IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 23, 2021, accepted April 21, 2021, date of publication April 23, 2021, date of current version May 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3075293

Hierarchical Classification of Very Small Objects: Application to the Detection of Arthropod Species

PAUL TRESSON[1,2,3,4], (Student Member, IEEE), DOMINIQUE CARVAL[1,2,3], PHILIPPE TIXIER[1,2], AND WILLIAM PUECH[4], (Senior Member, IEEE)
[1]CIRAD, UPR GECO, F-34398 Montpellier, France
[2]GECO, Univ. Montpellier, CIRAD, 34000 Montpellier, France
[3]CIRAD, UPR GECO, F-97455 Saint-Pierre, France
[4]LIRMM, Univ. Montpellier, CNRS, 34000 Montpellier, France

Corresponding author: William Puech (william.puech@lirmm.fr)

This work was supported in part by a grant from the European Community (ERDF) and the Conseil Régional de La Réunion, as part of the CIRAD DPP COSAQ agronomical research programme (activities 2015–2021), and in part by the French National Research Agency through the Investments for the Future Program, referred to as ANR-16-CONV-0004. The work of Paul Tresson was supported by CIRAD and #DigitAg grants.

ABSTRACT Automated image analysis and deep learning tools such as object detection models are being used increasingly by biologists. However, biological datasets often have constraints that are challenging for the use of deep learning. Classes are often imbalanced, similar, or too few for robust learning. In this paper we present a robust method relying on hierarchical classification to perform very small object detection. We illustrate our results on a custom dataset featuring 22 classes of arthropods used to study biodiversity. This dataset shows several constraints that are frequent when using deep learning on biological data with a high class imbalance, some classes learned on only a few training examples and a high similarity between classes. We propose to first perform detection at a super-class level, before performing a detailed classification at a class level. We compare the obtained results with our proposed method to a global detector, trained without hierarchical classification. Our method succeeds in obtaining a mAP of 75 %, while the global detector only achieves a mAP of 48 %. Moreover, our method shows high precision even on classes with the less train examples. Confusions between classes with our method are fewer and are of a lesser impact. We are able achieve a more robust object classification with the use of our proposed method. This method can also enable better control on the model's output which can be particularly valuable when handling ecological, biological or medical data for example.

INDEX TERMS Very small object detection, deep learning, robustness, biodiversity, taxonomy.

I. INTRODUCTION

Recently, there has been a wide adoption of deep learning techniques in various fields of study. For example, deep learning has recently been taken up by the medical and biological sector, biological and used in ecological research [1]–[3]. For several cases however, deep learning methods still have a lot of limitations and constraints that hinder their proper usage. Datasets may be limited, classes imbalanced or similar and the labelling task too heavy to build a robust model.

While such constraints maybe overcome with transfer learning or data augmentation for image classification, these methods might not be sufficient when dealing with object detection. As detectors tend to have a larger number of parameters than classifiers, they also require more training data to achieve satisfactory performances. This makes reliable object detection and classification a major challenge when working with a custom dataset [4]. Numerous cases work with imbalanced datasets *e.g.* in medical research, the number of sane examples can outnumber the diseased ones, in particular for very rare diseases. An unsufficiently robust deep learning model could generate false positives, if this is not taken into account [5].

The availability of a large enough dataset to ensure the robustness of the predictions remains a barrier to the use of deep learning with niche tasks. The training of a detector, with only a few training examples is known as ''low-shot object detection'' or ''few-example object detection''. Several methods have recently been developed. Multi-modal labelling, *i.e.* referring to objects with several classes is commonly

The associate editor coordinating the review of this manuscript and approving it for publication was Qiang Lai.

used. Dong *et al.* propose a method relying on multi-modal labellisation and different learning paces depending on class difficulty [6]. Low-Shot Transfer Detector on the other hand relies on transfer learning [7]. Another method is to rely on distant metric learning [8]. Adding to the constraint of few examples, in many practical cases, objects featured in a dataset may be similar, requiring fine classification tasks, which may be challenging for a deep learning model, particularly when trained on imbalanced data. Furthermore, many cases in biology or medicine need control over the predictions of a model to avoid false negatives or false positives [2]. In such situations, the control and robustness of a prediction model over precise classes are more important than efficiency or inference time on general datasets.

In this paper, we propose a method to detect and hierarchically classify very small objects on large input images. We illustrate the usefulness of our method working on a custom dataset showing typical constraints that could be encountered when working on a fine detection task: limited training examples, class imbalance and fine classification.

The main contributions and key-points of our proposed paper can be summarized as follows:

- We propose a robust method of hierarchical classification when performing very small object detection.
- A pre-processing step is applied to perform a super-class detection of very small objects within large images.
- The detection and classification steps are independent, allowing us to work with sparse datasets.

