

Article

Vision-Based Module for Herding with a Sheepdog Robot

Virginia Riego del Castillo ¹, Lidia Sánchez-González ^{1,*}, Adrián Campazas-Vega ¹
and Nicola Strisciuglio ²

¹ Departamento de Ingenierías Mecánica, Informática y Aeroespacial, Universidad de León, 24071 León, Spain; vriegc@unileon.es (V.R.d.C.); acamv@unileon.es (A.C.-V.)

² Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7522 NB Enschede, The Netherlands; n.strisciuglio@utwente.nl

* Correspondence: lidia.sanchez@unileon.es

Abstract: Livestock farming is assisted more and more by technological solutions, such as robots. One of the main problems for shepherds is the control and care of livestock in areas difficult to access where grazing animals are attacked by predators such as the Iberian wolf in the northwest of the Iberian Peninsula. In this paper, we propose a system to automatically generate benchmarks of animal images of different species from iNaturalist API, which is coupled with a vision-based module that allows us to automatically detect predators and distinguish them from other animals. We tested multiple existing object detection models to determine the best one in terms of efficiency and speed, as it is conceived for real-time environments. YOLOv5m achieves the best performance as it can process 64 FPS, achieving an mAP (with IoU of 50%) of 99.49% for a dataset where wolves (predator) or dogs (prey) have to be detected and distinguished. This result meets the requirements of pasture-based livestock farms.



Citation: Riego del Castillo, V.; Sánchez-González, L.; Campazas, A.; Strisciuglio, N. Vision-Based Module for Herding with a Sheepdog Robot. *Sensors* **2022**, *22*, 5321. <https://doi.org/10.3390/s22145321>

Academic Editors: Diego González-Aguilera, Jose A. Jiménez-Berni, Ittai Herrmann, Shangpeng Sun and Monica Herrero-Huerta

Received: 30 May 2022

Accepted: 12 July 2022

Published: 16 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: computer vision; threat identification; wolf recognition; herding; sheepdog robots; precision livestock farming

1. Introduction

Nowadays, people are more concerned about sustainability and biodiversity, and they demand eco-products and healthier food [1]. This trend also helps to enhance regions that have experienced a huge depopulation in recent years. For example, traditional jobs such as sheep herding have become again popular, initiating extensive livestock farming in rural areas as a form of entrepreneurship. Industrial livestock production systems are mainly indoors and have a higher profitability [2]. On the contrary, pasture-based livestock farming increases animal welfare as they can behave naturally and move freely, and it is also positive for biodiversity [3,4]. However, grazing and monitoring the welfare of cattle and sheep is an arduous and time-consuming task, as the animals are often scattered over large areas and require year-round attention [5]. For this reason, assisting the required tasks by deploying autonomous systems is of utmost timeliness, although it presents challenges.

Precision Livestock Farming (PLF) offers technological tools to assist farmers in livestock management [6]. Through the use of sensors and data-driven systems, the herdsmen are able to manage and control several stages of the production flow [7]. In this sense, the benefits of adding new technologies to improve herder productivity are being explored, such as in the case of pasture-based livestock farming. Several different approaches in the literature use certain sensors such as accelerometers, cameras or GPS collars among others to obtain data useful to understand the animal behaviour and monitor them in order to detect diseases, supervise feeding and weight gain control [8].

Pasture-based livestock production remains an important sector in the European Mediterranean basin. It contributes to preserving large agricultural areas of high nature value, which are often located in less industrialised regions with low productive capacity,

such as mountainous regions, e.g., in southern Europe [9]. These grazing systems present environmental advantages, facilitating biodiversity and encouraging cultural and landscape diversity [9]. Furthermore the shepherd's working conditions have personal benefits, such as working in a natural environment. On the downside, they may suffer from inclement weather.

However, pasture-based livestock farming has several trade-offs [9]. Exclusively pasture-feeding animals ensures they maintain nutrient cycles, but it implies landscape modification. Another issue to consider is the prevention of environmental risks, such as wolf attacks. They might indeed harm the herd, causing a reduction in profits. Therefore, the introduction of a mobile robot capable of monitoring the herd, performing grazing tasks and sending information to the farmers when they cannot cover the whole herd would help to improve current PLF solutions. It also helps biodiversity as it can locate wolf packs and send their position to the herder to avoid encountering them.

Robots have applications in many fields as they have a multitude of functionalities such as the ability to swim [10], navigation in dark underground mines [11], assistance in subterranean search-and-rescue [12], and others. In livestock production, many commercial solutions help farmers in their daily routines. One example is robotic milking farms, which allow experts to analyse herd behaviour during the summer, enabling them to control how temperature affects milk production as well as contributing to animal welfare [13]. Other tasks have also been automated to assist farmers, such as feeding robots (<https://www.lely.com/solutions/feeding/vector/>, accessed on 28 May 2022), forage pushers (<https://milkingparlour.co.uk/portfolio/joz-moov-robotic-silage-pusher/>, accessed on 28 May 2022), scraping robots (<https://www.lely.com/gb/centers/english/farm-business-improvement-scheme/>, accessed on 28 May 2022) or herd-monitoring robots (<https://www.gea.com/en/articles/data-for-better-fresh-cow-management/index.jsp>, accessed on 28 May 2022), which provide data to monitor animal welfare, control feeding and prevent disease. However, most of them are used indoors, as they require a specific infrastructure.

Although the use of a four-legged robot as a sheepdog has been reported in the news, it is still an unsolved problem [14]. The reason is basically that the challenges of deploying a robot into the wild (perception of the environment, herd control or communication problems, among others) still requires investigation to allow its use in real-world scenarios. In this work, we focus on the development of a perception system that can build automatically a dataset with the desired predators using the iNaturalist API and determine the presence of such potential threats in order to prevent damages and respecting biodiversity. Otherwise, the proposed system can be deployed on an autonomous robot operating as a sheepdog in an outdoor farm.

We present a vision-based system that provides herders with valuable information from on-site sheepdog robots in real time. This information helps to increase the profitability of the sheep farm by avoiding some threats and helping the shepherd in their daily tasks. The proposed system identifies threats such as the presence of wolves, which helps make decisions about which grazing areas to drive the flock to. Moreover, we have developed a method to automatically build datasets of images of certain potential predators of a region by using the iNaturalist API [15]. The proposed system can be thus adapted to any region, including species of animals specific of a certain area or part of the world.

This paper is organised as follows. In Section 2, the related works are discussed. Section 3 presents the proposed system. The dataset considered to conduct the experiments and all the experimental setup is given in Section 4. Section 5 shows the obtained results. Finally, Section 6 includes a comparison with existing techniques, and Section 7 gathers the achieved conclusions.

2. Related Works

PLF involves data-driven systems to control animals, supervising all the related aspects such as their welfare or health, improving the production process. As sensor technology has advanced enormously in recent years, many measures related to physiological, behavioural

and productivity aspects can be acquired [16]. Most of them are focused on monitoring and tracking animal behaviour [17,18] as it is related to livestock diseases and also allows for the analysis of extensive and hilly pastures [19].

Different sensors are employed to develop these tasks such as GPS, cameras, accelerometers or thermographic devices [20,21]. Generally, these sensors are classified in two groups: sensors worn by animals such as ear tags, collars, leg straps or implants [5], and sensors placed in the animal's surroundings as in the case of cameras. Sensors placed on animals have several drawbacks. For example, the use of GPS collars can harm animals or even become stuck in forest. This is solved with sensors placed in the environment, which have more advantages by allowing the tracking of many animals simultaneously instead of a single one.

Among wearable sensors, there are approaches that use accelerometers or gyroscopes. These sensors are attached to the ear or collar to classify grazing and ruminant behaviour in sheep, providing information about their health or even detecting the lack of pasture quality [22].

With non-wearable sensors, in [23], a system for tracking sheep that detects if they are standing or lying with infrared radiation cameras and computer vision techniques is proposed. Using video cameras and deep learning, wild animals can be successfully identified, counted and described [24,25] as well as other particular species such as Holstein Friesian cattle [26]. A quadcopter with a Mask R-CNN architecture has been used to detect and count cattle in both extensive production pastures and in feedlots with an accuracy of 94% [27,28]. A complete review of the use of images from satellites, manned aircraft and unmanned aerial systems to detect wild animals is given in [29].

