Semi-Supervised Classification With Pairwise Constraints: A Case Study on Animal Identification from Video

Ludmila I. Kuncheva[a], José Luis Garrido-Labrador[b], Ismael Ramos-Pérez[b], Samuel L. Hennessey[a], Juan J. Rodríguez[b]

[a] *Bangor University, Bangor, UK*
[b] *Universidad de Burgos, Burgos, Spain*

Abstract

Mainstream semi-supervised classification assumes that part of the available data are labelled. Here we assume that, in addition to the labels, we have pairwise constraints on the unlabelled data. Each constraint links two instances, and is one of Must Link (ML, belong to the same class) or Cannot Link (CL, belong to different classes). We propose an approach that uses the labelled data to train a classifier and then applies the ML and CL constraints in subsequent labelling. In our approach, a set of instances are labelled at the same time. Our case study is on animal re-identification. The dataset consists of five free-camera video clips of of animals (koi fish, pigeons and pigs), annotated with bounding boxes and animal identities. The proposed approach combines the representations or classifiers predictions from the bounding boxes of consecutive frames. We demonstrate that our approach outperforms standard classifiers, constrained clustering, as well as inductive and transductive semi-supervised learning, using five feature representations.

Keywords: animal re-identification, computer vision, classification, semi-supervised learning

1. Introduction

The aim of semi-supervised learning is to train classifiers using both labelled data and (typically abundant) unlabelled data. Semi-supervised learning has two principal branches [1]:

Email addresses: l.i.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)

transductive learning and inductive learning. The main difference between the two is that inductive learning returns a classifier, while transductive learning aims at labelling the unlabelled data in an optimal way. Transductive learning may or may not produce a classifier model in the process.

As a rule, the unlabelled data are assumed to be independent, identically distributed (iid). Known relationship between instances are not considered in mainstream semi-supervised classification or transductive learning, except for graph data [2]. Here we assume that, in addition to the feature representation of the unlabelled data, we have also two sets of constraints: Must Link (ML) constraints in the form of pairs of instances that belong to the same class, and Cannot Link (CL) constraints in the form of pairs of instances that should not be labelled in the same class. Pairwise constraints are the main gist of constrained clustering [3, 4, 5, 6, 7, 8]. However, state-of-the-art classifiers trained on the labelled data are hardly used within constrained clustering, leading to mediocre classification accuracy.

The main contribution of this study is a method which falls in the area of semi-supervised/ transductive learning, that includes pairwise constraints on the unlabelled data. Unlike semi-supervised learning, we do not use unlabelled data to improve on the initial classifier model. Also, unlike most transductive learning methods, our method does not apply an iterative algorithm to relabel the data. Once the labelled data has been used to train a classifier, they are no longer needed for classifying the unlabelled data. Our method is based on classification of *sets* of instances simultaneously, using data-generated constraints. We offer an experiment that demonstrates the benefit of using constraints.

Figure 1 shows a flow diagram of the task at hand. In conventional classification, we use a training dataset consisting of instances with corresponding labels to train a classifier. Once the classifier has been trained, it can be used to predict labels for testing instances. The training instances are not needed for using this classifier. Therefore, the classifier can be used to classify new instances without requiring the original training data.

Conventional classification can be seen as the initial step in the task that we are considering. The difference is that, in this case, constraints can be automatically obtained when given a set of instances to classify. These constraints are then used to modify the predictions

given by the classifier, taking them into account.[1]

As a case study, we chose identification of individual animals https://doi.org/10.5281/zenodo.7322820 [9]. Examples of annotated frames from the five videos are shown in Figure 2. The choice of subject and data for our case study was prompted by two aspects. First, the ML and CL constraints can be obtained automatically from the video material, and do not require human annotation. Second, animal identification from image collections and videos has received little attention in the literature compared to human [10, 11] or vehicle [12] identification. Global concerns about a looming ecological catastrophe require multidisciplinary effort in monitoring and managing of animal populations and ecosystems [13, 14, 15, 16, 17, 18]. We view our work on animal identification as a step in this direction.

Figure 3 illustrates how constraints can be obtained from frames. For each frame, several bounding boxes have been previously identified.[2]. Instances for the classification task are the images in these bounding boxes. Different instances from the same frame cannot be the same individual, so there are CL constraints between them. Given two bounding boxes from consecutive frames, if they are in similar positions and with similar sizes, they are likely to be from the same individual. Therefore, a ML constraint between them can be established. Constraints obtained in this way may be incorrect, but if the proportion of errors is manageable this constraints could improve the predictions.

The rest of the paper is organised as follows. Related work is summarised in Section 2. Section 3 explains the proposed methodology. Experimental results are shown in Section 4, and a conclusion is offered in Section 5.

2. Related work

Re-identification from videos is a task that involves recognizing elements (e.g., persons, vehicles, animals) from a video. As such, re-identification is a different task from classifi-

[1]Actually, not only the predictions but also the probabilities assigned by the classifier to each class are used. This detail is not shown in Figure 1 for the sake of simplicity.

[2]This work assumes that these bounding boxes and the corresponding instances are already available. For the considered case study, some approaches are evaluated in [19].

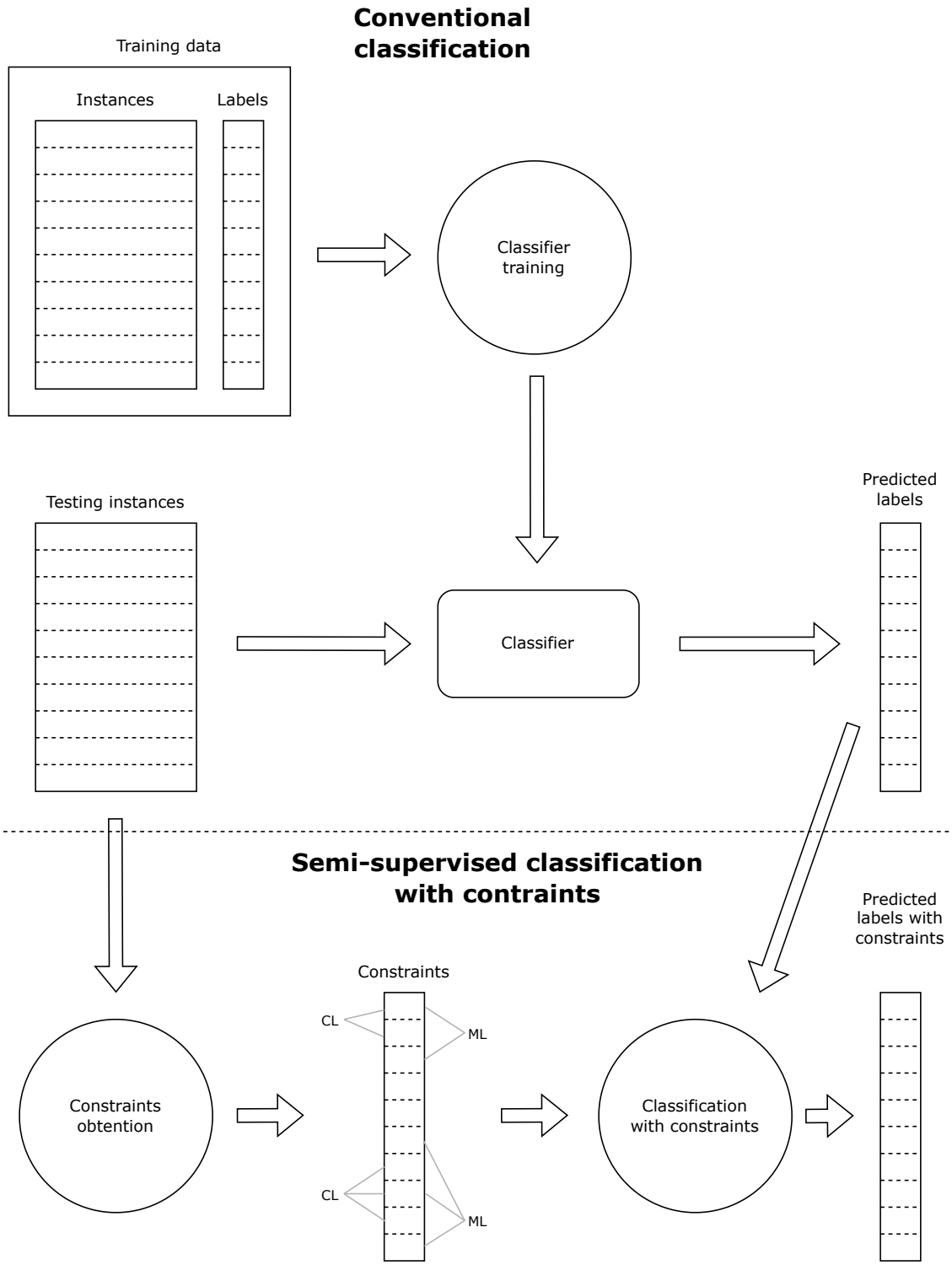

Figure 1: Flow diagram of the considered classification task.

(a) Koi (b) Pigeons (square) (c) Pigeons (pavement) (d) Pigeons (curb) (e) Pigs

Figure 2: Examples of annotated frames from the animal re-identification database used as our case-study.

cation. Nevertheless, they are related as classification can be, and is commonly, used for re-identification [10, 20].

