Recognition of Endangered Pantanal Animal Species using Deep Learning Methods

Mauro dos Santos de Arruda
Computer Departament – UFMS
79907-414, Ponta Porã, MS, Brasil
email: mauro.arruda@aluno.ufms.br

Gabriel Spadon
ICMC/USP
13566-590, São Carlos, SP, Brasil
email: spadon@usp.br

Jose F Rodrigues-Jr
ICMC/USP
13566-590, São Carlos, SP, Brasil
email: junio@icmc.usp.br

Wesley Nunes Gonçalves
Computer Departament – UFMS
79907-414, Ponta Porã, MS, Brasil
email: wesley.nunes@ufms.br

Bruno Brandoli Machado
Computer Departament – UFMS
79907-414, Ponta Porã, MS, Brasil
email: brunobrandoli@gmail.com

***Abstract*—Pantanal is one of the most important biomes of the world, with a large number of wild animal species, some of them are in extinction. The automatic identification of wild animals is extremely important for the estimation of the species' population within Pantanal. However, digital processing techniques for the identification and tracking of species have faced great challenges due to clumsy light and pose conditions present in images taken in the wild. To overcome such problems, we propose a methodology that, by combining regular RGB images and thermal images, improves the identification of species even in images taken in rough circumstances. We use the SLIC segmentation algorithm to identify the regions of the images where animals are present; after that, we apply convolutional neural networks to classify the identified regions according to eight possible animal species. We experiment on a real-world dataset composed of 1,600 images. Our results showed an average gain between 6% and 10% when compared to the method *Fast R-CNN*.**

1. Introduction

Pantanal is one of the richest ecosystems in the world, housing a large number of animals living in ecological balance. This region is the world's largest tropical wetland with an area of approximately 138,183 km^2. Junk et al. [1] list 263 fish species, 96 reptile species, 40 amphibian species, 390 bird species, and 130 mammal species. The number of species diverges in several works, for example, the number of bird species is usually given as something between 600 and 700 species [2], although only 390 were confirmed [1]. Because of its importance and ecological diversity, Pantanal is considered by UNESCO as a World Heritage Site, and as a Biosphere Reserve[1].

As a result of unsustainable land uses, Pantanal is constantly in danger of losing its biodiversity and associated natural habitats. Endangered species occur in the region, such as the giant armadillo (*Priodontes maximus*), marsh deer (*Blastocerus dichotomus*), giant anteater (*Myrmecophaga tridactyla*), bush dog (Speothos venaticus), among others. The major challenge for Pantanal is to balance the growing demand for social and economic development (e.g., cattle ranching, agriculture, tourism) with the conservation of the biodiversity [3].

However, getting to know the biodiversity present in the vast area of Pantanal comes to be a big challenge faced by researchers working to protect species from extinction. The identification of species in Pantanal is a laborious time-consuming task due to the wetland environment and to the eating habits of the animals, making it difficult to effectively estimate the species and their densities by km^2. To track and monitor such animal population, researchers spend days in Pantanal observing the species with the aid of technologies, such as camera traps, and aerial images. The current identification process is extremely cumbersome, requiring special training and the manual inspection of captured images and videos.

In this paper, we propose a new approach to automatically detect and identify the animal species of Pantanal using convolutional neural networks (CNN). In contrast to the other region-oriented CNN (R-CNN) proposals, here, regions of interest are identified using the segmentation of thermal and RGB images by means of the SLIC algorithm [4]. Such regions are projected onto the network feature map of the Fast R-CNN [5] network; after that, a maximum pooling is performed to adjust the final size of the features with the fully-connected layer. We use the deep neural network VGGNet [6] architecture with 16 layers for the final species identification. Our approach achieved better results when compared with the original Fast R-CNN for 8 animals species.

The rest of the paper is organized as follows. In Section 2, we review the related papers on automatic animal identification. In Section 3, we described the methods that we use and compare. The proposed approach is detailed in Section 4. Section 5 presents experiments and results. Finally, the

1. More details at http://whc.unesco.org/en/list/999.

conclusions and future directions are presented in Section 6.

