Application to assist in the recognition and identification of lost pets using the Artificial Neural Networks technique

José P Marinho, Kalil B Rocha, Patrick L Teixeira, Victor A Barbosa, Espec Alexander Barreira

Journal of Software Engineering and Applications

OPEN ACCESS

Application to assist in the recognition and identification of lost pets using the Artificial Neural Networks technique

Orientador: Espec Alexander Barreira [2]

[1,2] **Faculdade São Paulo Tech School - São Paulo, SP, Brasil**

[1] ***Student's Bachelor's degree in Computer Science at the São Paulo Tech School, Brazil***
[2] ***Professor of Computer Science at the São Paulo Tech School, Brazil***

E-mail: secretariageral@sptech.shool

IdHal: hal-04483047, version 1

DOI: **10.5281/zenodo.10724842**

Accepted 28 Feb 2024 for publish Vol 1, 2024

Abstract

The number of abandoned pets in Brazil increased significantly between 2018 and 2020, with a 126% increase at the start of the pandemic (IPB, 2022). This study explores the emotional impact of abandonment on dogs and the happiness of reunions with their owners. Dogs suffer from abandonment, waiting for their owners to return (Bradshaw, 2016). This waiting period weakens them, as it breaks their routine and affects their self-care. Losing a pet can be compared to losing a loved one.

In Brazil, the process of reuniting pets with their owners is still precarious. However, when reunions do happen, the happiness is immense for both parties, improving physical and mental health. The connection at this moment transcends the senses of life. This work aims to implement a pattern recognition tool using Artificial Intelligence (AI) techniques to help identify lost pets. The tool will use image recognition and machine learning algorithms to match lost pets with their owners. The tool will be developed using open-source technologies and will be made available to the public free of charge. The expected benefits of the tool include: Increased chances of reuniting lost pets with their owners; Reduced stress and anxiety for both pets and owners; Increased awareness of the problem of pet abandonment and, Encouragement of pet adoption.

The development of this tool is a significant step towards addressing the problem of pet abandonment in Brazil. By leveraging the power of AI, we can help to create a more compassionate and responsible society for all animals.

Keywords: Pet abandonment, Reunion, Emotional impact, Happiness, Mental health, Connection, AI, Pattern recognition, Image recognition, Machine learning.

1. Introduction

In Brazil, the number of helpless pets increased sharply between 2018 and 2020. According to IPB (2022) - Instituto Pet Brazil, during the first survey carried out in 2018, the survey showed that animals in this situation reached 3.9 million, with a 126% increase at the start of the pandemic in 2020.

We can assume that some of these abandoned pets are lost from their owners, which is a great pain for both the animal and the owner. According to British biologist John Bradshaw (2016), dogs suffer from this certain "abandonment" in their lives, waiting for their owners to reappear. This wait, which often doesn't even happen, leaves the pet weak, as they create this expectation of a reunion and stop taking care of themselves, feeding themselves, not resting, becoming sad and anxious. All of this is because dogs are bound to a routine of living with and caring for their owners. They have their own time to rest, eat and play. When this routine is altered, it's difficult for them to adapt, as they realize that something isn't right.

This behavior comes from a genetic heritage, where thousands of years ago, dogs were domesticated by humans, and from then on, they began to see us as members of their packs. And that's precisely why the loss of an animal can be compared to the pain of losing a loved one in our families.

Pets give meaning to our lives. Over time, pet ownership has begun to change in Brazilian families. They are part of our lives, our routine, they occupy a space in our homes, we see many examples of pets being raised indoors, no longer in backyards, and they even sleep with their owners on their beds and sofas. Because of this, the pain of losing a pet, as much as the death of a pet or the loss of a pet, is a huge pain in the owners' lives.

In Brazil, we still don't have a process for reuniting pets with their owners - it's a delicate situation. But when fate makes this reunion happen, it is a moment of extreme happiness for pets and owners and improves the health of both. We can see that the connection of this moment goes beyond all the senses of life.

2. Rationale

Reuniting lost pets with their owners can be made easier by integrating Machine Learning and Pattern Recognition. By analyzing data and identifying patterns, it is possible to develop intelligent systems capable of identifying found animals that are similar to registered lost animals. This approach uses classification and clustering algorithms to establish matches and promote the happiness and health of lost animals, but it also provides us with a more efficient and comprehensive approach to dealing with the alarming increase in helpless animals, both pets and owners. It is important to point out that *Deep Learning* models or artificial intelligence models chosen inappropriately, such as Simple Linear Regression, K-Means, Naive-Bayes, Decision Trees, Deep Neural Networks, among others, could be harmful for a number of reasons, such as complexity, insufficient models for analyzing data, etc.

2.1 Pattern recognition

Nowadays, pattern recognition is widely used all over the world, including in the field of technology.

With lots of new data being generated all the time and the evolution of

pattern recognition has become a powerful tool for creating new products on the market and thus solving various problems. To understand this better, you must first understand what recognition is and what a pattern is.Recognition is an association, identification or understanding of something you already know,even if you don't remember it, a pattern will refer to some memory of yours and you will "know it again". An example of this is when you hear the voice of a well-known or famous person and already associate them with that figure.

A standard is a set of characteristics and behaviors that form a concept. According to the Michaelis dictionary (2023), a standard is an established model whose approval by general consensus or by official authority serves as a basis for comparison. Standards vary according to the need or purpose of the solution, i.e. if the purpose changes, so will thestandard.

So you could say that pattern recognition is a way of describing an object among a population, using the search for significant attributes and behaviors among the members of that population. This is how we'll be recognizing dogs and cats in this article's system, looking for significant characteristics that help identify a specific pet.

2.2 Machine Learning

The field of Machine Learning has played a significant role in solving complex problems ina number of areas, including data analysis and pattern recognition. Machine Learning consists of algorithms and techniques that allow a computer system to learn from data, identifying patterns and making decisions or performing ideate predictions based on these patterns. In the context of reuniting lost pets with their owners,Machine Learning could be a promising approach to help with this task.

By analyzing the data collected, such as the animals' physical and behavioral characteristics and records of lost and found animals, it is possible to develop models that identify patterns and establish correspondences between found pets and theirrespective owners.

2.2.1 Supervised Learning

Supervised machine learning consists of working with data that has already been labeled, predicted or classified as training for amodel, so that it can classify or predict new data that it has never seen before.

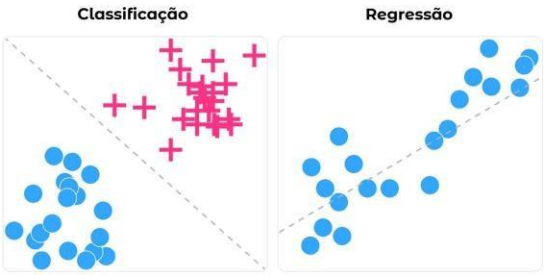


Figure 1: Example of classification and regression.