Regarding herding, most of the existing approaches are based on monitoring animals from a distance, since it results in a performance increase of the livestock farm, improves animal welfare and reduces the environmental impact. This is called Virtual Fencing (VF), and it is used for handling free-ranging livestock, as it optimises the rangelands and acts like a virtual shepherd [30]. By combining GPS information from the collars and remote sensing, it is possible to monitor animal interactions with the landscape to predict animal behaviour and manage rangelands, protecting those regions that are sensitive or likely to suffer degradation due to overgrazing [31]. Other approaches use drones to herd and move animals, especially in areas dangerous for herders [5,32]. Regarding the impact of drones to animals, there are studies that conclude that terrestrial mammals are more tolerant to unmanned aircraft systems [33], becoming accustomed to them [34], but other species, such as penguins, react differently [35]. Moreover, while unmanned ground vehicles (UGVs) can operate in adverse weather conditions, drones cannot and have problems in forested areas [36].

As the proposed system is developed outdoors, wireless coverage is not guaranteed. In [37,38], a review of the existing wireless sensor networks that can be employed in Precision Agriculture is gathered. Among them, the Long-Range Radio (LoRa) Protocol is discussed as well as its use in different approaches such as smart irrigation systems or the monitoring of agricultural production [38,39].

The proposed method goes a step further, focusing not only on the herd but also on potential threats such as the presence of predators such as wolves, which is a major challenge as it involves both livestock safety and wolf conservation [40,41]. In other regions, predators are jaguars [42] or bears [43,44]. Thus, an adaptive method should be automatically configured to work with different species of predators. In Section 4.1, we show how to configure the method to detect other predators. In addition to this, contact between wildlife and livestock can also be studied as it can potentially transmit zoonotic diseases [45].

In order to detect the presence of wolves, computer vision techniques are employed. Traditionally, features such as Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) or Histogram of Oriented Gradients (HOG) were extracted from images and then were classified using Support Vector Machines, among other classifiers, in order

to detect objects in an image [46]. These HOG features have been employed, i.e., for human detection [47], animal detection to avoid animal–vehicle collisions [48], and also in real time [49].

Modern approaches use deep learning techniques such as Convolutional Neural Networks (CNN) that extract features from images, which were subsequently classified using several dense layers. There are also solutions that combine traditional features such as HoG with CNN [50,51]. An extensive review of the object detection techniques using deep learning is provided in [52]. Different CNN architectures are used for object detection and classification, which are trained for large datasets—a time-consuming task [53]. As it is not possible to handle certain problems if the available dataset is not so large or the hardware requirements do not allow obtaining results in a short period of time, a technique named transfer learning is used to take advantage of those existing models that have been trained to detect certain objects and adapt them to a different domain [54].

Through transfer learning, pre-trained models can be adapted to new domains in a way that takes advantage of the knowledge extracted from the set of millions of images on which the network was trained and adapts it to the new problem [54]. They also reduce training time as only the added layers, which introduce the information of the particular problem, are trained. The model layers are frozen or fine-tuned, so the existing knowledge is maintained but adapted to the new problem. Most of the popular solutions for object detection are: YOLO (You Only Look Once), Single Shot Detector (SSD) and R-CNN. The existing architectures with the pre-trained models have been applied to animal classification with an accuracy over 90%. Some examples are fish classification [55], bird classification [56], camera trap images classification [57] or wild animal image classification [39].

3. System Architecture

The imitation of sheepdog behaviours with a shepherd robot requires a decision-making system that is able to manage information collected by its sensors and to generate actions through its actuators, taking into account the changing characteristics of the environment. As this system will be deployed in a real environment, some conditions, such as lighting and obstacles that must be detected for a proper navigation, are not fixed, since they vary over time. Moreover, these decisions have to offer long and short-term opportunities to be reactive to any expected and unexpected behaviour in the scenario.

This research uses MERLIN [58], which combines deliberative and behavioural capacities. Deliberative capacities define the characteristics of planning to infer long-term tasks. Behavioural capabilities provide the set of actions capable of responding to changes in the environment more quickly. Thus, Merlin allows us to set a certain goal but is also able to react if an unexpected event occurs.

Once the robot is deployed in a real context, it is crucial to handle unexpected behaviours, and for this, the use of different sensors is required. Traditionally, the literature proposes the use of camera sensors, such as the one proposed in this research. This sensor provides information about the context, such as dangerous species, identification of certain animals or predator attacks. The information provided by the proposed method in the robot's closed environment will promote alternative navigation routes, keep herd control updated or trigger an alarm process to alert the herder remotely if a wolf is detected. The proposed vision module can also be deployed in fixed cameras near grazing areas.

The proposed vision module is included along with the other reactive components. It assumes a background approach, feeding a knowledge database and keeping the deliberative monitoring system that would interrupt the active task up to date. The architecture proposed in [58] requires to be updated by including the new vision-based module proposed in this paper in the reactive layer. As this reactive layer gathers all the sensors of the robot, with the proposed vision module, the images acquired by the robot camera are processed. The obtained information is sent to the rest of the layers of the architecture. The useful provided knowledge about what the robot has seen helps the rest of the architecture to make decisions about the subsequent actions to carry out (see Figure 1).

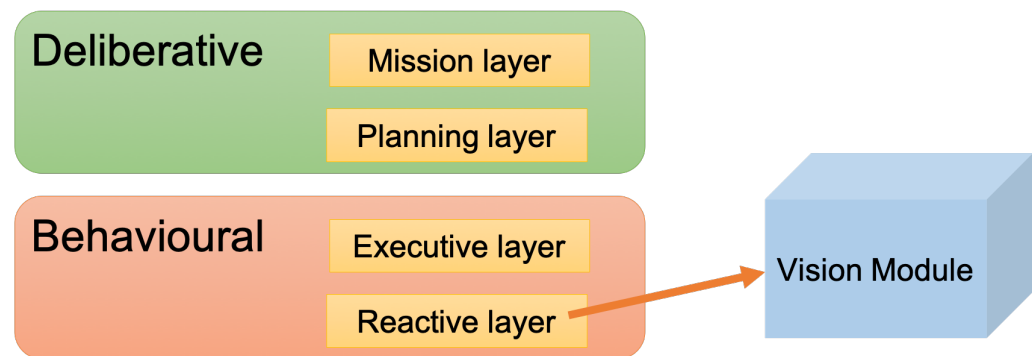


Figure 1. MERLIN architecture and proposed vision module.

In some scenarios, the autonomous features of the robot are not sufficient for the continuous monitoring of the herd, and there are some specific events where the herder needs to access the robot in real time to monitor dangerous external contexts or improve productivity. The point of view of the shepherd and of the sheepdog are not always the same, so the robot can respond to the commands of the shepherd, who has previous experience and knows how to deal with situations such as weather conditions or grazing specific areas. Thus, images from the robot can be sent to the shepherd's mobile phone on demand.

The acquired images are processed in the vision module in order to obtain information about the environment. This information is sent to the shepherd by using the LoRaWan® networking protocol. It is widely used in those areas where there are not wireless connections satisfying Internet of Things requirements. The LoRaWan® network architecture allows bi-directional communication, and messages are sent to a gateway that functions as a bridge to the Internet network [59]. Those gateways can be located at different points of the region, making it possible to cover a huge area (≈ 15 km).

4. Vision Module

The vision module belongs to the perception system, which plays an important role in the behaviour of a robot. The use of a Unitree A1, which is a robot dog, has been proposed to detect victim and pedestrians in emergency rescue situations by using thermal and color cameras [60]. These systems usually use a Robot Operating System (ROS) with a vision module to acquire images and detect the existing objects, i.e., smart glasses [61] or cameras on drones [62]. Cameras can also be used to help in robot navigation, exploration and mapping as in [63,64].

We propose a vision module that can be used in fixed cameras near pastures and villages or in cameras built into SheepDog robot systems. Figure 2 shows the complete pipeline of the process, where images are acquired and labelled in order to train the object detection models.

There are datasets that are very frequently used in object detection problems [65]. PASCAL Visual Object Classes (VOC) [66] includes 20 categories with animals such as birds, cats, cows, dogs, horses and sheep. ImageNet [67] provides an important source of data for object detection with 200 labelled categories in the 2017 version. Common Objects in Context (COCO) [68] includes the localisation, label and semantic information for each object in one of the 80 categories. In this work, the proposed system can automatically generate species-specific animal datasets through an API.