2. Review of the Literature

Some approaches have been proposed to perform the automatic identification of animal species [7], [8], [9]. Yu et al [7] proposed an automated species identification for pictures captured by remote-camera traps using *Scale-invariant feature transform* (SIFT) and spatial pyramid matching. They tested their method on a database with 18 species from two different biomes (tropical rainforest and temperate forest); they achieved an average accuracy of 82%. More related to the present work, Chen et al. [8] and Gómez et al. [9] proposed methods for animal species identification using convolutional neural networks. Chen et al. [8] used an image database with 20 species common in North America, reaching an accuracy of only 38% with a CNN with 6 layers. On the other hand, Gómez et al. [9] achieved 98% of accuracy on an image database with 26 species from Tanzania. They have tested more advanced CNN architectures, including AlexNet, VGGNet, and GoogLeNet.

Similarly, our work also uses convolutional neural networks, but it is different in two main aspects. First, the animal species are specific to the Pantanal, which is important to support ecologists and biologists on the animal conservation of this specific biome. Second, we investigate a new strategy to propose regions of interest in the task of animal identification. Different from the first region-oriented CNN architecture proposed by Girshick et al. [10], named R-CNN, our approach uses a single CNN instead of multiple CNNs for each candidate region in the image. In the second version of the architecture proposed by Girshick et al. [5], named Fast R-CNN, our approach is different in terms of the region identification methodology; that is, it differs in how the regions of interest (ROIs) are selected using the segmentation algorithm SLIC, and in how the region coordinates are mapped into the feature map; lastly, similar to the Fast R-CNN, feature vectors are pooled into a fixed-size feature vector and connected to the fully connected layers.

3. Materials and Methods

3.1. VGGNet Architecture

Convolutional networks have recently received a great deal of attention due to the success in world-class competitions on large-scale image classification [11]. Simonyan and Zisserman proposed the VGGNet architecture, which brought a substantial improvement in the classification accuracy of images by increasing the depth of the network along with a clever architecture in between layers [6]. Basically, the input of VGGNet is a fixed-size 224×224 RGB image. First, a pre-processing step is performed by subtracting the mean RGB value, computed on the training set, from each pixel. VGGNet passes each image through convolutional layers, with small 3×3 filters in all layers, and the convolution stride and padding are fixed to 1 pixel. Along the convolution process, the spatial pooling is carried out by five max-pooling layers, performed over a 2×2 pixel window, with stride 2. The architecture is then composed of three fully-connected layers. In our work, we adapted the last fully-connected layer to the number of classes used in the experiments, in our case, 8 classes of wild animals. The last layer of the VGGNet is the softmax gradient-log-normalizer of the classes' probability distribution. It is worth saying that all the hidden layers are equipped with the rectification (ReLU) non-linearity [12]. Thereby, as the spatial size of the filter decreases along the convolutional layers, the number of filters (depth) increases, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512. In our paper, we used the VGGNet with 16 layers; it was considered the best variation of the proposed architecture in the original paper.

3.2. Fast R-CNN Architecture

Recently, the performance of object-detection techniques has been boomed by the object-oriented methods [13] and region-oriented convolutional neural networks (R-CNN) [10], [14]. The basic idea of R-CNN is to propose multiple candidate regions by using an auxiliary method to detect regions, or objects, of interest. Next, it proceeds by computing convolutional features from each region (using, for example, VGGNet [6]); then, it obtains the classification rate for each region using a binary linear Support Vector Machine. In contrast, Fast R-CNN computes convolutional features only once over the entire image. After that, for each region of interest, features are directly extracted from the global features map with a fixed-length vector over the region of interest (ROI) pooling layer. Finally, feature vectors are connected to fully-connected layers that finally branch into two output layers: (i) classification and (ii) regression.

In more details, the first layer estimates a discrete probability distribution for each region of interest, $p = (p_0, ..., p_C)$, over $C + 1$ categories, where p is computed by a softmax probability over $C + 1$ outputs of the fully-connected layer. The second layer obtains the bounding-box regression, $t^c = (t^c_x, t^c_y, t^c_w, t^c_h)$, of an object belonging to a certain class C. Therefore, each training region is associated to a class u, as well as the bounding-box regression value v. For training the network, each region has its loss function defined as:

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v) \qquad (1)$$

where $L_{cls}(p, u) = -logp_u$ is the log loss for the true class u. L_{loc} is defined over a true bounding-box regression target for class u, $v = (v_x, v_y, v_w, v_h)$, and a predicted tuple $t^u = (t^u_x, t^u_y, t^u_w, t^u_h)$ for the class u.

3.3. SLIC Segmentation

The *Simple Linear Iterative Clustering* (SLIC) is a clustering-based segmentation algorithm proposed by

Achanta et al. [4]. The method employs the k-means algorithm for the generation of regions, called superpixels. The parameter k is the desired number of superpixels in the image. The SLIC algorithm is considered a fast approach with linear runtime, and it yields the state-of-the-art adherence to image boundaries, which outperforms existing methods when used for image segmentation.

