


Article

Insect Pest Image Recognition: A Few-Shot Machine Learning Approach including Maturity Stages Classification

Jacó C. Gomes ¹  and DÍbio L. Borges ^{2,*} 

¹ Department of Mechanical Engineering, University of Brasília, Brasília 70910-900, Brazil; cirino.jaco@gmail.com

² Department of Computer Science, University of Brasília, Brasília 70910-900, Brazil

* Correspondence: dibio@unb.br

Abstract: Recognizing insect pests using images is an important and challenging research issue. A correct species classification will help choosing a more proper mitigation strategy regarding crop management, but designing an automated solution is also difficult due to the high similarity between species at similar maturity stages. This research proposes a solution to this problem using a few-shot learning approach. First, a novel insect data set based on curated images from IP102 is presented. The IP-FSL data set is composed of 97 classes of adult insect images, and 45 classes of early stages, totalling 6817 images. Second, a few-shot prototypical network is proposed based on a comparison with other state-of-art models and further divergence analysis. Experiments were conducted separating the adult classes and the early stages into different groups. The best results achieved an accuracy of 86.33% for the adults, and 87.91% for early stages, both using a Kullback–Leibler divergence measure. These results are promising regarding a crop scenario where the more significant pests are few and it is important to detect them at earlier stages. Further research directions would be in evaluating a similar approach in particular crop ecosystems, and testing cross-domains.

Keywords: few-shot learning; insect pest classification; insect maturity stages; RGB images



Citation: Gomes, J.C.; Borges, D.L. Insect Pest Image Recognition: A Few-Shot Machine Learning Approach including Maturity Stages Classification. *Agronomy* **2022**, *12*, 1733. <https://doi.org/10.3390/agronomy12081733>

Academic Editor: Louis Kouadio

Received: 28 June 2022

Accepted: 15 July 2022

Published: 22 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Crop yields are subject to many threats and conditions, such as pathological agents, mismanagement of soil nutrients, and climate changes to name a few. Insect pests inflict high damage on every crop, if not controlled, and a warming climate scenario may increase insect infestations and losses, especially in tropical areas [1].

Insect pests are a major cause of concern in crops because of yield losses and the intensive use of broad-spectrum insecticides [2]. Although, integrated pest management (IPM) practices have attained importance, there are still lacks of precision on timely identifying hazardous species of insects during a crop cycle [3]. If identified more precisely, and at early stages, monitoring and controlling mitigation strategies could be brought in avoiding economic losses, and helping in more sustainable practices [2].

The similarities between insect species, especially at the same maturity stages, make conventional manual identification imprecise, time-consuming, and inefficient in most cases, even for experienced agronomists [4]. Visual-based machine learning algorithms can effectively address this issue. Using images to help classify insects for pest management is a major research topic lately since the advance of machine learning techniques [5,6]. Deep learning is one of the most widely used approaches for insect classification tasks in agriculture as demonstrated in [7–10]. However, supervised models require large labeled data sets for training these models, which are scarce, very demanding, and are still far from being able to bridge the gaps in insect classes variability [11]. Besides, computer vision and deep learning methods may supply novel cost-efficient and automated sensor techniques to the field of entomology [4].

In recent years, many automated recognition systems, based on computer vision and machine learning, were proposed to manage insect pests in agriculture. Karar et al. [12] proposed a mobile application to classify five classes of insect pests using deep learning in cloud computing. Chen et al. [13] proposed an embedded drone system and deep learning to recognize insects in a tree. Li et al. [7] studied five state-of-the-art deep learning architectures for image recognition of ten categories of crop pests. Thenmozhi and Reddy [14] proposed an improved deep convolutional network, outperforming fine-tuned models in insect pest recognition. Deep learning have been the most used method for visual insect recognition using image data sets, but no one has yet approached it separating maturity stages and using few samples.

Recently, the data set IP102 [15] with 75,000 images of 102 categories of insects, mixing samples in different life stages such as egg, larva, pupa, and adults, has been put together and made available for researching this topic. Although being a major advance as data availability for insect pest recognition, the IP102 is out of proportion in many species [15]. Moreover, the IP102, with different life stages of insect classes together, makes the automatic visual recognition task even more difficult mainly due to its structural intra-class large morphological samples [15].

Learning from a Small Amount of Data: Few-Shot Learning

Machine learning is a sub-area of artificial intelligence where computer programs are designed to solve tasks T , based on gathering experience E , and approximating an objective function using a performance measure P [16]. Despite its success in approaching data-intensive applications, getting big amounts of supervised data (i.e., the experience E) is not always feasible. Learning from a small amount of samples may be possible though, if prior knowledge of few categories can be grouped and subsequently applied to further categories [17]. Few-shot learning (FSL) refers to this problem of learning using few samples, with interesting scenarios, approaches, and learning issues depending on the area addressed [18].

Few-shot learning (FSL) [18] is a learning approach that seeks to define a relative approximation between machine and human learning considering the challenging task of learning from very few samples. One important category of FSL methods is metric-based meta-learning [19]. Figure 1 provides an overview of FSL metric-based with meta-learning paradigm. Given a labeled data set for training, from a particular problem, the goal is to learn concepts in embedding space, through training tasks, to generalize classes in test tasks from a novel problem by using a similarity metric. Convolutional neural networks (CNN) are commonly used as embedding functions f and g for image feature extraction.

In this paradigm, an FSL model is typically trained through several N -way and K -shot classification tasks. A classification task is referred to as a training episode. In an episode, the support set S is composed of N classes containing K samples from each of them (i.e., $S = N \times K$), and the query set Q consists of q samples from the same classes (i.e., $Q = N \times q$). In Figure 1, a task is composed of two-way, two-shot, and $Q = 1$ from a particular class for demonstration. The model goal is to label Q images into N classes of the task. Furthermore, in meta-learning guidelines, a source set is used for training n tasks and a Target set for test tasks, there is no overlap between classes in Source and Target sets.