The examples that can be passed to the model can be continuous values, such as real numbers,or classificatory values, such as binary values. When continuous values are used as an input example for the model, it will try to predict a future value, for example: a set of attributes defining the value of a car insurance policy is passed in, along with the values that have already been established for each type of insurance. The model's task will be to predict, with the input of new attributes, what the insurance value will be for that type of insurance.

In specific case. This technique is called regression.

Now, if you use features that define whether a flower is a rose, daisy or sunflower as input data to train the model and then pass it a new chain of features, the model will predict the typeof flower based on the 3 types it has learned. This technique is called classification.

The graphical difference between these twotypes can be seen in figure 1.

2.2.2 Deep Learning

Deep learning is a technique for simulating how human intelligence works, synthesizing human neurons so that computer models can bedeveloped that are capable of learning, understanding and recognizing complex patterns on their own in various types of data such as images, sounds, texts, etc (Amazon-a).It uses several layers with

several neurons to dothis, which is why it is called deep learning, since with each layer, the algorithm is able to understand more complexity within the data.

Deep learning is made up of an input layer, one or more hidden layers and an output layer, in which the classification is made. For example, in the input layer you can pass an image of a cat, and so the first layer can identify some edges, colors, simpler things, while the next layers will learn more complex parts and characteristics of the animal, so in the end you can classify whether it's a cat or not. An example of how *deep learning* works is shown in figure 2.

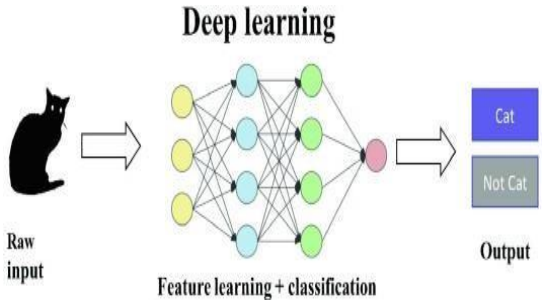


Figure 2: Example of deep learning.

Artificial Neural Networks

Artificial neural networks (ANN) are networksmade up of artificial neurons that try to imitate human biological neurons, sending signals to each other and keeping the learning for future use, thus acquiring the ability to learn and gaining knowledge.

An artificial neuron is made up of a set of synapses, where it is represented by the value of the synaptic weight, the sum function responsible for multiplying the input values and their respective weights and the activation function which will limit the output value so that learning between the layers of neurons is optimized.

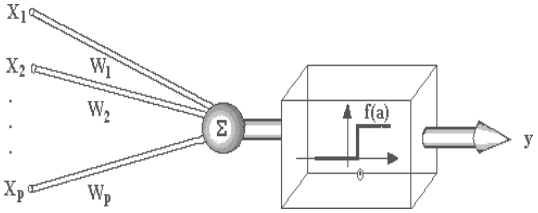


Figure 3: Example of an artificial neuron.

Figure 3 shows what makes up an artificial neuron. The x1, x2 and xn are the input values that are connected to the sum function (Σ). The link values (w1, w2, wn) represent the weight values of these links. The activation function, represented by f(a) and, finally, we have y which will be the output value of this neuron and, if it exists, will be the input value for a new neuron.

Each neuron has these components. In a neural network, there can be several layers with several neurons, all interconnected so that knowledge can be passed on.

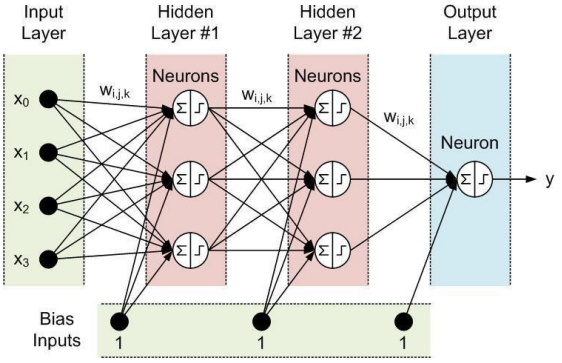


Figure 4: Example of a simple artificial neural network

Figure 4 shows the architecture of a simple ANN. It consists of an input layer, where the image data will be loaded into the model and connected to the next layer, which will be the first hidden layer. In the example, there are twohidden layers, but it is possible to have several hidden layers, which can generate intriguing pattern recognition features in the images, it alldepends on the most appropriate solution to theproblem. It should also be noted that in additionto the neuron connections, there are *Biases*, which are values that help with the values generated by the weighted sum and the output values of the activation functions. An ANN learns by passing the data several times throughthe layers of neurons. As these passes are made, the weights (w) are readjusted to help determine the importance of the variables generated by any neuron and to contribute to abetter model output result (IBM).

2.2.3 Convolutional Neural Networks

CNNs (Convolutional Neural Networks) are networks that have great accuracy in solving computational visualization problems in images. It all started when Hubel and Wiesel (2018) carried out an experiment in 1962 in which it was shown that neurons are activated when exposed to lines and edges, thus producing visual recognition. Matrices are used to input an RNC, which is the conversion of the image so that the computer can understand and process this data. Figure 5 shows that matrices can vary in size dependingon the color channel of the images, which are usually RGB, i.e. 3 color channels.

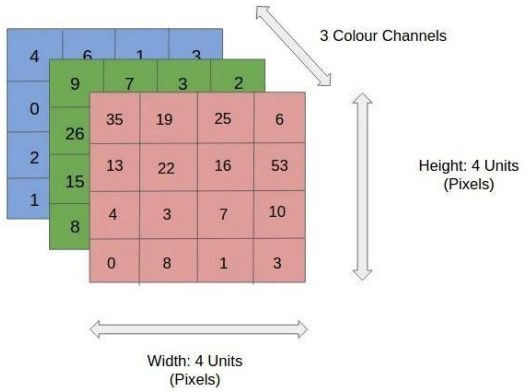
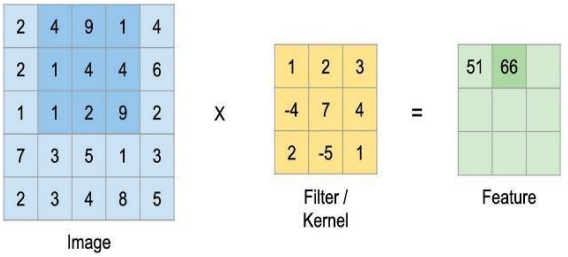


Figure 5: Example of a 3-dimensional matrix.

RNC's with convolution layers are basically filters passed through the matrix that does the feature extraction. These filters, usually called *kernels*, have defined sizes when instantiating the layers and they "slide" through the matrix, analyzing the pixels of each column and row and resulting in a feature extraction. This sliding process is called *stride*, in which it defines how many pixels the kernel will jump over as it walks through the matrix. At the end of this process, a feature map is generated, which is a new matrix with the features extracted by the kernels. Figure 6 shows an example of how this process is done.