The first step of the SLIC algorithm is to convert the original color image into the CIELAB color space. It uses the color components L, a, b, and the spatial position x and y as features of the pixels. For an image with N pixels, the algorithm is initialized with even-distributed k cluster centers, and each region composes an initial superpixel at a grid interval of $S = \sqrt{N/k}$. Then, the distances between the pixels and the clusters' centers around a $2S \times 2S$ region are calculated. Pixels are labeled to the cluster with the nearest centroid. The centers are moved to the lowest gradient value over a 3×3 pixel neighborhood, avoiding centroid positioning in edge regions or having noisy pixels. The distance is calculated by the Euclidean norm of the pixel i to the kth cluster center, defined as follows:

$$d_c(i,k) = \sqrt{(L_k - L_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \quad (2)$$

$$d_s(i,k) = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad (3)$$

$$d = d_c + m\frac{d_s}{S} \quad (4)$$

where d_c is the color distance, d_s is the spatial distance, d is the distance between two pixels, S is the spacing between cluster centers, and m is a factor parameter, which is used to weight the proportion of color values and spatial information in the distance measurement. SLIC also requires a post-processing in order to group the connectivity of the superpixels, forming a region of interest.

4. A New Proposed Approach for Wild Animal Recognition

In this section, we detail our method, which is divided into two parts: (1) identification of regions of interest using the SLIC algorithm; and (2) feature map projection onto the CNN for animal recognition. Following, we detail the two parts of the methodology.

4.1. Region Detection

The proposed approach for region detection is based on the SLIC segmentation algorithm, as described in Section 3.3. Basically, the idea is to group the pixels of the image into regions that have similarity based on the body temperature of the animals. Since the temperature of the animals tends to be higher than the average ambient temperature, the color attributes are emphasized according to the color matrix given by a thermal camera.

Formally, an RGB image is defined as $I \in \Re^{w \times h}$, w and h standing for the width and height of the images. Its corresponding thermal image is defined as $T \in \Re^{w \times h}$. For each cell of the matrix $T(x,y)$, the thermal camera provides the temperature in Celsius degrees (°C). In order to detect the animal in the region, we first calculate the average temperature of the image $Temp_m$, by means of the equation:

$$Temp_m = \frac{1}{w \times h}\sum_{x=1}^{w}\sum_{y=1}^{h} T(x,y) \quad (5)$$

We then apply the SLIC algorithm on the thermal image T to obtain a set of superpixels $S = \{s_1, s_2, ..., s_k\}$, where k is the number of superpixels. Thereby, we calculate the average temperature for each superpixel s_i by:

$$Temp_{s_i} = \frac{1}{N_{s_i}}\sum_{(x,y)\in s_i} T(x,y) \quad (6)$$

where N_{s_i} is the number of pixels associated with the superpixel s_i.

The proposed approach detects the regions where the animals are present in the images by checking if a given superpixel s_i has temperature higher than the average ambient temperature $Temp_m$:

$$Temp_{s_i} + L > Temp_m \quad (7)$$

where L is a threshold used for separating the superpixels belonging to the animal or to the background. Once we have the target superpixels, we group the superpixels that satisfy to Equation 7 in regions through the 8-connected component labeling algorithm [15], [16]. As a result, we have a set of regions $R = \{r_1, r_2, ..., r_z\}$, where z is the total number of clustered regions, and each region is defined by $r_i = \{s_1 \cup s_2 \cup \ldots \cup s_{N_{r_i}}\}$, where N_{r_i} is the number of superpixels of region r_i. Thus, a superpixel s_i belongs to a region r_i if, for a given j, we have that $s_j \in r_i$ and s_i is considered a neighborhood of 8-connected pixels of s_j and if it satisfies to Equation 7.

Finally, we analyze the regions with the largest number of pixels with the aim of eliminating small noisy regions that do not represent any animal in the image, e.g., rocks or ant nests that may be warmer than the ambiance. To this end, we first obtain the region with the largest number of pixels of the set R with $M_r = \max_{i=1}^{z}(Tam_{r_i})$, where Tam_{r_i} is the size in pixels of the region r_i. Then, we check for every region r_i, if $Tam_{r_i} > \frac{M_r}{p_{min}}$, where p_{min} is a variable that defines the lowest accepted ratio between Tam_m and M_r. As an example, if $p_{min} = 4$, all regions r_i that have size Tam_{r_i} greater than 25% related to the region M_r will be defined as a region of interest of the image I.