Although classification can be a part of the re-identification task, there are also other parts that are not classification, such as object detection and tracking. For the videos in the considered case study, they have been studied in [19]. These are not considered in the current paper, as it focuses in the classification part.

Re-identification problem falls in the realm of *weakly labelled* data [21], and specifically the compound decision problem [22] and the Restricted Set Classification problem (RSC) [23, 24].

The real-life scenario where such a problem occurs is labelling objects in video frames or collections of time-lapse images. As several instances (identities) are present in each frame, they must belong to different classes. Also, we can combine instances which are related through ML constraints by examining the proximity of the bounding boxes and the instance appearance.

Standard transductive learning [1] does not include any information about dependencies of the unlabelled data, hence we are using it in this study as a baseline. Label propagation belongs to the transductive branch of semi-supervised learning [25]. A graph is built where the nodes are the instances and the edges are obtained from the distances between the instances. Subsequently, the labels of the labelled nodes are propagated to the unlabelled nodes. Co-training belongs to the inductive branch of semi-supervised learning [26, 27]. In co-training, two learning algorithms are trained separately on different "views" of the labelled data. Typically, different views come from different feature sets. Then each algorithm's predictions on the unlabeled examples are used to enlarge the training set of the other

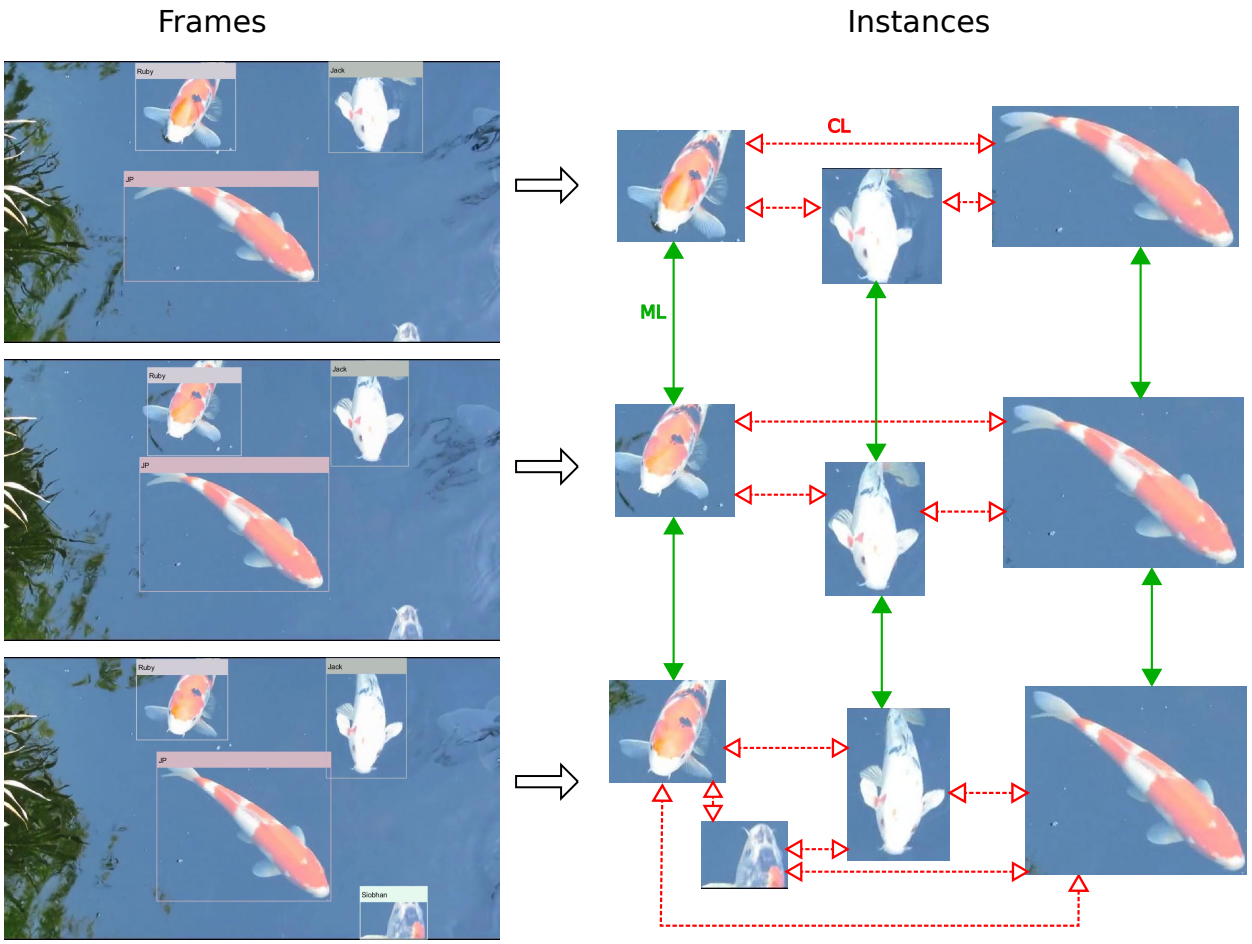

Figure 3: Example of constraints obtained from consecutive frames.

algorithm. This process is repeated iteratively.

Another approach that has been used for person re-identification is metric learning [28, 29]. The labeled instances are used to learn a metric and this metric is used to classify or cluster the unlabeled instances.

Combining label constraints with pairwise constraints (ML and CL) has been discussed in the literature under different guises. Collective classification [30, 31] offers a solution to classification of nodes of a network, where a limited number of nodes have class labels, and the network structure serves as ML constraints. The basic collective classification algorithms label the nodes iteratively, in a label propagation fashion. Provisional labels are obtained and re-used for the unlabelled part until convergence. There is no mechanism for enforcing CL constraints, though. Propagating labels over a graph has been studied extensively [25, 32]. In principle, our type of constraints can be converted to a graph structure with two types of edges: ML and CL (positive and negative). While graph structures have been explored for constrained clustering [33, 34], we did not come across studies combining label and pairwise constraints.

Including pairwise constraints is considered by Wang et al. [35], who propose a method for generating such constraints from the propagated labels and using constrained spectral clustering thereafter. Zhang and Yan [36] propose a binary classification method where the pairwise constraints are used to identify an optimal classification boundary. They assume that the number of constraints is significantly larger than the number of instances to be labelled. The labels are used at a later stage to associate the two sides of the boundary with the class label. Interesting as it is, this method would not be suitable for data with many classes. Nguyen and Caruana [37] propose a method termed PCSVM (pairwise constraints SVM) which incorporates ML and CL in the multi-class SVM criterion function. Basu et al. [38], on the other hand, use the labelled data to seed the clusters. Assuming that there is at least one representative from each class, the initial means for the k-means constrained clustering algorithm are calculated as the mean of each class. Two variants are proposed: one where the constrained clustering algorithm is allowed to re-label the originally labelled instances, and another, where the originally labelled instances are kept intact. In all cases,

using both types of constraints leads to improved accuracy of the clustering. We take these two methods in our experimental study.

Similar but not identical problems are considered in multi-instance classification [39, 40, 41], set classification [42], and relaxation labelling [43]. The context of our problem also relates it to tracking of multiple objects in video [44]. Some tracking algorithms include simultaneous classification of a set of instances. An example is tracking of individual moving parts [45, 46, 47] or people [45, 48]. The classification is dominated by ML constraints derived using the spatial location of the object/part. Appearance (extracted features) is deemed much less important in video tracking [47]. Indeed, some objects are indistinguishable, and the only way to identify them is using their predicted and observed locations. For example, the idTracker models [49, 50] as well as several related studies [51, 52, 53, 54], report experiments on identification of simultaneously moving, practically indistinguishable animals such as ants, mice, fruit fly, zebra fish, honeybees and crabs. None of these animals presents clear biometric markers. The videos are taken in a non-cluttered lab environment and the individual recognition is mostly based on the trajectories. Thus far, using constraints in video tracking is done in an implicit way, in a niche and problem-specific form. In this study, we propose a generic framework which will allow for a wider use.

Aside from the problem set-up, there are several aspects that distinguish our study from the related works where both labels and constraints are used:

- Previous studies make an assumption (explicitly or implicitly) that the labelled data are insufficient to train a classifier of reasonable accuracy. Conversely, we will assume that the data is of adequate size for this task. Linear models, the nearest neighbour, and heavily pruned decision trees can be used for small-size data.

- Most other studies demonstrate their methods on real or synthetic datasets where the constraints are sampled as i.i.d. or created from the candidate labels assigned to the unlabelled data. Our prime example are the animal videos, where the constraints come from the data. The constraints are not i.i.d, in that the CL constraints come from single frames and ML constraints come from time and location proximity of the

bounding boxes. We adopt the data-generated constraints because they come at no extra cost, and reflect the real-life scenario.

- Previous studies use the label constraints at the initialisation stage [38], as a part of the criterion guiding the classifier training algorithm [37] or as a part of the label propagation iterative procedure. We propose a novel approach to using the constraints. First, a classifier is trained on the labelled data, and then the unlabelled data are classified, one set at a time, where data-generated constraints are applied.