In our context, the aim of this work is to robustly detect several species to latter study the interactions between them. Knowing these interactions is crucial for several purposes, such as resilient pest control. Our application highlights several constraints of the application of deep learning in real world situations with custom dataset.

The rest of this paper is organized as follows. First of all, Section II reports current state-of-the-art methods on hierarchical classification when detecting objects. Section III describes with details the proposed method. Experimental results are provided in Section IV. Finally, the conclusion is drawn and future work is proposed in Section V.

II. RELATED WORK

In this section we describe several related work concerning low-shot object detection, then hierarchical classification, small object detection and finally, applications in ecology.

A. LOW-SHOT OBJECT DETECTION

Low-shot object detection, *i.e.* the training of a detector with only a few examples per class is a recent but active research field. In this area, several methods have been developed [6]–[8]. A commonly used method is to refer to objects with several classes (multi-modal labelling) which is the approach we selected. Otherwise, Dong *et al.* propose a method communicating between model training and sample selection [6]. Then, using this method the most challenging yet reliable training samples are selected. This approach can be combined with multi-modal learning and varying learning paces, given the difficulty of the classes.

Chen *et al.* rely on transfer learning with a Low-Shot Transfer Detector [7]. While transfer learning might be prone to overfitting with only a few training examples, they address this challenge. The authors propose to combine SSD [9] and Faster RCNN [10] architecture, separating bounding box regression and object classification.

Another method is to rely on distant metric learning [8]. The extraction of low dimensional representations enables the learning of more generic features such as class only features. The losses are designed in an embedding space so that different categories are not only distinct, but that similar categories are close as well.

B. HIERARCHICAL CLASSIFICATION

Hierarchical classification has been used in deep learning for the handling of large datasets with numerous classes [11]–[13]. For classification tasks, Katole *et al.* achieved 3.2 % error rate on the ImageNet 10K dataset [11] that features over 10,000 classes using hierarchical classification [12]. Hierarchical classification was also used for detection tasks on ImageNet. For instance, Redmon *et al.* achieved a mAP of 76.8 % on over 9000 classes [13].

C. SMALL OBJECT DETECTION

Small objects are defined by the MS COCO (Microsoft Common Objects in Context) dataset as objects occupying areas under 32×32 *pixels* [14]. This is a challenging problem in computer vision and several methods can be used to address it [15]. One solution is to slice the large input image and perform detection on slices separately before merging all results. This method can be used for satellite imagery analysis [16] or insect detection for example [17].

D. APPLICATIONS IN ECOLOGY

Hierarchical classification is useful for small datasets requiring precise classification, typically for a use in biology [18], medicine [19] or ecology [20], [21]. Indeed, hierarchical classification enables to better control the error rates of the classifier [22]. For species identification, particularly, this approach has been used following the taxonomy of different species. While hierarchical classification has already been developed for species classification, it has to our knowledge not been carried out for fine detection tasks.

III. PROPOSED METHOD

In this section, we develop our proposed method for hierarchical classification of very small objects. An overview of our method is presented in Fig. 1. Objects detected on the input images belong to N super-classes C_i, with $1 \leq i \leq N$. Each super-class C_i contains k_i classes c_{ij}, with $1 \leq j \leq k_i$.

During the training step, a detector is trained with objects that have been labeled at the super-class level. The areas of these objects are then cropped from the original images and

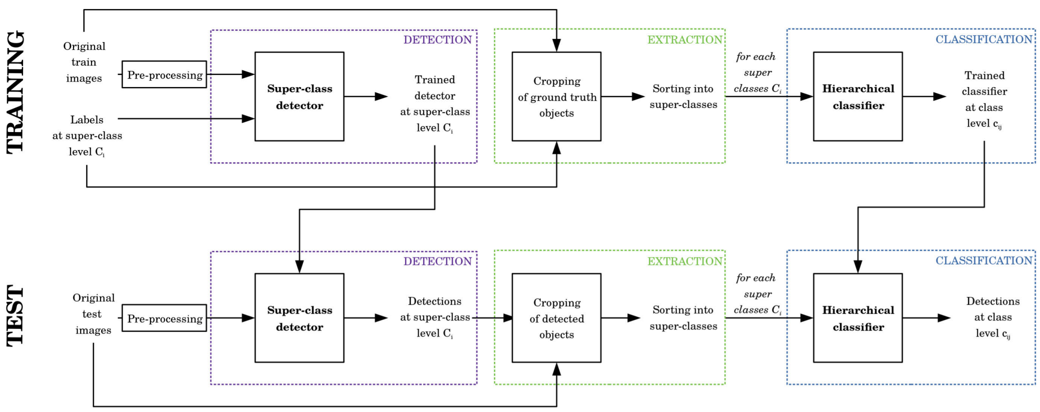


FIGURE 1. Overview of our proposed hierarchical classification method.

for each super-class C_i, a classifier is specifically trained to identify the objects within the super-class into c_{ij} classes.