Therefore, the set of regions of interest that detect the animals in the image is defined as $ROI = roi_1, roi_2, \ldots, roi_n\}$, where each region roi_i is a region of interest. In turn, each region of interest is defined as $roi_i = \{x_{roi_i}, y_{roi_i}, w_{roi_i}, h_{roi_i}\}$, where x_{roi_i} and y_{roi_i}

define the starting point (upper left corner of the region), and w_{roi_i} and h_{roi_i} represent the width and height of the region.

4.2. Animal Recognition

After the animal detection in the image, that is, the region proposal, the proposed approach for the identification of animal species is performed by using a convolutional neural network. We chose convolutional neural networks due to the recent performance in various computational vision tasks [6], [12], obtained mainly through the AlexNet networks [12], and the VGGNet [6].

Our animal species classification surpasses the competitor methodology Fast R-CNN by using the regions of interest obtained by the SLIC algorithm over the thermal images. Fast R-CNN works differently, it detects regions of interest directly over the RGB images not counting on the SLIC pre-processing. Figure 1 shows the diagram of the proposed approach.

On the training set, we performed a pre-processing step by resizing each RGB image I to a fixed-size of $224 \times 224 \times 3$. In this paper, we use the convolutional network VGGNet; however, any other convolutional network can be used. In addition, another pre-processing on the training set is performed by subtracting the mean RGB value from each pixel, as given by:

$$\begin{aligned} I(x,y,R) &= I(x,y,R) - \mu_R \\ I(x,y,G) &= I(x,y,G) - \mu_G \\ I(x,y,B) &= I(x,y,B) - \mu_B \end{aligned} \tag{8}$$

where $\mu_{\{.\}}$ is the mean color of all images of the training set.

Each image is processed up to the last convolutional layer to obtain the feature map, named here as MP_c. Once we obtain the feature map after Step 3, we use the detected set of regions of interest ROI to perform the feature vector extraction over the feature map of each individual roi_i. However, since the $ROIs$ were extracted from the input image I, and the features map MP_c has dimensions smaller than the dimension of I (due to the output format of VGGNet), we also have to project each roi_i on the map. We use the dimensions x_{roi_i}, y_{roi_i}, w_{roi_i} e h_{roi_i} to obtain the projection over MP_c. To achieve this goal, we use a max pooling to convert each roi_i into a feature vector with a dimension of 7×7 pixels, called hyper-parameters. The result is a projected region $r_{proj} = \{x_{proj}, y_{proj}, w_{proj}, h_{proj}\}$, an appropriate input to Step 4. After Step 5, the network has two output vectors per ROI: the softmax probabilities and the bounding-box regression. Our approach is trained end-to-end with a multi-task loss.

5. Experimental Results

5.1. Image Datasets

Our image datasets are divided into training and test sets, as explained in the following subsections.

Training dataset: the training dataset was obtained from the well-known IMAGEnet dataset [11], comprised of images from the web. Images were taken with different illumination and points of view, as well as with different image resolutions. We used a subset of IMAGEnet with 8 classes, 600 images each class, totalizing 4,800 images of animals species from the Pantanal biome. The classes of animal species are: Brazilian Tapir (*Tapirus terrestris*), Blue-and-yellow Macaw (*Ara ararauna*), Capybara (*Hydrochoerus hydrochaeris*), Collared Peccary (*Pecari tajacu*), Cougar (*Puma concolor*), Turquoise-fronted Amazon (*Amazona aestiva*), South American Coati (*Nasua nasua*), and Giant Anteater (*Myrmecophaga tridactyla*).

Test dataset: the test dataset is comprised of thermal and RGB images with dimension of 640×480 pixels. Notice that the thermal images are used to detect regions of interest in the images; the animal identification occurs over RGB images similar to those of the training dataset. The thermal images were captured with a thermal camera FLIR SC640. Parameters of the camera were set to the standard values, including 0.98 of emissivity, 60% of humidity and $25°\mathrm{C}$ temperature. We used distances ranging from 1 to 5 meters from an orthogonal direction. The total of images sums up to 1,600 thermal and RGB images, divided into 8 classes of animal species taken from the Pantanal biome.

