Abstract —Automatic image-based recognition systems have been widely used to solve different computer vision tasks. In particular, animals' identification in farms is a research field of interest for computer vision and the agriculture community. It is then necessary to develop robust and precise algorithms to support detection, recognition, and monitoring tasks to enhance farm management. Traditionally, deep learning approaches have been proposed to solve image-based detection tasks. Nonetheless, databases holding many instances are required to achieve competitive performances, not to mention the hyperparameter tuning, the noisy images, and the low-resolution issues. In this paper, we propose a transfer learning approach for image-based animal recognition. We enhance a pre-trained Convolutional Neural Network based on the ResNet101 model for animal classification from noisy and low-quality images, holding few samples. First, a dog vs. cat task is tested from the well-known CIFAR database. Further, a cow vs. no cow database is built to test our transfer learning approach. The achieved results show competitive classification performance using different types of architectures compared to state-of-the-art methodologies.

Index Terms —Animal recognition, Computer vision, deep learning, image processing, transfer learning.

Introduction

In recent years, advances in deep learning have encouraged the generation of novelty learning algorithms based on Convolutional Neural Networks—(CNN) to solve a wide variety of problems focusing on computer vision tasks such as visual tracking, segmentation, and image classification [1–4]. Indeed, relevant applications such as smart farming and medicine require robust computer vision systems to support diagnosis and monitoring tasks [5–8].

However, the amount of data, the storage, and the long training time are some limitations that CNN-based models provide. Namely, the volume of information is directly related to the overfitting and performance decaying when evaluating new sample sets [9,10]. Several approaches solve the overfitting issues by creating synthetic samples, penalizing the loss function, and regularizing the architectures [12–13]. Still, many samples are required, not to mention the hyperparameter tuning drawbacks [14,15].

Transfer learning has recently emerged as an alternative to reuse pre-trained models (on large databases) to improve a specific task's performance and robustness [16–18]. Such a strategy aims to exploit the ability to directly assign the knowledge acquired by a network model to solve similar problems as an alternative to lead machine learning problems.
image-based recognition, from small datasets. Notwithstanding, the main problem relies on identifying an effective use of this approach by enhancing a pre-trained network's parameters based on the new instances [19]. In particular, for image-based object recognition problems, some CNN-based approaches based on pre-trained architectures are proposed. AlexNet, GoogleNet, VGG-16, and ResNet-based networks are commonly applied on the well-known CIFAR and ImageNet datasets. Nonetheless, low accuracy is obtained, and complex hyperparameter tuning is required, besides the low resolution and noisy data challenges [21,22].

Here, we propose an image-based animal recognition approach based on transfer learning. We aim to classify cat vs. dog (from the well-known CIFAR database) and cow vs. no-cow (from a custom-built database) for concrete testing. In short, our strategy aims to: i) attain competitive accuracy results regarding image-based animal recognition tasks, even for small datasets, ii) properly tune the required hyperparameters, and iii) deal with the low resolution and noisy sample issues. In this sense, our approach employed the ResNet101 architecture and compared the achieved performance against other well-known state-of-the-art architectures, e.g., GoogleNet-2014, Vgg16, and ResNet50. Achieved results demonstrate how ResNet101 coupled with transfer learning favors the discrimination of images in small datasets. Besides, obtained performances show that the transfer learning method is more effective in classifying pixelated images. Our proposal could be an alternative to support computer vision tasks, i.e., medical image processing and intelligent farming systems, from databases with few samples.

The rest of the manuscript is organized as follows. Sections II and III presents the methods. Sections IV and V describe the samples. The training of a Deep Learning II and III presents the methods. Sections IV and V describe the samples. and intelligent farming systems, from databases with few

## II. IMAGE-BASED RECOGNITION USING DEEP LEARNING

Let \( \{X_i \in R^{W \times H \times C}, y_i \in [N]\}_{i=1}^I \) be an input image set, holding \( I \) samples in \( C \) channels, sizing \( WH \) pixels, and equipped with output labels \( y_i \). The training of a Deep Learning-based image recognition model is twofold: feature mapping mapping based on convolutional neural networks-(CNN) and multilayer perceptron-classification.