Figure 6: Example of convolution on a matrix.



In the convolution process, *padding can be* defined. This attribute is used to add a border to the matrices, which can be filled with zeros or with pixel intensity similar to what is around the image. *Padding* allows the layers not to shrink too quickly and makes it possible to take better advantage of the details in the image, if there is a need not to lose these details.

Activation functions allow the system to learn any type of functionality, since it is applied to non-linearity to the data. There are various types of activation functions such as *sigmoid*, *tahn* and *softmax*, but the most widely used in convolution layers is *ReLu* (Medium-a).

The *pooling* process in a CNN is the technique for reducing the data in a matrix, thus simplifying learning in the dense layers at the end of the network.

As with the *kernels* and convolution process, we determine an area, usually made up of 2x2, to traverse the matrix. There are two types of techniques to be used here: the first is to select the largest value within the area being traversed and add it to the final matrix generated by the process. The second is to obtain the average of the values in the area covered. At the end of *pooling*, a new matrix is generated. In the 2x2 example, in a 32x32 matrix, when *pooling* is applied, the final matrix will be 16x16. An example can be seen in figure 7.

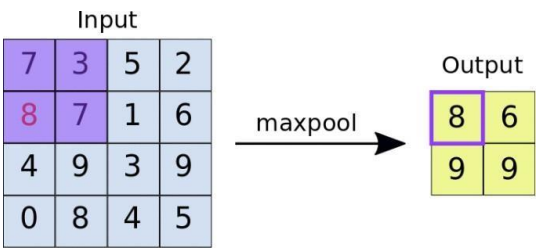


Figure 7: Example of *max pooling* in a matrix.

3. Related work

Facial recognition has been widely explored due to its relevance in various fields, from security to authentication in electronic devices. In the middle of this process is facial detection, initiallycarried out by algorithms such as Viola-Jones or, more recently, by Convolutional Neural Networks (CNNs). By identifying patterns of facial features, these methods establish the basis for subsequent analysis and identification.

However, uncontrolled challenges, such as variations in lighting and angle, present themselves as obstacles to be overcome. The robustness of facial recognition is tested by the ability to cope with different lighting conditions, since sudden variations can compromise the effectiveness of the process.

This is a critical aspect, especially in uncontrolled environments. The emergence of deep learning, notably through CNNs, has revolutionized the field of facial recognition. These convolutional neural architectures have the unique ability to learn hierarchical features, providing a more abstract representation that is invariant to smallvariations. This approach has proven particularly effective in mitigating the aforementioned challenges, standing out as a robust solution.

A notable example of a successful application is Apple's Face ID. This system incorporates not only advanced hardware, such as depth sensors, but also deep learning algorithms to create a three-dimensional model of the user's face. This approach demonstrates the synergy between hardware and software, providing a facial recognition solution that is not only secure, but also capable of operating in a variety of environmental conditions.

Although facial recognition has advanced significantly, it still faces complex challenges. The use of CNNs, together with strategies that consider three-dimensional aspects, illustrates the ongoing evolution of these systems.

4. Methodology

This work could only be completed after the studies and tests carried out in section 2 of this work and also with the use of the dataset of images collected, mentioned in section 3.1.

4.1 *Dataset* used

To complete this work, volunteers collected 4 photos of their pets, and if they had more than one pet, they sent the same number of photos of each pet. We collected 60 images of 15 different pets. These 60 images will serve as training for the neural network model, so that itlearns who these dogs are.
The name of each folder will serve as a class, which defines which dog is which. So, when the model is asked to recognize a pet in a new test photo, it will provide one of these classes, thus classifying which dog is in the test photo.

Figure 8: All the pets collected from volunteers.

To validate the model, detailed in section 5.2 of this paper, another image database was used, which was collected from public profiles on the internet.

In all, this database contains 143 images of 5 different dogs which were used to train and test the recognition and identificationmodels.

Figure 9: Set of images of dogs collected from public profiles on the internet.

4.2 Technologies used

A well-structured architecture is fundamental to the success of projects, especially in complex domains such as facial recognition. The structure of the project not only directly influences the efficiency and effectiveness of recognition, but also affects aspects such as maintenance, scalability and performance. From a time perspective, the continuous evolution of the architecture is crucial to keep up with the ever-changing demands of the technological environment.

In this context, cloud infrastructure, as offered by AWS (Amazon Web Services), plays a crucial role. AWS provides a comprehensive range of services and resources, allowing developers to build and deploy solutions in a scalable and efficient manner. This approach is especially advantageous for facial recognition projects, where the processing of large data sets and the need for robust computing resources are common.

Based on how beneficial a cloud infrastructure can be for the project, the decision was made to make it the main cloud for this project and throughout this section, it will be described which technologies were used within it.

4.2.1 Infrastructure

Starting with data entry, represented by figure 10, the user will use an application made in React Native, a framework developed by Meta, which helps efficiently and effectively in the development of multiplatform mobile applications. In this application, users will be able to register themselves, their pets and their photos.

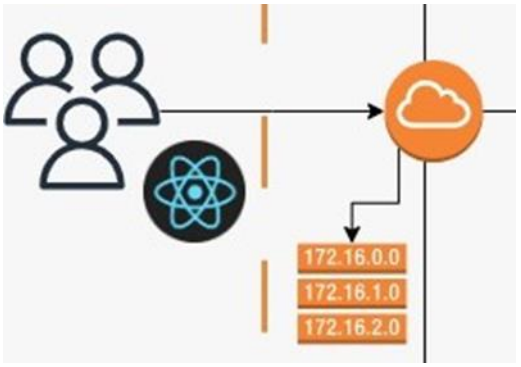


Figure 10: Part of the architecture showingthe project's data input.

This will be the data input in production, for the model to get to know and/or recognize pets by registering or searching for a lost pet through image recognition.

As can be seen in figure 11, before the application connects to the backend, a proxy application was developed using *GoLang* that allows all requests to be filtered, thus providing monitoring, security and access control.

Figure 11: Public subnet used in the project, containing an EC2.

When using AWS as an option for allocating cloud infrastructure, it is possible to access the following network services: VPC, in which an internal network is set up for hosting virtual machines containing containers with allocated applications. According to Amazon-b (2023), VPC offers total control over the virtual network environment, including resource placement, connectivity and security.

The *Public Subnet* is a network group open to the Internet so that the application can connect to the API to obtain data from the backend, where there is a proxy service that connects to the *Private Subnet*, where the backend is located.

Private Subnet contains a private network group where the machines with the recognizer, the backend and the database are hosted.