During the testing step, objects are first detected at super-class level. Detected objects are then cropped and classified with the corresponding classifier.

The super-class detection part of the process is detailed in Section III-A, the hierarchical classification in Section III-B and the evaluation metrics used to assess the performances our method are described in Section III-C.

A. OBJECT DETECTION

With large input images, detection might require a pre-processing phase. For this step we propose to apply an approach based on the method proposed by Tresson *et al.* [17]. The pre-processing step is summarized in Algorithm 1. Original images are split into small slices for the input of a deep learning model. Slicing is performed with an overlap O in order to reduce the risk of an object being cut between two neighboring slices. Coordinates of the objects are recomputed within the referential of the slices and empty slices showing background only are discarded. The detector is then trained on the sliced dataset. Likewise, during testing, original images are sliced with the same parameters as during the training. Detections of slices belonging to the same original image are then merged together and refined to suppress potential false positives due to overlap. This pre-processing phase allows the detection of very small objects within very large input images. Detection is performed with objects identified at a super-class level.

B. OBJECT HIERARCHICAL CLASSIFICATION

For each super-class, a classifier is trained independently on cropped ground truth objects. The model is then able to classify objects into the k_i classes c_{ij} within this super-class as illustrated in Fig. 1. During the test step, the detected objects are cropped according to their coordinates obtained in the coordinate system of the entire original image. Cropped objects are then identified with the classifier model corresponding to their detected super-class.

Algorithm 1 Image and Label Pre-Processing

input : Original image of size $n_W \times n_H$ pixels (I_{orig}),
Original labels ($c_{ij}, x_i, y_i, w_i, h_i$),
Slice of size $n_{slice} \times n_{slice}$ pixels,
Overlap O
output: Sliced image, and recomputed labels
($c_{ij}, x'_i, y'_i, w'_i, h'_i$)
$I_{orig} \leftarrow$ slice(n_{slice}, O)
for ***every slice*** **do**
  if *x_i, y_i within slice or 50 % of the object on the slice or 40 % of the slice covered by object* **then**
    recompute label
  else
    Discard

C. EVALUATION METRICS

To assess detection performances, the model predictions on the test dataset are compared with ground truth labels. The *IoU* (Intersection over Union) is used to compare bounding boxes. Detections are accepted as True Positive (TP) if $IoU > 0.5$ and if the detected class is correct. Otherwise, the detection is counted as FP. As well, duplicates are counted as False Positive (FP). If a ground truth object is missed, it is counted as FN. Performances are assessed with precision, recall and F1-score:

$$precision = \frac{TP}{TP + FP}, \tag{1}$$

$$recall = \frac{TP}{TP + FN}, \tag{2}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}. \tag{3}$$

For each class, the Average Precision (AP) is computed as the area under the precision-recall curve. AP is used to compare performances between classes.

The mAP is favored as this indicator reacts strongly to performance loss on a class, regardless of the number of objects within the class. This is a good indicator of the performances

TABLE 1. Classes and corresponding Super-classes.

Super-classes	Train/Test examples	Classes	Train/Test examples
C_1 Arachnida	1048/252	$c_{1,1}$ *Isometrus maculatus*	109/23
		$c_{1,2}$ *Lycosidae* msp. 1	464/117
		$c_{1,3}$ *Lycosidae* msp. 2	237/52
		$c_{1,4}$ *Salitcidae* msp. 1	145/39
		$c_{1,5}$ *Salitcidae* msp. 2	93/21
C_2 Coccinellidae	722/147	$c_{2,1}$ *Cheilomenes sulphurea*	127/25
		$c_{2,2}$ *Exochomus laeviusculus*	386/74
		$c_{2,3}$ *Psyllobora variegata*	209/48
C_3 Coccoidea	8779/2929	$c_{3,1}$ *Ceroplastes sinensis*	132/24
		$c_{3,2}$ *Dysmicoccus brevipes*	394/147
		$c_{3,3}$ *Icerya seychellarum*	8253/2758
C_4 Formicidae	13400/3219	$c_{4,1}$ *Brachymyrmex aphidicola*	189/37
		$c_{4,2}$ *Cyphomyrmex rimosus*	1114/290
		$c_{4,3}$ *Paratrechina longicornis*	560/129
		$c_{4,4}$ *Pheidole megacephala* major	3076/878
		$c_{4,5}$ *Pheidole megacephala* minor	1814/461
		$c_{4,6}$ *Solenopsis geminata* minor	2982/620
		$c_{4,7}$ *Tapinoma melanocephalum*	1176/286
		$c_{4,8}$ *Technomyrmex albipes*	2197/428
		$c_{4,9}$ *Tetramorium bicarinatum*	292/90
C_5 Myriapoda	3018/862	$c_{5,1}$ *Pachybolidae* msp.	2545/774
		$c_{5,2}$ *Paradoxosomatidae* msp.	473/88

of rare classes. However, the F1-score however still provides information on the overall detection performances of a model.