The first stage exploits the local spatial correlation of input images through a convolutional filter arrangement \( \{K_{n,l}\}_{l=1}^{N_l} \), for which a square-shaped layer kernel, sizing \( P \times P \times C \), explores the spatial relationships between pixels. Note that the number of kernels depends on the number of layers, \( n \in \{1,2,\ldots,N_l\} \). Thus, the convolutional operation projects stepwise a given image \( X \), as follows:

\[
\hat{X}_l = (\varphi_{l} \circ \cdots \circ \varphi_1)(X), \tag{1}
\]

where:

\[
\varphi_i(\hat{X}_{l-1}) = y_i(K_{n,l} \otimes \hat{X}_{l-1} + B_{n,l}). \tag{2}
\]

The convolutional layer in Eq. (2) holds the nonlinear activation \( y_i: R^{W \times H \times C} \rightarrow R^{W \times H \times C} \), \( \hat{X}_l \in R^{W \times H \times C} \) is the \( l \)-th CNN feature map (being \( \hat{X}_0 = X \)), and \( B_{n,l} \in R^{W \times H \times C} \) is the bias matrix (\( l \in \{0,1,\ldots,L\} \)). Notations \( \otimes \) and \( \circ \) stand for convolution operator and function composition, respectively. Besides, \( W_l < W \) and \( H_l < H \). Overall, the feature maps allow extracting relevant patterns concerning the spatial relationships among pixels.

In turn, a multi-layer perceptron-based classifier is applied on the \( L \)-CNN-based feature map, yielding:

\[
\hat{y} = (\varphi_D \circ \cdots \circ \varphi_1)(u_0), \tag{3}
\]

where \( \varphi_d(u_{d-1}) = \eta_d(A_d u_{d-1} + \alpha_d) \) is a dense layer ruled by the non-linear activation function \( \eta_d: R^{d} \rightarrow R^{d'}, d' \in N \) is the number of neurons at the \( d \)-th layer, \( d \in \{0,\ldots,D\}, \) \( (d = 0 \) is the initial concatenation before the classification layer). \( A_d \in R^{d \times d' - 1} \) is a weighting matrix that contains all connection weights between the preceding neurons, \( \alpha_d \in R^{d'} \) is a bias vector, and \( u_d \in R^{d'} \) is the \( d \)-th hidden layer vector that is iteratively updated as \( u_d = \Phi(u_{d-1}) \), from the input flattened vector \( u_0 = [\text{vec}(\hat{X}_L)] \), sizing \( \xi = W' H' \sum_{i} N_i \), with \( W' < W, H' < H \), after concatenating all matrix rows across the \( C \) color channels.

Note that the Deep Learning-based image recognition model estimates the predicted labels under the optimized trainable parameters \( \theta = \{K_{n,l}, A_d, b_{n,l}, \alpha_d\} \), as seen in Eqs. (1-3), which are minimized in terms of the output labels as follows:

\[
\theta^* = \arg \min_{\{K_{n,l}, A_d, b_{n,l}, \alpha_d\}} (L(\hat{y}_i, y_i | \theta); \forall i \in I), \tag{4}
\]

where \( L: N \times N \rightarrow R \) is a given loss function, i.e., mean square error or cross-entropy, that is solved through a mini-batch based gradient descend procedure using back-propagation and automatic differentiation [22,23].

## III. IMAGE-BASED ANIMAL RECOGNITION USING TRANSFER LEARNING

Transfer learning aims to reuse trained models across similar tasks. It is recently a popular approach in deep methods, where pre-trained models on large databases are used as the starting point on similar problems. In particular, such a process repeats the architecture, and fixing the weights of low-level layers, demonstrates exciting results for computer vision and natural language processing tasks. Concerning the image-based recognition problem, the most common CNN-based architectures include the VGG-16, the GoogleNet-2014, and the ResNet. Namely, the VGG-16 holds a low-complex architecture compared to the remaining networks since it contains a relatively small number of sequential convolutional layers. The GoogleNet-2014, also known as InceptionVn, where \( n \) refers to the Google updated version, presents a startup module that acts as extractors of multi-level features. At last, the ResNet, with its two versions ResNet50 and ResNet101, contains a more in-depth architecture than VGG-16 and
GoogleNet-2014, combining convolutional layers with residual modules. Table I summarizes the essential properties of the networks above for image-based recognition.

