Harmful Insects Detection Using Convolutional Neural Networks (Faster R-CNN) *

Abderrahim Akarid, Samir El Adib, Naoufal Raissouni

Harmful Insects Detection Using Convolutional Neural Networks (Faster R-CNN)*

Abderrahim Akarid
Remote sensing systems and telecommunications
University Abdelmalek Essaadi
National School for Applied Sciences of Tetuan, Morocco
a.akarid@uae.ac.ma

Samir El Adib, Naoufal Raissouni
Remote sensing systems and telecommunications
University Abdelmalek Essaadi
National School for Applied Sciences of Tetuan, Morocco
seladib@uae.ac.ma, naoufal.raissouni.ensa@gmail.com

Abstract **–** Insect detection is a crucial task in various fields, including agriculture, entomology, and biodiversity conservation. Among the problems we encountered was the difficulty of identifying insects due to the great similarity of appearance of certain species. Currently, the Convolutional Neural Networks (CNNs) have been widely adopted for insect detection due to their ability to accurately classify objects in images, using recent advances methods in deep learning and computer vision algorithms. In this paper, we focused only on seven types of insects most harmful to agricultural crops in Morocco, such as olive and wheat… We propose a CNN-based architecture specifically Faster RCNN to processing our model. The purpose of this research is to determine the type of insect and monitor it, which can allow us to identify and reduce the chemical pesticides used but also to take timely preventive measures and avoid economic losses.

Index Terms - Agriculture, Insect, Object detection, Deep learning, Image processing, ***Convolutional neural network.***

I. INTRODUCTION

Here are over a million species of insects in the world. Manual categorization and identification of these species is time consuming and requires extensive knowledge of field crops. Traditionally, Identification and classification of insects in the field of entomology and biology is a challenging and important task Insects play a crucial role in ecosystems, acting as pollinator, predators, and decomposers, among other functions.

Accurate identification and classification of insects are necessary for understanding the diversity and distribution of species, as well as for monitoring changes in their populations while predicting damage in agriculture.

However, the task of identifying and classifying insects can be difficult due to the large number of species and the variability in their physical characteristics. In addition, many insects are similar in appearance, making it difficult to differentiate between species. To address these challenges, scientists and researchers have developed methods for identifying and classifying insects, including various techniques, with relying on advanced computer vision technique [1] has made it possible to overcome these challenges and to gain a deeper understanding. Currently computer vision techniques play a crucial role in many fields of research such as entomological sciences (Weeks et al., 1999) [2], environment (Larios et al., 2008) [3] and agricultural engineering (Zhao et al., 2012)[4]. The computer vision methods could be a feasible way to solve the problem of automated insect categorization and identification. Therefore, there is a need to find an efficient and fast technique for automatic classification and detection of harmful insects.

Deep Learning has been extensively used for insect detection in recent years with features including image classification, and object detection. These deep learning models have shown remarkable results in object detection and classification tasks, making them a popular choice for insect detection. In such applications, CNNs are trained on large datasets of insect images to learn the features and patterns unique to different species. The trained model can then be used to classify new images of insects with high accuracy. The use of CNNs in insect detection has been effective in automating the process and reducing the time and effort required for manual identification and allows predicting and taking the decision.

II. RETATED WORK

A. HISTORICLLY

Historically, the first attempts to use computers to analyse images back to 1940s and 1950s with the first mathematical model of a neuron [5]. And the intention was to simulate the human brain system to solve general learning problems. It was popular in the 1980 and 1990 with the algorithm is named "proposal propagation" [6], proposed by Hinton, one of the pioneers of deep learning. The proposal propagation algorithm is a type of artificial neural network (ANN) that is designed to overcome some of the limitations of traditional neural networks. To this day, it widely used in a variety of

applications, including image classification, and natural language processing. In the late 1990s and early 2000s, the proposal propagation algorithm fell out of favor as other deep learning algorithms, such as convolutional neural networks (CNNs), gained popularity [7].The rise of deep learning, with the introduction of deep neural networks (DNN) with many layers and development of recurrent neural networks (RNNs) and long-short term memory (LSTM) networks, which have been used in applications such as natural language processing (NLP) and speech recognition [8][9].