We compare the performances obtained with our proposed method with the performances obtained with a global detector trained directly across all n classes without hierarchical classification.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the different results obtained with our hierarchical classification method for very small objects are discussed. First we describe the dataset we built in Section IV-A and the deep learning models and parameters used in Section IV-B. In Section IV-C our method is applied to a large image from the test dataset and presented in detail. Section IV-D gives overall performance analysis of our method and compares the results with a global classification.

A. DATASET

For our experiments we have developed a custom dataset featuring various insects on large input images ($3,000 \times 4,000$ *pixels*). This dataset shows typical constraints when performing object detection in biology.

Indeed, as illustrated in Fig. 2, some arthropods featured in this dataset are easily distinguishable, while other classes are visually very close one to another. For instance, *Pachybolidae* msp. ($c_{5,1}$) (Fig. 2.a) is easily distinguishable. On the other hand, ant species such as *Technomyrmex albipes* ($c_{4,8}$) (Fig. 2.b), *Solenopsis geminata* minor ($c_{4,6}$) (Fig. 2.c), *Pheidole megacephala* major ($c_{4,4}$) (Fig. 2.d) or *Pheidole megacephala* minor ($c_{4,5}$) (Fig. 2.e) are very similar. While pictures were taken in controlled conditions, dirt and branches were added to the background to add noise and additional complexity for the deep learning models.

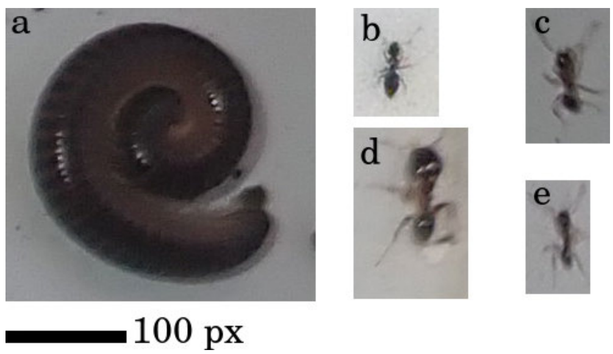


FIGURE 2. Examples of objects found in our custom dataset: a) *Pachybolidae* msp. ($c_{5,1}$), b) *Technomyrmex albipes* ($c_{4,8}$), c) *Solenopsis geminata* ($c_{4,6}$), d) *Pheidole megacephala* major ($c_{4,4}$), e) *Pheidole megacephala* minor ($c_{4,5}$).

A total of 4,102 images are used as a training dataset and 955 as a test dataset. After pre-processing, the actual training dataset for the detector is 27,160 slices and the test dataset is comprised of 7,000 slices.

The dataset features 26,967 objects belonging to the 5 super-classes (C_1 to C_5) and 22 classes in total (see Table 1). Classes are imbalanced as some species are more frequent than others, between and within super-classes. For instance, there are 16 times more *Pheidole megacephala* major ($c_{4,5}$) training examples as *Brachymyrmex australis* ($c_{4,1}$) and 62 times more *Icerya sechellarum* ($c_{3,3}$) as *Ceroplasted sinensis* ($c_{3,1}$). The most frequent class is *Icerya sechellarum* ($c_{3,3}$) which shows 8,253 training examples, whereas Salticidae msp. 2 ($c_{1,5}$) features only 93 training examples.

The objects featured on the dataset are very small with an average width of 102.54 *pixels* and an average height of 81.9 *pixels*, derived from images of $3,000 \times 4,000$ *pixels*, hence the utility of the pre-processing step.

