ISSN 2007-9737

Classification of Domestic Dogs Emotional Behavior Using Computer Vision

Víctor Ocyel Chávez-Guerrero[1], Humberto Pérez-Espinosa[2], María Eugenia Puga-Nathal[1], Verónica Reyes-Meza[3]

[1] Instituto Tecnológico Nacional de México, Mexico

[2] Centro de Investigación Científica y de Educación Superior de Ensenada-UT3, Mexico

[3] Universidad de Autónoma de Tlaxcala, Centro Tlaxcala de Biología de la Conducta, Mexico

{M19291005, maria.pn}@cdguzman.tecnm.mx, hperez@cicese.mx, vrmeza@gmail.com

Abstract. Dogs are the most common companion animals worldwide, motivated by their exceptional social behavior with humans. Unlike many animals, dog learn vocal commands, identify moods, maintain eye contact, and recognize facial expressions. Besides, dogs have great agility and senses of smell and hearing superior to humans, so dogs have been trained for crucial tasks like search, rescue, and assistance. Therefore, it is relevant to do scientific research to understand the fundamentals of behavior and communication that increase the use of its capabilities for the benefit of the human being, guaranteeing the animal's welfare. In this work, a computational method for analyzing dog behavior based on artificial vision techniques was developed. A video database recorded in positive and negative stimuli that induced different emotional states was used. The proposed method determines the dog's emotional state at a given time, which opens a promising field to develop new technologies that trainers and users can take advantage of, to improve the processes of selection, training, and execution of the tasks of working dogs. Using the proposed method, the best test accuracy value we obtained was 0.6917 on the best model trained using transfer learning over the architecture MobileNet, getting good but not perfect results. The training process was carried out using 1067 images distributed among four categories, aggressiveness, anxiety, fear and neutral. The proposed method obtained acceptable results but can still be improved in technical and methodology terms. However, this method can be used as a baseline for exploring and expanding the canine behavior study using computational models.

Keywords. Computer vision, machine learning, canine behavior.

1 Introduction

The study of dog behavior is a task that has become an essential part of society nowadays, since some dogs may be assistance or service dogs, whether they are rescue dogs, guide dogs, or guard dogs, among others. Behavioral assessment and analysis are carried out in various ways, including dog owner questionnaires, expert evaluation, standardized measures, and observational studies [13]. However, because sometimes it is possible that some details are missed or that the human does not detect some variation, it is necessary to generate or create tools that support the evaluation of the dogs' behavior [3, 2]. Since a computer can analyze a whole day of dog activities and for a human it is complicated, it would be valuable to capture those details

and report them to the experts, who will make a complete decision since they will have more information regarding the behavior of the dogs.

Also, since dogs cannot express emotions through words, it is necessary to determine their internal state and behavior using other aspects [5] like visual or auditory actions, for example, showing their teeth, barking, growling, moving their tail, and adopting a particular posture. Then it is essential to develop technology that uses this abstract data given by the dogs and transform it into information that humans can understand and use.

This work aims to develop a tool to evaluate dog characteristics and behavior, more specifically, dogs that are candidates or dogs that are already in the process of training to perform rescue work, thus having an impact on society, helping dog trainers, and providing a tool to the field of computational sciences in the field of animal behavior analysis. This solution can be achieved through machine learning since it can solve various problems classifying and evaluating specific features from different media, such as images, audio, or video. Then, using a machine learning model to extract features from images we gathered, we were able to make a computational model that uses these data and can describe a related emotion or behavior and give information about it that helps to pay more attention to the care of dogs, even for someone who is not familiar with these animals and their emotions.

Section two presents a review of research work on posture detection and classification in animals and specifically in dogs. Section three describes the data used. In section four, the proposed and developed method we describe in detail. We describe the results obtained in section five. In section six, the results are discussed, and finally, a conclusion of the work is made in section seven.

2 Related Work

First, it was necessary to carry out exhaustive research on the works related to the problem to be solved in order to find similarities and differences, and thus be able to set the guideline for the methodology to be designed by understanding what already exists and what can be improved in this field, obtaining the following information.

In this section, we present an analysis of works tackling tasks related to the analysis of human and animals behaviors using computer vision.

2.1 Detection and Classification of Postures by Means of Computer Vision

In the field of posture detection and classification through the use of artificial vision, there are several studies with their particular methods focused on detecting specific aspects, for example, some studies have been conducted to detect full-body postures; an example of this is the work by [12] in which the approach is to develop a system that detects and classify postures in humans using a Kinect device by implementing convolutional neural networks for a 2D and 3D model of postures. In this work, they were in charge of using RGB images, and depth images from the Kinect device to train a model that works with flat or 2D images, but also implement 3D models of the human skeleton of each posture to be classified.

In this case of the postures Standing, Bent, Sitting, Walking, Crouching, plus the depth images to make calculations of geometric angles and distance sets to classify and recognize the postures subsequently. Continuing with the advantages and conditions provided by Kinect, other posture recognition works have been proposed, such as the proposed by [21]. In this work, the authors proposed using technology that allows us to obtain more visual information by using the RGB and infrared cameras to analyze and classify the postures of the dogs through a semi-supervised model. The method is limited to classifying postures such as standing, sitting and lying down or lying down.

2.2 Computer Vision Applied to Animals

The use of depth or infrared images is not always necessary to detect and classify postures as demonstrated by [23] where the main point was to determine which two-dimensional image classification system in combination with deep learning techniques can be used to detect postures

in pigs living in commercial farms. This task was done by analyzing detection methods based on deep learning and convolutional neural networks, focusing mainly on the architectures Faster R-CNN, SSD, R-FCN combined with ResNet and Inception ResNet V2 in order to perform the classification and identification of 3 main postures, standing, lying on their bellies and lying on their sides.

In this work, it was only necessary a camera that provided regular RGB images, extracting a total of 4900 images taken aerially, which were divided into training, validation, and testing sets resulting in an accurate system for the classification of the postures obtaining the best classification results at a learning rate (Learning Rate) of 0.003 during training, obtaining accuracy of 0.93 in standing posture, 0.92 lying on its side, and 0.89 in lying on its belly posture.

