 41.73 | 30.00   |
|    | Average                         | 17.00 | 29.01 | 16.92 | 23.55 | 33.78 | 24.06   |

Table 8: Classification accuracy of the classifiers for the feature representations for the Pigeons (pavement) video.

|    | Classifier                      | AE    | HOG   | LBP   | MN2   | RGB   | Average |
|----|---------------------------------|-------|-------|-------|-------|-------|---------|
| 1  | Largest prior (baseline)        | 6.26  | 6.26  | 6.26  | 6.26  | 6.26  | 6.26    |
| 2  | Bernoulli (Naïve Bayes)         | 12.77 | 11.68 | 12.66 | 6.39  | 8.11  | 10.32   |
| 3  | Calibrated CV                   | 14.09 | 14.63 | 13.60 | 11.25 | 17.18 | 14.15   |
| 4  | Gaussian Naïve Bayes            | 14.12 | 12.78 | 15.27 | 3.30  | 9.96  | 11.08   |
| 5  | Linear Discriminant Analysis    | 13.63 | 14.76 | 13.59 | 12.29 | 16.50 | 14.15   |
| 6  | Linear SVM                      | 13.77 | 14.31 | 14.23 | 11.04 | 17.29 | 14.13   |
| 7  | Logistic Regression             | 12.51 | 15.27 | 15.35 | 10.31 | 15.35 | 13.76   |
| 8  | Nearest Centroid                | 15.45 | 11.09 | 14.93 | 6.72  | 8.47  | 11.33   |
| 9  | Passive Aggressive Classifier   | 14.31 | 14.03 | 13.27 | 11.51 | 16.28 | 13.88   |
| 10 | Perceptron                      | 14.44 | 12.80 | 13.14 | 10.85 | 14.58 | 13.16   |
| 11 | Ridge Regression                | 13.97 | 13.98 | 11.81 | 12.03 | 15.66 | 13.49   |
| 12 | Ridge Regression CV             | 13.94 | 14.21 | 11.81 | 10.96 | 15.60 | 13.30   |
| 13 | SGD                             | 11.22 | 15.44 | 16.36 | 11.78 | 11.35 | 13.23   |
| 14 | DecisionTree (C45)              | 11.81 | 10.46 | 11.68 | 8.64  | 11.10 | 10.74   |
| 15 | Extra Tree                      | 12.68 | 9.27  | 11.05 | 6.78  | 10.27 | 10.01   |
| 16 | K-nn                            | 14.54 | 13.67 | 13.97 | 7.59  | 11.22 | 12.20   |
| 17 | Quadratic Discriminant Analysis | 13.93 | 5.70  | 13.96 | 7.02  | 18.41 | 11.81   |
| 18 | SVM                             | 14.17 | 16.44 | 14.65 | 8.62  | 12.39 | 13.26   |
| 19 | AdaBoost                        | 9.81  | 3.82  | 10.04 | 8.16  | 7.34  | 7.83    |
| 20 | Bagging                         | 13.11 | 11.71 | 13.50 | 7.86  | 10.26 | 11.29   |
| 21 | Extra Tree Ensemble             | 15.95 | 14.50 | 14.99 | 8.14  | 12.15 | 13.14   |
| 22 | LGBM                            | 15.34 | 14.60 | 13.93 | 9.68  | 11.82 | 13.08   |
| 23 | Random Forest                   | 16.26 | 15.30 | 14.09 | 8.00  | 12.18 | 13.17   |
|    | Average                         | 13.40 | 12.47 | 13.22 | 8.92  | 12.60 | 12.12   |

Table 9: Classification accuracy of the classifiers for the feature representations for the Pigeons (curb) video.

|    | Classifier                      | AE    | HOG   | LBP   | MN2   | RGB   | Average |
|----|---------------------------------|-------|-------|-------|-------|-------|---------|
| 1  | Largest prior (baseline)        | 7.59  | 7.59  | 7.59  | 7.59  | 7.59  | 7.59    |
| 2  | Bernoulli (Naïve Bayes)         | 22.94 | 16.49 | 12.60 | 11.97 | 25.55 | 17.91   |
| 3  | Calibrated CV                   | 26.31 | 23.11 | 18.92 | 23.70 | 38.53 | 26.12   |
| 4  | Gaussian Naïve Bayes            | 24.93 | 17.34 | 12.98 | 11.89 | 28.04 | 19.03   |
| 5  | Linear Discriminant Analysis    | 27.76 | 21.57 | 19.83 | 25.22 | 32.54 | 25.39   |
| 6  | Linear SVM                      | 25.98 | 21.75 | 19.46 | 23.67 | 37.06 | 25.59   |
| 7  | Logistic Regression             | 26.34 | 22.27 | 19.18 | 21.93 | 37.23 | 25.39   |
| 8  | Nearest Centroid                | 24.64 | 15.45 | 14.21 | 15.29 | 26.94 | 19.30   |
| 9  | Passive Aggressive Classifier   | 22.22 | 22.66 | 13.82 | 20.25 | 36.65 | 23.12   |
| 10 | Perceptron                      | 24.12 | 20.54 | 13.36 | 19.18 | 30.79 | 21.60   |
| 11 | Ridge Regression                | 26.98 | 21.01 | 19.57 | 24.28 | 32.67 | 24.90   |
| 12 | Ridge Regression CV             | 27.03 | 21.38 | 19.69 | 24.41 | 31.34 | 24.77   |
| 13 | SGD                             | 24.74 | 21.27 | 16.25 | 20.14 | 30.96 | 22.67   |
| 14 | DecisionTree (C45)              | 23.23 | 13.66 | 12.58 | 13.81 | 21.59 | 16.97   |
| 15 | Extra Tree                      | 22.64 | 13.28 | 12.48 | 14.47 | 20.80 | 16.73   |
| 16 | K-nn                            | 28.64 | 21.94 | 14.42 | 19.34 | 30.04 | 22.88   |
| 17 | Quadratic Discriminant Analysis | 27.34 | 7.57  | 18.60 | 8.36  | 29.48 | 18.27   |
| 18 | SVM                             | 29.41 | 21.06 | 16.86 | 23.73 | 31.47 | 24.51   |
| 19 | AdaBoost                        | 15.55 | 7.48  | 11.37 | 12.20 | 12.72 | 11.86   |
| 20 | Bagging                         | 26.00 | 18.36 | 14.76 | 17.83 | 26.89 | 20.77   |
| 21 | Extra Tree Ensemble             | 31.30 | 19.72 | 15.69 | 21.43 | 31.97 | 24.02   |
| 22 | LGBM                            | 30.32 | 20.67 | 15.84 | 21.81 | 30.57 | 23.84   |
| 23 | Random Forest                   | 30.12 | 22.03 | 16.23 | 21.56 | 30.30 | 24.05   |
|    | Average                         | 25.05 | 18.18 | 15.49 | 18.44 | 28.77 | 21.19   |