The recovery of deep learning can be attributed to the emergence of training data like ImageNet [10] and development of high-performance computing systems, such as GPU clusters. With data augmentation and some initialization provided such as unsupervised learning pre-training by Auto-Encoder (AE)[11] or Restricted Boltzmann Machine (RBM)[12]. The learning problem in the training was relieved [13], and with batch normalization (BN), the neural network training becomes quite efficient [14].

Meanwhile, various network structures have been extensively studied to improve performance, such as AlexNet [15] Overfeat [16] GoogLeNet[17], VGG [8], ResNet[18]. This revolution results to the training of a large CNN on 1.2 million labeled images as well as some techniques [19](such as Relu[20]).

All these proposed improvements in detection performance are based on standard large-scale dataset, such as MS COCO [21], PASCAL VOC [22] et ILSVRC [23], with the appearance of multi-stage (Two-stage) object detection methods that's use first model to extract regions of interest (ROIs) , and second model is used to classify and further refine the localization of the object. Namely R-CNN, R-FCN et FPN [24]. After that, massive improved models have been proposed with multiple convolution layers, which require huge computational capacity. As a solution, Faster R-CNN implements Region Proposal Networks (RPN) which solves also the speed problem of R-CNN and Fast R-CNN. Hereafter, R-FCN is proposed to further improve performance.

Although these models can be used for object detection, it is difficult to find a model that achieves a balance between accuracy and speed, because it is necessary to take into consideration other factors. Is the computationally expensive approach useful or not? For example, The "Automatic Butterfly Detection System" [25] developed by Ding and Taylor based on the sliding window method, a traditional object detection technique. In this method, the image is divided into multiple overlapping regions or windows, and each window is evaluated to determine if it contains an object of interest, in this case, a butterfly. The sliding window method scans the image at multiple scales, allowing the detection of objects at different sizes. This method can be computationally expensive and may not be as accurate as more recent deep learning-based object detection methods.

Recently, Advantage of deep learning-based object detection methods have been used in agriculture, mainly for disease and pest detection. And among the big changes in the field of object detection, and the original sliding window approaches have been replaced by region proposals [26].For example, to achieve efficient detection of major tomato organs, Zhou et al. [27] integrated a classification network model based on VGGNet, designed TD-Net with Fast-RCNN to obtain an effective detection of the main organs of the tomato. The average accuracy (AP) of the detector for fruit, flower and stem was 81.64%, 84.48% and 53.94%, respectively. This has improved performance and speed compared to R- CNN and Faster R-CNN.

B. CNN

Object detection can be categorized into two types: Non-CNN methods, which rely on traditional computer vision techniques like Haar cascades, HOG-based detection, and sliding windows, and are computationally efficient but less accurate than CNN-based methods. CNN-based methods use Convolutional Neural Networks (CNNs), a specific type of neural network [28] that processes and transmits information through interconnected nodes called artificial neurons. This allows the network to calculate a unique output with high accuracy.(Touzet, 1992)[29]

Convolutional neural networks (CNNs) are optimized for image processing. They apply filters to extract features like edges and textures. A pooling layer then reduces the feature map size for efficiency and to combat overfitting.

Each filter has a dimension of size F×F, applied to an input containing C channels, the formula of the input volume (I), with (F) the size of the filter and the size (O) of the output feature map without a stride (S) along this dimension is such that in (1) and with stride (S) in (2) and (3) is formula by adding padding (P) that allows more accurate analysis of images and more space for the kernel to cover the image.

$$O = (I - F) + 1 \quad ; \quad O = \frac{(I - F)}{S} + 1 \qquad \textbf{(1) (2)}$$

$$O = \frac{(I - F + 2P)}{S} + 1 \qquad \textbf{(3)}$$

Common convolution layers in a CNN vary based on task and requirements. Other useful layers include activation (ReLU), dropout & batch normalization for improved performance and stability.

C. REGION-BASED CNN

Generally, CNN is a type of deep learning network architecture that is specifically designed for image data processing. The first family is the region-based convolutional neural network includes R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN.

R-CNN

The R-CNN is an object detection architecture start by extracting interesting regions from image, and then it uses

these regions as data input for a CNN. This separation into regions makes it possible to detect several objects of several different classes in the same image, this solution proposed by GIRSHICK et al., 2013[30].