5.2. Implementation Details

Our approach was implemented on MATLAB, using the package matconvnet [17]. The training step was performed over two images each time, since proposed by [5]. We used the default value of 75 epochs for training the neural network. Each epoch corresponds to one training using all the images from the training dataset. We used a learning rate of $\frac{1e-3}{64}$ for the first 50 epochs and $0.1 * \frac{1e-3}{64}$ for the 25 remaining epochs. The experiments ran on a Nvidia Titan X GPU. For the testing step, we used the testing dataset described in Section 5.1, comparing our approach to the Fast R-CNN method. For detecting regions of interest, we used the technique SLIC, setting the number of superpixels to $k = 1,500$, and the level of compactness to $m = 5$. In our work, we adapted the last fully-connected layer for the number of classes used in the experiments, in the case, 8 classes of wild animals.

5.3. Evaluation Metrics

This paper adopted the same metrics described in the competitor work of Girshick [5], which are widely used in region detection by convolutional networks: (i) average precision and (ii) f-measure. The evaluation of the method considers ground-truth images annotated by an expert. To determine if the target region was correctly classified, we considered true positives (tp) and false positives (fp), the accuracy is calculated according to the intersection over union (IoU). Furthermore, with the labels of each region of interest, four metrics can be calculated to measure the performance of the algorithms: precision (PR), recall (RE),

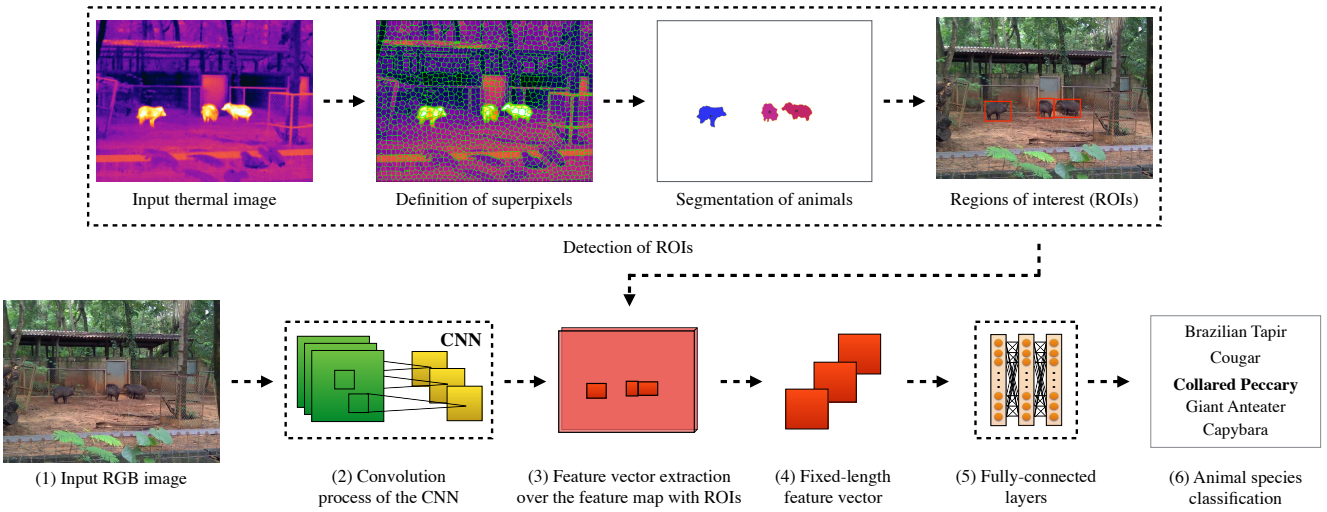


Figure 1. The proposed approach: (1) input RGB image into a fully convolutional network; (2) convolution process using the entire image to obtain the feature map of the last convolutional layer; (3) feature map extraction using the regions of interest proposed by our approach; (4) fixed-length feature vector scaled using a max pooling layer; (5) fully-connected layers with the scaled feature vectors of step (4); and (6) recognition of the animal species performed using the softmax classifier and bouding-box regression for each ROI.

f-measure (FM), and the average precision (AP). Such measures are formalized in Equations 9 and 10.

$$PR = \frac{tp}{tp + fp} \quad \text{and} \quad RE = \frac{tp}{p} = \frac{tp}{tp + fn} \tag{9}$$

$$AP = \frac{PR}{RE} \quad \text{and} \quad FM = 2 \times \frac{PR \times RE}{PR + RE} \tag{10}$$