For the application to be able to connect to our model, an API developed in NestJS is used, which is hosted in the AWS cloud. This API provides information to the backend via user and pet registration. It also stores new photos for recognition. The service provides information to the application to visualize user and pet data.

MongoDB plays a key role in storing information securely, including user data, specific details of each pet in relation to its respective owner, and the results of pet recognition.

This non-relational database offers flexibility and scalability to efficiently manage information related to the system, guaranteeing robust and accessible storage.

By keeping user data, such as credentials and personal information, MongoDB contributes to the security and integrity of the information. In addition, by storing specific details of each pet in relation to its owner, an organized structure is created that makes it easier to access and retrieval of information related to pets and their owners.

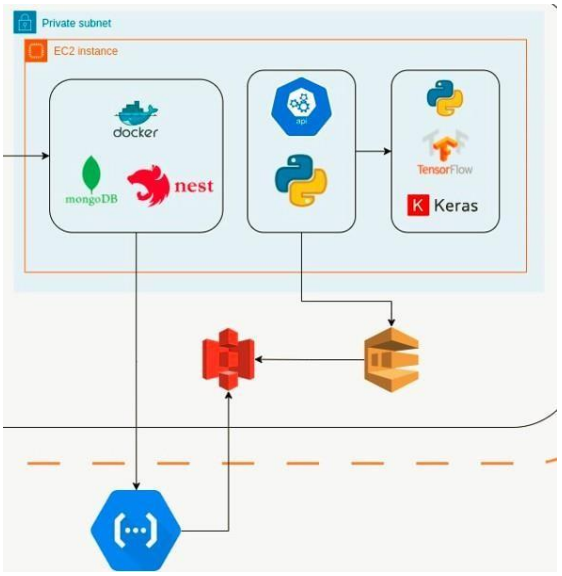


Figure 12: Private subnet where the backend, storage and deep learning models are concentrated.

MongoDB's ability to handle unstructured data and support for complex queries make it a solid choice for storing animal recognition results. This provides effective management of this information, enabling detailed analysis and continuous improvement of the recognition system. As seen in figure 12, Google Function is used as an alternative to AWS Lambda, playing the role of a lambda function. Whenever an attempt is made to save an imagein Amazon S3, a request is made to Google Function. In this process, the function is createdand tasked with collecting essential information, such as the name of the documentand its size. When this stage is completed, the data is then saved on Amazon S3.

This approach allows for efficient integration between Google Function and Amazon S3, guaranteeing proper execution of the operations required for storing images.

Previously mentioned, S3 is the storage of files and especially images of pets, separated by directories that correspond to each registered pet. Amazon S3 plays the essential role of storing files, especially images of pets. These images are organized into directories

corresponding to each individual registered pet.

The integration between S3 and the Simple Queue Service (SQS) is crucial for operational efficiency. When the need arises to recognize a pet, an event is automatically triggered in SQS via S3. The messages, formatted in *JSON*, are queued, containing information such as the directory in S3 where the pet's image was stored. To process these messages, we use a *Python* API which, on receiving the SQS message, extracts the directory, downloads the corresponding image and then invokes the recognition algorithm.

As for the infrastructure, Amazon EC2 acts as a virtual machine hosting several layers of the application, including the applications themselves, the database and the recognition model. This approach provides a consolidated solution for the operations required, guaranteeing scalability and adequate system performance. The Figure 13 shows the architecture of the cloud infrastructure as a whole.

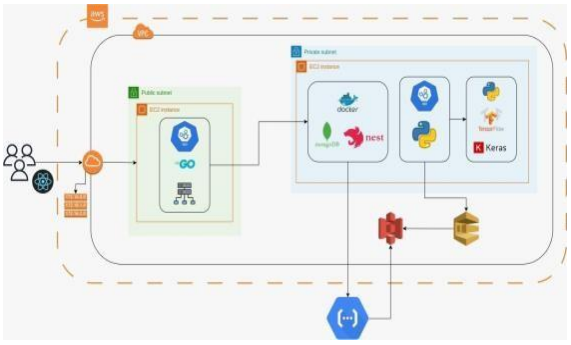


Figure 13: Cloud infrastructure architecture.

4.3 Image Recognition

An **End To End** architecture was used for image recognition. End To End architecture (also known as E2E) is a type of system or application design in which the entire processing flow, from data input to results output, is handled in a continuous and integrated manner. In this way, it is possible to follow the flow of image recognition, a flow very similar to that of PIX. As soon as the APP calls the functionality to recognize the **Pet**, an ***EndToEnd*** is generated which is made up of 3 parts:

- Indicator of what will be used, in thiscase: **REC** for Recognize;
- 5 random characters
- Date Time Place Concatenated, e.g. 2509232345

So it looks something like this:

RECnnij2509232345

This technique makes it possible to analyze theresult of the image recognition, so that at the *end* of the flow, when it queries this *endToEnd*, it will return the result of the recognition provided by the neural network.

5. Animal Recognizer

During the development of the project, following research, the decision was made to use a pre-trained model to perform unique animal recognition.
The model in question is the RNC called VGG16. This detection and classification network has one of the most popular algorithms for classifying more than 1000 images from 1000 classes, with an accuracy of92.72% (Medium-b).

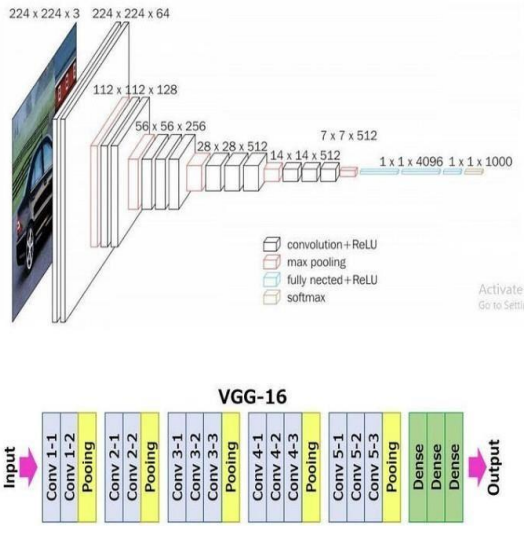


Figure 14: VGG16 network architecture.

Designed by the Visual Geometry Group (VGG) team at Oxford University, VGG16 is notable for its deep architecture with deep convolutional layers and small filters (3x3), which make it effective at extracting complex features from images.

VGG16 is made up of 16 layers, including 13 convolutional layers, 5 *pooling layers* and 3 fully connected layers. As input, the network requires images to have a size of 224x224 and with 3 color channels.

Figure 14 shows that there is a sequence of two convolutional layers and a *pooling* layer. The reason for having this convolution sequence is because it uses a 3x3 filter area, which combined with two convolution layers, manages to cover a 5x5 filter area, but using two non-linear functions (ReLu) in sequence, obtaining the most discriminating decision capacity.