Fast R-CNN

Fast R-CNN [31] is a variant of R-CNN that reduces computation and memory required in R-CNN, by passing the entire image through a deep CNN to produce a feature map. This map is passed through a region proposal network (RPN) that generates object proposals, which are then used as input to a ROI pooling layer. Finally, a sequence of fully connected layers predicts class probabilities and bounding box regression.

Faster R-CNN

Faster R-CNN [32] is an improvement of R-CNN that uses a pre-trained CNN model trained on a large image classification dataset, such as ImageNet. It uses an RPN to generate ROIs and extracts features for each of the ROIs. The ROIs are then resized to a fixed size and passed to the classification and regression subnetworks, which output a probability distribution over predefined classes and bounding box coordinates for the object, respectively.

Mask R-CNN

Mask R-CNN [33] extends the Faster R-CNN architecture by adding a third stage for generating object masks. It consists of a backbone network for feature map extraction, an RPN for generating region proposals, and a detection and segmentation head.

Single-stage methods YOLO and SSD

Nowadays,object detection has become increasingly demanding for faster speed and higher efficiency, particularly in single-stage methods such as YOLO and SSD. These methods use a single network forward pass to identify all objects in an image instead of the previous two-stage approach [34], as seen in Faster R-CNN. YOLO, which is exceptionally fast, is ideal for real-time object detection. SSD uses a feedback convolution neural network to produce a set of bounding boxes and scores for the presence of object classes [35]. Multi-stage detection using the VGG-16 architecture for feature extraction is used to detect multiple scales. The prediction includes the coordinates of the bounding boxes, such as the coordinates center, width, and height of the box.

R-FCN

FCN [36] is a fully convolutional network that uses position-sensitive score maps to reduce computation. R-FCN increases speed by sharing calculation on the entire image, solving the contradiction between image classification translation-invariance and object detection translation-variance. It achieves a good balance between speed and accuracy, with comparable results to Faster R-CNN in shorter running times.

III. METHODOLOGIES:

The methodology illustrated in Fig.1 involves data preparation and augmenting data with defined parameters, selecting a model architecture, and evaluating results through testing and prediction

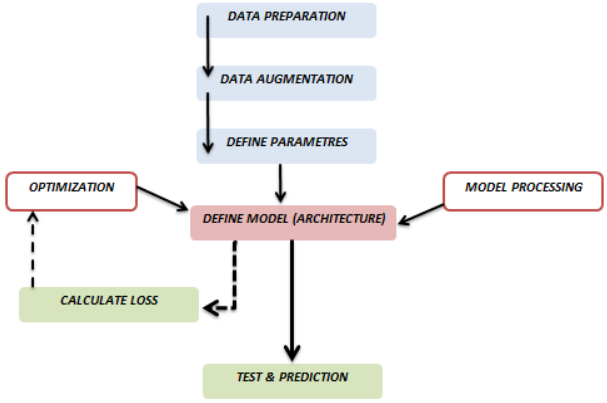

Fig. 1: Methodology

DATASET DESCRIPTIONS.

The dataset used in this study has been retrieved from the internet; we collect over 4000 images using common image search engines, which are weakly labelled. The detailed process of creating a dataset is the presented earlier in the methodologies section. Our dataset covers 7 common crop pest species with over 19000 images.

Indeed, as illustrated in Table 1, the scientific names and numbers for each insect and number of objects for each image.

TABLE 1: THE NUMBER OF IMAGES AND THE NUMBER OF OBJECTS FOR EACH SPECIES.

SPECIES	NUMBER OF IMAGE	NUMBER OF OBJECT
Diptera	2030	2215
Coleoptera	2159	2257
Araneae	2419	2610
Hemiptera	2446	3539
Lepidoptera	2048	2103
Hymenoptera	2106	2278
Odonata	2277	2301
TOTAL	15485	16303

Images collected from the internet aime to improve the generalization ability of this model, more images collected from the internet were used with a data augmentation technique. And as it is impossible to collect all varieties of insects, this is due to the presence of tens of thousands of varieties of species and classes of the database ArTaxOr [40] was adopted in this work.

The image annotation tool used is LabelImg (v1.8.3), which is an image labelling tool that facilitates the creation of objects, the generated annotations are saved in XML format in Pascal VOC or YOLO format, to mark categories and rectangular delimitation frames of stray images.

DATA AUGMENTATION

Data augmentation is crucial in computer vision. Generating additional data without additional labeling costs is key to improving data volume.