This study uses the pre-trained parameters of the networks exposed in Table I, obtained from the reduced version of the ImageNet collection (including 1000 categories, each with approximately 1000 images), as a common reference point for evaluating large-scale image classification models [4]. The proposed methodology consists of using the learned parameter from the ImageNet set to initialize the parameters of a particular network. Namely, we fix the lower layers parameters to initialize the feature mapping generation stage. Then, we solve the following optimization problem:

$$\theta^* = \arg \min_{\{K_{n,l}, A_d, b_n, a_d\}} \{L(\hat{y}_i, y_i(\theta); \forall i \in I)\},$$

(5)

where $\theta^* = \{K_{n,l}\} \cup \{K_{n,l}, A_d, b_n, a_d\}$, holds the fixed low-level CNN kernels $\{K_{n,l}\}$, the high-level CNN kernels $\{K_{n,l}\}$ ($l' < l'$); and the multi-layer perceptron parameters (fully connected layers) $\{A_d, b_n, a_d\}$. It is worth mentioning that the transfer learning is encoded in the low-level CNN kernels, which are trained for each architecture in Table I from the ImageNet dataset. Meanwhile, the remaining parameters (high-level CNN kernels and multi-layer perceptron variables) must be optimized on the animal image-based recognition dataset of interest through Eq. (5). Therefore, the recognition model will not have to learn from scratch all the low-level structures in most pictures; it will only have to know the higher-level structures. Additionally, it will not only speed up training considerably but will also require much less training data. Finally, note that the output layer must be updated concerning the number of considered classes.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>DEEP LEARNING MODELS FOR IMAGE-BASED RECOGNITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Trained layers</td>
</tr>
<tr>
<td>VGG-16</td>
<td>16</td>
</tr>
<tr>
<td>GoogleNet-2014</td>
<td>22</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>50</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>101</td>
</tr>
</tbody>
</table>

IV. EXPERIMENTAL SETUP

The proposed image-based animal recognition scheme is presented in Fig. 1. First, a preprocessing stage is carried out to adjust the image dimension required by each studied architecture. We fix the input dimension as 224x224 for VGG16, ResNet50, ResNet101 architectures, and 229x229 for the GoogleNet-2014. Then, we extract the spatial features selecting trainable and non-trainable CNN-based layers for each network, as presented in Table II. Further, as explained in Section III, we carry out the training and validation procedure, optimizing the trainable parameters based on the loss function chosen. Finally, the classification performance is measured.

Our approach is tested on two databases. Firstly, we use the widely known CIFAR10 dataset as a benchmark for image-based object recognition from noisy and low-resolution samples [1,3,11]. CIFAR10 collects images of size 32x32 pixels, holding ten classes (airplane, automobile, bird, cat, deer, dog, frog, horse, boat, and truck), for a total of 60.000 images, divided into 50.000 samples for training and 10.000 for testing. We build a data subset composed of 2.000 images belonging to two classes: cat and dog. Then, we split this subset, 80% for training, and 20% for testing [24,25]. In turn, we collect a small cow database, termed IMGCOW, holding cow and no cow instances captured from farm-based scenarios. Chiefly, the IMGCOW dataset is composed of 1.500 images, as follows: cow (500 samples), chicken (200 samples), and horse (200 samples), all capture from the Animals-103 dataset. The remaining 600 illustrations were taken from the training set of the CIFAR10 collection. Again, 80% of the samples are randomly selected as the training set, and the remaining 20% for testing. Fig. 2 illustrates some examples of the studied datasets.