2.2.1 Detection of Postures in Dogs by Means of Computer Vision

The posture detection in dogs has been studied in several works such as the one by [21] where the authors propose to design a semi-supervised algorithm capable of detecting basic postures such as standing, sitting and lying down using the technology of the Kinect device, which has infrared, depth and regular cameras. Several videos are captured using an infrared and a regular camera, the videos are processed to detect the required postures. The authors of the work carried out the following steps to perform the posture detection:

1. Capture multiple static depth images of the background without the dog being present.
2. Pool all previously captured images of the background using the mean and average of the neighboring pixels in each image to reduce the noise produced by the Kinect camera.
3. For each depth image with a dog on the scene, the background is subtracted, considering that everything 20 millimeters or less from the ground is part of the background.
4. Contour detection is used to find the exact location of the dog in the scene.
5. Calculate the average distance between the dog and the Kinect using the information extracted from the depth camera and the dog's contour detection.
6. The three postures to be evaluated were captured at 10-second intervals between each posture, and then the images resulting from the recording of these postures were manually labeled.
7. The labeled images were used to train the semi-supervised algorithm and subsequently, its execution is performed in real-time.

Under this method, the best results obtained were accuracy of 0.94 for standing posture detection, 0.91 for sitting posture, and 1.00 for lying or lying down posture.

Another related work by [22] aimed to reconstruct and analyze the position of a dog in a scenario using an unsupervised algorithm, again using the Kinect and making use of the depth camera and infrared camera, plus by using the information coming from the depth camera, infrared camera, and a binary image, the authors were able to reconstruct with better accuracy the silhouette or body of a dog within the scene, eliminate the extra or unnecessary elements and thus determine the orientation of the dog, that is, if the dog was looking straight ahead, if the dog was on its side, if it was being viewed from the back of the body or if it was leaning in a particular direction. For this the authors conducted three experiments under different conditions and following three different methodologies. However, each methodology consisted of changing the camera's position, changing the angle at which the dog was located and finally segmenting the dog's body parts. By combining the two works of these authors, it would be possible to create a system capable of detecting basic dog postures with greater accuracy and distinguishing the dog's orientation during the posture.

2.3 Canine Behavioral Studies

On the other hand, there are many studies on canine behavior; however, not all of them are

standardized nor use a common language that allows cross investigations among all the studies carried out. In Jones and Gosling's work [16], an analysis and a compendium of articles and works that have been developed on canine behavior between 1934 and 2004 are made, presenting the following results:

- 43 studies analyzed fear in dogs, mostly related to reactivity to some external stimulus or impulse.
- 31 studies discussed sociability in dogs, focusing on interest in training dogs, interaction with other people, and interaction with other dogs.
- 34 studies focused on studying responsiveness to training in dogs, focusing on tendencies classified as "Distraction" or "Concentration", including "problem-solving, cooperation and willingness to work".
- On the other hand, 30 studies talk about dogs' aggressiveness, carrying out somewhat invasive tests such as the approach of a stranger and then the attack by the stranger towards the dog or the dog's owner to record the animal's reaction. It should be noted that depending on the authors of the research, there were subcategories of aggression.
- 16 studies focused on the dominance and submission that a dog could present in different scenarios, for example, not moving out of a person's way.
- 15 studies were oriented to analyze activity behavior, i.e., the movements a dog performs, its locomotor coordination, among others.
- In addition, authors Jones and Gosling found 23 articles that were not members of the above categories but also studied dog behavior.

2.4 Studies on Canine Behavior with Application of Computational Modeling and Computer Vision

The postures of dogs have not been the only field studied through computational models, also their behavior. In the work by [26], it is mentioned that during an emergency or in an unexpected situation, dogs tend to present actions that, under normal conditions, would not show; for example, they may start to run, knock down people who are nearby or on the contrary not act or perform any action in an emergency, so the authors mention that by monitoring the behavioral patterns of dogs during an emergency, technology can be developed capable of asking for help during the unexpected situation.

The authors propose that one solution could be to capture the dog's vital signs and movement through sensors placed on the dog since dogs can react through postures or movement and irregular vocalizations that indicate stress or an adverse condition. They also mention that if this biometric data is combined with cameras that indicate their position within a specific area, a system can send a distress signal to the emergency services to arrive at that location or to know that something unexpected happened in the area that requires human attention. The authors mention other works developing other solutions to the problem and how certain technologies can complement such systems.

On the other hand, estimating or predicting the movements and behaviors that a dog may have in certain situations is what was done in the work by [11], where they present a computational system in which, from sensors and cameras attached to a dog, the behavior of a dog is modeled, extrapolating the recorded movements to a 3D model using digitization techniques to study the possible behavior and actions that a dog could carry out when performing certain activities; This analysis is performed using a computational model that analyzes images captured on video and recreates the probable movements that the dog will perform. These images are recorded with a GoPro camera mounted on the dog at head height, and information is also collected on the dog's movement and body position during the journey using Inertial Measurement Units or IMUs. Four IMUs collect information from the limbs, one IMU measures the activity of the tail and one more is responsible for measuring the position of the body; the IMUs allow capturing the dog's movement in

terms of angular displacement. Then this data is analyzed by a ResNet convolutional neural network. The information is encoded and sent to the computational model in charge of recreating the dog's movements during the journey. Once the model is sufficiently trained, the system can predict the future movements that a dog might make given certain conditions or in the face of specific stimuli with a certain degree of accuracy.

As it can be seen, many works use computational models and artificial vision to solve posture and gesture classification problems in both humans and animals, each from their particular approaches but attacking the same shared problem. In addition, there is an identifiable trend towards face detection in different species. Although face detection in humans is a widely researched and applied field, animal faces have particular characteristics that make it necessary to carry out exhaustive work and research focused on each species to create a face detection model extracting those not shared between species.

Although some works on canine behavior tackle some particular issues about domestic dogs, more research and development of new technologies are needed to classify and detect common behaviors and attitudes related to its internal state and its responses in certain particular situations.

This paper attempts to venture into the field of domestic dog behavior classification, in particular, detecting some emotion-related postures. We take advantage of computer vision techniques to detect postures or faces and consider studies on canine behavior, besides adding other machine learning techniques. The proposed method is described in detail in the proposed method section.

3 Used Data

The present work uses a video database collected during a previous project carried out during 2015 and 2017. These data belong to the company Mescalina who allowed its use for research and development purposes.

This database comprises 386 videos of 121 different dogs, and each video is of varying duration; some can last from a few seconds to hours.

In addition, they were recorded in uncontrolled environments, mainly being the private homes of the dogs' owners, so the number of objects in the scene, camera movement, background color, lighting, among other factors, can be highly variable. This variation can positively or negatively affect processing and development. Then it is expected that not all the videos in question can be used because of the difficulty in processing them. Therefore, it was necessary to perform pre-processing to filter out the videos that were not useful, either because they did not contain relevant information or because the visibility was null. The following explains how this pre-processing was carried out. These databases are available for research purposes by requesting them directly from the corresponding author of this article.

In addition to the videos, a public database named Dogs vs. Cats[1] was used, which contains 25,000 images in total between the categories of dogs and cats; this dataset served as support to have even more images of dogs for the training of the model.