In our work, we decided to use the python library Imgaug allowing to easily modify the images with multiple transformations, and the augmentations used on the images listed in Table 2.The different augmentations are applied in sequence independently of each other. The addition and multiplication of light are excluded, because their combination was too disruptive for the model. With 9 distinct augmentations, we can generate 512 different images from the same data, not counting the variable ranges.

TABLE 2: CONFIGURING OF DATA AUGMENTATION USED.

Augmentation	Probability	Variable	Description
Fliplr	50%	-	Symétrie verticale
Flipud	50%	-	Symétrie horizontale
Rot 90	50%	-	Rotation 90 degrés
Translate	50%	-90/ to 50%	Geometric translation
Rotation	50%	-10 to 10	Rotation in degrees
Scale	50%	50% to 100%	Reduce image scale Continued
Warping mode	50%	all	Fill empty areas
Color add	50%	-45 to 45	Adds to the image value
Color multiply	50%	50% to 150%	Multiplies the image value

Augmentation during model learning allows for multiple image variants, but each base image is only shown once per cycle. Multiple distinct data is still important for the model to learn effectively.

ARCHITECTURE.

In our proposed approach is based on the study of existing works in the literature and research papers in the field of " Deep Learning for Image Recognition " [37] and object detection, specifically insect detection. For example Fuentes et al [38] compared the performance of different deep network architectures for insect detection and showed that Faster R-CNN could effectively recognize insect detection and plant pests with the ability to cope with complex scenarios in an environment. Fig.2, Illustrate the method Faster R-CNN used in our work is an improvement of R-CNN in its accuracy and speed a training (Ren, He, Girshick Sun, 2015)[32].

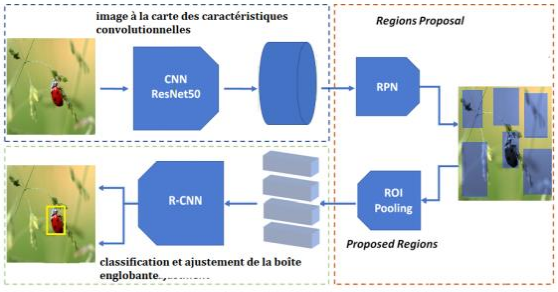

Fig. 2: Faster RCNN architecture used in our work

The method used obtains a trained CNN (ResNet50) is used to extract feature maps of the last convolution layer. A region proposal network (RPN) generates anchor boxes and potential ROIs, which are classified by a separate network. An ROI pooling layer extracts features for each insect, which are passed to classification and regression subnetworks to output a probability distribution and bounding box coordinates, respectively.

Faster-RCNN object detection is Faster in the TensorFlow library [**39**], the configuration of the TensorFlow pipeline can be divided into different steps. First, the model is configured while different parameters are defined in this step, such as the number of classes in the model, the characteristics of the feature extractor, the meta-architecture and the characteristics of the loss function. The best parameter selection will depend on the application.

IV. EXPERIMENTS AND RESULTS

Among the various multi-class metrics that have been studied in [19], we selected the most representative to assess the model's performance. The mean average precision (map) values are calculated over the area under the Precision-Recall curve (PR).

The formula of precision and recall are given as such:

$$\text{Precision} = \frac{TP}{TP+FP} \; ; Recall = \frac{TP}{TP+FN} \quad \textbf{(4) (5)}$$

Precision and recall are two commonly used metric to judge the performance of model, the precision value is given as the ration TP and the total number of predicted positives, the recall, true positive rate, is defined as the ration of TP out of all predictions (TP+FN).

With *TP*, *FP*, and *FN* are the number of true pixel-level positives, false positives, and false negatives, and may have been computed for each semantic class.

The 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. (5)

$$mAP = \frac{\sum_{C=0}^{C} AP(c)}{C} \quad \textbf{(6)}$$

The results illustrate in Table 3 shows the insect detection model's performance with a total mAP accuracy of 79.49%. The data was split into a training set (80%) and testing set (20%) randomly.

TABLE 3: ACCURACY TABLE FOR EACH CLASS.