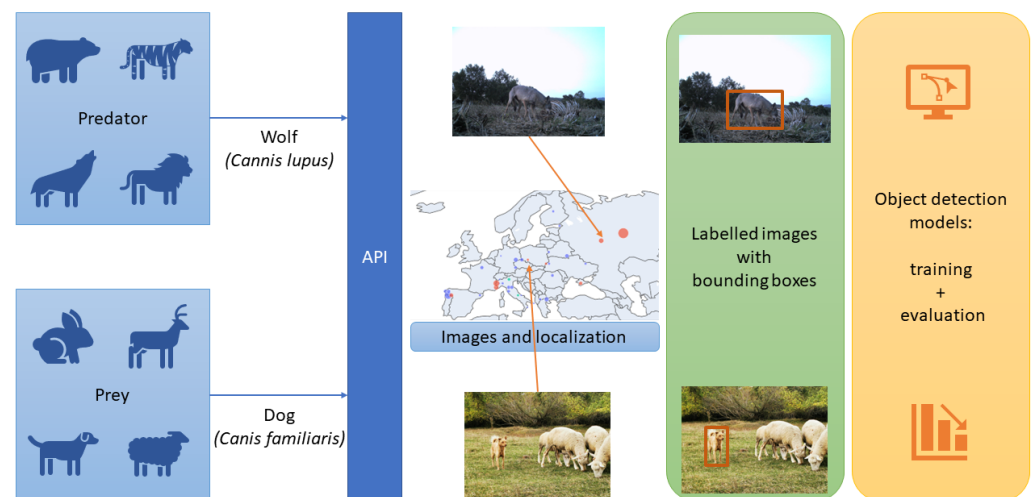


Figure 2. Pipeline of the vision module. First, images and their location are acquired using an API; then, images are labelled to train object detection models.

4.1. Data Acquisition And Labelling

Shepherds have to deal with predator attacks on the herd, so it is necessary to anticipate this situation by evading the threat and distinguishing if there is a potential risk [69]. We consider the presence of predatory animals as a potential risk, which can be a bear, tiger, lion or wolf. In the proposed system, prey animals are distinguished from predators to determine suitable grazing areas to maintain distances from predator locations.

We focus on a predatory species of the northwest of the Iberian Peninsula: the Iberian wolf. In [70], a study of the diet of the Iberian wolf shows that it tends to eat goats, cattle, sheep and rabbits, which are some of the animals that farmers raise in the area. Iberian wolves have phylogenetic proximity to other European wolf populations (*Canis lupus*), being considered as a sub-specie of it (*Canis lupus signatus*). Otherwise, dogs are the domesticated descendants of the wolf, presenting similarities as specie (*Canis familiaris*). In this paper, we have created a dataset to differentiate a predator (wolf) from a prey (dog) that can be implemented for more species diversity.

We have used the iNaturalist API [15] to create the dataset, obtaining images from two species: *Canis lupus* (wolf) and *Canis familiaris* (dog). As the code is available in [71], the vision module can be adapted to other predators of other regions by using the notebook `get_inaturalist` and choosing the desired species. Localisation has allowed us to divide the images into two groups: Europe and Outside Europe. Then, images were labelled manually by experts, removing images with low quality. Figure 3 shows how 925 images and 1137 detections are split by species and location.

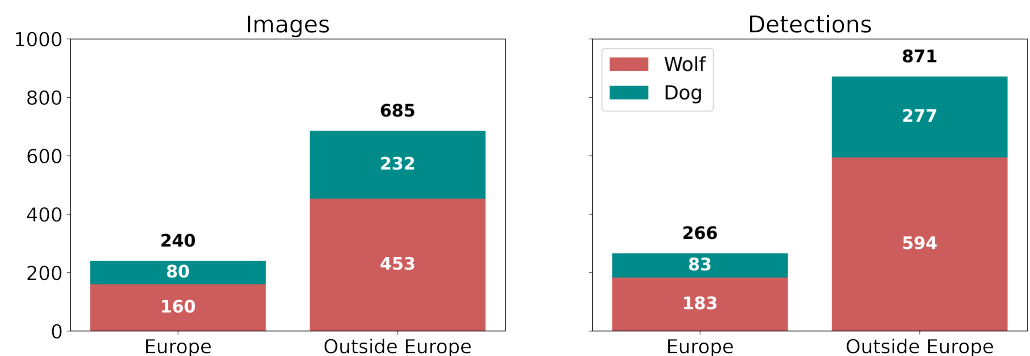


Figure 3. Information of the dataset disaggregated by species and location in number of images (left side) and number of detections (right side).

Images present diversity, as different animals can appear on the images (Figure 4a), lying (Figure 4b), looking to the camera (Figure 4c), partially occluded (Figure 4d), with different lighting conditions (Figure 4e), with multiple detections (Figure 4f) or just one (Figure 4g), feeding (Figure 4h) with different illuminations and different distances from the camera. Due to this information, in Europe, we can observe wolves in couples or alone, as can be shown in Figure 5.

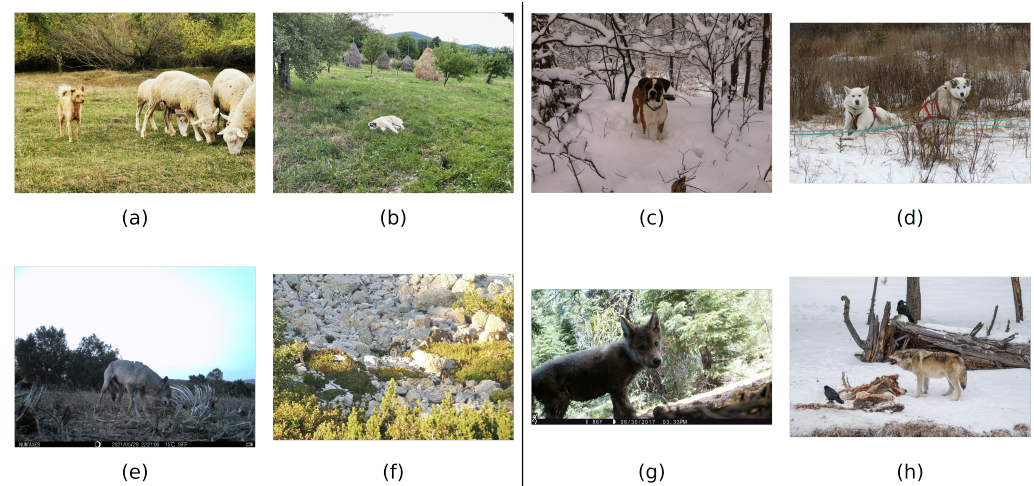


Figure 4. Samples of dogs (**upper row**) and wolves (**bottom row**) in Europe (**left**) and the rest of the world (**right**).

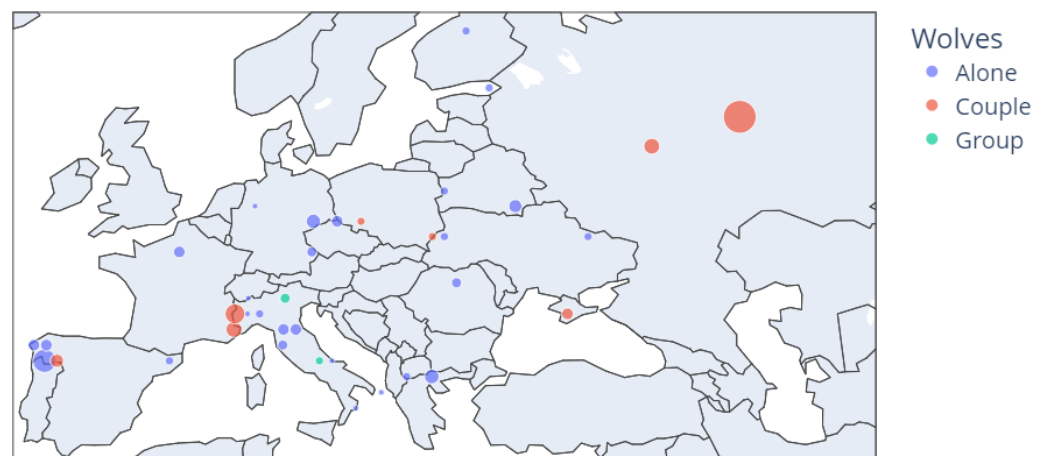


Figure 5. Iberian wolves detected in Europe. Bubble size depends on the number of images.