The binary cross-entropy (BCE) is used to solve the optimization problem in Eq. (5). Then, the loss in logistic between the real probability and the predicted one is computed as follows:

$$L(\hat{y}_i, y_i) = \frac{1}{N} \sum_{i=1}^{N} y_i \log (p(\hat{y}_i)) + (1 - y_i) \log (1 - p(\hat{y}_i))$$

(6)

To evaluate our transfer learning proposal, we carried out three experiments for each architecture described in Table 1. The first experiment aims to compare the model's performance for the CIFAR10 data subset. The second experiment tests each model in the IMGCOW dataset. Finally, a method comparison is presented with state-of-the-art approaches [20,21].

In all experiments, we set the following hyperparameters: The optimizer is fixed based on the adaptive estimation of moments, called the Adaptive Moment Estimation (ADAM) algorithm [4], using a learning rate value of 0.001 and 250 epochs. Moreover, the pooling size is fixed as 64. The experimental codes were developed using TensorFlow 2 and are publicly available on Github⁴. Concerning the performance criteria, the following metrics are considered:

---

² https://www.cs.toronto.edu/~kriz/cifar.html
³ https://www.kaggle.com/alessiocorrado09/animals10
⁴ https://github.com/gloria256/EXPERIMENTOS-ARTICULO-1
Fig. 1. Image-based animal recognition sketch using transfer learning. The VGG-16, GoogleNet-2014, ResNet-50, and ResNet-101 architectures are tested as trained networks. As a Reference Dataset, ImageNet collection is employed, and as Target Dataset, a dog vs. cat and a cow vs. no cow databases are built.

\[
\text{Acc} = \frac{100(TP + TN)}{TP + FP + TN + FN} \quad [%] \\
\text{Acc}_{c} = \frac{100TP}{N_{e}} \quad [%] \\
\text{Re} = \frac{100 TP}{TP + FN} \quad [%] \\
\text{Pr} = \frac{100 TP}{TP + FP} \quad [%] \\
F_{1} = \frac{200(Pr \times Re)}{Pr + Re} \quad [%]
\]

where \( \text{Acc}, \text{Acc}_{c}, \text{Re}, \text{Pr}, F_{1} \in [0,100] [%] \) stand for Accuracy, Accuracy per class, Recall, Precision, and F1-score, respectively. Namely, \( TP, TN, FN, \) and \( FP \) represent the True Positive, True Negative, False Negative, and False Positive rates. Specifically, the \( Pr \) metric gives us the quality of the prediction. But this would not be very useful since the classifier would ignore all but one positive instance. But this would not be very useful, since the classifier would ignore all but one positive instance. So \( Pr \) is typically used along with the \( Re \) metric, also called sensitivity or the true positive rate, which is the ratio of positive instances that are correctly detected by the classifier and gives us the quantity about what percentage of the positive class have we been able to identify. Finally, the \( F_{1} \) score combines precision and recall in a single measure and consists of the harmonic mean between them. Whereas the regular mean treats all values equally, the harmonic mean gives much more weight to low values. As a result, the classifier will only get a high \( F_{1} \) if both recall and precision are high.

V. RESULTS AND DISCUSSION

Table III shows the results of the first experiment (cat vs. dog from CIFAR10 subset). The ResNet101 network obtains a performance of 76.2% and outperforms in 1.2%, 3%, and 16.2% the other ResNet50, Vgg16, and GoogleNet-2014 architectures, respectively. Besides, the obtained results report that the transfer learning approach generates better performance in architectures with residual units: ResNet101 and ResNet50, whose difference in all the evaluation metrics does not exceed 1.3%. The overall ResNet101’s account (our approach) exceeds the remaining architectures by 6.8% on average.

Table IV shows the achieved performances of the second experiment (IMGCOW dataset). As seen, the GoogleNet-2014 network obtains the lowest classification accuracy. Again, models composed of residual units, i.e., ResNet50 and ResNet101, achieve the highest performance. However, both achieve similar accuracy, indicating that the number of residual units does not drastically influence the IMGCOW data's recognition. Despite the networks Vgg16, ResNet50, and ResNet101 being competitive, the ResNet101 is ranked as the best performing, with an achieved accuracy of 98.3%.