3. Methodology

Consider a labelled dataset $X_L = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with class labels $l_i \in \Omega$, $i = 1, \ldots, N$, where Ω is a set of class labels. Assume that there is an unlabelled dataset in the same feature space, coming from the same distribution, $X_U = \{\mathbf{x}_1^U, \mathbf{x}_2^U, \ldots, \mathbf{x}_M^U\}$. The goal is to label all instances in X_U as accurately as possible. Without further information, this is a transductive learning task. However, we assume that there is more information in X_U, and this can improve the accuracy of the labelling. Instead of classifying one instance $\mathbf{x}_j^U$ at a time, we consider a set $S^U \subseteq X^U$ at a time. Within S^U, we know of certain dependencies among the testing instances in the form of "these instances are all from different classes" or "there are at most k" instances from class i in S^U.

3.1. Restricted Set Classification (RSC)

Definition 1. *The* restricted set classification problem *is defined as follows. Let* $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ *be a set of instances such that at most* k_i *instances come from class* $\omega_i \in \Omega = \{\omega_1, \ldots, \omega_c\}$. *Find labels for all elements of* X *so that the restriction holds.*

Note that $k_1 + \ldots + k_c = k \geq m$.

Definition 2. *A* base classifier D *is a classifier that assigns a class label to an instance* $\mathbf{x} \in \mathbb{R}^n$. *We also require that* D *provides estimates of the posterior probabilities* $P(\omega_1|\mathbf{x}), \ldots, P(\omega_c|\mathbf{x})$.

In our case, the instances that come from a frame constitute X. They must have different class labels, hence $k_i = 1$.

The RSC uses the posterior probabilities obtained from D to infer class labels for each element of X. The standard RSC uses the Hungarian assignment algorithm [55] to find the labels.[3]. The input to the algorithm is the matrix $-LP = -\{l_{ij}\}$, $i = 1, \ldots, m$ (instances in X), $j = 1, \ldots, c$ (classes in Ω) where l_{ij}, are the *logarithms* of the posterior probabilities obtained from D. As the Hungarian algorithm minimises cost, while we seek *maximum* sum of logarithms of posterior probabilities, the input needs to be negated. Optimality of this label assignment has been proven in previous studies [23, 24]. RSC outperforms D because it incorporates extra information in the form of CL constraints. We propose to label the video frame by frame in order to enforce the CL constraints most effectively.

3.2. Why use one frame at a time?

The RSC method will not work effectively if applied to the whole of X_U. It will lose its main strength coming from the CL constraints if we apply it even just over two frames. To illustrate this, consider the straightforward application of RSC to a set of two consecutive frames. Suppose that there are two objects in each frame, a_1 and b_1 in frame 1, and a_2 and b_2 in frame 2. Assume that objects a come from class A and objects b, from class B. Assume also that no pair of instances can be matched to generate an ML constraint. The CL constraints enforced by RSC, applied separately on the two frames, will result in the following possible label set for the four objects: $ABAB, ABBA, BAAB, BABA$. The true labelling is among the four options. If we pool the two frames and try to label the four objects together through RSC, we will have the restriction that there are at most two objects from each class. This will introduce two more label possibilities, $AABB, BBAA$, with non-zero probabilities, which will violate the CL constraints of both frames.

To illustrate this point numerically, consider the following simulation experiment. We will call a 2-class classifier *reasonable* if its area under the ROC curve (AUC) is strictly

[3]Further developed by Kuhn and Munkres, also known as Kuhn-Munkres algorithm. Proposed originally for $c \times c$ matrices, the Hungarian algorithm has been extended for rectangular matrices [56].

greater than 0.5. Assume that we have trained a reasonable classifier D. Let $P_A(x)$ be the posterior probability that D assigns to the hypothesis that the object x belongs to class A. Then, with probability $p > 0.5$, $P_A(a_1) > P_A(b_1)$ and $P_A(a_2) > P_A(b_2)$. We set the AUC to be $p = \{0.51, 0.60, 0.80, 0.95\}$ and generated 50,000 scenarios of posterior probabilities for four points a_1, b_1, a_2, b_2. Let x be the accuracy of D. To sample a pair of posterior probabilities, we first generated a random number $P_A(a)$. With probability x, $P_A(a) > 0.5$, and with probability $1 - x$, $P_A(a) < 0.5$. For instance b, with probability x, $P_A(b) < 0.5$, and with probability $1 - x$, $P_A(b) > 0.5$. If both probabilities were simultaneously greater than 0.5 (b is mislabelled) or less than 0.5 (a is mislabelled), with probability p we put the two probabilities in correct order, that is $P_A(a) > P_A(b)$, to ensure that the AUC of D is p.

Subsequently, we applied the RSC while pooling the frames (RSC_pooled), and the RSC, to each frame individually (RSC_ind). The classification accuracy for the two methods and $x = 0.6$ (accuracy of D) is shown in Table 1. The results demonstrate the advantage of applying the RSC frame by frame instead of pooling multiple frames. We observe that, while inferior to RSC_ind, RSC_pooled still improves on D.

Table 1: Classification accuracy [in %] of the simulation experiment for accuracy of D fixed at $x = 0.6$ and four values of p.

p (AUC)	RSC_pooled	RSC_ind
51.00	64.60	72.36
60.00	65.68	74.28
80.00	67.91	79.01
95.00	69.75	82.85

We can verify the accuracy of RSC_ind by a simple equation. As the frames are considered separately, this accuracy is the same as the accuracy of a single frame. If a and b are labelled correctly, accuracy is 1. The probability of this event is x^2. The accuracy will also be equal to 1 if b is mislabelled $P_A(b) > 0.5$ but the two probabilities are in the correct order, $P_A(a) > P_A(b)$. The probability for this event is $x(1 - x)p$. the same calculation is valid if a is mislabelled but b is correct. Finally, we have accuracy of 0.5 (one correct, the other

wrong) if both $P_A(a)$ and $P_A(b)$ are greater than 0.5 or both are less than 0.5, but the AUC condition does not work. Therefore, the contribution to the accuracy calculation is $0.5 \times 2 \times x(1-x)(1-p)$. Finally, the RSC_ind accuracy is,

$$Acc_{\text{frame}} = x^2 + 2\,x\,(1-x)\,p + x\,(1-x)\,(1-p)\,.$$

Calculating the accuracy of RSC_pooled is more involved for this example because of the need to apply the Hungarian algorithm to a 4-by-4 matrix.

3.3. Semi-supervised classification using constraints

3.3.1. Formulating of the ML constraints

Up to now, we only considered CL constraints in this study. They come from the fact that two instances in the same frame cannot have the same identity. However, due to time contingency, video data allow for easily obtainable, highly plausible, ML constraints. In our case study, we construct ML constraints based on proximity of bounding boxes. We identify *tracks* by calculating the Intersection-Over-Union (IoU) between all pairs of bounding boxes in two consecutive frames. A matrix of IoUs is constructed with rows corresponding to the bounding boxes in the first frame and columns for the second frame. The Hungarian algorithm is applied to connect bounding boxes between the two frames, amounting to ML constraints. If the IoU exceeds a given threshold (0.5 in our case), and ML constraint is generated. The ML constraints in consecutive frames generate a track.

3.3.2. The proposed method

Assuming that the input to our task is a video footage, we need to apply the following *preprocessing* steps.

Semi-supervised Classification with Constraints – PREPROCESSING.

1.) Identify instances as the bounding boxes (BB) surrounding each identity of interest in each frame. This can be done by hand, by an object detector, or as a byproduct of a tracker software.

2.) Annotate the first N BB with class labels.

3.) Extract features from all BB in the video to form the two sets using a chosen feature extraction method, thereby forming the two sets: set $X_L = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with class labels $l_i \in \Omega$ and $X_U = \{\mathbf{x}_1^U, \mathbf{x}_2^U, \ldots, \mathbf{x}_M^U\}$ with unknown labels.

4.) Using IoU metric between all pairs of BB in consecutive frames, identify the tracks in the video. Thus, for each $\mathbf{x}_j^U \in X_U$, we have a set of indices $s(\mathbf{x}_j^U) \subseteq \{1, \ldots, M\}$ of the ML instances associated with $\mathbf{x}_j^U$. Note that $s(x_j^U)$ may also be a set of one element. Denote the collection of the index sets as $S = \{s(\mathbf{x}_1^U), \ldots, s(\mathbf{x}_M^U)\}$.

5.) Train the chosen model of classifier D on X_L. (Note that no constraints are used on the labelled training data.)

6.) Prepare a frame association set $F = \{f_1, \ldots, f_M\}$, where f_j is the frame number that instance x_j comes from. The frame numbering is immaterial here; it can be a consecutive number starting from the beginning of the video, from the beginning of the unlabelled data, or can be any other number collection where each frame has a unique number. This number is needed only to select all instances *in the same frame* for set classification.

From here onward, we propose two variants of our semi-supervised algorithm depending on how we use the ML constraints, as illustrated in Figure 4 and detailed in Algorithms 1 and 2.

4. Experimental study

4.1. Description of the experiment

The purpose of the experiment is to demonstrate that the proposed methods for semi-supervised classification with constraints are better than: (a) standard classification, (b) semi-supervised classification without constraints, (c) transductive leaning, and (d) classification through constrained clustering. To accomplish this, we carried out a case study on animal re-identification using a dataset consisting of five short videos. [57]

Algorithm 1 Semi-supervised Classification with Constraints. **A: Feature fusion.**

Input: Trained classifier D, unlabelled data X_U, index set collection S, and frame association set F.