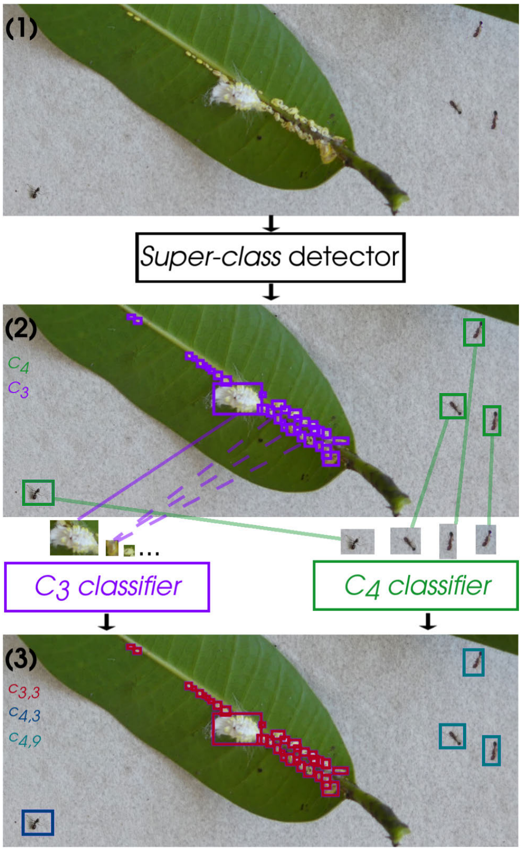


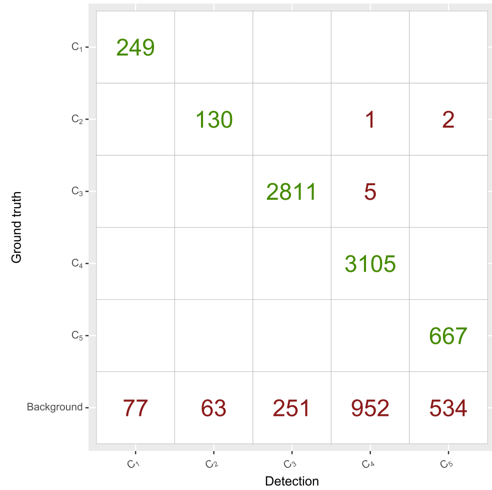
FIGURE 3. A full example illustrating the two main steps of our hierarchical classification method: 1) A part of an original image of our dataset. 2) Super-class detection: Objects of super-classes C_3 and C_4 are detected. 3) Obtained hierarchical classification: Objects of classes $c_{3,3}$, $c_{4,3}$ and $c_{4,9}$ are classified.

TABLE 2. Overall performances of our hierarchical classification method compared to a global classification.

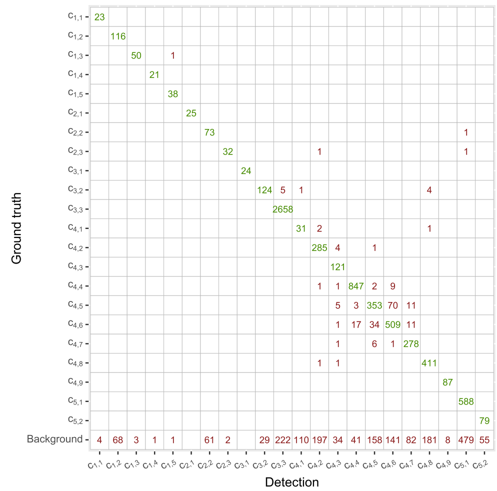
Metric	Global classification	Hierarchical classification	Gain
Precision	0.46	**0.77**	0.31
Recall	**0.92**	0.89	-0.03
F1-score	0.61	**0.83**	0.22

B. PARAMETER DESCRIPTION

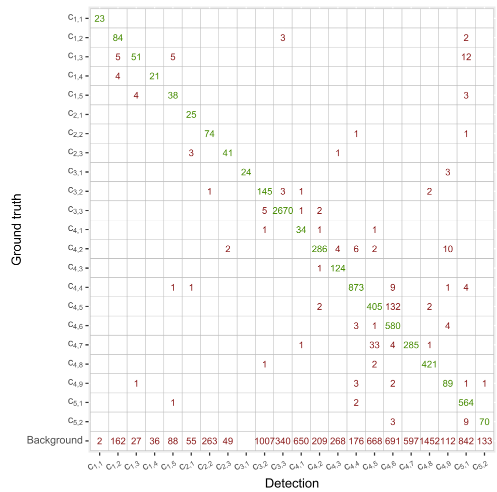
Several networks were trained to obtain our experimental results. For our method, a detector over the five super-classes presented in Table 1 (Arachnida, Coccinellidae, Coccoidea, Formicidae and Myriapoda) and specific classifiers for each super-class. A global detector over the 22 classes was trained as a comparison. All networks states were chosen to maximise mAP on test dataset. We use YOLOv3 [23] as a super-class detector. Squeezenet [24] was chosen as a classifier after a benchmark for this task. YOLOv3 features 106 fully convolutional layers. The model was trained for 39 epochs for our super-class detector (66 epochs for the global detector as a comparison) with a batch size of 4 and a learning rate of 0.001, using Adam as an optimizer. Squeezenet features 11 layers and was trained for each class. All training cycles lasted between 20 and 30 epochs with a batch size of 8 and a learning rate of 0.001, using SGD as an optimizer.



(a) Super-class detection for our hierarchical classification



(b) Hierarchical classification



(c) Comparison with a global classification

FIGURE 4. Confusion matrices obtained from our proposed approach: a) At the super-class detection step; b) After our proposed hierarchical classification; c) Comparison with a global classification.

TABLE 3. Detection and classification performances for the different Classes and Super-classes.