The network convolution in all layers uses a 3x3 *kernel* and a *stride* of 1. The convolution activation function uses the *ReLu function.* This function is used because it is more computationally efficient, as it results in fast learning and possibly reduces the problems of disappearing gradients (Lima Collection). The *pooling* layers use 2x2 filters with a *stride of* 2.

The number of filters in the convolution layers increases proportionally as the data is passed through the network. The first convolution uses 64 filters, the second 128 filters, the third 256 filters and the fourth andfifth 512 filters.

As for the fully connected layers, the first two use 4096 neuron units each and *ReLu* is also used as the activation function. The third layer, on the other hand, uses 1000 neuron units, as this network was used to classify thisnumber of objects. The activation function used is *softmax,* which is the mostrecommended for the final layers, when the network will actually do the classification.

Convolutional layers help to extract relevant features of the images, while thefully

connected layers perform the final classification (Simonyan & Zisserman).

The network has approximately 183 million parameters, making it a relatively largenetwork compared to other architectures. This large number of parameters allows the network to learn different and detailed representations of images, thus achieving high accuracy. This is a reflection of the network's ability to learn discriminative representations of objects in complex images.

During the development and training of the model, unexpected results were obtained, which will be discussed in more detail in section 4.2 of this paper. For this reason, a new RNC architecture was created using the classes and methods provided by Keras, with the aim of achieving greater performance than a pre-trained network.

Based on VGG 16 and its concept of using a 3x3 kernel to cover larger feature areas, three convolution layers were used, containing 64,128 and 256 filters respectively, and between them two 2x2 *MaxPooling* layers.

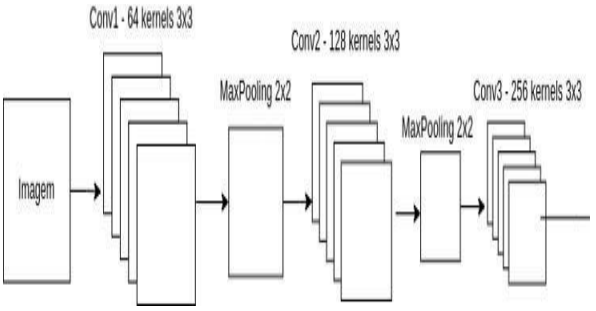


Figure 15: Convolution layers and *pooling* architecture of the new model.

In the dense layers, 3 layers are used, each containing 256, 128 and 64 neuron units respectively. Finally, we added the classificationlayer, which contains the same number of neurons for each class in our dataset.

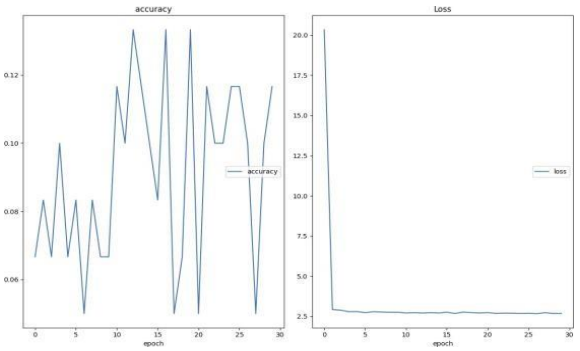


Figure 16: Dense layers of the new model's architecture.

5.1 Model training

To train the model, a local environment was used, i.e. a machine containing an Nvidia GTX 1650 GPU and using the VSCode IDE integrated with Jupyter Notebook.
The *TensorFlow framework* was also used to develop the model, which has several essential tools and integration with the Keras API for efficient training of neural network models. As already mentioned, the VGG16 network requires the input images to be 224x244 and contain 3 color channels, in this case, the images must be RGB (Red, Blue and Green), with pixels varying between 0 and 255.

To get a more assertive image, the Keras *ImageDataGenerator* class is used. It allows you to increase the number of images in the dataset by generating new images at different angles. It is also possible to modify thesenewly generated images by distorting their height and width.

For the actual training of the model, the *transfer* learning technique was used. and its weights, acquired from one problem, tosolve a new problem (Keras). This makes it possible to reduce training time and use the techniques already developed to solve a new problem.

Using the VGG16 network, provided by the Keras library, the network and its convolution and *pooling* layers are obtained. Then, to adapt the model to the problem, fully connected layers were added to the end of the architecture. After obtaining the results of the pre-trained model in relation to the dataset, the model itself, described in section 4, was trained. The same pre-processing of the images for VGG 16 in the model itself.

5.2 Results

The *EarlyStopping* and *Model Checkpoint* classes were used to train the pre-trained model, provided by Keras itself. With these two classes it is possible to monitor the model's training and save the best weights generated during training. Using *EarlyStopping*, we defined the metric to be monitored: in the case of training the models in this work, we monitored the *loss*, which is the error rate that the model can have in predictions. It was also decided that if this measure did not decrease over 10 epochs, the training would automatically be interrupted, as it would not be viable to spend processing on training that was not performing as well as we would like.

Together with *ModelCheckpoint*, itis possible to save the best weights based on measures that are performing better. Again, the*loss* rate was used as a measure to save the bestweights, i.e. the lower the *loss*, the better the model can perform for future predictions, thus saving the weights from the time of the lowest computed value of this measure.

By training the model using the dataset specified in section 4 and saving the best training weights, it achieves a maximum accuracy of 13% and a minimum *loss of* 2,671.

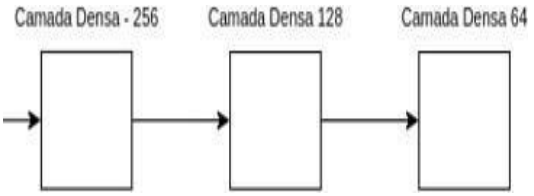


Figure 17: Accuracy and *loss of* the model pre-trained with the dataset from section 4.

The Figure 18 shows an example of the model's prediction. When the model passed the image of one of the dogs, it made a mistake and predicted a cat.

Figure 18: Prediction of the pre-trained model.

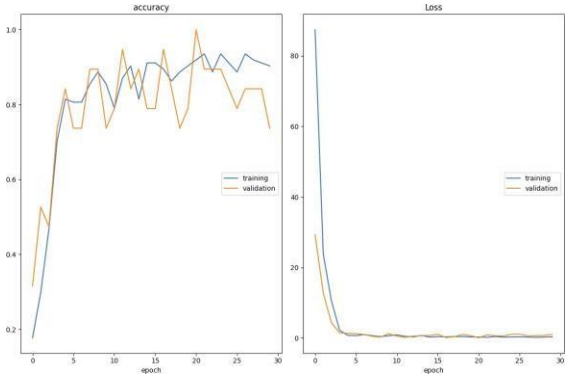
Looking more closely at t h e probabilities provided by the model, we can see that they are very low for any class of pet, with the highest probability, even if wrong, being no more than 10%. See figure 19.

```
1/1 [==============================] - 0s 126ms/step
cat1
[[0.09100069 0.07446556 0.07188999 0.05309477 0.05341394 0.07825414
  0.0640643  0.07878179 0.085691   0.05598125 0.06961314 0.06249922
  0.05108512 0.06909426 0.0410708 ]]
```

Figure 19: Probability of the 15 classes for the petimage passed to the model trained with our dataset.