Output: Labels for all instances in X_U.

for each frame k in the testing sequence **do**
  Use F to find all $\mathbf{x}^U$ in frame k and place them in set I_k.
  $D_k = \emptyset$ (a set to hold all outputs for set I_k)
  for each $\mathbf{x}^U \in I_k$ **do**
    if $|s(\mathbf{x}^U)| > 1$ **then**
      Feature Fusion: Replace $\mathbf{x}^U$ with the average of all $\mathbf{z} \in s_(\mathbf{x}^U)$.*
    end if
    Apply D to $\mathbf{x}^U$ and store in D_k.
  end for
  Apply RSC to D_K and store the assigned labels.
end for
return the labels of X_U

* The feature fusion is done once for each track and then retrieved for each frame of that track.

Algorithm 2 Semi-supervised Classification with Constraints. **B: Probability fusion.**

Input: Trained classifier D, unlabelled data X_U, index set collection S, and frame association set F.

Output: Labels for all instances in X_U.

for each frame k in the testing sequence **do**
  Use F to find all $\mathbf{x}^U$ in frame k and place them in set I_k.
  $D_k = \emptyset$ (a set to hold all outputs for set I_k)
  for each $\mathbf{x}^U \in I_k$ **do**
    Apply D to $\mathbf{x}^U$.
    if $|s(\mathbf{x}^U)| > 1$ **then**
      Probability Fusion: Replace the output of D for $\mathbf{x}^U$ with the average of the outputs of all $\mathbf{z} \in s(\mathbf{x}^U)$.*
      Add the revised output of D to D_k.
    end if
  end for
  Apply RSC to D_k and store the assigned labels.
end for
return the labels of X_U

* The probability fusion is done once for each track and then retrieved for each frame of that track.

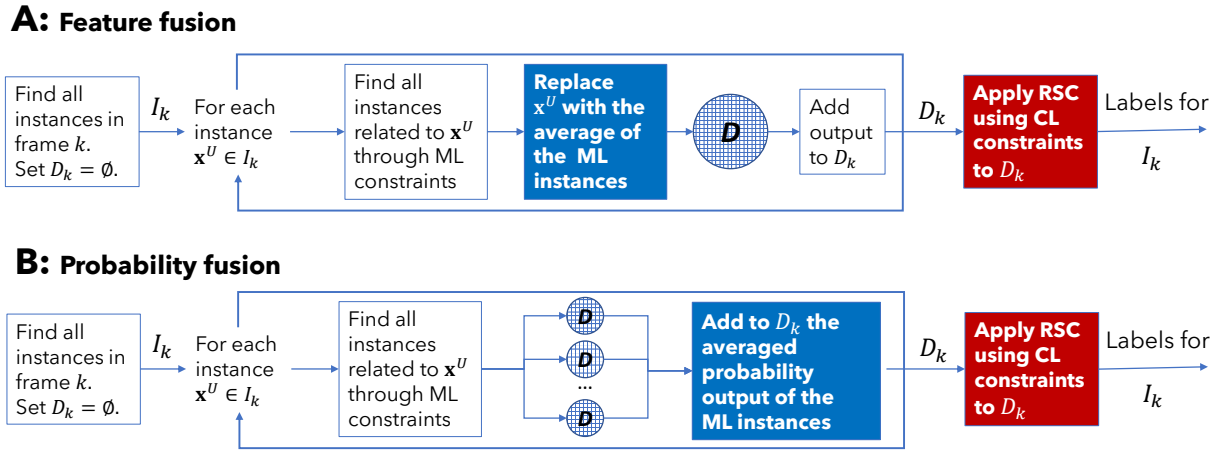

Figure 4: Diagrams of methods A and B for a single video frame. ML denotes Must Link constraints and CL denotes Cannot Link constraints. D is a classifier that outputs posterior probabilities.

We implemented the two proposed methods in Python and sourced the remaining classification models from *Scikit-Learn* [58] and the semi-supervised learning extension *Semi-Supervised Learning Library* [59] (`https://github.com/jlgarridol/sslearn`). The code for the experiment is available at `https://github.com/admirable-ubu/semi-supervised-animal-re-identification`.

In all experiments, we applied a two-fold cross-validation where the video frames were split into time-contingent halves. As we have the bounding box annotations already, for the purpose of this experiment, we are not relying on object detection or multi-object tracking. We are interested in showing how using constraints can improve classification accuracy.

4.2. Data

The database includes five videos and is available in full at `https://doi.org/10.5281/zenodo.7322820` [9]. Bounding-box and animal identity annotations are also available at `https://github.com/LucyKuncheva/Animal-Identification-from-Video`. The characteristics of the five videos are summarised in Table 2. We have a total of 2379 frames, 20,490 clips, and 93 identities. 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. As mentioned in the previous work [60], the features are extracted using various techniques. Three sets of features are based on color relationships (RGB), shape (HOG), and textures

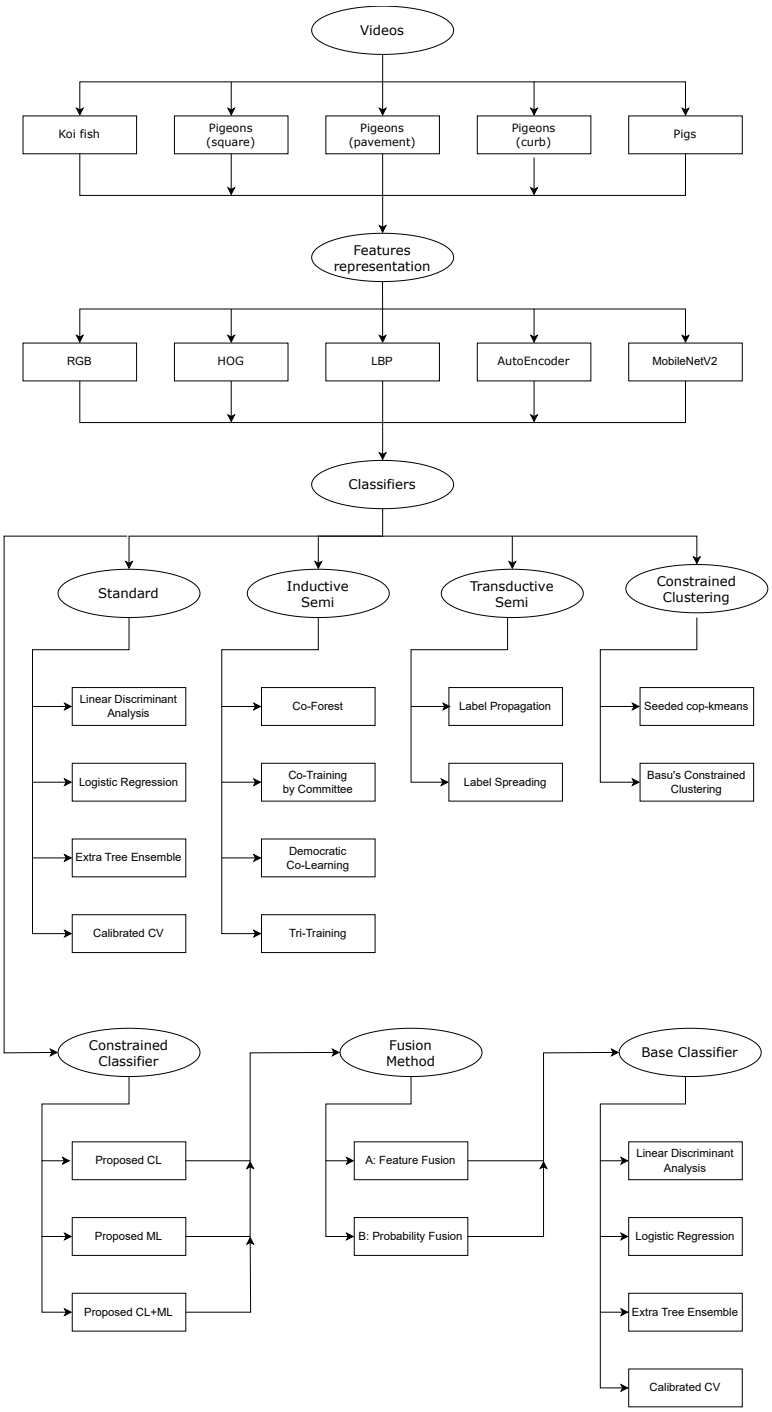

Figure 5: Experiments overview. There are five videos and for each one five features representations are used. For the resulting datasets, several classifiers are considered. These classifiers are grouped in five categories: standard, inductive, transductive, constrained clustering and the proposed constrained classifiers. For the latter, it can be used with only the CL constraints, only the ML constraints or both. Then, two fusion methods can be used: from features or from probabilities. The proposed method can be used with any supervised method, and the methods used are the ones in the standard group.