4 Proposed Method

In this section we will describe the proposed method in detail, and a general outline of each of the steps followed to complete the methodology is shown in figure 1.

4.1 Data preprocessing

First, performing a manual review of each video was necessary to identify which videos contained valuable information and which could be discarded. The manual review process consisted of the following:

1. Two tables were made, one for the 2015 videos and one for the 2017 videos in a spreadsheet. In these tables, the names of each dog made up the rows, while the columns belonged to the behaviors. It is important to note that the number of videos for each dog could vary, i.e., a dog could have only one video while another could have two or more.

[1], 2014.

ISSN 2007-9737

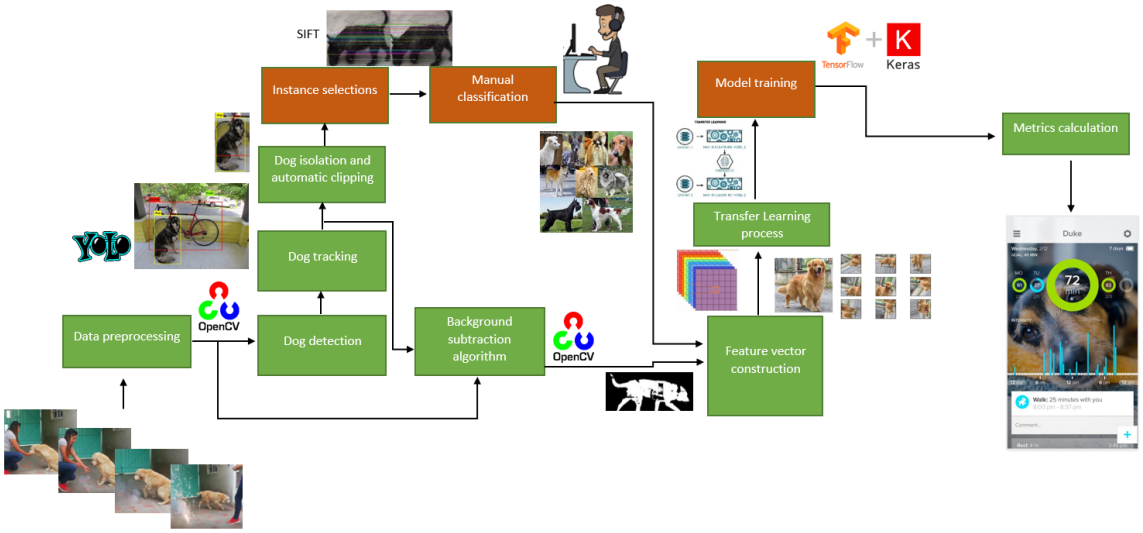

Fig. 1. General outline of the proposed method

Table 1. Number of videos in each category according to manual review and classification

Year	Aggressiveness	Anxiety	Fear
2015	57	44	12
2017	128	72	15
Total	185	116	27

2. Once the tables were structured, each video of each dog was played one by one; that is, all the videos belonging to each dog were manually reviewed to evaluate the usefulness of each one.

3. In order to evaluate, the entire video was viewed, and things such as the visibility of dogs, actions of them and what behavior each action could incur, as well as issues such as the lighting of the scene, the movement, and focus of the camera, were evaluated. For example, suppose there was insufficient lighting in a video and the dog is not visible, that video is discarded, as well as if there are too many obstacles in front of the dog or the dog's gestures and movements are not visible due to camera movement. On the other hand, if the dog remained without performing any action during the whole video, even if the visibility was perfect, it was also discarded.

4. Once an action belonging to behavior was identified and verified that other elements such as visibility or lighting were not a problem, the name of the video was noted in the column corresponding to the identified behavior. If the name consisted only of digits, the last 3 or 4 digits of the name were recorded; if the name was alphanumeric, it was recorded as such. If the video was discarded due to lack of actions by the dog, the line with its name was left blank; on the other hand, if discard was due to elements such as visibility, lighting, obstacles, etcetera, an annotation was made indicating the reason for discard.

   It is important to note that the same video could be classified in more than one behavioral category, i.e., if, in the course of the video, the dog was aggressive for a few minutes and then became anxious or fearful, the name of the video was noted in the corresponding columns.

The number of videos classified in each category is shown in the table 1.

4.1.1 Sample Generation

Once the videos were listed and cataloged manually, the next step consisted of coding a

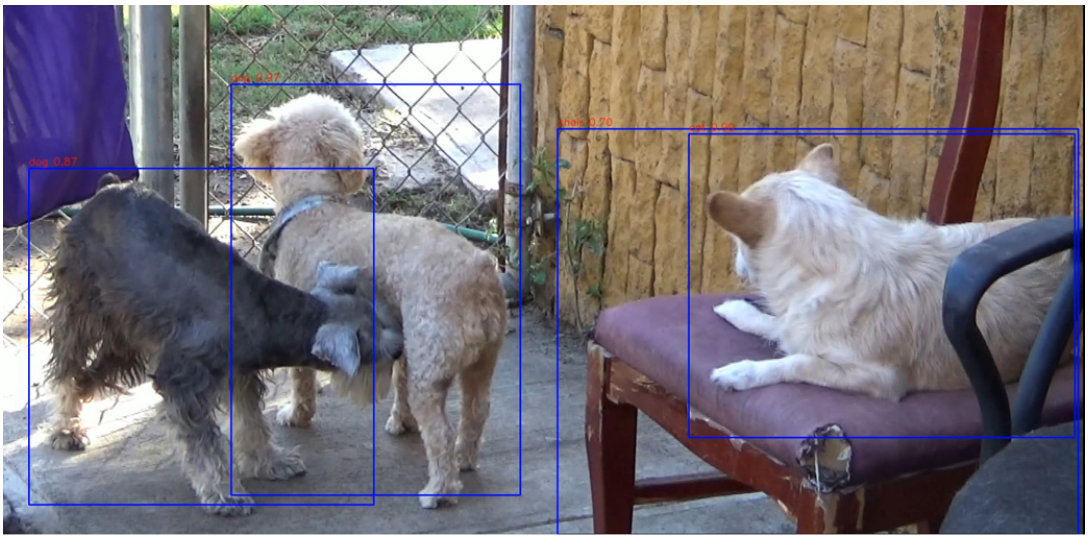

Fig. 2. General detection of objects in scene

Fig. 3. Isolation in detection by focusing only on the dog

program that automatically detects the dogs within the videos mentioned in the previous section. It once detected, automatically cropping the detected dog(s), focusing or framing exclusively on the dog, and eliminating the rest of the scene as best as possible. The program is coded/designed in the Python programming language on its version 3.7 using the OpenCV library on its version 4.1.2 with tools for artificial vision and was implemented on the Google Colaboratory platform given the facilities offered by this platform. This program had several stages and evolutions, as described in the following subsections.