5.4. Recognition of Animal Species

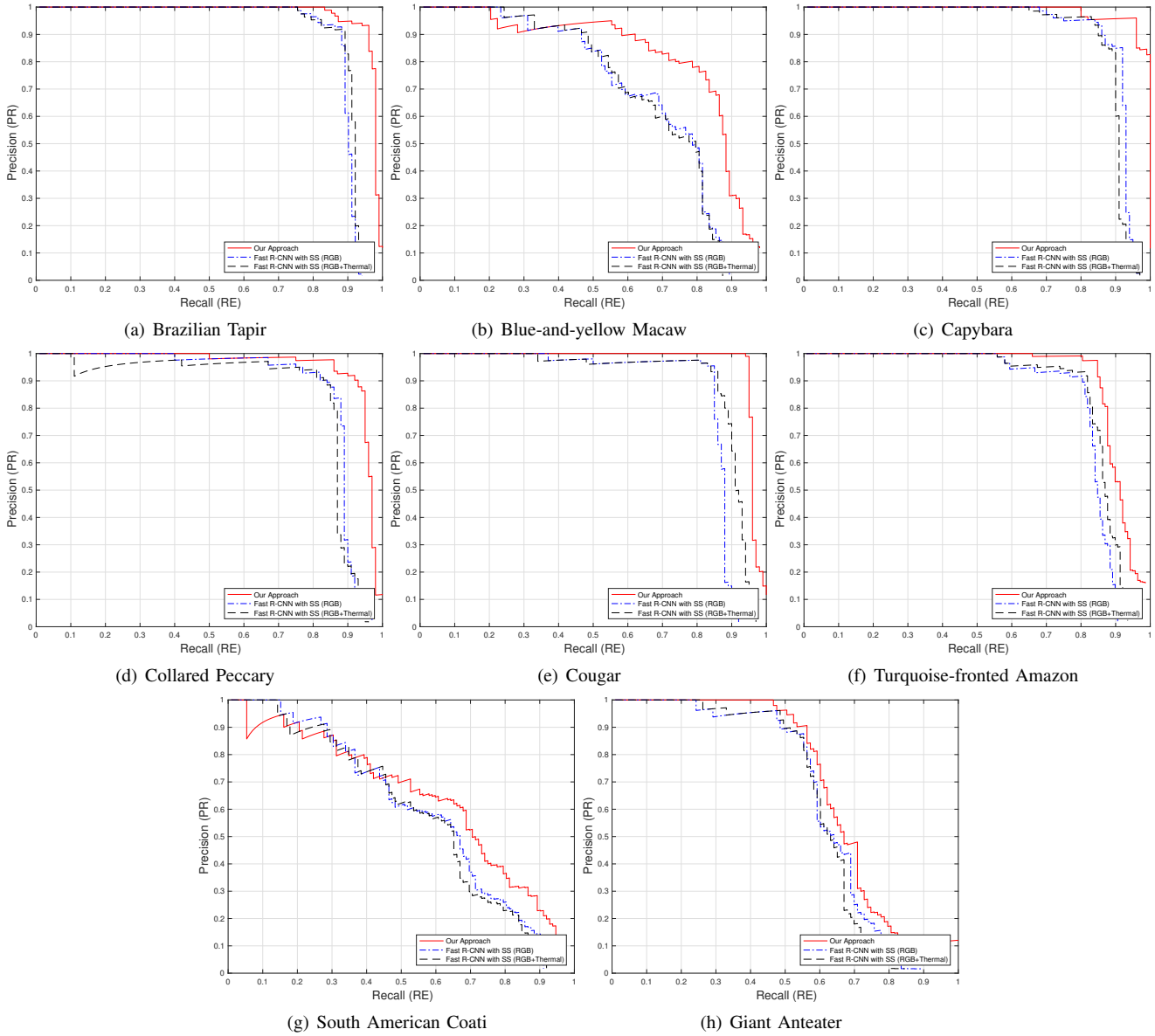
Figure 2 shows the results of our approach using the precision and recall curves, compared to the Fast R-CNN method under two versions: (i) the traditional Fast R-CNN with Selective Search over RGB images only, and (ii) the Fast R-CNN with Selective Search over RGB images, as well as over thermal images. All candidate regions are then used in the test step. In this experiment we used 8 classes of animal species. For all the plots, the x-axis corresponds to the recall values (RE), while the y-axis corresponds to the precision values (PR), both of them varying from 0 to 1. The value of the average precision (AP) is calculated by the area under the curve. It is worth noticing that the higher the area under the curve, the higher is the average precision value.