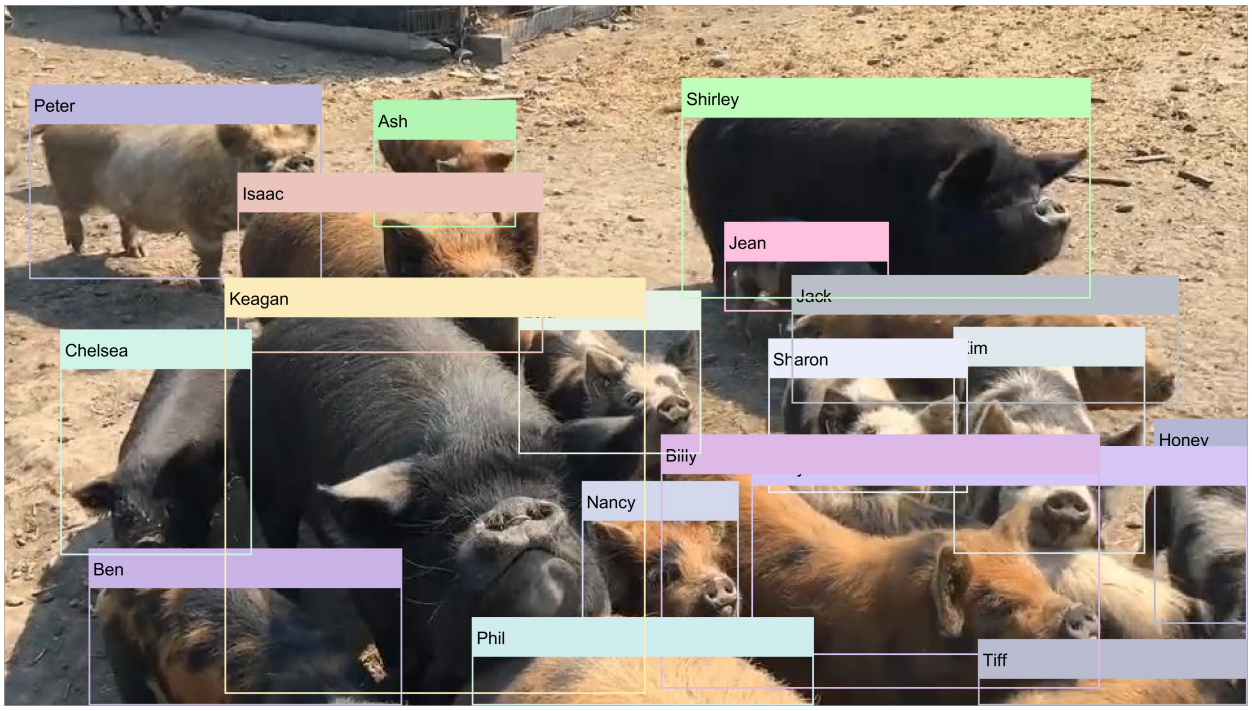

Figure 6: Annotated frame from the Pigs video.

(LBP). Two additional sets of features are obtained through deep learning, one using an Autoencoder with a size of 10 and the other using the pretrained MobilNetV2 model with the final activation layer.

Table 2: Characteristics of the videos: N is the number of objects (individual animal clips); 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.

Video	#Frames	Length [s]	N	c	Min p/f	Max p/f	Avr p/f	Imbalance
Koi fish	536	22	1635	9	1	6	3.1	2.8
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
Pigs	500	16	6184	26	4	20	12.4	10.5

Figure 6 shows an annotated frame from the Pigs video. The large overlap between the bounding boxes illustrated the difficulty of the dataset.

4.3. Feature representation

In a previous study, we carried out an extensive experimental comparison of state-of-the-art classifiers on the same data [60]. We experimented with five feature representations: colour data (RGB), shape-related features (Histogram of Oriented Gradients (HOG) [61]), texture-related features (Local Binary Patterns (LBP) [62]), Autoencoder [63] features and MobileNetV2 [64] features. We used MATLAB function `trainAutoencoder` with default parameters for the Autoencoder features. For the MobileNetV2 features, we used the Keras MobileNetV2 model pre-trained on Imagenet. The last layer was cut off, and replaced with a GlobalAveragePooling layer. The RGB moments calculated from each bounding box in the following way. The image with the animal was divided into 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. MATLAB and Python functions for feature extraction are provided in the GitHub repository `https://github.com/admirable-ubu/animal-recognition/` (We note that the previous experiment did not include constraints or the RSC classifier.)

We treated each video and feature representation as an individual dataset. Thus, we have $5 \times 5 = 25$ *datasets* in total. The classifiers will be compared by their ranks on the 25 datasets.

4.4. Competing classifiers

4.4.1. Standard classifiers

To choose classifier D, in a previous study [60], we ran a large experiment with five groups of classifiers: baseline, linear, non-linear, ensembles, and deep learning. The classifiers that performed best, were: **1. (ST-LDA)** Linear Discriminant Analysis, **2. (ST-LR)** Logistic Regression, **3. (ST-ETE)** Extra Tree Ensemble [65], and **4. (ST-CV)** Calibrated CV [58] (which with default options is a linear SVM). In this case study, we use the four selected classifier models as our individual classifier D. We compare the proposed methods to their "vanilla" alternatives as D. The Python code was sourced from scikit-learn [58].

The objective of the experiment is to see if the constraints obtained from the data to be classified improve the predictions of the classifiers. The proposed approach can be used

with any classification method, with any parameters adjustement.

4.4.2. Inductive semi-supervised learning

Based on a survey by Triguero et al [66], we chose the following models from the semi-supervised group:

5. (SSI-CoF) Co-Forest [67] is a co-training method where the training is done iteratively. At each iteration, each base classifier C is re-trained using an augmented training set. A sub-ensemble is constructed from all base classifier except C. The instances included in the augmented training set for C are those elements of X_U, whose sub-ensemble prediction is most confident. The method was called Co-Forest because it was originally proposed for random trees, but it can be used with any other classifier. Preliminary experiments with several base classifier models singled out 5-NN as the best base classifier, hence we will use Co-Forest with 5NN as a representative of the Co-Forest method.

6. (SSI-CoTC) Co-Training By Committee [68] is another iterative ensemble method from the semi-supervised group. At each iteration, an ensemble is trained, and an augmented training set is constructed for the next iteration. At each iteration, a random sample from X_U is labelled by the ensemble. The instances with the highest confidence of the prediction are added to the training set, while maintaining the class proportions. The main difference from Co-Forest is that the classifiers in the ensemble are not adapted individually.

7. (SSI-DeCo) Democratic Co-Learning [69] combines three different classifiers, i.e., Naïve Bayes, a k-nn and a decision tree. Each classifier has its own enlarged labelled set. The new instances to be added to the training set are chosen from the unlabelled set if the majority vote and the weighted vote of the classifiers are equal. The difference between SSI-DeCo and SSI-CoF is in the criteria for adding new instances in the extended dataset, and in SSI-DeCo each classifier has their own enlarged labelled set.

8. (SSI-Tri) Tri-training [70] is a combination of three classifiers trained, with the same method, initially on bootstrap samples from X, and updated iteratively. The training set for a classifier is updated by including instances from X_U for which the other two classifiers agree in the predictions and meet certain conditions. In contrast to the previously mentioned

methods, the training sets are not augmented but updated at each iteration starting with the respective training sample from X_L. This means that instances added at a previous iteration may no longer be present in successive iterations.

As with Co-Forest, our preliminary experiments with several base classifier models favoured 5-NN as the best base classifier, hence we will use Tri-training with 5NN as a representative of the Tri-training method.

4.4.3. Transductive semi-supervised learning

The transductive graph-based semi-supervised models, implemented in *scikit-learn* [58] and used in this study are:

9. (SST-LP) Label Propagation [25] creates a graph with the instances according to distance criteria, and propagates the label according to the community structure of the network [71]. The community structure is the natural division of the graph into different groups of nodes with highly dense connection between them.

10. (SST-LS) Label Spreading [72] is a modification of Label Propagation whose initially labelled nodes can change labels to make the network more consistent.

4.4.4. Constrained clustering

Label-type constraints have been used in constrained clustering mostly for initialising (seeding) [38]. However, in our problem, we have two different types of constraints. On the one hand, we have labelled data (label-type constraints) and, on the other hand, ML and CL (pairwise constraints).

To combine the two types, we use Basu's algorithm to incorporate the label constraints, and the cop-kmeans [73] to implement the pairwise constraints. To enforce the label constraints, the clusters are seeded with the class centroids of the labelled data. Then X_L and X_U are clustered together using cop-kmeans while not allowing the labels of X_U to change in the process (we call this method Basu2). The classification accuracy is calculated only on the testing data. Viewed this way, we can liken our approach to transductive learning. The two 'classifiers' from the constrained clustering group are detailed below.

11. (CC-SCK) Seeded cop-kmeans works by using X_L for calculating the cluster centroids. Cop-kmeans [73] is applied thereafter on X_U. Cop-kmeans extends the standard k-means by including constraint validation at each iteration.

12. (CC-BA) Here we apply Basu's method by combining the two types of constraints. First, X_L is used to calculate the initial means. Then cop-kmeans is applied to $X_L \cup X_U$. After each cop-kmeans iteration, the labels of X_L are returned to the true labels.

The total number of pairwise constraints for each testing dataset is far too large for the cop-kmeans algorithm. Therefore, we experimented with random samples of 20, 100, 200, 600, and 1400 constraints, half of each type (e.g., 10 ML and 10 CL constraints). With each number of constraints, we carried out 5 runs with randomly chosen constraints, and then averaged the results. We observed that increasing the number of constraints did not affect the accuracy by much, therefore we report only results with 1400 constraints (700 of each type).