Classes	mAP
Araneae	0,839435
Coleoptera	0,747686
Diptera	0,713036
Hemiptera	0,767915
Hymeno	0,634429
Lepidoptera	0,885707
Odonata	0,976389
TOTAL = 0.794942	

The model was evaluated on seven classes of insects, the results of each class listed in table 3.Adonata having the highest mAP score of 97,63% and hymenoptera being the class with which the model struggled the most, achieving an mAP of 63,4%. The model achieved mAP scores of 83,94%, 74,76%, 71,3%, 76,79% and 88,57% for Araneaer, Coleoptera, Diptera, Hemiptera, and Lepidoptera, respectively.

The model was evaluated on seven classes of insects. Adonata had the highest mAP score of 97.63%, while hymenoptera was the class with the lowest mAP score of 63.4%. The results for each class are listed in Table 4.

TABLE 4: GENERAL INFORMATION TABLE OF OUR MODEL.

Fonction	accuracy
Loss/BoxClassifierLoss/classification-loss	0.12712845
Loss/BoxClassifierLoss/localization-loss	0.056957208
Loss/RPNLoss/localization-loss	0.17811374
Loss/RPNLoss/objectness-loss	0.09159783
Loss/total-loss	0.45379785
PascalBoxes-Precision/mAP@0.5IOU	0.7949424
global-step	60000
Learning-rate	2e-06
loss	0.45379785
Loss for final step	0.24441578
Inference time per image (s)	0.195
Training time (h)	70.1

The model used achieved an mAP of 79,49% after 60,000 iterations of training, but it took around 70 hours for training, and the detection time per image was 0.195 seconds, indicating a significant requirement of computational resources and time for insect detection compared to the proposed method.

The loss function measures the deviation between the model predictions and the actual observations of the dataset used during training results a total loss of 45,37% and classification loss of 12,71% , and localization loss of 5,69%, an RPN for localization loss of 17,81%, an RPN loss/objectness loss of 9,15%, and a loss for the final step of 34,4%.

V. PREDICTION AND DISCUSSION.

Our model provided good results and correctly detected the object in some images, as shown in fig.4, where HYMENOPTERA was well detected.

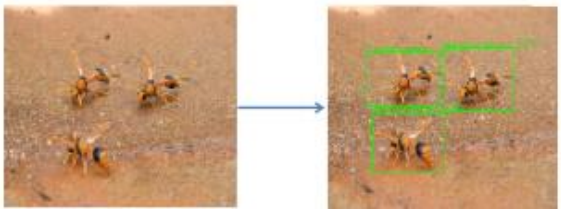

Fig. 4: Correct detection of HYMENOPTERA.

Fig. 5: Correct detection of COLEOPETRA.

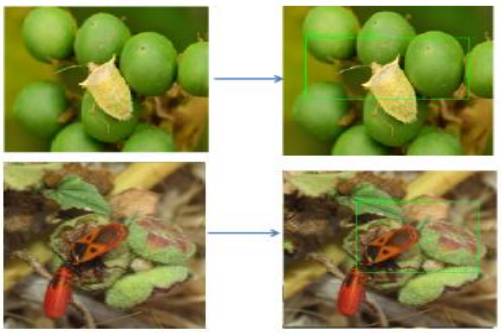

Fig. 6: Correct detection of HEMIPETRA.

The model's effectiveness in handling the differences in shape, color, and background within the insect classes is shown in figures 4 and 5 and 6. Feature extraction and occlusion processing are essential elements for multi-object detection, and the insects' diversity is taken into account.

The model can accurately detect objects in blurred image representations, as shown in fig.5 for DIPETERA.In order to test other positions, fig.6 contains two images of HEMIPETRA species. The first one is well detected, but in the second image which contains two species, we note that despite the insects being separated, the model fails to detect one of them. This may be due to the incorrect positioning of the insect.

The results showed that the detection system is effective and feasible, particularly in cases where the image contains only one object class. The model is capable of accurately detecting objects in different backgrounds. However, false detections were present in our model, and its detection ability is insufficient in some cases.

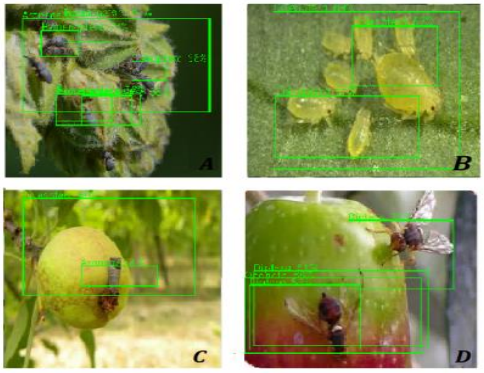

Fig. 7:.Incorrect detection

Fig .7 (a), (b) and (c) brings multiple insects cross from the same pest and the model badly detected accurately. This

indicates that when the overlapping area of adjacent pests is large, the results of multi-object detection are not good. It is difficult to distinguish whether the overlapping objects are independent individuals or not.