4.2. Object Detection Architectures

Object detectors are evaluated based on accuracy, speed and complexity. Two-stage detectors have two steps: extract features from the input image (feature extractor) and recognise the features (classifier). Meanwhile, one-stage detectors combine the feature extractor and classifier into one, reducing complexity and improving speed, but accuracy may be reduced. As the proposed module is deployed in real-time environments, it is based on one-stage detectors. The considered state-of-the-art algorithms [72] based on one stage are You Only Look Once (YOLO) in different versions (YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5) and Single-Shot MultiBox Detector (SSD). SSD has improved versions such as Deconvolutional SSD (DSSD) that includes large-scale context in object detection, Rainbow SSD (RSSD) that concatenates different feature maps using deconvolution and batch normalisation [73], and Feature-fusion SSD (FSSD) that balances semantic and positional information using bilinear interpolation to resize feature maps to the same size to

be subsequently concatenated [74]. The comparison of different architectures for real-time applications presented in [75] also mentions RetinaNet because it has higher accuracy, but it is not recommended for real-time applications, as it has a frame rate lower than 25 frames per second (FPS). EdgeEye [76] proposes an edge computing framework to analyse real-time video with a mean of 55 FPS as inference speed.

For evaluation, different metrics have been considered. First, Intersection over Union (*IoU*) measures the overlapping area between the predicted bounding box and the ground-truth divided by the union of the areas. *IoU* is fixed by a threshold (*t*) generating the confusion matrix as:

- True Positives (*TP*) are those objects detected by the model with an *IoU* greater than the considered threshold ($IoU \geq t$);
- False Positives (*FP*) are the detected objects whose *IoU* is less than the fixed threshold ($IoU < t$);
- False Negatives (*FN*) stand for those objects that are not detected;
- True Negatives (*TN*) are the number of objects detected by the model when actually the image does not have such objects.

Detector models use performance metrics computed from the confusion matrix mentioned above as follows:

- *Precision*: measures how many of the predicted outputs labelled as true predictions are correctly predicted:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

- *Recall*: measures how many of the real true predictions are correctly predicted:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

In order to compare the results of different authors, there are well-established metrics based on mean Average Precision (*mAP*), which is the average of the accuracy obtained in the object detection over all the dataset categories. Specifically, metrics are related to datasets mentioned previously [77]:

- COCO metric (mAP_{COCO} or $mAP@50 : 5 : 95$): evaluates 10 *IoUs* between 50% and 95% with steps of 5% of mean Average Precision as

$$mAP_{COCO} = \frac{mAP_{0.50} + mAP_{0.55} + mAP_{0.60} + \dots + mAP_{0.95}}{10} \quad (3)$$

- PASCAL VOC metric (mAP_{VOC} or mAP_{50}): evaluates *IoU* at 50%.

4.3. SSD

SSD is composed by two components: a backbone model (in this case, a pre-trained VGG16) and an SSD head with convolutional layers to obtain the bounding boxes and categories of the detected objects. From an image of 300 by 300 pixels, SSD achieves 72.1% mAP_{VOC} on a VOC2007 test at 58 FPS on a Nvidia Titan X [78]. The model has been trained from the PAZ (Perception for Autonomous Systems) library with the object detection module [79].

4.4. YOLO

YOLO is also based on a classification backbone with new headers to obtain the bounding box and the assigned class of the object. There are multiple implementations of the architecture, for example, YOLOv3 [80], which uses a DarkNet framework and makes predictions in three different scales. It achieves 57.9% mAP_{VOC} in 51 ms (around 20 FPS). An improvement was made with YOLOv4 [81] obtaining 65.7% of mAP_{VOC} and a speed of 65 FPS. After this, a tiny version of YOLOv3 [82] was proposed with higher speed but

lower precision. The newest version, which is YOLOv5 [83], can achieve 68.9% of mAP_{VOC} with more than 80 FPS. Moreover, it includes different versions of complexity, obtaining more precision and more inference time when the model is more complex. The YOLOv5 models are: Nano (YOLOv5n), Small (YOLOv5s), Medium (YOLOv5m), Large (YOLOv5l) and Extra-Large (YOLOv5x).

5. Results

The goal of the following experiment is to establish which of the existing architectures is better to detect predators as wolves in images taking into account performance and speed due to it being a real-time vision module.

On the one hand, an SSD model was trained with pre-trained weights from VGG and Stochastic Gradient Descent (SGD) optimizer (as in [78]) with a learning rate of 0.0001. Training was completed during 100 epochs with a batch size of 16. The model achieves 92.90% of mAP_{VOC} in the training set and 85.49% in the test set. Inference takes 80 ms on average, which corresponds with 12.5 FPS (in an NVIDIA GeForce RTX 2060).

On the other hand, multiple YOLO models were trained with an SGD optimizer configured with a learning rate of 0.01 (as in [80]) and a batch size of 4 during 50 epochs. Table 1 shows the obtained results of the different models. As it can be observed, YOLOv3 achieved the best results in mAP_{COCO} with 88.63%, while the tiny model is the fastest one with 64 FPS also with an NVIDIA GeForce RTX 2060. With the newest version of YOLO, the extra-large model YOLOv5x yielded the highest mAP_{COCO} with 88.24% but with the lower frame rate, whereas the nano, small and medium architectures are faster, achieving 64 FPS and keeping a mAP slightly lower. Small architectures (nano YOLOv5n, small YOLOv5s and medium YOLOv5m) are lighter in weight, and therefore, the execution time is lower. Heavier architectures (large YOLOv5l and extra-large YOLOv5x) take longer to run but have higher precision in the results.

Table 1. Results of the models over the test set, where mAP_{VOC} is the PASCAL VOC metric and mAP_{COCO} corresponds with the COCO metric. Inference time is measured in ms, and YOLOv3t is YOLOv3-tiny. Best results are highlighted in bold.

Model	Precision	Recall	mAP_{VOC}	mAP_{COCO}	Inference	FPS
SSD	85.49	93.33	85.49	47.87	80	12.5
YOLOv3	99.35	99.56	99.49	88.63	31.24	32.01
YOLOv3t	94.86	94.97	98.63	71.98	15.62	64.01
YOLOv5n	95.88	96.13	99.07	75.86	15.62	64.02
YOLOv5s	98.54	98.54	99.47	82.78	15.62	64.01
YOLOv5m	99.17	99.52	99.49	85.53	15.62	64.02
YOLOv5l	99.47	99.61	99.50	87.57	31.24	32.01
YOLOv5x	99.38	99.73	99.50	88.24	46.86	21.34

Figure 6 shows the scheme of the final module. Images are divided into training and validation using localisation: species from outside Europe are used for training and validation is performed with images of European species. First, the YOLO model is trained and then, metrics are calculated using the validation set (2nd step). Once the model is trained and evaluated as it is proposed in this paper, further steps involve integrating the model in an autonomous system such as a robot or in any device with a fixed camera, which acquires images (3rd step). The acquired images are processed by the model (4th step). Finally, identified threats such as the presence of wolves can be used to raise an alarm and inform the shepherd to avoid the area where they are located (5th step).

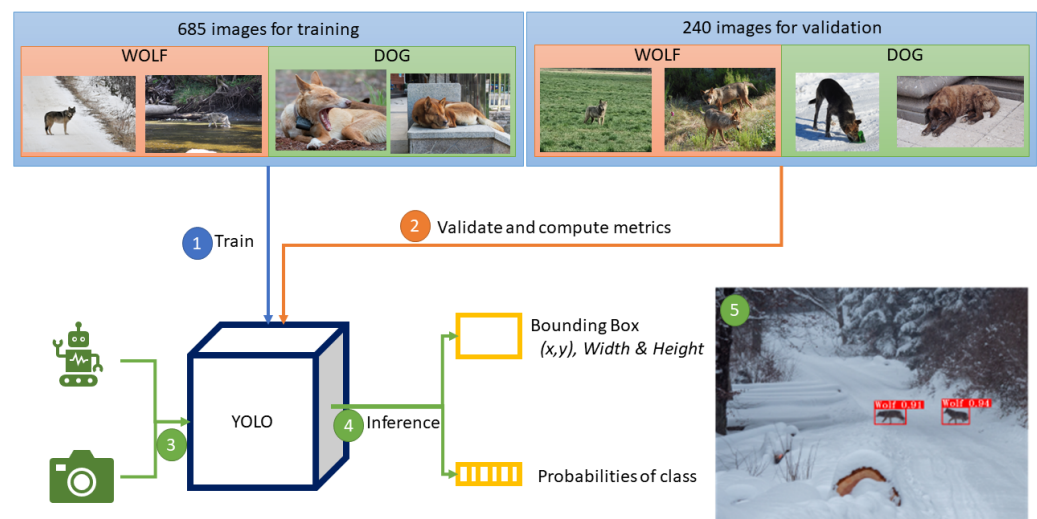


Figure 6. Complete scheme of the proposed vision module.