4.1.2 Dog Detection and Tracking

The first phase consisted of making the program capable of detecting and tracking dogs in real-time within the video scene; this was achieved by using state of the art in computer vision called YOLOv3 [25], which is a real-time object detection system based on the open-source neural network Darknet: Open source neural networks in C [24], YOLOv3 served as the base object detection system due to its flexibility to adapt to custom classes as well as high accuracy to detect and track objects [25].

A pre-trained version of YOLOv3 was used that in addition to detecting dogs was able to identify other objects, the pre-trained version of YOLOv3 is the one provided on its Web page[2] and the pre-trained weights were also provided in the same Web page.

However, the code was adapted so that only information related to the detection of the dog was extracted, which was achieved by code that filtered

[2] pjreddie.comdarknetyolo

ISSN 2007-9737

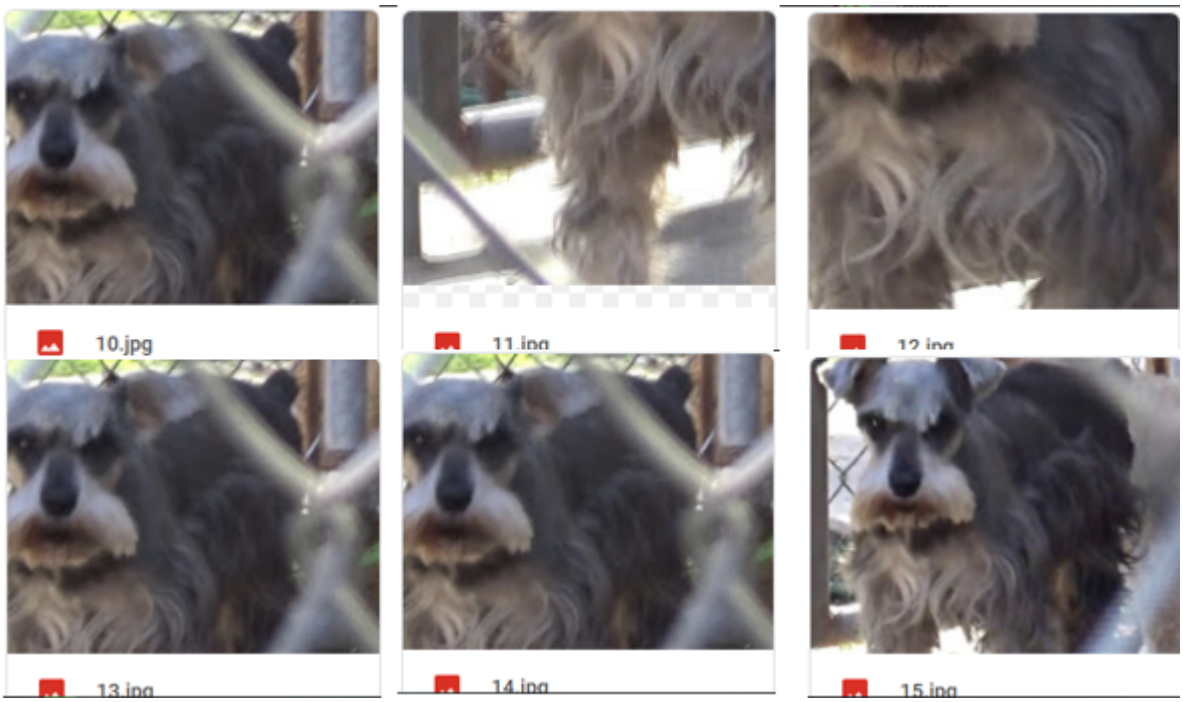

Fig. 4. Clippings stored with consecutive numbering

the class identifiers provided by YOLOv3 during detection, so that each time YOLOv3 detected a dog, the Python code would retain the location in the scene for that detection based on the X and Y positions that YOLOv3 uses during its execution to draw a rectangle around the detected object, so using this information the program is able to exclude the other detected objects but focusing now exclusively on the detected dog. The results can be seen in figures 2 and 3, wherein figure 2 the general detection of the objects in the scene is observed and later in figure 3 the isolation in the detection focusing only on a dog can be observed.

Once the object is detected, the object tracking process was also assigned to YOLOv3 by analyzing the video frame by frame, that is, every time the video advances one frame during the analysis, YOLOv3 performs the detection of objects in the scene, then all the frames with the detections are joined, thus achieving the tracking of the objects as the video advances.

Once this first stage was achieved, the next step was to get the system to perform automatic clipping and storage of the detected dogs while ignoring the rest of the objects in the scene.

4.1.3 Dog Isolation and Automatic Clipping

The next step was to make the program capable of automatically isolating, trimming, and storing the dogs detected in each frame of the video, given that the number of frames in a video could be huge to perform manual trimming (for example, a video with a duration of 3 and a half minutes could contain an average of 7000+ frames), in addition to the fact that not in all frames the dogs could be visually appreciated so that some frames were of no value as material for the database. Using the information provided by the detection and isolation obtained in the initial phase, the program perform automatic clipping and storage, initially storing all the clipped images of the frames where a dog was detected. In order to store each clipping automatically, the names of the stored images were limited to consecutive numbers starting with number one, as shown in the figure 4.

However, the number of images similar to each other was very high, i.e., there could be ten or more clippings where the change in the dog's posture or action was minimal, and the clippings were the same or at least similar, due to the time elapsed in the video or simply because the dog did not move or perform any other action for a while. Hence, the next step was to get the program to discard the clippings that were highly similar to each other.

4.1.4 Instance Selections

At this stage, the system compares consecutive clippings, and if the compared clippings were similar, one of the clippings would be discarded; on the other hand, if the clippings were different, they would be stored. In order to achieve this task, a Scale Invariant Feature Transforms (SIFT) comparison algorithm [19] was implemented, thus reducing the number of similar images stored.

The SIFT algorithm is one of the existing invariant point extraction methods; its name comes from the fact that it performs transformations of image data in invariant coordinates to changes such as scale, rotation, illumination (to some extent), and 3D viewpoint if implemented in a problem with these features [18]. The extracted data is considered an image feature; such a feature is a single point of interest, taking into account its position, scale, orientation, and descriptive vector. Thus, during the initial stages of the project, it was necessary to implement an algorithm that, utilizing the descriptors it would be possible to compare two images and determine how similar they were to each other. Due to the characteristics of the images used in this work, the algorithm needed to be able to detect two points of interest that were identical despite a slight change in lighting, the relative dimensions between the images, or if the image had rotated slightly due to the movement of the camera with which the videos were recorded, not to mention the noise that may exist in the images extracted from a video originated by the same movement.