Super-classes	Super-class detection AP	Classes	Global classification AP	Hierarchical classification AP	Gain
C_1 Arachnida	0.66	$c_{1,1}$ *Isometrus maculatus*	**0.98**	0.93	-0.05
		$c_{1,2}$ Lycosidae msp. 1	0.24	**0.72**	0.48
		$c_{1,3}$ Lycosidae msp. 2	0.84	**0.93**	0.09
		$c_{1,4}$ Salitcidae msp. 1	0.44	**0.96**	0.52
		$c_{1,5}$ Salitcidae msp. 2	0.31	**0.96**	0.65
C_2 Coccinellidae	0.59	$c_{2,1}$ *Cheilomenes sulphurea*	0.32	**1.00**	0.68
		$c_{2,2}$ *Exochomus laeviusculus*	0.22	**0.54**	0.32
		$c_{2,3}$ *Psyllobora variegata*	0.41	**0.65**	0.24
C_3 Coccoidea	0.84	$c_{3,1}$ *Ceroplastes sinensis*	**1.00**	**1.00**	0.00
		$c_{3,2}$ *Dysmicoccus brevipes*	0.18	**0.73**	0.55
		$c_{3,3}$ *Icerya seychellarum*	**0.89**	**0.89**	0.00
C_4 Formicidae	0.80	$c_{4,1}$ *Brachymyrmex aphidicola*	0.04	**0.21**	0.17
		$c_{4,2}$ *Cyphomyrmex rimosus*	**0.77**	0.70	-0.07
		$c_{4,3}$ *Paratrechina longicornis*	0.36	**0.77**	0.41
		$c_{4,4}$ *Pheidole megacephala* major	0.87	**0.94**	0.07
		$c_{4,5}$ *Pheidole megacephala* minor	0.38	**0.63**	0.25
		$c_{4,6}$ *Solenopsis geminata* minor	0.39	**0.66**	0.27
		$c_{4,7}$ *Tapinoma melanocephalum*	0.45	**0.75**	0.30
		$c_{4,8}$ *Technomyrmex albipes*	0.22	**0.67**	0.45
		$c_{4,9}$ *Tetramorium bicarinatum*	0.51	**0.89**	0.38
C_5 Myriapoda	0.45	$c_{5,1}$ Pachybolidae msp. 1	0.36	**0.44**	0.08
		$c_{5,2}$ Paradoxosomatidae msp. 1	0.29	**0.59**	0.30
mAP	0.67		0.48	**0.75**	0.27

C. A FULL EXAMPLE

A full example of the processing for our proposed hierarchical classification method is detailed in Fig. 3. The image on Fig. 3.1 is a detail of 980 × 400 *pixels* from an image of our test dataset. When fed to the super-class detector, scale insects (C_3 Cocooidae) and ants (C_4 Formicidae) are detected. The objects are then cropped given their coordinates and these cropped images are fed in to their respective classifiers for hierarchical classification into the *I. seychellarum* ($c_{3,3}$), *P. longicornis* ($c_{4,3}$) and *T. bicarinatum* ($c_{4,9}$) classes.

D. PERFORMANCE ANALYSIS

The overall performance of our method compared to a global detector trained on all classes is available in Table 2. First considering only the detection task, our method achieves an overall precision of 0.77, a recall of 0.89, meaning a F1-score of 0.83. However, global detection and classification achieves a precision of 0.46 and a recall of 0.92. This leads to an overall F1-score of 0.61 (see Table 2). So, when considering the detection task only, hierarchical classification leads to better results with a gain of 0.22 on the F1-score, thanks to the super-class detector having better generalised than a global detector.

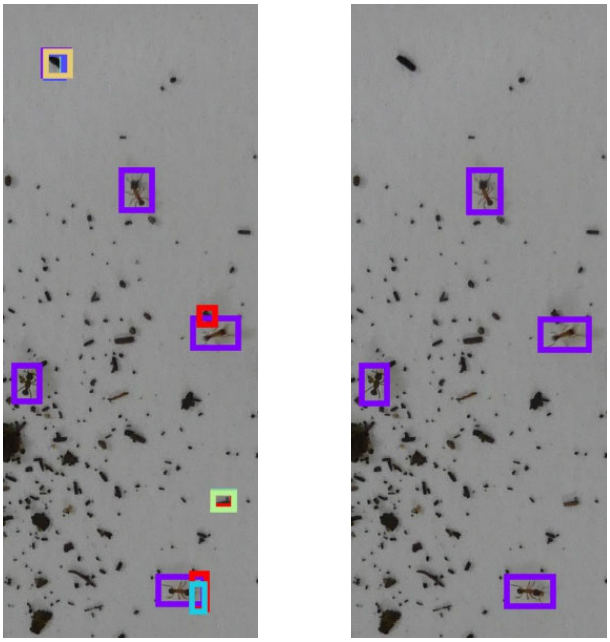
As shown in Table 3, global classification achieves a mAP of 0.48 with strong disparities among classes. Indeed, some classes such as *I. maculatus* ($c_{3,3}$), *C. sinensis* ($c_{3,1}$) and *P. megacephala* ($c_{4,4}$) show very good AP, whereas others, such as *B. australis* ($c_{4,1}$) or *T. albipes* ($c_{4,8}$) show very poor AP. With our proposed hierarchical classification, we achieve a far greater mAP of 0.75. Furthermore, disparities between classes are lower. These performance differences are particularly noticeable for classes with less training examples. Indeed, the average AP of the global classifier on classes with less than 500 training examples ($c_{2,1}$ *Cheilomenes sulphurea* or $c_{3,2}$ *Dysmicoccus brevipes* for instance) is of 0.44, whereas it is of 0.77, with our hierarchical classification.