The transfer learning effectiveness is presented in Tables III and IV. These results show that transfer learning provides better performance in classifying images that do not offer visual distortion, such as pixelation. Specifically, Table III reports the obtained results dealing with fully pixelated images (CIFAR10 subset), whose classification accuracy ranges from 60% to 70.2% in all considered evaluation metrics for each model. While Table IV shows the achieved performance using 60% of images without distortion (IMGCOW). In this case, the models yield a classification performance fluctuating between 78% and 98.3%. Of note, ResNet-based transfer learning allows dealing with imbalance issues and noisy scenarios, as reported in the Precision, Recall, and F1 scores, mitigating false positive and false negative predictions. Then, ResNet-based representation favors the model generalization to code each class’s relevant patterns.

Finally, Table V shows the results of the third experiment. In this case, we carried out a method comparison between the architecture that yields the highest performance in the guidelines of Table III, e.g., ResNet101, and deep learning-based state-of-the-art classifiers. Specifically, we consider the

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>DEEP LEARNING MODELS FOR IMAGE-BASED RECOGNITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Not trainable Layers</td>
</tr>
<tr>
<td>VGG-16</td>
<td>1 to 13</td>
</tr>
<tr>
<td>GoogleNet-2014</td>
<td>1 to 81</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>1 to 49</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>1 to 100</td>
</tr>
</tbody>
</table>
results for cat vs. dog reported in [1,3]. The attained results demonstrate that our ResNet101-based transfer learning approach can obtain comparable performance compared to ResNet50 and GoogleNet architectures without transfer learning. It is worth mentioning that our method employs a much lower number of samples than those used in the studied state-of-the-art.

In this study, we introduce an image-based animal recognition approach based on a transfer learning strategy. To this end, we couple a ResNet101-based scheme within a transfer learning framework and compare our approach with widely known architectures, such as GoogleNet-2014, Vgg16, and ResNet50. We assess all involved models into two datasets: a subset of samples extracted from the CIFAR10 database (cat vs. dog classes) and a database composed of cow and no cow images (IMGCOW). The results show that our transfer learning approach can fulfill the following aspects: i) achieve reasonable classification accuracy from small datasets, ii) properly fit the required hyperparameters without overfitting, and iii) deal with the low resolution and noisy sample issues. Indeed, our strategy yields competitive performance against state-of-the-art methods concerning the number of input samples used. Furthermore, in the subset formed from CIFAR10, we employ a low number of instances concerning the compared architectures without transfer learning. Then, residual units favor transferring the knowledge between similar tasks, e.g., ResNet models. Of note, the experimental codes were developed using TensorFlow 2, and the codes for method comparison are publicly available.

As future work, we plan to test our approach to smart farming environments to support a real-time vision system for cow counting. Also, medical imaging tasks from few samples could be tested. In turn, trying different loss functions and architectures is a research field of interest.

ACKNOWLEDGMENTS

A. Álvarez-Meza thanks to the project “Sistema de visión por computador para el monitoreo automático de variables de productividad en plantas industriales” (HERMES 46185 - FIA - Universidad Nacional de Colombia - Manizales).

REFERENCES


Gloria Stephany Gómez Gómez received his undergraduate degree in electronic engineering (2020) from the Universidad Nacional de Colombia. Research interests: deep learning.

ORCID: https://orcid.org/0000-0001-5895-0496

Diego Fabian Collazos Huertas received his undergraduate degree in electronic engineering (2014), and his M.Sc. (2016) from the Universidad Nacional de Colombia. Currently, he is a Ph.D. (c) at the same university. Research interests: deep learning and signal processing.

ORCID: https://orcid.org/0000-0002-0434-3444

Andrés Marino Álvarez Meza received his undergraduate degree in electronic engineering (2009), his M.Sc. (2011), and his Ph.D. in automatics from the Universidad Nacional de Colombia. Currently, he is a Professor in the Department of Electrical, Electronic and Computation Engineering at the same university. Research interests: machine learning and signal processing.

ORCID: https://orcid.org/0000-0003-0938-9576