All experiments were carried out on a Windows10 HP Pavilion laptop with core i5 processor CPU @ 1.60GHz and GeForce GTX 1050 GPU. MATLAB code for the experiment is available at `https://github.com/admirable-ubu/semi-supervised-animal-re-identification`.

4.5. Details of the implementation of the proposed methods A and B

To allow for a fair analysis of the two versions of the proposed method (A and B), we run the methods in the following versions *for each of the standard classifiers 1-4 as D*:

- **13. (PRO-CL-LDA), 14. (PRO-CL-LR), 15. (PRO-CL-ETE), 16. (PRO-CL-CV)** This is the implementation of RSC, frame-by-frame, in its original form. Only CL constraints are included.

- **17. (PRO-MLA-LDA), 18. (PRO-MLA-LR), 19. (PRO-MLA-ETE), 20. (PRO-MLA-CV)** Only ML constraints are included. To apply this method, we replaced each instance with the feature average according to method A, but we did not apply the RSC on each frame.

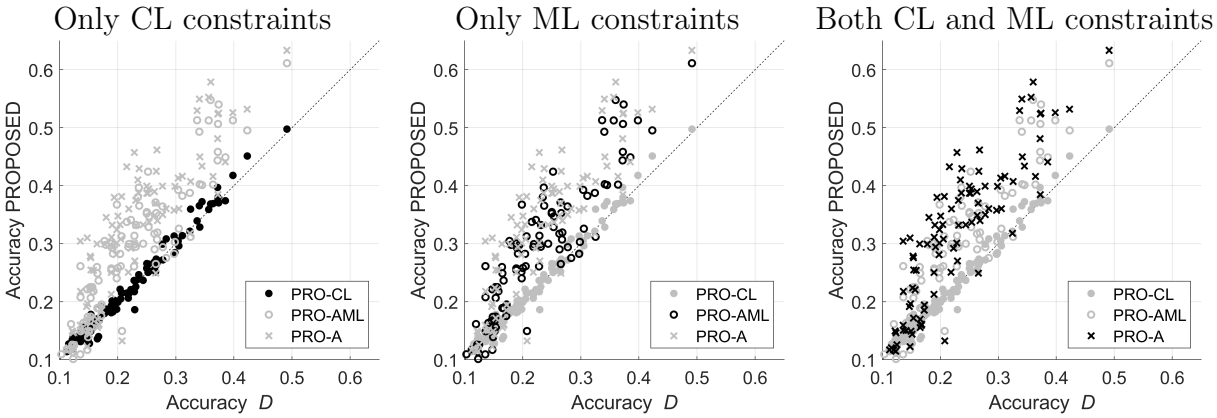

Figure 7: Accuracy of the proposed Method A (feature fusion) versus the independent classifier D.

- **21. (PRO-A-LDA), 22. (PRO-A-LR), 23. (PRO-A-ETE), 24. (PRO-A-CV)** This is the complete method A.
- **25. (PRO-MLB-LDA), 26. (PRO-MLB-LR), 27. (PRO-MLB-ETE), 28. (PRO-MLB-CV)** Only ML constraints are included. To apply this method, we replaced each instance with the probability average according to method B, but we did not apply the RSC on each frame.
- **29. (PRO-B-LDA), 30. (PRO-B-LR), 31. (PRO-B-ETE), 32. (PRO-B-CV)** This is the complete method B.

The methods marked with black labels are the ones we expect to perform best.

4.6. Results

A full set of results is available at [`https://github.com/admirable-ubu/semi-supervised-animal-re-identification/tree/main/results`] and given in a table form as supplementary material.

4.6.1. Comparison between the variants of the proposed methods

Figures 7 and 8 show scatterplots of the classification accuracy of the RSC models versus the independent classifier D for methods A and B, respectively.

Each point in the figures corresponds to an instance of the triplet (video, classifier, feature representation). Consider, for example, standard classifier SC-LDA as D. Suppose that it

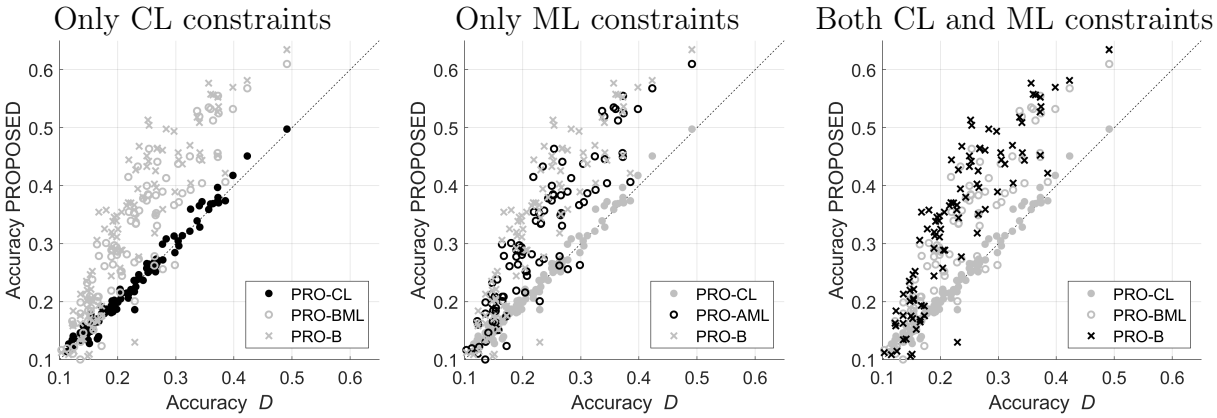

Figure 8: Accuracy of the proposed Method B (probability fusion) versus the independent classifier D.

has been applied to the RGB representation of the Pigs video. The accuracy of D defines the x-coordinate of a point. Then we apply methods PRO-CL-LDA, PRO-AML-LDA, and PRO-A-LDA, which generates three y-coordinates of points in the figure for method A, with the same x coordinates. If our proposed methods work well, the CL and the ML variants will be better than D, and will lie above the diagonal $y = x$. Then the point for the method which combines both constraints (PRO-A-LDA, in the example here), should lie higher than the other two, on the same x coordinates. As there are 5 (videos) $\times$ 5 (feature representations) $\times$ 4 (SC classifier models), there will be 100 values of x. With three points for every x, each scatterplot contains 300 points. The difference between the three subplots in each of Figures 7 and 8 is only in which of the three options is highlighted: only CL, only ML, or both.

It can be observed that both methods A and B improve on the accuracy of the individual classifier D. Interestingly, the CL constraints alone are the weaker of the two heuristics for both methods. Using only ML constraints results in markedly better accuracy. The reason for the weaker contribution of CL could be that they can only be used when the same identity is predicted for several individuals in the same frame. For the used videos this could happen rarely, the number of animals per frame can be well bellow the total number. The best results are obtained using the complete methods A and B which combine both type of constraints.

To evaluate which of the two proposed methods is better, we plot in Figure 9 the accuracies for the complete A and the complete B methods. The x axis are accuracies of the 100 instances of (video, classifier, feature representation) for method A (feature fusion), and the y-axis are the corresponding accuracies for method B (probability fusion).

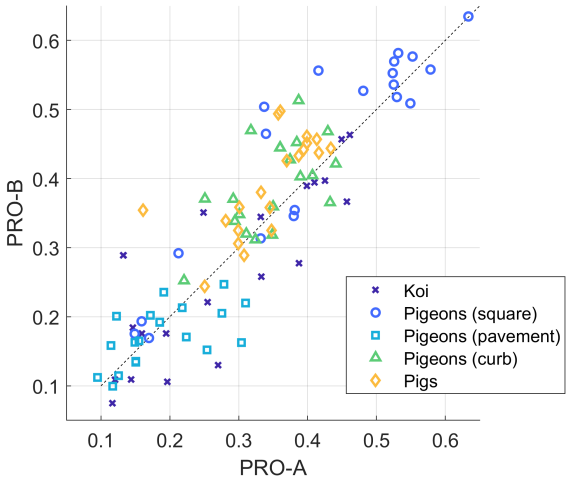

Figure 9: Comparison of methods A and B with both CL and ML constraints.

At first glance, the methods seem to be on a par, with a slight prevalence of B for higher accuracies. For a more detailed look, we plotted the points for the different videos with different colours and markers. It can be seen that Pigs video and Pigeons (curb) video benefit significantly from method B, while the two methods are similar for the other three videos.

4.6.2. Overall ranking

Next, we compare all 32 classifier models.

The overall best accuracies are shown in Table 3. As expected, the methods using both type of constraints took the top spots for all datasets. Method B won on four of the videos, and method A, on one (Pigeons (square)). Interestingly, different combinations of SC models and feature representations happen to achieve the best accuracy.

Table 4 shows the average ranks for the 32 classifiers. To calculate the rank, of a classifier for a given video and a given feature representation, we sorted all 32 accuracies in descending order. The top method was assigned rank 1, and the bottom method, rank 32. In case of a

Table 3: Overall best accuracies for the five videos out of the 32 classification methods.