After a few tests with other parameters, the results were similar.
The idea was proposed to use more than 4 photos of a single pet and re-train the model, to see if it is possible for a neural network to distinguish pets from each other. Thus, the second image base, described in section 4.1, was used to re-train the model.

As there are more images than the original dataset, it was possible to divide the images into training and validation and observe a little better how the model learns during the process.




```
1/1 [==============================] - 0s 126ms/step
cat1
[[0.09100069 0.07446556 0.07188999 0.05309477 0.05341394 0.07825414
  0.0640643  0.07878179 0.085691   0.05598125 0.06961314 0.06249922
  0.05108512 0.06909426 0.0410708 ]]
```

Figure 20: Performance of the model trained with more images.

In the end, looking at the model saved with the lowest *loss*, 0.063, gave an accuracy of 93%. Looking at figure 21, the probabilities that this model returns are still not enough for it to be sure which dog is which. In one test, in which it gets the prediction of a dog right, it offers only 34% certainty.

```
1/1 [==============================] - 1s 1s/step
heidi_pitbull
[[0.18050623 0.3451435  0.11151943 0.14932342 0.2135074 ]]
```

Figure 21: Probability of the 5 classes for the predicted pet

For academic research purposes, it was decided to create a model from scratch, as described in section 5, and by training it with the project dataset, a better performance than the pre-trained network w a s achieved, with a maximum accuracy of 61% and a minimum *loss of* 1.303.

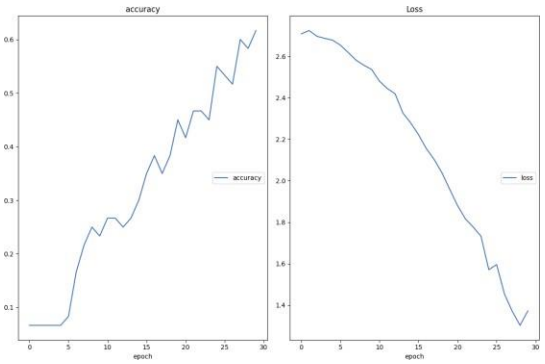


Figure 22: Performance of the model created forthe project.

By making a prediction with this model, passing the same image tested in the pre-trained model, it was possible to have the correct pet passed for recognition.

Figure 23: Prediction of the model createdfor the project.

In order to better evaluate the model, it was also trained with the 5 dogs chosen, which, as already mentioned, have more images than the dataset originally proposed for the project.

Then, when training, the model underperformed
pre-trained with the same dataset, reaching a maximum accuracy of 47% and a minimum *loss of* 1,322, as seen in the graphs in figure 24.

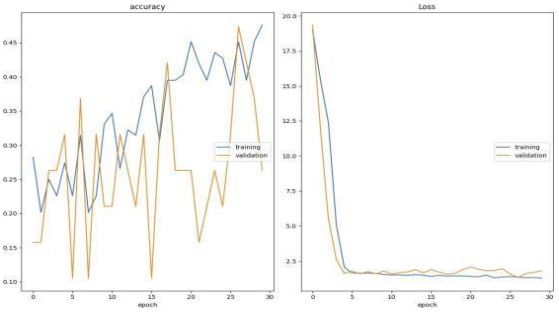


Figure 24: Performance of the model created for the project with more images.

When making predictions with it, there were no hits, and you can see an example of a prediction in figure 25.

Figure 25: Prediction of the model created for the project with the dataset made up of more images.

6. Application developed

Called "Achei o bicho" (I found the pet), its main objective is to register pets by their guardians, using images from four different positions and recognizing them using the mobile device's rear camera. The application requires user authentication so that all their pets and their data are stored securely in our database, as mentioned above in the project architecture. It will be necessary for the user of the app to grant permissions to use the device's camera so that the main functionality comes into play (pet recognition via the mobile device's camera); this will be one of the first steps when opening the app for the first time. Once permissions have beengranted, the user will be presented with the login screen, shown in figure 26.

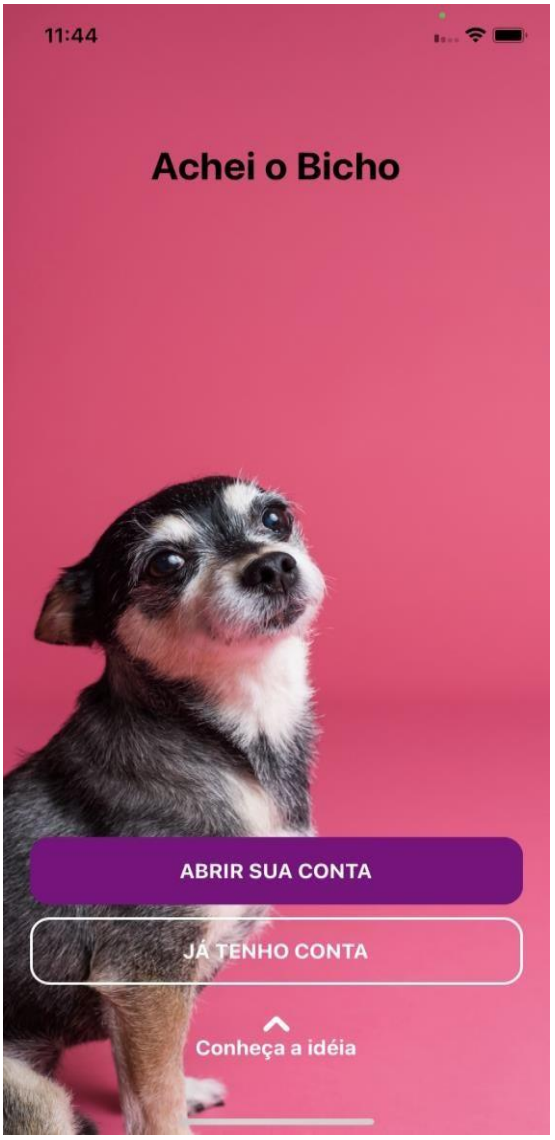


Figure 26: Login screen.

In the "Get to know the idea" area, the objective is briefly explained in text format. In "Open your account", through a short intuitive questionnaire in the chatbot model, you can create your account to access the application.

By accessing "I already have an account" you can connect to your account registered with us, entering your email address and password, and then enjoy all the features of the application.Once you have successfully authenticated, you will be redirected to the internal part.