Table 10: Classification accuracy of the classifiers for the feature representations for the Pigs video.

|    | Classifier                      | AE    | HOG   | LBP   | MN2   | RGB   | Average |
|----|---------------------------------|-------|-------|-------|-------|-------|---------|
| 1  | Largest prior (baseline)        | 6.65  | 6.65  | 6.65  | 6.65  | 6.65  | 6.65    |
| 2  | Bernoulli (Naïve Bayes)         | 17.15 | 23.66 | 15.90 | 11.07 | 20.48 | 17.65   |
| 3  | Calibrated CV                   | 20.03 | 25.31 | 16.44 | 26.51 | 31.13 | 23.88   |
| 4  | Gaussian Naïve Bayes            | 20.98 | 25.18 | 16.60 | 7.26  | 25.17 | 19.04   |
| 5  | Linear Discriminant Analysis    | 18.90 | 25.02 | 17.73 | 24.99 | 34.51 | 24.23   |
| 6  | Linear SVM                      | 20.06 | 24.13 | 16.93 | 24.67 | 30.89 | 23.33   |
| 7  | Logistic Regression             | 20.51 | 27.71 | 18.01 | 24.38 | 30.41 | 24.20   |
| 8  | Nearest Centroid                | 16.19 | 23.19 | 15.70 | 11.60 | 21.67 | 17.67   |
| 9  | Passive Aggressive Classifier   | 13.69 | 25.13 | 9.79  | 24.46 | 27.99 | 20.21   |
| 10 | Perceptron                      | 13.47 | 22.74 | 10.78 | 24.69 | 25.98 | 19.53   |
| 11 | Ridge Regression                | 15.02 | 24.30 | 15.05 | 26.60 | 30.67 | 22.33   |
| 12 | Ridge Regression CV             | 15.04 | 24.34 | 15.05 | 26.72 | 31.46 | 22.52   |
| 13 | SGD                             | 16.17 | 25.50 | 13.90 | 19.04 | 27.11 | 20.34   |
| 14 | DecisionTree (C45)              | 18.86 | 14.97 | 11.78 | 13.55 | 19.96 | 15.82   |
| 15 | Extra Tree                      | 15.33 | 12.09 | 10.98 | 13.30 | 16.09 | 13.56   |
| 16 | K-nn                            | 18.70 | 29.92 | 14.01 | 17.96 | 27.17 | 21.55   |
| 17 | Quadratic Discriminant Analysis | 23.08 | 4.65  | 15.87 | 7.12  | 22.73 | 14.69   |
| 18 | SVM                             | 20.47 | 27.72 | 17.36 | 21.24 | 27.51 | 22.86   |
| 19 | AdaBoost                        | 12.36 | 6.90  | 9.08  | 9.54  | 9.84  | 9.55    |
| 20 | Bagging                         | 19.99 | 20.09 | 14.00 | 17.22 | 25.03 | 19.27   |
| 21 | Extra Tree Ensemble             | 22.45 | 29.08 | 16.75 | 19.56 | 30.36 | 23.64   |
| 22 | LGBM                            | 23.01 | 26.51 | 16.48 | 19.69 | 29.32 | 23.00   |
| 23 | Random Forest                   | 22.78 | 29.15 | 16.32 | 19.52 | 29.30 | 23.42   |
|    | Average                         | 17.86 | 21.91 | 14.40 | 18.15 | 25.28 | 19.52   |

Table 11: Classification accuracy of the tuned classifiers with the RGB representation.

|                                 | Koi<br>fish  | Pigeons<br>(square) | Pigeons<br>(pavement) | Pigeons<br>(curb) | Pigs         |
|---------------------------------|--------------|---------------------|-----------------------|-------------------|--------------|
| Linear Discriminant Analysis    | <b>29.30</b> | <b>45.86</b>        | <b>15.73</b>          | 31.65             | <b>35.10</b> |
| Quadratic Discriminant Analysis | 19.39        | 35.46               | 15.65                 | <b>32.01</b>      | 25.72        |
| SVM                             | 28.32        | 39.65               | 11.98                 | 28.88             | 27.28        |
| Extra Tree Ensemble             | 27.71        | 43.91               | 12.74                 | 31.80             | 30.16        |
| Random Forest                   | 27.84        | 42.75               | 11.34                 | 30.63             | 29.65        |