Video	Classifier	Feature set	Accuracy
Koi fish	**32.** PRO-B-CV	RGB	0.46
Pigeons (square)	**29.** PRO-B-LDA	RGB	0.63
Pigeons (pavement)	**21.** PRO-A-LDA	HOG	0.31
Pigeons (curb)	**29.** PRO-B-LDA	MN2	0.51
Pigs	**31.** PRO-B-ETE	HOG	0.50

tie, the ranks are shared. Having calculated the 25 ranks for each method, we averaged them and sorted them in ascending order of the ranks (best method first). The rows of Table 4 follow this order. The columns of the table correspond to the feature representations. The entries show the average rank across videos. For example,

The table demonstrates that the proposed methods outperform the competitor methods from the related areas. The "complete" methods A and B, which use both CL and ML constraints are mostly at the top of the table, which indicates that the two type of constraints should be used together. None of the inductive and transductive methods, nor the constrained clustering show any reasonable accuracy. This is chiefly due to the fact that our proposed methods process the data frame-by-frame, i.e., set-by-set. We argued in Section 3.2 that processing larger sets of instances together (two consecutive frames) or processing the data instance-by-instance will not be as effective as processing one set at a time.

5. Conclusion

Pairwise constraints between instances arise naturally in some real-life problems. An example is object re-identification from video (animals, people, or inanimate objects), especially when there are multiple object in camera view simultaneously. To deal with this type of problems, we need a labelled data set for building an initial classifier, as well as a method to include CL and ML constraints in the classification process thereafter. We propose a solution where a set of objects are labelled together, applying CL and ML constraints for that set. In method A (feature fusion), the ML constraints are aggregated through averaging the features for all instances linked through an ML constraint chain, while in method

Table 4: Average ranks of the 32 classifiers.

Classifier	AE	HOG	LBP	MN2	RGB	Avg
21. PRO-A-LDA	12.25	3.50	4.50	8.50	8.75	**7.50**
24. PRO-A-CV	14.25	10.75	2.75	5.25	6.50	**7.90**
32. PRO-B-CV	18.75	8.25	3.75	7.75	4.50	**8.60**
31. PRO-B-ETE	3.50	7.25	10.75	14.00	8.25	**8.75**
29. PRO-B-LDA	18.50	7.50	9.00	6.75	8.00	**9.95**
30. PRO-B-LR	21.50	10.25	5.00	7.75	6.25	**10.15**
27. PRO-MLB-ETE	4.50	11.25	12.50	13.00	11.50	**10.55**
22. PRO-A-LR	18.00	11.00	3.50	9.75	11.00	**10.65**
18. PRO-MLA-LR	18.00	6.00	8.25	11.00	11.00	**10.85**
17. PRO-MLA-LDA	15.25	13.50	7.75	9.25	9.75	**11.10**
20. PRO-MLA-CV	10.50	17.88	7.12	13.75	10.00	**11.85**
28. PRO-MLB-CV	22.00	12.88	10.62	9.75	9.50	**12.95**
26. PRO-MLB-LR	20.00	15.00	8.25	10.25	14.50	**13.60**
25. PRO-MLB-LDA	20.50	14.00	12.50	15.50	5.75	**13.65**
23. PRO-A-ETE	4.50	11.00	18.25	25.00	15.25	**14.80**
19. PRO-MLA-ETE	5.50	15.50	17.25	23.00	17.00	**15.65**
15. PRO-CL-ETE	7.75	21.00	20.75	20.75	20.50	**18.15**
1. ST-LDA	19.75	22.75	20.75	14.50	15.00	**18.55**
3. ST-ETE	7.50	22.50	22.25	19.75	21.00	**18.60**
13. PRO-CL-LDA	22.25	22.00	19.50	14.50	15.00	**18.65**
16. PRO-CL-CV	22.25	20.75	19.00	17.00	15.75	**18.95**
4. ST-CV	21.25	22.00	18.75	19.75	17.00	**19.75**
2. ST-LR	23.50	20.00	19.25	18.50	22.25	**20.70**
14. PRO-CL-LR	26.00	21.25	19.50	17.75	20.00	**20.90**
8. SSI-Tri	13.88	18.75	25.75	26.62	25.25	**22.05**
5. SSI-CoF	14.62	21.00	25.75	26.38	27.00	**22.95**
10. SST-LS	20.75	12.75	29.88	23.00	31.50	**23.57**
6. SSI-CoTC	16.00	28.00	25.75	22.50	27.00	**23.85**
7. SSI-DeCo	14.75	23.25	27.75	27.25	26.25	**23.85**
9. SST-LP	18.75	17.00	29.62	25.00	31.50	**24.38**
11. CC-SCK	30.25	27.75	30.75	19.50	27.25	**27.10**
12. CC-BA	21.25	31.75	31.25	25.00	28.25	**27.50**

B (probability fusion), we average the output probabilities coming from the independent classifier D for those instances. The CL constraints are enforced through the Restricted Set Classification method (RSC). The animal re-identification experiment demonstrates the advantage of our approach over including constraints through constrained clustering and over not including constraints at all.

Video tracking through computer vision necessarily includes handling ML and CL constraints, but this is done in implicit and problem-specific manner. Here we propose a generic method which can be widely applied, not only to video tracking but to any problem which would benefit from using pairwise constraints.

There are several interesting directions for future work. For data sets with natural ordering of the sets of instances (e.g., frames in video), on-line variants can be constructed, inspired by the ideas of semi-supervised learning, especially by classifier ensemble learning.

Concept drift is a likely complication for this type of data. Hence, adaptive classification can be considered. Note that the constraints in the video example are generated by the data itself, and this will not change in the presence of concept drift, which is a particularly useful property.

Constrained clustering did not work well in our case study, mostly because we set aside a significant number of labelled data to train D. The labelled data is only used for initialisation in CC-SCK and CC-BA. However, constrained clustering may help build the labelled dataset with much less effort compared to manual annotation.

The proposed approach can be used with classifiers obtained with any method. In particular, these classifiers could also be obtained with semi-supervised learning methods. This combination could be worthwhile.

The study has a limitation in that it only considers videos where the animals to be identified are captured by a single camera positioned relatively fixed throughout the continuous recording. It is important to note that the method can potentially be applied to videos comprising multiple scenes, involving different camera positions, and even different cameras. However, it should be acknowledged that the performance of the method on this type of dynamic and varied video footage has not been studied thus far.

When multiple cameras are recording simultaneously, the proposed method can be applied to each video independently. However, this does not take advantage of the knowledge of camera positions. For example, if two cameras do not overlap, the same animal cannot be seen simultaneously on both cameras. However, if two cameras do overlap, the correspondence between animals in both cameras could be established.

Given the raising concern for animal welfare, the area of animal re-identification is expected to expand in the future. This calls for expanding the type of tasks and scenarios, and looking for solutions that are robust and widely applicable. We consider our study to be a step in this direction.

Acknowledgments

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 of Spain under the projects PID2020-119894GB-I00/AEI/10.13039/501100011033, 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] J. E. van Engelen, H. H. Hoos, A survey on semi-supervised learning, Machine Learning 109 (2) (2020) 373–440. doi:10.1007/s10994-019-05855-6.
URL https://doi.org/10.1007/s10994-019-05855-6

[2] M. Ding, J. Tang, J. Zhang, Semi-supervised learning on graphs with generative adversarial nets, CIKM '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 913–922. doi:10.1145/3269206.3271768.
URL https://doi.org/10.1145/3269206.3271768

[3] P. Gançarski, T.-B.-H. Dao, B. Crémilleux, G. Forestier, T. Lampert, Constrained clustering: Current and new trends, in: A Guided Tour of Artificial Intelligence Research: Volume II: AI Algorithms, Springer International Publishing, Cham, Switzerland, 2020. doi:10.1007/978-3-030-06167-8_14.