The process for comparing clippings is as follows:

1. Two snippets are taken, being firstly the most recent stored snippet, i.e., the last one that was stored by the system (hereafter referred to as snippet 'A') and secondly the snippet that was just obtained by automatic detection during program execution (hereafter referred to as snippet 'B').
2. The dimensions of both clippings got checked; if they are equal in dimensions, both clippings pass to the next stage; otherwise, both clippings get resized to the same dimensions.
3. If the percentage of coincidence between both images is higher or equal to 40%, cutout 'B' (i.e., the one just obtained during the program execution) is discarded since it is considered to be very similar to cutout 'A' and it is considered unnecessary to store two such similar cutouts, so cutout 'A' is kept for the next comparison, this threshold was selected through several experiments, where a threshold of 40% gave the best results while maintaining a balance between the number of stored images and the differences over each stored image, because a higher threshold negatively reduced the number of stored images, and a lower threshold increased the number of images that were similar to each other.
    (a) However, if cutout 'B' is kept below 40%, it is considered to have a significant difference from cutout 'A' so it is stored, and now cutout 'B' is used for the next comparison, i.e., being the last cutout stored it becomes the new cutout 'A' and the next cutout generated by the system becomes the new cutout 'B'.

In addition, employing operations based on the time elapsed between each frame and, therefore, the relative time between each compared clipping, reducing the number of stored images without losing those that may contain relevant information for the project.

The number of images stored before the instance selections may vary depending on the number of times a dog is present during the video or the length of the video, it also depends on

the number of frames per second in which the video was recorded, but on average the number of images stored could be about 6500 images for a three and a half minute video in which dogs appear constantly, and after instance selections this number could be reduced to 5000 +-300 images or more. There were instances where the number of stored images was reduced to 2000 or less images in a video with the same duration.

4.1.5 Background Subtraction Algorithm

When it comes to detecting postures, some works have chosen to extract the background of the scene and work with the resulting silhouette [4], [10], [22], [12] this under the principle that the resulting silhouette contains the necessary information to distinguish between one posture and another.

In order to extract only the silhouette, it is necessary to separate the foreground (front of the scene) and the background (back of the scene), and for this, there are background subtraction algorithms, these algorithms get applied to a video or a sequence of images that are part of the same sequence, where the background changes very little or nothing and instead, in the foreground objects are moving along the scene constantly.

Considering the results of the works as mentioned earlier, we decided to implement a Mixture Of Gaussian (MOG) background subtractor algorithm while performing the regular detection and comparison, thus obtaining conventional RGB images and images of the dog silhouettes in black and white, numbered at the same time as the conventional RGB images. That is, image 50 obtained by background subtraction corresponds to image 50 in RGB and so on. This algorithm is included in the OpenCV library, the base code for its implementation can be found on its Web page (tutorial for background subtraction).

4.1.6 Manual Classification

Once the program was able to generate the samples automatically, the next step consisted of manually classifying the samples into the categories that would get used during the training, which are:

— Aggressiveness,

— Anxiety,

— Fear,

— Neutral.

The classification process was carried out by manually reviewing each of the images generated from each video, where each image was reviewed visually, and each detail was checked against a list of the most common visual actions that could indicate that a dog is displaying a specific behavior. This definition of actions or behaviors was developed by the authors based on the following references [1, 30, 6], the table of behaviors can be seen in table 2.

The ”neutral” category was included to represent the postures or behaviors that are not associated with any other categories. This action was done in order to avoid false positives by the classification system.

After the manual classification the number of samples was about 1067 images distributed among the four categories, 343 in the Aggressiveness category, 212 in the Anxiety category, 160 in the Fear category, and 352 in the Neutral category, mixing the obtained images during the videos processing and the ones extracted from the Dogs vs Cats dataset.

The final dataset can be requested directly from the corresponding authors of this article.

4.1.7 Feature Vector Construction

Once all the images were obtained and classified into their respective categories, the step before training the model consisted of generating the feature vector by applying Data Augmentation techniques, which consists of generating synthetic samples from rotations, biases, partial or total focus and blur on an area of the image, mirror effect, among other techniques [32].

After applying data augmentation techniques such as rotation, biases, blur and mirror effects to the data set, more samples became available during training, this whole process was achieved using the OpenCV library which already contains data

Table 2. List of behaviors to observe

Category	Behavior				
Aggressiveness	Show teeth	Bristling back hair	Growl	Forward facing ears	Bark
	Excessive salivation	Tail lift	Throwing against the bars		
Anxiety/Nervousness/Quietness	Howl	Intense scratching	Nibbling/Biting	Liplicking	Slight tail lift
	Excessive preening	Persistent licking of objects	Shaking the body	Yawn/Yawning	Licking ribs
	Hyperactivity	Snorting			
Fear	Neck retraction	Low posture	Hidden tail	Ears down	Howling with certain frecuency
	Complain	Curling up	Trembling	Scratching the cage	

augmentation techniques, then the transformation of the images to numerical vectors is implemented, this vectors contain the information of each color, the number of channels of each image, adjusted dimensions for all images, class labels, and also applied normalization operations on each vector, thus having normalized vectors for each category and per image.

Once the feature vectors have been normalized and pre-processed, the model training got performed, the training process it is explained in the subsection Model training.

To summarize, the final dataset consisted on 1067 images distributed among the four categories, 343 in the Aggressiveness category, 212 in the Anxiety category, 160 in the Fear category, and 352 in the Neutral category, it was follow a criteria of 70%/30% distribution for the train and validation sets, while for the classification tests set, 52 images were used in the Aggressiveness category, 32 in the Anxiety category, 24 in the Fear category and 53 in the Neutral category.

4.2 Model Training

For model training, it was decided to use Transfer Learning [31] to take advantage of the properties of robust neural network architectures tested and pre-trained on similar categories. Several models were trained based on different architectures to compare and choose the model that best meets the purpose of this work. Our motivation to test the incorporation of pre-trained models in a learning transfer scheme was the poor performance in classifying our data. The assumption made was that our database was not big enough to learn the most characteristic attributes of dog postures and expressions. For this reason, we decided to use trained models with data sets from a domain close to the one of interest.