The "My Pets" tab shows all the pets registered by their owner. They will be listed with their basic information and there is also an option to register a new pet, which will ask for data such as name, date of birth, gender (Male or Female) and description (optional). And finally, photos of the pet that follow a requested pattern, as shown in Figure 27.

Figure 27 User interaction

The user profile screen displays basic accountinformation, along with the option to interrupt your application authentication.

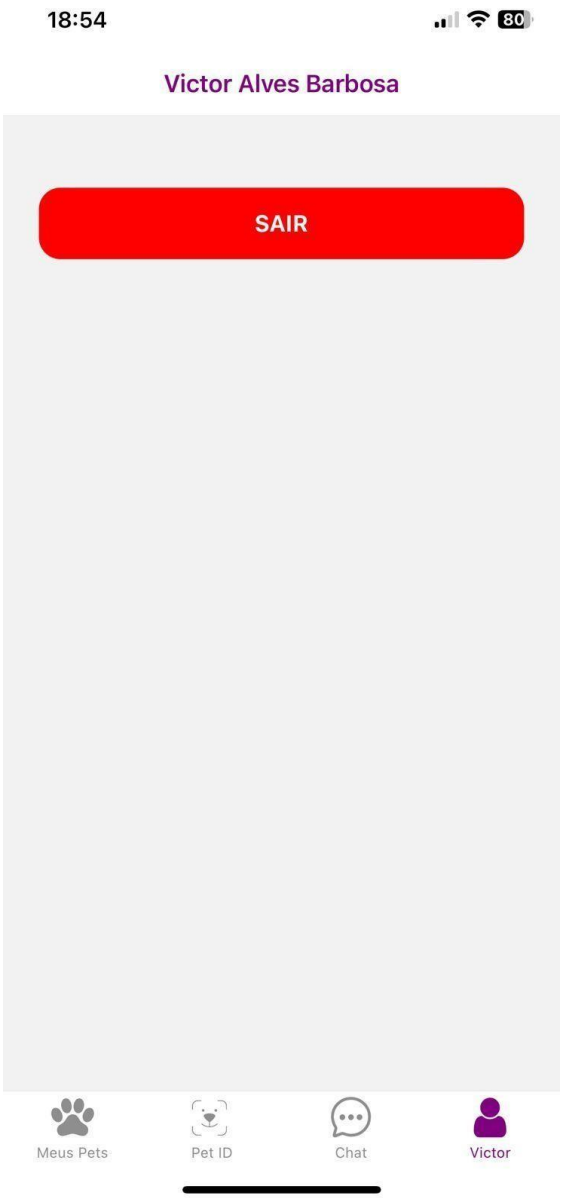


Figure 28: User profile tab

6.1 Recognizer in the application

By clicking on "Pet ID", with the camera permission granted, a screen with the current device's camera preview will be displayed, waiting for the user to capture a photo of a cat or dog.
Once the animal's photo has been captured, the file will be sent to the recognizer and the user will be redirected to a screen containing a list of pets that may be the pet that was attempted to be recognized.

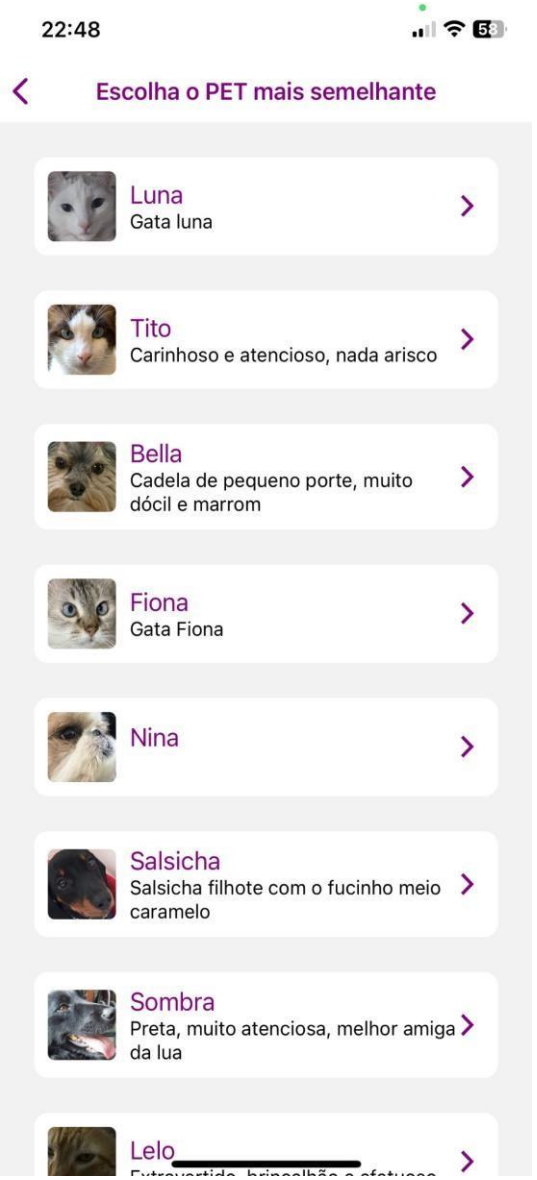


Figure 29: Screen with the list of pets identified as most likely to be the lost pet.

Clicking on the pet chosen by the user will open a screen containing photos, name, gender and description. This information helps the user to identify whether it matches the pet recognized by our application. If the user doing the recognition decides that the pet passed by the recognizer and the pet chosen in the list are the same, it is possible to contact the pet's owner using the "Talk to owner" button and a screen will open where messages can be exchanged.

between users, in order to be sure that the pet has really been found and to be able to return it to its owner.