The testing results are shown in Table 5.we notice that the results obtained for each class are separated. The table displays information about the insects that require predictions, along with the associated agriculture categories and the final prediction results of our model.

Overall, The model achieved a 100% insect detection accuracy for the ARANEAE class of the LATRODECTUS insect and for the ODONATA class of dragonfly insects. However, lower results were obtained for other insect classes, such as 52.77% for the Hymenoptera class of VESPA insects and 55.5% for the Coleoptera class of Parallelelepipedus insects. The overall average detection rate for all insect classes in the model was 82.19%.

TABLE 5: PREDICT FINAL RESULTS

Insects	Class	image	Obje -ct	Detected Objects	Percent a-ge
Tuta Absoluta	Lepidoptera	14	16	15	93.75%
Cydia	Lepidoptera	18	22	16	72.72%
Castaneum	Coleoptera	10	21	16	76.19%
Parallelepipedus	Coleoptera	10	18	10	55.5%
Latrodectus	Araneae	7	7	7	100%
Tetranychus	Araneae	10	11	10	90.90%
Dragonfly	Odonata	16	18	18	100%
Bactrocera	Diptera	21	28	27	96.42%
Tephritoidea	Diptera	11	15	14	93.33%
Vespa	Hymenoptera	10	36	19	52.77%
Acyrthosiphon	Hemiptera	10	29	20	68.96%
Myzus	Hemiptera	9	20	18	90%
Total		146	241	190	78.83%

CONCLUSION

In this article, we present our model that uses deep learning techniques, specifically the Faster R-CNN algorithm, computer vision algorithms, and image processing, to identify and classify insects.

The results of the model suggest that there are still some limitations in accurately detecting insects due to issues like image quality, lighting, object positioning, and the need for more effective data augmentation techniques. Therefore, there is a need to further improve the detection system by addressing these factors.

However, the proposed method has limitations, including target detection errors and a low ability to detect small objects. To improve the method's performance, we can increase the insect database and try more suitable models for extracting useful insect areas from images. Additionally, the classification of insects should be more detailed, including information on their growth periods. In future work, we will focus on developing new attention modules and incorporating hardware components, specifically in the field of the Internet of Things (IoT), to reduce computational costs. This will also help reduce the use of chemical pesticides in agriculture, while taking timely preventive measures to avoid economic losses.

REFERENCES

[1] Larios, N., Deng, H., Dietterich, T.G., et al., 2008. Automated insect identification through concatenated histograms of local appearance features. Mach. Vis. Appl. 19 (2), 105–123.

[2] Yaakob, S.N., Jain, L., 2012. An insect classification analysis based on shape features using quality threshold ARTMAP and moment invariant. Appl. Intell. 37, 12–30.

[3] Wang, J., Lin, C., Ji, L., Liang, A., 2012. A new automatic identification system of insect images at the order level. Knowl.-Based Syst. 33, 102–110.

[4] Weeks, P.J.D., O'Neill, M.A., Gaston, K.J., Gauld, I.D., 1999. Species identification of wasps using principal component associative memories. Image Vis. Comput. 17 (12), 861–866.

[5] Zhao, Y., He, Y., Xu, X., 2012. A novel algorithm for damage recognition on pest-infested oilseed rape leaves. Comput. Electron. Agric. 89, 41–50.

[6] Geoffrey E. Hinton, Simon Osindero.Fitch, Fast Learning Algorithm for Deep Belief Nets F.:1944, 'Review of McCulloch and Pitts 1943', Journal of Symbolic Logic 9(2), 49–50.

[7] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In European Conference on Computer Vision (ECCV), 2018. https://arxiv.org/abs/ 1712.00559.

[8] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher kernel for large-scale image classification. In European Conference on Computer Vision (ECCV),2010 https://www.robots.ox.ac.uk/~vgg/rg/papers/peronnin_etal_ECCV10.pdf.