Further study of the results indicates that YOLO performs best in terms of accuracy and speed; that is, it is able to recognise wolves or wolf packs and distinguish them from dogs with high accuracy and in real time. Figure 7 shows a graph with the COCO metric and speed (FPS), where the best results are in the top right. We can determine that YOLOv5m (medium) is the best model in terms of speed and mAP , as it can process 64 FPS with 99.49% of mAP_{VOC} and 85.53% of mAP_{COCO} . Observing these results, YOLOv5m is chosen to form the vision module.

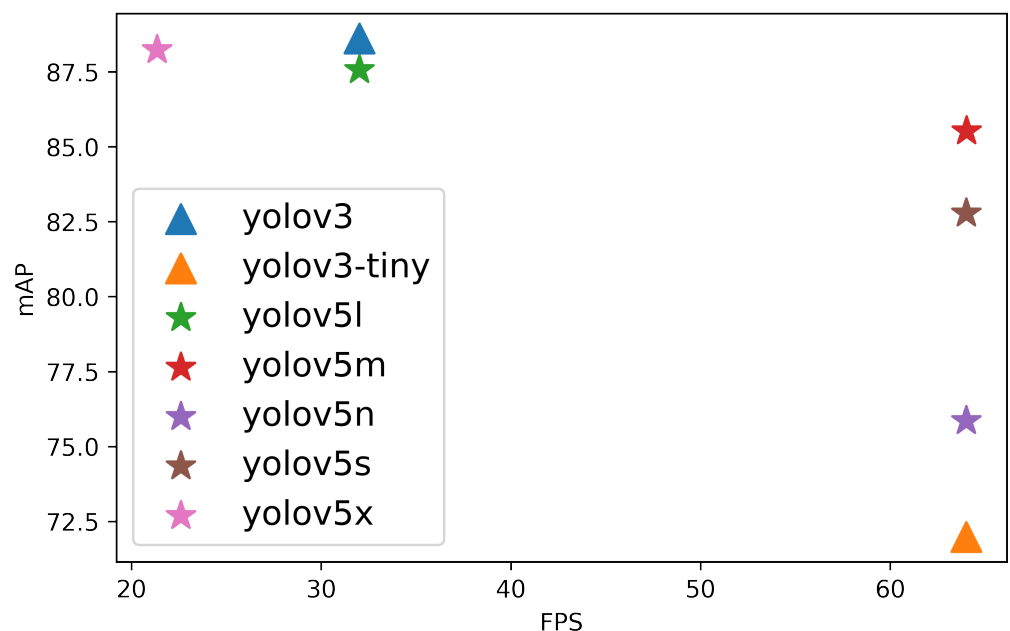


Figure 7. Results of the mAP_{COCO} over the speed for YOLO models.

YOLOv5 employs a loss function composed by three losses: the bounding box loss, which uses a regression loss for object location (Mean Squared Error, MSE), the object loss, which obtains the confidence of object presence (Binary Cross-Entropy) and classification loss, which determines that the classification is correct (Cross-Entropy). Figure 8 displays the evolution of the loss functions during the training that shows the good behaviour of

the model. Moreover, an evolution of the obtained metrics is shown in Figure 9, achieving a high performance after 30 epochs.

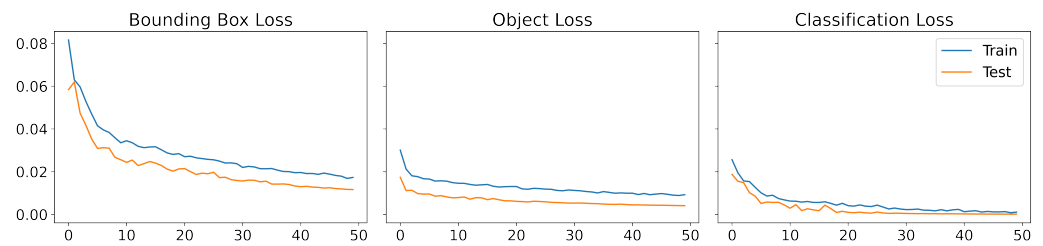


Figure 8. Loss functions of YOLOv5m during the training: bounding box loss with Mean Squared Error (**first graph**), object loss with Binary Cross-Entropy (**second graph**) and classification loss with Cross-Entropy (**third graph**).

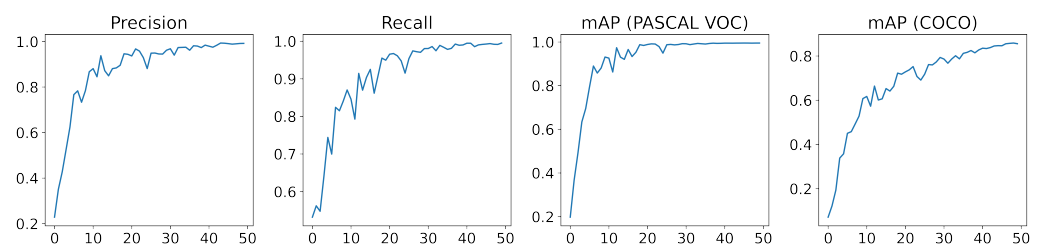


Figure 9. Metrics of YOLOv5m: precision (**first graph**), recall (**second graph**), mAP_{VOC} which corresponds with PASCAL VOC metric (**third graph**) and mAP_{COCO} which corresponds with COCO metric (**fourth graph**).

Finally, Figure 10 gathers some samples of the dataset with the objects detected by the YOLOv5m model. As it can be pointed out, there are images of the considered categories in which the objects to be detected, wolves and dogs, are at different distances to the camera and also deal with occlusions. Figure 11 shows the confusion matrix for training and validation data with an IoU greater than 0.5.



Figure 10. Samples of dogs (**upper row**) and wolves (**bottom row**) in Europe (**left**) and the rest of the world (**right**) with the detections yielded by YOLOv5m.

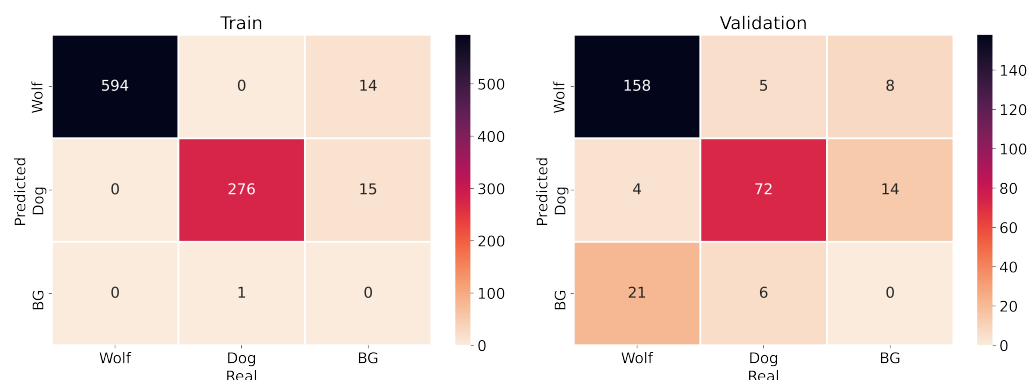


Figure 11. Confusion matrix for training (left) and validation (right) of YOLOv5m.

6. Discussion

There are approaches in the literature where researchers address similar problems of animal detection and classification. For instance, in [27], cattle are counted by analysing the images with a Mask R-CNN, obtaining 92% accuracy and AP for the detection of 91%, which is outperformed by the proposed method. In addition, using image processing and Mask R-CNN for counting animals, in [28], livestock, sheep and cattle are counted and classified, achieving a precision of 95.5% and mAP_{40} values of 95.2%, 95% and 95.4% for livestock, sheep and cattle, respectively. Using thermal images, animals such as dogs, cats, deer, rhinos, horses and elephants are detected and classified with an mAP_{VOC} of 75.98% with YOLOv3, 84.52% with YOLOv4 and 98.54% with Fast-RCNN [25]. In [24], the problem is turned into a classification as follows. First, a binary classification is performed to decide whether or not there are animals in the image, with an accuracy of 96.8%. If animals are detected, then a classification of the number of animals in the image is carried out, considering 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11–50, or +51 individuals and achieving 63.1% top-1 accuracy.

According to the previously discussed methods, the proposed vision module outperforms the state-of-the-art results not only in precision and accuracy but also in speed to be able to couple to real-time systems. Table 2 summarises the comparison between the state-of-the-art methods explained above and the proposed method.

Table 2. Comparison of the state-of-the-art results with the proposed method for animal detection and count of different species.