One of the models was trained under the ResNet50 architecture [14], loading weights from ImageNet50 [9] and using Transfer Learning with a neural network ResNet50. We based on the example given by the Web page Towards Data Science [27], taking this architecture to classify between two categories: Dogs and Cats. Taking that basis the pre-trained ResNet50 architecture was used as a feature extractor, some layers were frozen, and the final layers that were fully connected were removed and replaced by the fully connected custom layers dedicated to performing the classification required for the project. Subsequently, a Fine-Tuning process was performed on the model already trained with the specific dataset for this work following the indications in the Towards Data Science Web page [27].

Another model trained for the project was created using Transfer Learning under the VGG-16 [28] architecture using pre-computed bottleneck features; these features are an abstract representation of the images after having passed through the immutable (frozen) layers of the neural network. These bottleneck features came from a dog breed classifier, classifying in total 133 dog breeds ([8]); the specific adaptation process for the project was similar to the one performed on the ResNet50 architecture, most of the layers were frozen to be used as feature extractors, the final layers were removed and replaced by fully connected layers to perform the particular classification task of this work.

Finally, another model was made using the MobileNet architecture, which is a model designed to be used in mobile applications, and it is TensorFlow's first mobile computer vision model [15], this model uses depthwise separable convolutions, which significantly reduces the number of parameters when compared to the network with

regular convolutions with the same depth in the nets. This process results in lightweight deep neural networks.

The depthwise separable convolution consists of two operations:

— Depthwise convolution, which is a map of a single convolution on each input channel separately. Therefore its number of output channels is the same as the number of the input channels.

— Pointwise convolution, this can be explained as a convolution with a kernel size of 1x1 that combines the features created by the depthwise convolution.

This MobileNet architecture has an accuracy of 70.6% across the ImageNet set compared to GoogleNet and VGG 16, which obtained 69.8% and 71% respectively [15]. Also, there exist some other thinner models that enhanced the MobileNet architecture even more called Thin MobileNet [29]. However, for this work, we used the standard MobileNet architecture under a tool called Teachable Machine [20] that trains a classifier in Tensorflow with the classes and images provided by the user, the tool allows you to modify specific parameters, but most of them are already predetermined.

This tool is developed in Tensorflow.js, Javascript, and built over MobileNet and pre-trained on 1000 categories over ImageNet [7]. This project is also part of Experiments with Google [17] and is an open-source project hosted on GitHub which can be accessed directly from their website. The advantage of using this tool is that it allows adding as many samples and classes as needed without having to spend too much time modifying the architecture, as this process had to be done in the other models previously described; in addition, at the end of the training, it allows to test the model in real-time with samples obtained through the webcam or local files. Finally, it allows downloading the trained model in TensorFlow + Keras, TensorFlow Lite, TensorFlow JavaScript format, or hosting it on the website to share the link so that other users can test the model.

5 Results

The selection of the architecture for the classifier consisted of an exhaustive investigation to achieve the implementation of Transfer Learning to save time and resources in the design of the convolutional network architecture; for this three architectures were chosen, ResNet50 with pre-trained weights from ImageNet, VGG-16 with weights in 'bottleneck features' format based on a classifier of 133 dog breeds and finally the architecture MobileNet pre-trained over 1000 classes over ImageNet provided in the Teachable Machine tool ([20]).

Two versions of the classifier were trained, one version with four classes on the 3 mentioned architectures and another version with three classes only on the MobileNet architecture as it turned out to be the best of the three architectures used.

5.1 4 Class Model

This model is made up of the following classes:

— Aggressiveness,

— Anxiety,

— Fear,

— Neutral.

The training process was carried out using 1067 images distributed among the four categories, 343 in the Aggressiveness category, 212 in the Anxiety category, 160 in the Fear category, and 352 in the Neutral category.

As can be seen, the Anxiety and Fear categories are the most unbalanced due to the particular visual characteristics or postures to be considered to classify such behaviors in these categories.

Table 3 shows a summary of the results described above.

During the training of the models, the ResNet50-based classifier was the one that obtained the worst accuracy value, with 0.3524 at best and 0.2948 at worst. In contrast, the validation accuracy value remained with an average value of

Table 3. 4-class classifier training results per model

Metric	ResNet50	VGG-16	MobileNet
Best Accuracy	0.3524	0.3920	0.9710
Worst Accuracy	0.2948	0.3056	0.9652
Best Validation Accuracy	0.3302	0.3683	0.6824
Worst Validation Accuracy	0.3209	0.3051	0.6795
Average Test Accuracy	0.3002	0.3243	0.6917

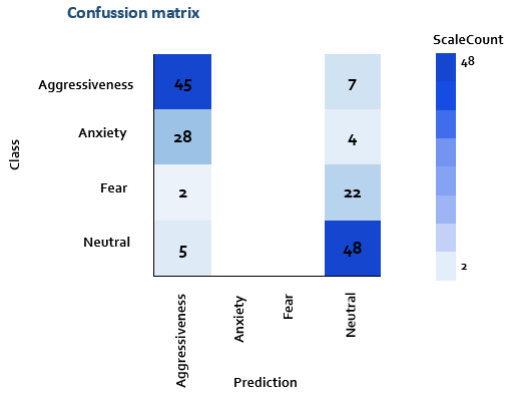

Fig. 5. 4-class ResNet50 model confusion matrix

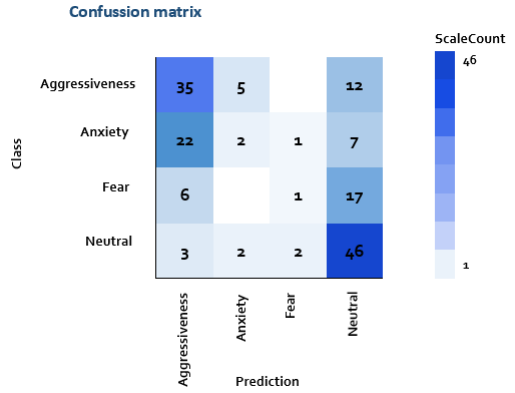

Fig. 6. 4-class VGG-16 model confusion matrix

0.3262 constant with slight variations to 0.3302 as maximum or 0.3209 as minimum.

This classifier had the most trouble classifying the Anxiety and Fear images, classifying all test images belonging to these categories in the Neutral category.

The VGG-16 based model obtained similar results to the ResNet50 based classifier during training, with slightly better values such as 0.3920 in accuracy at best and 0.3056 at worst; on the other hand, the validation accuracy obtained an average value of 0.3425 with slight changes to 0.3683 as maximum or 0.3051 as a minimum. Regarding the classification, the performance was not as good as expected since it presented difficulties in classifying the Anxiety and Fear categories; although it did not misclassify all the samples as the ResNet50-based model, the vast majority fell into the Aggressiveness or Neutral categories, correctly classifying only 5.3% of the Anxiety category and 3.1% of the Fear category.