Table 1 shows the evaluation of our approach compared with Fast-CNN for two metrics: (i) average precision (AP) and (ii) f-measure for all the animal classes. In addition, at the bottom of the columns, we present the mean average precision (mAP) of all the classes for both metrics. Our approach achieved a mean average precision of 83.89% vs 76.45% compared with Fast R-CNN, while for the mean f-measure, our approach achieved 77.95% vs 73.06% of Fast R-CNN. We also compared our methodology to the Fast R-CNN with Selective Search applied to RGB images and over themal images. Although the perfomance of both spectra with Selective Search has slightly improved related to the traditional Fast R-CNN, it did not outperform the results achieved by our approach. In terms of comparison, for animal classes, our approach outperformed the Fast R-CNN values for the average precision and f-measure metrics. The best results were achieved for the *Cougar* and *Blue-and-yellow Macaw* classes. On average, for both classes, our approach performed 10% better on the average precision and, 6% for f-measure. In contrast, results achieved by our approach for *Brazilian Tapir* and *South American Coati* classes were closer to the Fast R-CNN method, improving by 6% for average precision and 3% for f-measure.

To support our numerical analysis, in Figure 3, we visually present the detected regions of interest in a sample of images; we use our method and the Fast R-CNN method. We use a red frame to emphasize the results in each scene. The two first left-hand columns present three examples of multiple-animals identification, comparing Fast R-CNN to our approach. The two right-hand columns show a single animal identification. Our approach visually performed better in images with multiple animals. For single animals, although both methods can detect the animal, our approach can frame the animal with higher accuracy, as well as higher probability prediction, as indicated in the blue boxes on top of the red frames.

5.5. Discussion of Results

We tested our approach with images of animal species taken under real conditions in a wild forest. The accuracy for all species are compared with the competitor Fast R-CNN; Figure 2 presents the results. It is evident that our approach obtains the best accuracy for all the animal species using the average precision, which is based on the precision and recall



(a) Brazilian Tapir (b) Blue-and-yellow Macaw (c) Capybara

(d) Collared Peccary (e) Cougar (f) Turquoise-fronted Amazon

(g) South American Coati (h) Giant Anteater

Figure 2. Results of the average precision for the eight animal species of the Pantanal biome: Brazilian Tapir (a), Blue-and-yellow Macaw (b), Capybara (c), Collared Peccary (d), Cougar (e), Turquoise-fronted Amazon (f), South American Coati (g), and Giant Anteater (h).

curves. It should be noted that the thermal imaging improves the results if compared with the Fast R-CNN, locating the animals in the images and increasing the robustness for animal recognition.

Table 1 shows the accuracy results per animal class using our approach versus Fast R-CNN, evaluated by average precision and f-measure metrics. In all the classes, the accuracy reached a high performance. The lowest performance was observed in the case of species *Brazilian Tapir* and *South American Coati* with an average precision of 6%; the best results were 10% higher on average precision for the *Cougar* and *Blue-and-yellow Macaw* classes. This also illustrates how our approach has better and robust representation ability, superior to Fast R-CNN location and recognition using thermal infrared. Despite the fact that the region-based CNN usually has good results in detection performance, our approach also performs favorably in different kinds of pose and illumination conditions.

In addition, Figure 3 shows the results of the visual comparison of our experiments versus the Fast R-CNN method. The columns illustrate the identification of animals, along with the corresponding probability prediction (blue box on top). For multiple animals present in the image, our results show superior robustness in terms of location and recognition. Note that, in this work, the CNN was fine-tuned by means of a large training dataset. This proves that the automatic recognition using camera-trap is possible for helping biologists and ecologists, but it depends on having

Animal Species of the Pantanal	**Average Precision (AP) (%)**			**f-measure (FM) (%)**		
	Fast R-CNN		Our Approach	Fast R-CNN		Our Approach
	SS (Default)	SS (RGB+Thermal)		SS (Default)	SS (RGB+Thermal)	
Brazilian Tapir	87.08	88.94	91.58	87.96	87.50	93.60
Blue-and-yellow Macaw	68.38	68.28	78.89	48.57	49.65	53.79
Capybara	88.95	88.83	98.06	78.30	77.59	80.65
Collared Peccary	83.48	81.73	90.88	85.08	83.80	90.53
Cougar	82.16	87.52	92.27	89.47	89.12	96.94
Turquoise-fronted Amazon	80.97	83.14	86.49	78.15	79.32	83.68
South American Coati	59.45	57.57	63.72	54.08	54.82	59.90
Giant Anteater	61.16	60.96	69.30	62.89	63.75	64.52
Mean Average Precision	76.45	77.12	**83.89**	73.06	73.19	**77.95**

TABLE 1. COMPARISON OF THE RESULTS BETWEEN THE PROPOSED APPROACH AND THE COMPETITOR METHOD FAST R-CNN. EACH ROW SHOWS AN ANIMAL SPECIE, WHILE COLUMNS SHOW THE RESULTS FOR THE AVERAGE PRECISION AND f-MEASURE METRICS.