Problem	Results
Holstein Friesian cattle detection and count [24]	Animal detection: accuracy of 96.8% Counting animals: 63.1% top-1 accuracy
Cattle count [27]	Counting animals: accuracy of 92% Bounding box prediction (localisation): AP of 91%
Livestock, Sheep, Cattle detection [28]	Precision rate: 95.5%, 96% and 95% Recall with IoU of 0.4: 95.2%, 95% and 95.4%
Animal detection in thermal images [25]	mAP_{VOC} with YOLOv4: 75.98% mAP_{VOC} with YOLOv3: 84.52% mAP_{VOC} with Faster-RCNN: 98.54%
Wolf and Dog detection (Vision Module)	mAP_{VOC} with YOLOv3: 99.49% (FPS: 32) mAP_{VOC} with YOLOv5m: 99.49% (FPS: 64)

7. Conclusions

In this paper, a vision-based module to detect predators in pasture-based livestock farming and distinguish them from other species is proposed. This module can be deployed within on-site sheepdog robots and fixed cameras to assist shepherds in threat detection.

First, we propose a system that can automatically generate datasets of different species through the iNaturalist API in order to obtain a module that can be used in any region depending on the existing predators. Focusing on a predator specie of the northwest of the Iberian Peninsula, namely the Iberian wolf, a particular dataset is obtained. The generated benchmark has the aim of providing data and an evaluation framework to test different algorithms to detect wolves, as a predator specie, and differentiate from other animals such as dogs, which have a similar physical appearance. These data can be automatically extended with the new predator and prey species of the region.

Then, multiple object detection models have been trained to establish which one achieves better results in a real-time module. According to the obtained results, the best results are achieved with YOLOv5m yielding an inference time of 15.62 ms, which allows 64 FPS. This model achieves a precision of 99.17% and a recall of 99.52% on the considered benchmark, outperforming other existing approaches, with an mAP_{VOC} of 99.49% and mAP_{COCO} of 85.53%. These results fulfil the requirements of a real-time detection module and improve state-of-the-art methods.

Future development lines involve integration into autonomous systems and data collection in the field. Information about potential threats will enable early warning alerts to be managed for herders in difficult-to-access terrain.

Author Contributions: Conceptualization, N.S. and L.S.-G.; software, V.R.d.C.; validation, V.R.d.C., A.C.-V. and L.S.-G.; investigation, V.R.d.C. and L.S.-G.; resources, V.R.d.C. and A.C.-V.; data curation, V.R.d.C. and A.C.-V.; writing, V.R.d.C., L.S.-G., A.C.-V. and N.S.; supervision, N.S. and L.S.-G. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Universidad de León.

Data Availability Statement: All experiments and data are available in [71]. Images can be downloaded from iNaturalist API with the proposed system.

Acknowledgments: Virginia Riego would like to thank Universidad de León for its funding support for her doctoral studies. We also express our grateful to Centro de Supercomputación de Castilla y León (SCAYLE) for its infrastructure support.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
COCO	Common Objects in Context
FPS	Frame per second
IoU	Intersection over Union
LoRa	Long-Range Radio
mAP	mean Average Precision
PAZ	Perception for Autonomous Systems
PLF	Precision Livestock Farming
ROS	Robot Operating System
SGD	Stochastic Gradient Descent
SSD	Single-Shot MultiBox Detector
UGVs	Unmanned Ground Vehicles
VF	Virtual Fencing
VOC	Visual Object Classes
YOLO	You Only Look Once

References

- Delaby, L.; Finn, J.A.; Grange, G.; Horan, B. Pasture-Based Dairy Systems in Temperate Lowlands: Challenges and Opportunities for the Future. *Front. Sustain. Food Syst.* **2020**, *4*, 543587. <https://doi.org/10.3389/fsufs.2020.543587>.
- Campos, P.; Mesa, B.; Álvarez, A. Pasture-Based Livestock Economics under Joint Production of Commodities and Private Amenity Self-Consumption: Testing in Large Nonindustrial Privately Owned Dehesa Case Studies in Andalusia, Spain. *Agriculture* **2021**, *11*, 214. <https://doi.org/10.3390/agriculture11030214>.
- Lessire, F.; Moula, N.; Hornick, J.L.; Dufrasne, I. Systematic Review and Meta-Analysis: Identification of Factors Influencing Milking Frequency of Cows in Automatic Milking Systems Combined with Grazing. *Animals* **2020**, *10*, 913. <https://doi.org/10.3390/ani10050913>.
- Yu, H.; Li, Y.; Oshunsanya, S.O.; Are, K.S.; Geng, Y.; Saggar, S.; Liu, W. Re-introduction of light grazing reduces soil erosion and soil respiration in a converted grassland on the Loess Plateau, China. *Agric. Ecosyst. Environ.* **2019**, *280*, 43–52. <https://doi.org/10.1016/j.agee.2019.04.020>.
- Herlin, A.; Brunberg, E.; Hultgren, J.; Högberg, N.; Rydberg, A.; Skarin, A. Animal Welfare Implications of Digital Tools for Monitoring and Management of Cattle and Sheep on Pasture. *Animals* **2021**, *11*, 829. <https://doi.org/10.3390/ani11030829>.
- Schillings, J.; Bennett, R.; Rose, D.C. Exploring the Potential of Precision Livestock Farming Technologies to Help Address Farm Animal Welfare. *Front. Anim. Sci.* **2021**, *2*, 639678. <https://doi.org/10.3389/fanim.2021.639678>.
- Niloofar, P.; Francis, D.P.; Lazarova-Molnar, S.; Vulpe, A.; Vochin, M.C.; Suci, G.; Balanescu, M.; Anestis, V.; Bartzanas, T. Data-driven decision support in livestock farming for improved animal health, welfare and greenhouse gas emissions: Overview and challenges. *Comput. Electron. Agric.* **2021**, *190*, 106406. <https://doi.org/10.1016/j.compag.2021.106406>.
- Samperio, E.; Lidón, I.; Rebollar, R.; Castejón-Limas, M.; Álvarez-Aparicio, C. Lambs' live weight estimation using 3D images. *Animal* **2021**, *15*, 100212. <https://doi.org/10.1016/j.animal.2021.100212>.
- Bernués, A.; Ruiz, R.; Olaizola, A.; Villalba, D.; Casasús, I. Sustainability of pasture-based livestock farming systems in the European Mediterranean context: Synergies and trade-offs. *Livest. Sci.* **2011**, *139*, 44–57. <https://doi.org/10.1016/j.livsci.2011.03.018>.
- Chen, G.; Lu, Y.; Yang, X.; Hu, H. Reinforcement learning control for the swimming motions of a beaver-like, single-legged robot based on biological inspiration. *Robot. Auton. Syst.* **2022**, *154*, 104116. <https://doi.org/10.1016/j.robot.2022.104116>.
- Mansouri, S.S.; Kanellakis, C.; Kominiak, D.; Nikolakopoulos, G. Deploying MAVs for autonomous navigation in dark underground mine environments. *Robot. Auton. Syst.* **2020**, *126*, 103472.
- Lindqvist, B.; Karlsson, S.; Koval, A.; Tevetzidis, I.; Haluška, J.; Kanellakis, C.; Akbar Agha-mohammadi, A.; Nikolakopoulos, G. Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue. *Robot. Auton. Syst.* **2022**, *154*, 104134. <https://doi.org/10.1016/j.robot.2022.104134>.
- Osei-Amponsah, R.; Dunshea, F.R.; Leury, B.J.; Cheng, L.; Cullen, B.; Joy, A.; Abhijith, A.; Zhang, M.H.; Chauhan, S.S. Heat Stress Impacts on Lactating Cows Grazing Australian Summer Pastures on an Automatic Robotic Dairy. *Animals* **2020**, *10*, 869. <https://doi.org/10.3390/ani10050869>.
- Vincent, J. A Robot Sheepdog? 'No One Wants This,' Says One Shepherd. 2020. Available online: <https://www.theverge.com/2020/5/22/21267379/robot-dog-rocos-boston-dynamics-video-spot-shepherd-reaction> (accessed on 12 January 2022).
- Matheson, C.A. iNaturalist. *Ref. Rev.* **2014**, *28*, 36–38. <https://doi.org/10.1108/RR-07-2014-0203>.
- Tedeschi, L.O.; Greenwood, P.L.; Halachmi, I. Advancements in sensor technology and decision support intelligent tools to assist smart livestock farming. *J. Anim. Sci.* **2021**, *99*, skab038. <https://doi.org/10.1093/jas/skab038>.
- Porto, S.; Arcidiacono, C.; Giummarra, A.; Anguzza, U.; Cascone, G. Localisation and identification performances of a real-time location system based on ultra wide band technology for monitoring and tracking dairy cow behaviour in a semi-open free-stall barn. *Comput. Electron. Agric.* **2014**, *108*, 221–229. <https://doi.org/10.1016/j.compag.2014.08.001>.
- Spedener, M.; Tofastrud, M.; Devineau, O.; Zimmermann, B. Microhabitat selection of free-ranging beef cattle in south-boreal forest. *Appl. Anim. Behav. Sci.* **2019**, *213*, 33–39. <https://doi.org/10.1016/j.applanim.2019.02.006>.
- Bailey, D.W.; Trotter, M.G.; Knight, C.W.; Thomas, M.G. Use of GPS tracking collars and accelerometers for rangeland livestock production research. *Transl. Anim. Sci.* **2018**, *2*, 81–88. <https://doi.org/10.1093/tas/txx006>.
- Stygar, A.H.; Gómez, Y.; Berteselli, G.V.; Costa, E.D.; Canali, E.; Niemi, J.K.; Llonch, P.; Pastell, M. A Systematic Review on Commercially Available and Validated Sensor Technologies for Welfare Assessment of Dairy Cattle. *Front. Vet. Sci.* **2021**, *8*, 634338. <https://doi.org/10.3389/fvets.2021.634338>.
- Aquilani, C.; Confessore, A.; Bozzi, R.; Sirtori, F.; Pugliese, C. Review: Precision Livestock Farming technologies in pasture-based livestock systems. *Animal* **2022**, *16*, 100429. <https://doi.org/10.1016/j.animal.2021.100429>.
- Mansbridge, N.; Mitsch, J.; Bollard, N.; Ellis, K.; Miguel-Pacheco, G.G.; Dottorini, T.; Kaler, J. Feature Selection and Comparison of Machine Learning Algorithms in Classification of Grazing and Rumination Behaviour in Sheep. *Sensors* **2018**, *18*, 3532. <https://doi.org/10.3390/s18103532>.
- Ren, K.; Karlsson, J.; Liuska, M.; Hartikainen, M.; Hansen, I.; Jørgensen, G.H. A sensor-fusion-system for tracking sheep location and behaviour. *Int. J. Distrib. Sens. Netw.* **2020**, *16*. <https://doi.org/10.1177/1550147720921776>.
- Norouzzadeh, M.S.; Nguyen, A.; Kosmala, M.; Swanson, A.; Palmer, M.S.; Packer, C.; Clune, J. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, E5716–E5725. <https://doi.org/10.1073/pnas.1719367115>.