Finally, the last model was trained over the MobileNet architecture in the Teachable Machine tool; this model was the one that obtained better results compared to the other two since it obtained an accuracy value of 0.9710 while the validation accuracy value was 0.6824. The 4-class model correctly classified 86.5% of the test samples in the Aggressiveness category, 71.6% in the Neutral category, and 59.3% in the Anxiety category, having more difficulties in the Fear category classifying only 29.1% of the samples belonging to this category.

Of the three models, the one that obtained the best results was the model trained with the MobileNet architecture, obtaining an accuracy value during training of 0.971 compared to a value of 0.3524 in ResNet50 and 0.392 in VGG-16 using the same database belonging to the project, in addition to the fact that the model trained in MobileNet presented fewer problems in the confusion matrix when classifying the test images in the four categories.

The 4-class confusion matrix for the ResNet50 and VGG-16 models can be seen in figures 5 and 6 respectively, and the confusion matrix for the 4-class MobileNet can be seen in the figure 7.

5.2 3 Class Model

Seeing that the previous classifiers presented some difficulties in classifying the samples belonging to the Anxiety and Fear classes, it was decided to try to unite these classes and turn them into a single class, thus having a model with the following classes:

— Aggressiveness,

— Anxiety/Fear,

— Neutral.

Combining the Anxiety and Fear categories, the distribution of images was 343 in the Aggressiveness category, 372 in the unified Anxiety/Fear category, and 352 in the Neutral category, resulting in a better balance between the three classes.

For the training of this model, it was decided to use only the MobileNet architecture under the Teachable Machine tool since it was the one that yielded the best results during the training of 4 classes and obtained the best performance at the time of classification. The accuracy value obtained during training was 0.9921, while validation accuracy was 0.6861.

Although the accuracy and validation accuracy values were very similar to the 4-class model, a more noticeable change was observed in the classification. The accuracy per class was calculated for both the 4-class model and the 3-class model, obtaining the following results shown in the table 4:

During the classification of the test samples, the 3-class model correctly classified 71.1% of the Aggressiveness samples, 64.2% of the samples in the unified Anxiety/Fear category, and 64.1% in the Neutral category.

Compared to the 4-class model, the percentage of classification hits in the aggressiveness and neutral categories decreased from 86.5 and 71.6 to 71.1 and 64. On the other hand, in the Anxiety and Fear categories, more balanced values were obtained for the remaining classes, since when these classes were separated, a percentage of classification hits of 59.3% and 29.1% were obtained, respectively, and when they were unified into a single class, a percentage of hits of 64.2% was obtained.

The summary of percentages can be seen in tables 5 and 6. The 4-class and 3-class confusion matrix for the MobileNet model can also be seen in figures 7 and 8 respectively.

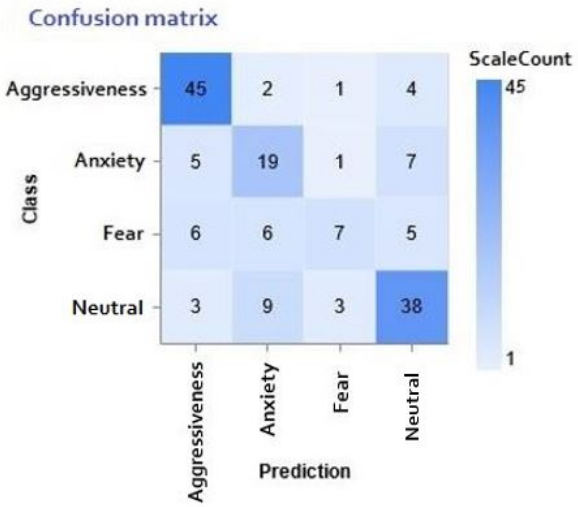

Fig. 7. 4-class MobileNet model confusion matrix

Table 4. Comparison of accuracy per class in the 4-class and 3-class models

Model	Aggressiveness	Anxiety	Fear	Neutral
4-class model	0.87	0.59	0.29	0.72
3-class model	0.71	.64		.64

Table 5. Percentage of successes during the classification test: model 4 classes

	Aggressiv	Anxiety	Fear	Neutral
Percentage	86.5%	59.3%	29.1%	71.6%

Table 6. Percentage of successes during the classification test: model 3 classes

	Aggressiveness	Anxiety/Fear	Neutral
Percentage	71.1%	64.2%	64.1%

6 Discussion

As we can see from the results section, the architecture and the data available for training the model play a crucial role. The resulting model will lack reliability and robustness if the used dataset does not have enough samples or such samples have poor quality. On the other hand, if there is a considerably good amount of data, but the architecture under which the model is trained is not suitable to solve the problem, the results obtained

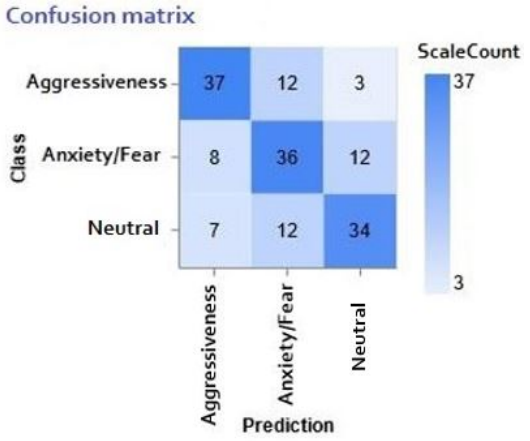

Fig. 8. 3-class MobileNet model confusion matrix

will not be satisfactory, so it is essential to improve both elements.

Thus, this research provides a method for classifying canine emotional behavior. The proposed approach differentiates from previous works in that they have mainly focused on detecting poses such as standing, sitting, or lying down. We aim to support the ethological study of canine behavior with this research work, providing computer science tools.

This research is limited to four basic emotional categories: aggressiveness, anxiety, fear, and neutral. It can get expanded to more categories or behaviors such as happiness, sadness, and discomfort, among other categories. However, it is limited to behaviors that have attitudes or actions visible to the human eye and can be captured or recorded with a camera. This research can be helpful as a basis for other researchers in the area who are interested in the subject and can improve or refine the proposed method to create more robust models or cover more behaviors and applications.

7 Conclusions

Given that dogs are an essential part of society today, their study and the development of new technologies that allow us to make better decisions regarding their care or help us better understand them are relevant.

This research provides a methodology with which it is possible to detect and classify dog behavior through artificial vision, machine learning, and transfer learning techniques. The classification of behaviors in this work is limited to aggression, anxiety, fear, and neutral being these four categories under which convolutional neural network models were trained, obtaining mixed results, opening the possibility of carrying out new technologies or more research in what to canine behavior respects combining computational models.