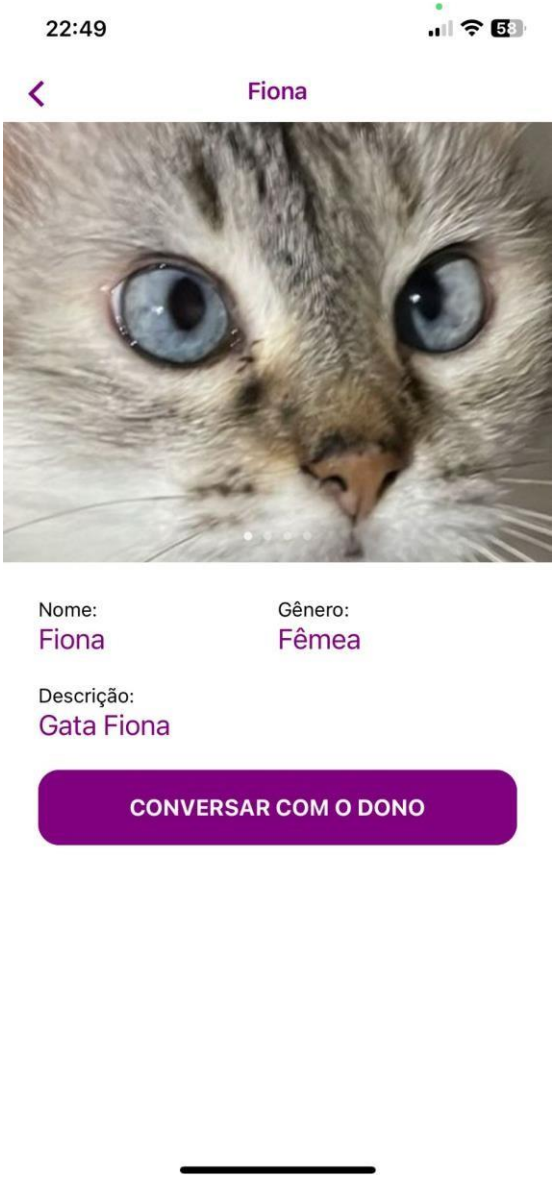


Figure 30: Screen with information and photos of the pet with a button to contact the owner.

6.2 Implementation features

The app was developed using React Native technology, along with the Expo tool to make it easier to use the device's native features. The main reason for using this framework is the possibility of cross-platform development,with our focus being solely on mobile devices(Android and iOS operating systems).

7. Conclusion

This work was created for a noble cause that moves many people and NGOs when dealing with the loss of domestic animals, especially dogs and cats. With this in mind, an application was proposed that can help in these situations by being able to recognize dogs and cats distinctly using an Artificial Neural Network. Throughout the research described in section 1, there are thousands of cases of animal loss and the difficulty of reuniting pets with their respective owners. Studies on Artificial Neural Networks and Convolutional Neural Networks were essential to realizing a solution that could solve this problem for society.

Looking at the results described in section 5.2,it is possible to say that this project could be a viable solution. Looking at the results of the model that performed best with the dataset proposed for the project, although it doesn't have a high degree of certainty, it is capable of correctly discriminating the recognition of a pet, and if it doesn't, among the 8 pets with themost possibilities, the recognized pet will be among them. Even so, the accuracy values of the proposed model are not acceptable for the application to be included in virtual *smartphone* stores, but there are points to cover: (i) the model pre-trained with the VGG16 network had an accuracy of 93% when trained with more images; (ii) if more than 4 images of each pet were requested, would it be possible for the model to offer greater possibilities?(iii) Is it possible to create an RNC in which you can delve more deeply intothe characteristics of the pets, such as the distance from the eyes to the muzzle, or even use the dog's muzzle as a recognizer?

This work does not contain such a large image base, which limits the performance of *Deep Learning* models trying to make however, it still has room for improvement ifit acquired more images for training and validation, used the video option and used *frames*, thus increasing the number of images acquired even further. Consequently, with more images, the infrastructure could be scaled up and the hardware increased to train more robust models.

Another implementation, which could be added in the future, is the comparison of the lost pet, which was sent for recognition, with the list of identified pets returned by the recognizer. In other words, after having the list of possible pets identified, a new comparison could be made within this set, going further by being able to make a one-to-one comparison of each pet within the set, thus increasing assertiveness even more, sincethe number of classes of the total number of pets is reduced to just two classes, which are being compared. This removes the user's responsibility for making the final recognition decision and leaves it solely to the recognition models.

An example of a solution that makes one-to- one comparisons of patterns are biometric systems, both facial and fingerprint. This is astrategy that can be adopted after acquiring larger and more robust computer processing. In conclusion, the objective of developing an application that can recognize a domestic animal from among others has been partially achieved, thus creating a classifier from a set of data extracted from images of pets. This isa classifier which does not yet have such assertive discriminatory power but which comes close to what is expected.

References

Lima Collection. VGG16 | CNN modelAvailable at: https://acervolima.com/vgg-16-modelo-cnn/

Amazon-a (2023). What is deep learning? Available at: https://aws.amazon.com/pt/what-is/deep-learni ng/

Amazon-b (2023). Amazon Virtual Private Cloud. Available at: https://aws.amazon.com/pt/vpc/

Bruce Palácio. Profile used in the bank of images collected from internet profiles. Images available at: https://www.instagram.com/doog.brucee/?utm_source=ig_web_button_share_sheet&igshid=OGQ5ZDc2ODk2ZA==

Facial Recognition using Deep Learning CNN in Python (2021). Available at: https://thinkingneuron.com/face-recognition-usi ng-deep-learning-cnn-in-python/

Heidi. Profile used in the bank of imagescollected from internet profiles. Images available at: https://www.instagram.com/heidi_pitbull/?utm_source=ig_web_button_share_sheet&igshid=OGQ5ZDc2ODk2ZA==

Hubel and Wiesel (2018). The NeuroscientificBasis for Convolutional Networks. Available at: https://medium.com/future-technologies-lab/th e-neuroscientific-basis-for-convolutional- netwo rks-3aa995919f5a

IBM (2023). What are neural networks? https://www.ibm.com/br-pt/topics/neural- netwo rks

IPB (2022). Number of pets in vulnerablesituations more than doubles in two years, saysIPB survey. Available at: http://institutopetbrasil.com/fique-por- dentro/nu number-of-pets-in-vulnerable-points-out-ipb-research/

John Bradshaw (2016). Find out why, evenwhen abandoned, dogs wait for their owners. Available at: https://veja.abril.com.br/ciencia/saiba-por- que-even-abandoned-caes-wait-for-their-owners s/

Keras (2020). Transfer Learning & Fine- tuning. Available at: https://keras.io/guides/transfer_learning/

Geron, Aurélio. Hands on: Machine Learning with Scikit-Learn, Keras & Tensor Flow. 2nd Edition. Alta Books: September 1, 2021.

Leona. Profile used in the bank of images collected from internet profiles. Images available at: https://www.instagram.com/leona.pet.princesa/ ?utm_source=ig_web_button_share_sheet&igshid=OGQ5ZDc2ODk2ZA==

Maria Eduarda Golden Retriever. Profile used in the image bankcollected from internet profiles. Images available at: https://www.instagram.com/madu_retriever/? utm_source=ig_web_button_share_sheet&igshid=OGQ5ZDc2ODk2ZA==

Medium-a (2018). Understanding Convolutional Networks (CNNs). Available at: https://medium.com/neuronio-br/entendendo- r andes-convolutional-cnns-d10359f21184

Medium-b (2021). Everything you need to know about vgg16. Available at: https://medium.com/@mygreatlearning/everyt h ing-you-need-to-know-about-vgg16- 7315defb5 918

Michaelis (2023). Dictionary definition of "Standard". Available at: https://michaelis.uol.com.br/busca?r=0&f=0& t= 0&word=standard

PIX Starter Standards Manual. https://www.bcb.gov.br/content/estabilidadefina