25. Khatri, K.; S, A.C.; D'Souza, J.M. Detection of Animals in Thermal Imagery for Surveillance using GAN and Object Detection Framework. In Proceedings of the 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 21–22 January 2022; pp. 1–6. <https://doi.org/10.1109/ICONAT53423.2022.9725883>.
26. Andrew, W.; Greatwood, C.; Burghardt, T. Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 2850–2859. <https://doi.org/10.1109/ICCVW.2017.336>.
27. Xu, B.; Wang, W.; Falzon, G.; Kwan, P.; Guo, L.; Chen, G.; Tait, A.; Schneider, D. Automated cattle counting using Mask R-CNN in quadcopter vision system. *Comput. Electron. Agric.* **2020**, *171*, 105300. <https://doi.org/10.1016/j.compag.2020.105300>.
28. Xu, B.; Wang, W.; Falzon, G.; Kwan, P.; Guo, L.; Sun, Z.; Li, C. Livestock classification and counting in quadcopter aerial images using Mask R-CNN. *Int. J. Remote Sens.* **2020**, *41*, 8121–8142. <https://doi.org/10.1080/01431161.2020.1734245>.
29. Wang, D.; Shao, Q.; Yue, H. Surveying wild animals from satellites, manned aircraft and unmanned aerial systems (UASs): A review. *Remote Sens.* **2019**, *11*, 1308.
30. Anderson, D.M.; Estell, R.E.; Holechek, J.L.; Ivey, S.; Smith, G.B. Virtual herding for flexible livestock management—a review. *Rangel. J.* **2014**, *36*, 205–221.
31. Handcock, R.N.; Swain, D.L.; Bishop-Hurley, G.J.; Patison, K.P.; Wark, T.; Valencia, P.; Corke, P.; O'Neill, C.J. Monitoring Animal Behaviour and Environmental Interactions Using Wireless Sensor Networks, GPS Collars and Satellite Remote Sensing. *Sensors* **2009**, *9*, 3586–3603. <https://doi.org/10.3390/s90503586>.
32. Rivas, A.; Chamoso, P.; González-Briones, A.; Corchado, J.M. Detection of Cattle Using Drones and Convolutional Neural Networks. *Sensors* **2018**, *18*, 2048. <https://doi.org/10.3390/s18072048>.
33. Schroeder, N.M.; Panebianco, A.; Musso, R.G.; Carmanchahi, P. An experimental approach to evaluate the potential of drones in terrestrial mammal research: A gregarious ungulate as a study model. *R. Soc. Open Sci.* **2020**, *7*, 191482. <https://doi.org/10.1098/rsos.191482>.
34. Ditmer, M.A.; Werden, L.K.; Tanner, J.C.; Vincent, J.B.; Callahan, P.; Iaizzo, P.A.; Laske, T.G.; Garshelis, D.L. Bears habituate to the repeated exposure of a novel stimulus, unmanned aircraft systems. *Conservation Physiology* **2019**, *7*, coy067. <https://doi.org/10.1093/conphys/coy067>.
35. Rümmler, M.C.; Mustafa, O.; Maercker, J.; Peter, H.U.; Esefeld, J. Sensitivity of Adélie and Gentoo penguins to various flight activities of a micro UAV. *Polar Biol.* **2018**, *41*, 2481–2493. <https://doi.org/10.1007/s00300-018-2385-3>.
36. Meena, S.D.; Agilandeswari, L. Smart Animal Detection and Counting Framework for Monitoring Livestock in an Autonomous Unmanned Ground Vehicle Using Restricted Supervised Learning and Image Fusion. *Neural Process. Lett.* **2021**, *53*, 1253–1285. <https://doi.org/10.1007/s11063-021-10439-4>.
37. Ruiz-Garcia, L.; Lunadei, L.; Barreiro, P.; Robla, I. A Review of Wireless Sensor Technologies and Applications in Agriculture and Food Industry: State of the Art and Current Trends. *Sensors* **2009**, *9*, 4728–4750. <https://doi.org/10.3390/s90604728>.
38. Jawad, H.M.; Nordin, R.; Gharghan, S.K.; Jawad, A.M.; Ismail, M. Energy-Efficient Wireless Sensor Networks for Precision Agriculture: A Review. *Sensors* **2017**, *17*, 1781. <https://doi.org/10.3390/s17081781>.
39. Gresl, J.; Fazackerley, S.; Lawrence, R. Practical Precision Agriculture with LoRa based Wireless Sensor Networks. In Proceedings of the Proceedings of the 10th International Conference on Sensor Networks—WSN4PA, Vienna, Austria, 9–10 February 2021; INSTICC, SciTePress: Setúbal, Portugal, 2021; pp. 131–140. <https://doi.org/10.5220/0010394401310140>.
40. Axelsson Linkowski, W.; Kvarnström, M.; Westin, A.; Moen, J.; Östlund, L. Wolf and Bear Depredation on Livestock in Northern Sweden 1827–2014: Combining History, Ecology and Interviews. *Land* **2017**, *6*, 63. <https://doi.org/10.3390/land6030063>.
41. Laporte, I.; Muhly, T.B.; Pitt, J.A.; Alexander, M.; Musiani, M. Effects of wolves on elk and cattle behaviors: Implications for livestock production and wolf conservation. *PLoS ONE* **2010**, *5*, e11954.
42. Cavalcanti, S.M.C.; Gese, E.M. Kill rates and predation patterns of jaguars (*Panthera onca*) in the southern Pantanal, Brazil. *J. Mammal.* **2010**, *91*, 722–736. <https://doi.org/10.1644/09-MAMM-A-171.1>.
43. Steyaert, S.M.J.G.; Sørensen, O.G.; Elfström, M.; Karlsson, J.; Lammeren, R.V.; Bokdam, J.; Zedrosser, A.; Brunberg, S.; Swenson, J.E. Resource selection by sympatric free-ranging dairy cattle and brown bears *Ursus arctos*. *Wildl. Biol.* **2011**, *17*, 389–403. <https://doi.org/10.2981/11-004>.
44. Wells, S.L.; McNew, L.B.; Tyers, D.B.; Van Manen, F.T.; Thompson, D.J. Grizzly bear depredation on grazing allotments in the Yellowstone Ecosystem. *J. Wildl. Manag.* **2019**, *83*, 556–566. <https://doi.org/10.1002/jwmg.21618>.
45. Bacigalupo, S.A.; Dixon, L.K.; Gubbins, S.; Kucharski, A.J.; Drewe, J.A. Towards a unified generic framework to define and observe contacts between livestock and wildlife: A systematic review. *PeerJ* **2020**, *8*, e10221. <https://doi.org/10.7717/peerj.10221>.
46. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>.
47. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. <https://doi.org/10.1109/CVPR.2005.177>.
48. Mammeri, A.; Zhou, D.; Boukerche, A.; Almulla, M. An efficient animal detection system for smart cars using cascaded classifiers. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; pp. 1854–1859. <https://doi.org/10.1109/ICC.2014.6883593>.