The methodology can be improved, and the model is not perfect; however, decent results were obtained that can serve as a basis to expand either the range of behaviors to detect, or on the contrary, focus on perfecting the detection of a single behavior according to the applications or uses that are required of a said computational model.

Acknowledgments

The first author of this paper thanks to *Consejo Nacional de Ciencia y Tecnología* for the master scholarship *(749915)*.

References

1. **Aloff, B. (2018).** Canine body language: a photographic guide. Dogwise Publishing.
2. **Amir, S., Zamansky, A., van der Linden, D. (2017).** K9-blyzer: Towards video-based automatic analysis of canine behavior. Proceedings of the Fourth International Conference on Animal-Computer Interaction, pp. 1–5.
3. **Barnard, S., Calderara, S., Pistocchi, S., Cucchiara, R., Podaliri-Vulpiani, M., Messori, S., Ferri, N. (2016).** Quick, accurate, smart: 3d computer vision technology helps assessing confined animals' behaviour. PloS one, Vol. 11, No. 7, pp. e0158748.
4. **Barros, P., Magg, S., Weber, C., Wermter, S. (2014).** A multichannel convolutional neural network for hand posture recognition. International Conference on Artificial Neural Networks, Springer, pp. 403–410.

5. **Bremhorst, A., Mills, D., Würbel, H., Riemer, S. (2021).** Evaluating the accuracy of facial expressions as emotion indicators across contexts in dogs. Animal Cognition, pp. 1–16.
6. **Campbell, E. J. (2016).** Owners' abilities to recognise and comprehend signs or displays of aggression in their canine companions outwith the home environment. Veterinary Nursing Journal, Vol. 31, No. 11, pp. 329–333.
7. **Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., Chen, A. (2020).** Teachable machine: Approachable web-based tool for exploring machine learning classification. Extended abstracts of the 2020 CHI conference on human factors in computing systems, pp. 1–8.
8. **DataSmarts (2018).** Aprovechando el conocimiento de otros. `https://datasmarts.net/es/aprovechando-el-conocimiento-de-otros/`.
9. **Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L. (2009).** Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition, Ieee, pp. 248–255.
10. **Doan, H.-G., Nguyen, V.-T., Vu, H., Tran, T.-H. (2016).** A combination of user-guide scheme and kernel descriptor on rgb-d data for robust and realtime hand posture recognition. Engineering Applications of Artificial Intelligence, Vol. 49, pp. 103–113.
11. **Ehsani, K., Bagherinezhad, H., Redmon, J., Mottaghi, R., Farhadi, A. (2018).** Who let the dogs out? modeling dog behavior from visual data. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4051–4060.
12. **Elforaici, M. E. A., Chaaraoui, I., Bouachir, W., Ouakrim, Y., Mezghani, N. (2018).** Posture recognition using an rgb-d camera: exploring 3d body modeling and deep learning approaches. 2018 IEEE life sciences conference (LSC), IEEE, pp. 69–72.
13. **Giuffrida, M. A., Brown, D. C., Ellenberg, S. S., Farrar, J. T. (2018).** Development and psychometric testing of the canine owner-reported quality of life questionnaire, an instrument designed to measure quality of life in dogs with cancer. Journal of the American Veterinary Medical Association, Vol. 252, No. 9, pp. 1073–1083.
14. **He, K., Zhang, X., Ren, S., Sun, J. (2016).** Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
15. **Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. (2017).** Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
16. **Jones, A. C., Gosling, S. D. (2005).** Temperament and personality in dogs (canis familiaris): A review and evaluation of past research. Applied Animal Behaviour Science, Vol. 95, No. 1-2, pp. 1–53.
17. **Lab, G. C. (2019).** Teachable machine, a fast, easy way to create machine learning models – no coding required. `https://experiments.withgoogle.com/teachable-machine`.
18. **Lowe, D. G. (1999).** Object recognition from local scale-invariant features. Proceedings of the seventh IEEE international conference on computer vision, volume 2, Ieee, pp. 1150–1157.
19. **Lowe, D. G. (2004).** Distinctive image features from scale-invariant keypoints. International journal of computer vision, Vol. 60, No. 2, pp. 91–110.
20. **Machine, T. (2017-2021).** Teachable machine, prepara a un ordenador para que reconozca tus imágenes, sonidos y posturas. `https://teachablemachine.withgoogle.com/`.
21. **Mealin, S., Domínguez, I. X., Roberts, D. L. (2016).** Semi-supervised classification of static canine postures using the microsoft kinect. Proceedings of the Third International Conference on Animal-Computer Interaction, pp. 1–4.
22. **Mealin, S., Howell, S., Roberts, D. L. (2016).** Towards unsupervised canine posture classification via depth shadow detection and infrared reconstruction for improved image segmentation accuracy. Conference on Biomimetic and Biohybrid Systems, Springer, pp. 155–166.
23. **Nasirahmadi, A., Sturm, B., Edwards, S., Jeppsson, K.-H., Olsson, A.-C., Müller, S., Hensel, O. (2019).** Deep learning and machine vision approaches for posture detection of individual pigs. Sensors, Vol. 19, No. 17, pp. 3738.
24. **Redmon, J. (2013–2016).** Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`.
25. **Redmon, J., Farhadi, A. (2018).** Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

26. Robinson, C., Mancini, C., van der Linden, J., Guest, C., Swanson, L. (2015). Exploring assistive technology for assistance dog owners in emergency situations. Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments, pp. 1–2.

27. Science, T. D. (2019). Deep learning using transfer learning-python code for resnet50. `https://towardsdatascience.com/deep-learning-using-transfer-learning-python\-code-for-resnet50-8acdfb3a2d38`.

28. Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

29. Sinha, D., El-Sharkawy, M. (2019). Thin mobilenet: An enhanced mobilenet architecture. 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE, pp. 0280–0285.

30. Tami, G., Gallagher, A. (2009). Description of the behaviour of domestic dog (canis familiaris) by experienced and inexperienced people. Applied Animal Behaviour Science, Vol. 120, No. 3-4, pp. 159–169.

31. Torrey, L., Shavlik, J. (2010). Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global, pp. 242–264.

32. Zoph, B., Cubuk, E. D., Ghiasi, G., Lin, T.-Y., Shlens, J., Le, Q. V. (2020). Learning data augmentation strategies for object detection. European Conference on Computer Vision, Springer, pp. 566–583.

Article received on 26/07/2021; accepted on 26/09/2021.
Corresponding author is Humberto Pérez-Espinosa.