(a) *Fast R-CNN* (b) Our Approach (c) *Fast R-CNN* (d) Our Approach

Figure 3. Visual comparison of regions detection achieved by Fast R-CNN and by our approach. In the left-hand first two columns, the detection performance for multiple animals on the scene; in the last right-hand two columns, we see examples of only one animal. Blue boxes on top of the animals refer to the animal recognition probability generated by the network architecture VGGNet.

enough data.

6. Conclusion and Future Works

In this paper, we presented a methodology to detect and recognize animal species observed in wild conditions. We used deep convolutional neural networks trained to distinguish between eight different animal species. The training of the network was based on images extracted from the IMAGENet dataset. For testing, we constructed a dataset with regular RGB images and images taken with a thermal camera. By combining regular RGB images and thermal images, we surpassed the results of the method Fast R-CNN, which had limitations in detecting the regions of a given image in which animals were present.

In our approach, we selected regions from the thermal images using the segmentation algorithm SLIC; then, we used the regions of interest, as indicated by their thermal signatures, to identify the corresponding regions as seen in the RGB images. These regions were input to a neural network capable of tracing the probability of finding a given species in each region of interest. We directly compared our

method to the Fast R-CNN method over metrics precision (PR), recall (RE), f-measure (FM), and average precision (AP). Our method outperformed the Fast R-CNN results in all the tests concerning the recognition of the animal species. As future works, we expect to increase the number of animal species and improve the success in large-scale animal recognition by using camera-trap images. In addition, we intend to compare with differents methods, including Faster-RCNN and Yolo2.

Acknowledgments

This research was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), by the Fundacao de Apoio a Pesquisa do Estado de Sao Paulo (Fapesp), and by the National Council for Scientific and Technological Development (CNPq). The authors are thankful to the Centro de Reabilitação de Animais Silvestres (CRAS) of Campo Grande - Mato Grosso do Sul/Brazil, and to the Instituto de Meio Ambiente de Mato Grosso do Sul (IMASUL), for their assistance during the execution of the experiments. We also thank NVIdia for generously providing equipment through its Academic Program.

References

[1] W. J. Junk, C. N. da Cunha, K. M. Wantzen, P. Petermann, C. Strüssmann, M. I. Marques, and J. Adis, "Biodiversity and its conservation in the pantanal of mato grosso, brazil," *Aquatic Sciences*, vol. 68, no. 3, pp. 278–309, 2006.

[2] F. A. Swarts, *The Pantanal of Brazil, Paraguay and Bolivia: Selected Discourses on the Worlds Largest Remaining Wetland System*. Hudson MacArthur Publishers, January 2000.

[3] C. Alho and J. Sabino, "A conservation agenda for the Pantanal's biodiversity," *Brazilian Journal of Biology*, vol. 71, pp. 327 – 335, 04 2011.

[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, pp. 2274–2282, November 2012.

[5] R. Girshick, "Fast r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, December 2015.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014.

[7] X. Yu, J. Wang, R. Kays, P. A. Jansen, T. Wang, and T. Huang, "Automated identification of animal species in camera trap images," *EURASIP Journal on Image and Video Processing*, no. 1, p. 52, 2013.

[8] G. Chen, T. X. Han, Z. He, R. Kays, and T. Forrester, "Deep convolutional neural network based species recognition for wild animal monitoring," in *IEEE International Conference on Image Processing (ICIP)*, pp. 858–862, Oct 2014.

[9] A. Gomez, A. Salazar, and F. Vargas, "Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks," *ArXiv e-prints*, mar 2016.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, June 2014.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[13] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 142–158, Jan 2016.

[15] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1992.

[16] A. Rosenfeld, "Connectivity in digital pictures," *J. ACM*, vol. 17, pp. 146–160, Jan. 1970.

[17] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 689–692, ACM, 2015.