Fig. 4 illustrates the confusion matrices obtained from our proposed approach, just after the super-class detection step, Fig. 4.a, and at the end of our hierarchical classification, Fig. 4.b. As illustrated in Fig. 4, although our proposed method shows less confusions between classes compared to a global classification, as illustrated in Fig. 4.c (197 confusions against 315 for the global detector), this alone does not explain the difference in precision between the two methods. Indeed, the vast majority of the false positives generated by the global detector are detections of non-existent objects in the background (7,981 out of the total 8,296 false positives) noted in red in the last row of the confusion matrices in Fig. 4. The better performances of our proposed method comes from a more robust detection than from a finer classification.

An example illustrating this difference is presented in Fig. 5. We can see in Fig. 5a that several false positives have been detected. On the contrary, with our approach, illustrated in Fig 5b, all the false positives have been removed, whilst keeping all true positives.

Furthermore, we observe on Figure 4 that most of the confusions by our hierarchical approach are within the same super-class (mostly ants from other ants). These confusions are arguably of a lesser impact than confusions between objects of different super-classes (ants for spiders, for example).

It should be noted from these results that the image resolution could be decreased for the detection step with minimal degradation to the results obtained. However, the specialized classifiers should always work on the best resolution images.



(a) Global classification (b) Hierarchical classification

FIGURE 5. Comparison of obtained results between our proposed hierarchical classification and a global classification on an image detail. Our method generates less false positives with noised backgrounds.

This would further reduce the inference time with minimal impact on performance. Regarding the inference time, with the parameters described in Section IV-B and the hardware used (GPU: Nvidia GeForce GTX 1080, 32 GB RAM), the inference time for the detection with YOLOv3 [23] is about 20 *ms* per slice (*i.e.*, 140 *ms* on average, per entire original image) and the inference time per object for classification is about 20 *ms*, including the cropping step.

V. CONCLUSION

In this paper, we described a hierarchical classifier which is a robust method to achieve small object detection and fine classification. We demonstrate the utility of our method on a custom dataset showing classical constraints that may limit the use of deep learning, such as class imbalance, few examples, similar or noisy objects. This method could be very valuable to researchers still encountering issues while working with deep learning on custom datasets.

The robustness and reliability of our proposed method could be further improved by adding specific confidence thresholds for the different classes and super-classes (see Villon *et al.* [22]).

ACKNOWLEDGMENT

The authors would like to thank Antoine Wyvekens and Bernard Abufera for their help during labelling.

REFERENCES

[1] B. G. Weinstein, "A computer vision for animal ecology," *J. Animal Ecol.*, vol. 87, no. 3, pp. 533–545, May 2018.

[2] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P. M. Agapow, M. Zietz, M. M. Hoffman, and W. Xie, "Opportunities and obstacles for deep learning in biology and medicine," *J. Roy. Soc. Interface*, vol. 15, no. 141, 2018, Art. no. 20170387.

[3] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Apr. 2018.

[4] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020.

[5] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *J. Big Data*, vol. 5, no. 1, p. 42, Dec. 2018.

[6] X. Dong, L. Zheng, F. Ma, Y. Yang, and D. Meng, "Few-example object detection with model communication," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1641–1654, Jul. 2019.

[7] H. Chen, Y. Wang, G. Wang, and Y. Qiao, "LSTD: A low-shot transfer detector for object detection," 2018, *arXiv:1803.01529*. [Online]. Available: http://arxiv.org/abs/1803.01529

[8] L. Karlinsky, J. Shtok, S. Harary, E. Schwartz, A. Aides, R. Feris, R. Giryes, and A. M. Bronstein, "RepMet: Representative-based metric learning for classification and one-shot object detection," 2018, *arXiv:1806.04728*. [Online]. Available: http://arxiv.org/abs/1806.04728

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[12] A. L. Katole, K. P. Yellapragada, A. K. Bedi, S. S. Kalra, and M. S. Chaitanya, "Hierarchical deep learning architecture for 10K objects classification," 2015, *arXiv:1509.01951*. [Online]. Available: http://arxiv.org/abs/1509.01951

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[15] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image Vis. Comput.*, vol. 97, May 2020, Art. no. 103910.