49. Komorkiewicz, M.; Kluczewski, M.; Gorgon, M. Floating point HOG implementation for real-time multiple object detection. In Proceedings of the 22nd International Conference on Field Programmable Logic and Applications (FPL), Oslo, Norway, 29–31 August 2012; pp. 711–714. <https://doi.org/10.1109/FPL.2012.6339159>.
50. Munian, Y.; Martinez-Molina, A.; Alamaniotis, M. Intelligent System for Detection of Wild Animals Using HOG and CNN in Automobile Applications. In Proceedings of the 2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA), Piraeus, Greece, 15–17 July, 2020; pp. 1–8. <https://doi.org/10.1109/IISA50023.2020.9284365>.
51. Munian, Y.; Martinez-Molina, A.; Alamaniotis, M. Comparison of Image segmentation, HOG and CNN Techniques for the Animal Detection using Thermography Images in Automobile Applications. In Proceedings of the 2021 12th International Conference on Information, Intelligence, Systems Applications (IISA), Chania Crete, Greece, 12–14 July 2021; pp. 1–8. <https://doi.org/10.1109/IISA52424.2021.9555562>.
52. Sharma, V.; Mir, R.N. A comprehensive and systematic look up into deep learning based object detection techniques: A review. *Comput. Sci. Rev.* **2020**, *38*, 100301. <https://doi.org/10.1016/j.cosrev.2020.100301>.
53. Ren, J.; Wang, Y. Overview of Object Detection Algorithms Using Convolutional Neural Networks. *J. Comput. Commun.* **2022**, *10*, 115–132.
54. Elgendy, M. *Deep Learning for Vision Systems*; Manning: Shelter Island, NY, USA, 2020.
55. Allken, V.; Handegard, N.O.; Rosen, S.; Schreyeck, T.; Mahiout, T.; Malde, K. Fish species identification using a convolutional neural network trained on synthetic data. *ICES J. Mar. Sci.* **2019**, *76*, 342–349.
56. Huang, Y.P.; Basanta, H. Bird image retrieval and recognition using a deep learning platform. *IEEE Access* **2019**, *7*, 66980–66989.
57. Zualkernan, I.; Dhou, S.; Judas, J.; Sajun, A.R.; Gomez, B.R.; Hussain, L.A. An IoT System Using Deep Learning to Classify Camera Trap Images on the Edge. *Computers* **2022**, *11*, 13. <https://doi.org/10.3390/computers11010013>.
58. González-Santamarta, M.A.; Rodríguez-Lera, F.J.; Álvarez-Aparicio, C.; Guerrero-Higueras, A.M.; Fernández-Llamas, C. MERLIN a Cognitive Architecture for Service Robots. *Appl. Sci.* **2020**, *10*, 5989. <https://doi.org/10.3390/app10175989>.
59. Alliance, L. LoRaWAN Specification. 2022. Available online: <https://lora-alliance.org/about-lorawan/> (accessed on 24 April 2022).
60. Cruz Ulloa, C.; Prieto Sánchez, G.; Barrientos, A.; Del Cerro, J. Autonomous thermal vision robotic system for victims recognition in search and rescue missions. *Sensors* **2021**, *21*, 7346. <https://doi.org/10.3390/S21217346>.
61. Suresh, A.; Arora, C.; Laha, D.; Gaba, D.; Bhambri, S. Intelligent Smart Glass for Visually Impaired Using Deep Learning Machine Vision Techniques and Robot Operating System (ROS). In *Robot Intelligence Technology and Applications 5*; Kim, J.H., Myung, H., Kim, J., Xu, W., Matson, E.T., Jung, J.W., Choi, H.L., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 99–112.
62. Lee, J.; Wang, J.; Crandall, D.; Šabanović, S.; Fox, G. Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles. In Proceedings of the 2017 First IEEE International Conference on Robotic Computing (IRC), Taichung, Taiwan, 10–12 April 2017; pp. 36–43. <https://doi.org/10.1109/IRC.2017.77>.
63. Puthussery, A.R.; Haradi, K.P.; Erol, B.A.; Benavidez, P.; Rad, P.; Jamshidi, M. A deep vision landmark framework for robot navigation. In Proceedings of the 2017 12th System of Systems Engineering Conference (SoSE), Waikoloa, HI, USA, 18–21 June 2017; pp. 1–6. <https://doi.org/10.1109/SYSOSE.2017.7994976>.
64. Reid, R.; Cann, A.; Meiklejohn, C.; Poli, L.; Boeing, A.; Braunl, T. Cooperative multi-robot navigation, exploration, mapping and object detection with ROS. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, 23 June 2013; pp. 1083–1088. <https://doi.org/10.1109/IVS.2013.6629610>.
65. Zhiqiang, W.; Jun, L. A review of object detection based on convolutional neural network. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 11104–11109. <https://doi.org/10.23919/ChiCC.2017.8029130>.
66. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2009**, *88*, 303–338. <https://doi.org/10.1007/S11263-009-0275-4>.
67. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis. (IJCV)* **2015**, *115*, 211–252. <https://doi.org/10.1007/s11263-015-0816-y>.
68. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.
69. Panda, P.K.; Kumar, C.S.; Vivek, B.S.; Balachandra, M.; Dargar, S.K. Implementation of a Wild Animal Intrusion Detection Model Based on Internet of Things. In Proceedings of the 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 23–25 February 2022; pp. 1256–1261. <https://doi.org/10.1109/ICAIS53314.2022.9742948>.
70. Figueiredo, A.M.; Valente, A.M.; Barros, T.; Carvalho, J.; Silva, D.A.; Fonseca, C.; de Carvalho, L.M.; Torres, R.T. What does the wolf eat? Assessing the diet of the endangered Iberian wolf (*Canis lupus signatus*) in northeast Portugal. *PLoS ONE* **2020**, *15*, e0230433. <https://doi.org/10.1371/JOURNAL.PONE.0230433>.
71. Github: VISORED. Available online: <https://github.com/ulerooboticsgroup/VISORED> (accessed on 28 May 2022).
72. Li, D.; Wang, R.; Chen, P.; Xie, C.; Zhou, Q.; Jia, X. Visual Feature Learning on Video Object and Human Action Detection: A Systematic Review. *Micromachines* **2022**, *13*, 72. <https://doi.org/10.3390/M13010072>.
73. Jeong, J.; Park, H.; Kwak, N. Enhancement of SSD by concatenating feature maps for object detection. *arXiv* **2017**, arXiv:1705.09587.
74. Li, Z.; Zhou, F. FSSD: Feature Fusion Single Shot Multibox Detector. *arXiv* **2017**, arXiv:1712.00960.

75. Tan, L.; Huangfu, T.; Wu, L.; Chen, W. Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. *BMC Med. Inform. Decis. Mak.* **2021**, *21*, 324. <https://doi.org/10.1186/S12911-021-01691-8>.
76. Liu, P.; Qi, B.; Banerjee, S. Edgeeye: An edge service framework for real-time intelligent video analytics. In Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking, Munich, Germany, 10–15 June 2018; pp. 1–6.
77. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A Survey on Performance Metrics for Object-Detection Algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242. <https://doi.org/10.1109/IWSSIP48289.2020.9145130>.
78. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2015**, arXiv:1512.02325v5.
79. Arriaga, O.; Valdenegro-Toro, M.; Muthuraja, M.; Devaramani, S.; Kirchner, F. Perception for Autonomous Systems (PAZ). *arXiv* **2020**, arXiv:cs.CV/2010.14541.
80. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767v1.
81. Bochkovskiy, A.; Wang, C.; Liao, H.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934v1.
82. Adarsh, P.; Rathi, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 687–694. <https://doi.org/10.1109/ICACCS48705.2020.9074315>.
83. Github: Ultralytics. yolov5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 28 May 2022).