[9] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath et al., "Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups," IEEE Signal Process. Mag., vol. 29, no. 6, pp. 82–97, 2012.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet : A large-scale hierarchical image database," in CVPR, 2009.

[11] L. Deng, M. L. Seltzer, D. Yu, A. Acero, A.-r. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep autoencoder," in INTERSPEECH, 2010. Ii

[12] G. Dahl, A.-r. Mohamed, G. E. Hinton et al., "Phone recognition with the mean-covariance restricted boltzmann machine," in NIPS, 2010.

[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing coadaptation of feature detectors," arXiv :1207.0580, 2012.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in NIPS, 2012.

[15] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat : Integrated recognition, localization and detection using convolutional networks," arXiv :1312.6229, 2013.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in CVPR, 2015.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv :1409.1556, 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in CVPR, 2016.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in ICML, 2010.

[20] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in ICIP, 2002.

[21] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," J. of Comput. Sys.Sci., vol. 13, no. 5, pp. 663–671, 1997.

[22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, pp. 1627–1645, 2010.
[23] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge 2007 (voc 2007) results (2007),” 2008.
[24] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[25] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu, “Predicting eye fixations using convolutional neural networks,” in CVPR, 2015.
[26] E. Vig, M. Dorr, and D. Cox, “Large-scale optimization of hierarchical features for saliency prediction in natural images,” in CVPR, 2014.
[27] Touzet, C. (1992). les reseaux de neurones artificiels, introduction au connexion nième. EC2.
[28] Lee, J., Kim, T., Park, J. Nam, J. (2017). Raw Waveform-based Audio Classification Using Sample-level CNN Architectures. CoRR, abs/1712.00866. arXiv : 1712.00866. de http://arxiv.org/abs/1712.00866
[29] He, K., Gkioxari, G., Dollar, P. Girshick, R. B. (2017). Mask R-CNN. CoRR, abs/1703.06870. arXiv:1703.06870. http://arxiv.org/abs/1703. 06870
[30] Girshick, R. B., Donahue, J., Darrell, T. Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524.arXiv: 1311.2524. :http://arxiv.org/abs/1311.2524
[31] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.
[32] Ren, S., He, K., Girshick, R. B. Sun, J. (2015). Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks. CoRR, abs/1506.01497.arXiv: 1506.01497. http://arxiv.org/abs/1506.01497.
[33] He, K., Gkioxari, G., Dollar, P. Girshick, R. B. (2017). MaskR-CNN.CoRR,abs/1703.06870.arXiv: 1703.06870 . arxiv.org/abs/1703. 06870.
[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.
[35] Wei Liu1, Dragomir Anguelov , Dumitru Erhan, Christian Szegedy. Single Shot MultiBox Detector : 1UNC Chapel Hill 2 Zoox Inc. Google Inc. University of Michigan, Ann-Arbor https://arxiv.org/pdf/1512.02325.pdf
[36] Object Detection via Region-based Fully Convolutional Networks Jifeng Dai,Microsoft Research,Yi Li,Tsinghua University,Kaiming He.R-FCN : Object Detection via Region-based Fully Convolutional Networks Microsoft Research. University of Michigan, Ann-Arbor de https://arxiv.org/pdf/1605.06409.pdf
[37] L. S. Fu, Y. L. Feng, E. Tola, Z. H. Liu, R. Li, and Y. J. Cui, “Image recognition method of multi-cluster kiwifruit in field based on convolutional neural networks,” Transactions of the Chinese Society of Agricultural Engineering, vol. 34, pp. 205–211, 2018.View at : Google Scholar
[38] A. Fuentes, D. H. Im, S. Yoon, and D. S. Park, “Spectral analysis of CNN for tomato disease identification,” in Proceedings of the International Conference on Artificial Intelligence and Soft Computing, pp. 40–51, Cham, Switzerland, June 2017. View at : Google Scholar
[39] Ren, S., He, K., Girshick, R. B. Sun, J. (2015). Faster R-CNN : Towards Real-Time bject Detection with Region Proposal Networks. CoRR, abs/1506.01497. arXiv : 1506.01497. dehttp://arxiv.org/abs/1506.01497
[40] Arthropod Taxonomy Orders Object Detection Dataset ,License CC BY-NC-SA 4.0 : https://www.kaggle.com/mistag/ arthropod-taxonomy-orders-object-detection-dataset.