[16] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," 2018, *arXiv:1805.09512*. [Online]. Available: https://arxiv.org/abs/1805.09512

[17] P. Tresson, P. Tixier, W. Puech, and D. Carval, "Insect interaction analysis based on object detection and CNN," in *Proc. IEEE 21st Int. Workshop Multimedia Signal Process. (MMSP)*, Kuala Lumpur, Malaysia, Sep. 2019, pp. 1–6.

[18] D. Wu, X. Han, G. Wang, Y. Sun, H. Zhang, and H. Fu, "Deep learning with taxonomic loss for plant identification," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–8, Nov. 2019.

[19] W. Xuan and G. You, "Detection and diagnosis of pancreatic tumor using deep learning-based hierarchical convolutional neural network on the Internet of medical things platform," *Future Gener. Comput. Syst.*, vol. 111, pp. 132–142, Oct. 2020.

[20] Z. Miao, K. M. Gaynor, J. Wang, Z. Liu, O. Muellerklein, M. S. Norouzzadeh, A. McInturff, R. C. K. Bowie, R. Nathan, S. X. Yu, and W. M. Getz, "Insights and approaches using deep learning to classify wildlife," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, Dec. 2019.

[21] J. G. Colonna, J. Gama, and E. F. Nakamura, "A comparison of hierarchical multi-output recognition approaches for anuran classification," *Mach. Learn.*, vol. 107, no. 11, pp. 1651–1671, Nov. 2018.

[22] S. Villon, D. Mouillot, M. Chaumont, G. Subsol, T. Claverie, and S. Villéger, "A new method to control error rates in automated species identification with deep learning algorithms," *Sci. Rep.*, vol. 10, no. 1, pp. 1–13, Dec. 2020.

[23] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

[24] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360

PAUL TRESSON (Student Member, IEEE) received the Diploma degree in agricultural engineering and the M.Sc. degree in ecology, in 2018. He is currently pursuing the Ph.D. degree in informatics and modeling with CIRAD (French tropical agricultural research institute) and the ICAR Team (Image and Interaction), LIRMM. He has published several articles in ecological and agronomical research journals and has presented results at IEEE Multimedia and Signal Processing 2019 Conference.

DOMINIQUE CARVAL received the Ph.D. degree in ecology from the University of Paris 6, in 2009, with a focus on modeling of host-parasite co-evolutionary processes. In 2011, he joined the CIRAD (French agricultural research and international cooperation organization), where he worked on disease and pest management in banana crops in Martinique. In 2018, his research activities turned to the use of deep learning for the study of animal communities. He is currently the author of more than 20 articles in ecology and agricultural journals. His current research interests include the study of interactions in animal communities by combining image analysis and machine learning and the agronomic evaluation of new banana cultivars in La Réunion.

PHILIPPE TIXIER graduated in ecology from the University of Paris 6, in 1999, and received the Ph.D. degree in agronomical sciences from the Agronomic School of Montpellier, in 2004. He is currently an Agroecologist, who uses empirical and modeling methods to study questions at the interface of population, community, agro-ecosystem ecology, and farms' performances. He has 20 years of research experience working on tropical agrosystems, especially in the Caribbean, Central America, and West-Central Africa. The central question he addresses is the role of biodiversity on the processes of agrosystems, with a focus on ecological regulations of pests and diseases of tropical crops, especially bananas and plantains. His research has basic and applied implications, particularly in the design of innovative biodiversification schemes to improve the sustainability of tropical agriculture. He has currently more than 80 articles published in agronomy, ecology, and modeling.

WILLIAM PUECH (Senior Member, IEEE) received the Diploma degree in electrical engineering from the Université Montpellier, France, in 1991, and the Ph.D. degree in signal-image-speech from the Polytechnic National Institute of Grenoble, France, in 1997, with a focus on image processing and computer vision. He served as a Visiting Research Associate with the University of Thessaloniki, Greece. From 1997 to 2008, he was an Associate Professor with Université Montpellier. Since 2009, he has been a Full Professor in image processing with Université Montpellier. He is currently the Head of the ICAR Team (Image and Interaction), LIRMM. He has published more than 45 journal articles and 140 conference papers. His current research interests include image forensics and security for safe transfer, storage, and visualization by combining data hiding, compression, cryptography, and machine learning. He was a member of the IEEE Information Forensics and Security TC, from 2018 to 2020. Since 2021, he has been a member of the IEEE Image, Video, and Multidimensional Signal Processing TC. Since 2017, he has been the General Chair of the IEEE Signal Processing French Chapter. He is also an Associate Editor for five journals, such as *JASP*, *SPIC*, *SP*, *JVCIR*, and IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, in the areas of image forensics and security.