[4] D. Dinler, M. K. Tural, A survey of constrained clustering, in: Unsupervised Learning Algorithms, Springer, Cham, Switzerland, 2016. doi:10.1007/978-3-319-24211-8_9.
[5] S. Basu, I. Davidson, K. Wagstaff, Constrained Clustering: Advances in Algorithms, Theory, and Applications, CRC Press, Boca Raton, USA, 2008.
[6] I. Davidson, S. Basu, A survey of clustering with instance level constraints, ACM Transactions on Knowledge Discovery from Data (2007) 1–41.
[7] L. Kuncheva, F. Williams, S. Hennessey, A bibliographic view on constrained clustering, arXiv (2022). doi:10.48550/ARXIV.2209.11125.
[8] G. González-Almagro, D. Peralta, E. De Poorter, J.-R. Cano, S. García, Semi-supervised constrained clustering: An in-depth overview, ranked taxonomy and future research directions, arXiv preprint arXiv:2303.00522 (2023).
[9] 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
[10] 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.
[11] N. Huang, J. Liu, Y. Miao, Q. Zhang, J. Han, Deep learning for visible-infrared cross-modality person re-identification: A comprehensive review, Information Fusion 91 (2023) 396–411. doi:https://doi.org/10.1016/j.inffus.2022.10.024.
[12] A. Zheng, X. Zhu, Z. Ma, C. Li, J. Tang, J. Ma, Cross-directional consistency network with adaptive layer normalization for multi-spectral vehicle re-identification and a high-quality benchmark, Information Fusion 100 (2023) 101901. doi:https://doi.org/10.1016/j.inffus.2023.101901.
[13] S. Kumar, S. K. Singh, Visual animal biometrics: survey, IET Biometrics 6 (3) (2017) 139–156. doi:10.1049/iet-bmt.2016.0017.
[14] H. S. Kühl, T. Burghardt, Animal biometrics: quantifying and detecting phenotypic appearance, Trends in ecology & evolution 28 (7) (2013) 432–441.
[15] 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.
[16] E. Bohnett, J. Holmberg, S. P. Faryabi, L. An, B. Ahmad, W. Khan, S. Ostrowski, Comparison of two individual identification algorithms for snow leopards (panthera uncia) after automated detection, Ecological Informatics (2023) 102214doi:https://doi.org/10.1016/j.ecoinf.2023.102214.
[17] M. Zuerl, R. Dirauf, F. Koeferl, N. Steinlein, J. Sueskind, D. Zanca, I. Brehm, L. v. Fersen, B. Eskofier, PolarBearVidID: A video-based re-identification benchmark dataset for polar bears, Animals 13 (5) (2023). doi:10.3390/ani13050801.
URL https://www.mdpi.com/2076-2615/13/5/801
[18] Z. He, J. Qian, D. Yan, C. Wang, Y. Xin, Animal re-identification algorithm for posture diversity, in: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5. doi:10.1109/ICASSP49357.2023.10094783.
[19] F. Williams, L. I. Kuncheva, S. L. Hennessey, J. J. Rodríguez, Combination of object tracking and object detection for animal recognition, in: Proc. of the Fifth IEEE International Conference on Image Processing, Applications and Systems (IPAS 2022), 2022.
[20] S. Schneider, G. W. Taylor, S. C. Kremer, Similarity learning networks for animal individual re-identification: an ecological perspective, Mammalian Biology 102 (3) (2022) 899–914. doi:10.1007/s42991-021-00215-1.
URL https://doi.org/10.1007/s42991-021-00215-1
[21] J. Hernández-González, I. Inza, J. A. Lozano, Weak supervision and other non-standard classification problems: a taxonomy, Pattern Recognition Letters 69 (2016) 49–55.
[22] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, 2nd Edition, John Wiley & Sons, NY, 2001.
[23] L. I. Kuncheva, Full-class set classification using the Hungarian algorithm, International Journal of

Machine Learning and Cybernetics 1 (1) (2010) 53–61. doi:DOI10.1007/s13042-010-0002-z.
[24] L. I. Kuncheva, J. J. Rodríguez, A. S. Jackson, Restricted set classification: Who is there?, Pattern Recognition 63 (2017).
[25] X. Zhu, Z. Ghahramani, Learning from labeled and unlabeled data with label propagation, 2002.
[26] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the eleventh annual conference on Computational learning theory, 1998, pp. 92–100.
[27] X. Ning, X. Wang, S. Xu, W. Cai, L. Zhang, L. Yu, W. Li, A review of research on co-training, Concurrency and computation: practice and experience (2021) e6276.
[28] X. Xin, J. Wang, R. Xie, S. Zhou, W. Huang, N. Zheng, Semi-supervised person re-identification using multi-view clustering, Pattern Recognition 88 (2019) 285–297.
[29] X. Chang, Z. Ma, X. Wei, X. Hong, Y. Gong, Transductive semi-supervised metric learning for person re-identification, Pattern Recognition 108 (2020) 107569.
[30] L. K. McDowell, K. M. Gupta, D. W. Aha, Cautious inference in collective classification, in: In Processdings of AAAI, 2007, pp. 596–601.
[31] P. Sen, G. M. Namata, Bilgic, L. Getoor, B. Gallagher, T. Eliassi-Rad, Collective classification in network data, AI Magazine 29 (2008) 93–106.
[32] C. Gong, D. Tao, W. Liu, L. Liu, J. Yang, Label propagation via teaching-to-learn and learning-to-teach, IEEE transactions on neural networks and learning systems 28 (6) (2016) 1452–1465.
[33] J. Gao, X. Tao, S. Cai, Towards more efficient local search algorithms for constrained clustering, Information Sciences 621 (2023) 287–307. doi:https://doi.org/10.1016/j.ins.2022.11.107.
URL https://www.sciencedirect.com/science/article/pii/S0020025522014128
[34] B. Kulis, S. Basu, I. Dhillon, R. Mooney, Semi-supervised graph clustering: a kernel approach, Machine learning 74 (1) (2009) 1–22. doi:10.1007/s10994-008-5084-4.
[35] X. Wang, B. Qian, I. Davidson, Labels vs. pairwise constraints: A unified view of label propagation and constrained spectral clustering, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 1146–1151.
[36] J. Zhang, R. Yan, On the value of pairwise constraints in classification and consistency, in: Proceedings of the 24th international conference on Machine learning, 2007, pp. 1111–1118.
[37] N. Nguyen, R. Caruana, Improving classification with pairwise constraints: A margin-based approach, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2008, pp. 113–124.
[38] S. Basu, A. Banerjee, R. Mooney, Semi-supervised clustering by seeding, in: Proceedings of the 19th International Conference on Machine Learning (ICML), 2002, pp. 19–26. doi:10.5555/645531.656012.
[39] T. G. Dietterich, R. H. Lathrop, T. Lozano-Perez, Solving the multiple-instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1997) 31–71.
[40] J. Wang, J.-D. Zucker, Solving the multiple-instance problem: A lazy learning approach, in: In Proceedings 17th International Conference on Machine Learning, 2000, pp. 1119–1125.
[41] Z. Fu, A. Robles-Kelly, J. Zhou, MILIS: multiple instance learning with instance selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 958–977.
[42] X. Ning, G. Karypis, The set classification problem and solution methods, in: In Proceedings of SIAM Data Mining, 2009, pp. 847–858.
[43] M. Pelillo, M. Refice, Learning compatibility coefficients for relaxation labeling processes, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (9) (1994) 933–945.
[44] 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.
[45] I. Haritaoglu, D. Harwood, L. S. Davis, W/sup 4: real-time surveillance of people and their activities, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 809–830.
[46] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (4) (2001) 349–361.
[47] P. F. Felzenszwalb, D. P. Huttenlocher, Pictorial structures for object recognition, International journal of computer vision 61 (1) (2005) 55–79.

[48] T. Zhao, R. Nevatia, Tracking multiple humans in complex situations, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1208–1221.
[49] 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.
[50] 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.
[51] Z. XU, X. E. Cheng, Zebrafish tracking using convolutional neural networks, Scientific Reports 7 (1) (2017). doi:10.1038/srep42815.
[52] 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.
[53] 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.
[54] K. Zhang, Y. Xin, Z. Xie, C. Shi, A swimming crab portunus trituberculatus re-identification method based on RNN encoding of striped key regions, Engineering Applications of Artificial Intelligence 120 (2023) 105900. doi:https://doi.org/10.1016/j.engappai.2023.105900.
[55] H. W. Kuhn, The Hungarian Method for the assignment problem, Naval Research Logistic Quarterly 2 (1955) 83–97.
[56] F. Bourgeois, J.-C. Lassalle, An extension of the Munkres algorithm for the assignment problem to rectangular matrices, Communications ACM 14 (12) (1971) 802–804.
[57] 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.
[58] 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.
[59] J. L. Garrido-Labrador, C. García-Osorio, J. J. Rodríguez, J. Maudes, jlgarridol/sslearn: Zenodo indexed (Jan. 2023). doi:10.5281/zenodo.7565222.
URL https://doi.org/10.5281/zenodo.7565222
[60] L. I. Kuncheva, J. L. Garrido-Labrador, I. Ramos-Pérez, S. L. Hennessey, J. J. Rodríguez, An experiment on animal re-identification from video, Ecological Informatics 74 (2023) 101994. doi:10.1016/j.ecoinf.2023.101994.
[61] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Vol. 1, Ieee, 2005, pp. 886–893.
[62] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on pattern analysis and machine intelligence 24 (7) (2002) 971–987.
[63] M. F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural networks 6 (4) (1993) 525–533.
[64] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
[65] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Machine learning 63 (2006) 3–42.
[66] I. Triguero, S. García, F. Herrera, Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study, Knowledge and Information Systems 42 (2) (2013) 245–284. doi:10.1

007/s10115-013-0706-y.

[67] M. Li, Z.-H. Zhou, Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 37 (6) (2007) 1088–1098. doi:10.1109/TSMCA.2007.904745.

[68] M. F. A. Hady, F. Schwenker, Co-training by committee: A new semi-supervised learning framework, in: 2008 IEEE International Conference on Data Mining Workshops, 2008, pp. 563–572, ISSN: 2375-9259. doi:10.1109/ICDMW.2008.27.

[69] Y. Zhou, S. Goldman, Democratic co-learning, in: 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, pp. 594–602. doi:10.1109/ICTAI.2004.48.

[70] Z.-H. Zhou, M. Li, Tri-training: exploiting unlabeled data using three classifiers, IEEE Transactions on Knowledge and Data Engineering 17 (11) (2005) 1529–1541. doi:10.1109/TKDE.2005.186.

[71] M. E. Newman, Detecting community structure in networks, The European physical journal B 38 (2) (2004) 321–330.

[72] D. Zhou, O. Bousquet, T. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, Advances in neural information processing systems 16 (2003).

[73] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, Constrained K-means clustering with background knowledge, in: Proceedings of the Eighteenth International Conference on Machine Learning (ICML), 2001, pp. 577–584. doi:10.5555/645530.655669.