At present, several areas have benefited from FSL approaches, including image classification [20], and object detection in images [21], with great potential for agricultural applications. Few-shot enables the construction of models with drastic parameters reduction, which facilitates the application in embedded, mobile systems. Li and Yang [22] classified cotton crop pests using prototypical nets in an embedded terminal. Yang et al. [23] improved the results of prototypical nets by combining recognition and object localization. They proposed a salient region detection mechanism, which represents the region with the highest discriminatory characteristics for insect classification. Li and Yang [19] analyzed the cross-domain few-shot classification problem in agriculture. They used insect and plant

leaf diseases data sets. Their results showed that the mixed domain, in which meta-training and meta-testing use classes of both types of data together, produces better results.

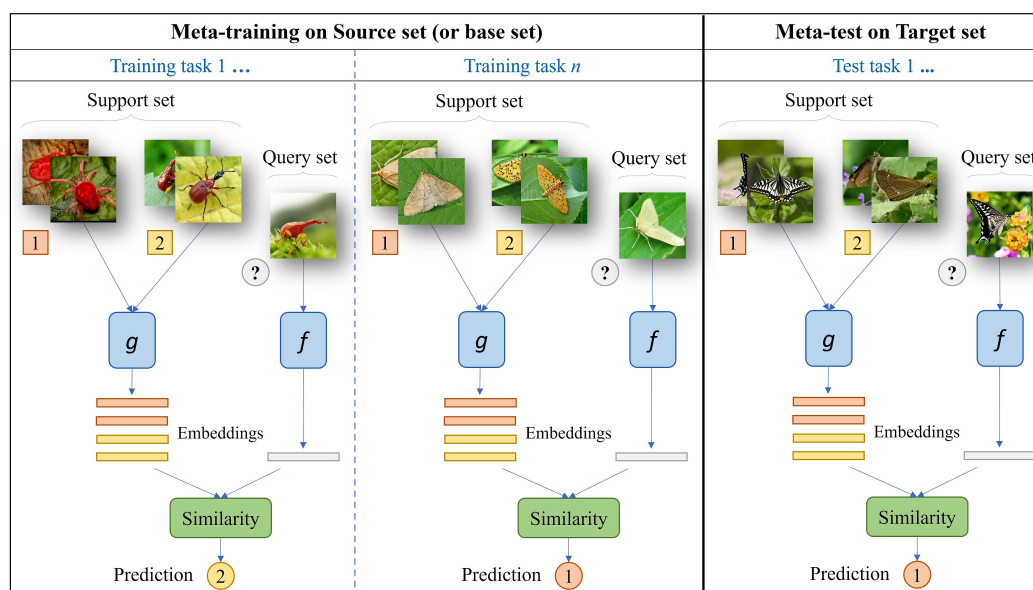


Figure 1. Meta-metric few-shot learning example representation. Illustrated settings of tasks are 2-way, 2-shot, and one query image. In some approaches, the embedding functions f and g may be the same. Query images are labeled according to a similarity score with support embeddings.

In this research we propose to address insect pest recognition by firstly putting together a different image data set (IP-FSL), derived of IP102, but distinguishing classes into two maturity stages: early and adults. The approach for classification is by framing the insect maturity stages classification problem in a few-shot learning paradigm, and then leveraging a prototypical network by including divergence measures as similarity functions. We see a need for an effective tool in agronomy for insect management to deal with rapid insect classification by maturity stages with field images. Our research approaches this problem by posing it in a few-shot learning paradigm. We achieved 86.33% and 87.91% of accuracy, respectively, for adult and early insect classes on the IP-FSL data set.

The remaining of this paper describes the materials and methods used, Section 2, and the Section 3 includes the description of the experiments. In Section 4, the results are shown; Section 5 discusses the results obtained, and finally conclusions are summarized in Section 6.

2. Materials and Methods

2.1. The Meta IP-FSL Data Set

IP102 [15] is a large insect pest data set that provides 75,222 images distributed in 102 classes. Most classes cover different stages of the insect life cycle such as egg, larva, pupa, and adult. Its taxonomy comprises two major agricultural crop groups: field crops (rice, corn, wheat, beet, and alfalfa), and economic crops (vitis, citrus, and mango).

We thoroughly analyzed the IP102 to rearrange classes, and to select samples, according to two biological stages of the pests, adult and early stages, and assembled a new data set, IP-FSL (insect pests for few-shot learning), for few-shot learning. Different maturity stages in the same class can make it difficult to learn patterns from a particular class, because visually they are far apart, and consequently, the resultant classifications of the learning algorithms may be misleading. By separating the biological maturity stages, we expect two advantages: (1) providing a more discriminative feature extraction for the classes, and (2) more accurate recognition in the early stages of the pest, which is important to control the spread of the insect pest.

We built IP-FSL by selecting a maximum of 50 images from each class in the IP102. This number was chosen because of the large diversity of the data set as a top limit, but the exact numbers of some categories are less as is shown in Table 1. For the Early stage subset, we considered those images containing the presence of egg, larva, or pupa. The subset Adult stage includes young and adult insects. As a selection criterion, images with field conditions were chosen. For the species containing images of both stages mentioned, we created new classes for the respective insect species in the two subsets.

The final configuration of the IP-FSL data set is presented in Figure 2. It has a total of 6817 insect images. The subset Early stage is composed of 45 insect classes, totaling 2050 images. The adult stage consists of 97 classes, totaling 4767 images. Figure 3 shows some examples of the IP-FSL data set, and class names and amounts are presented in Table 1.

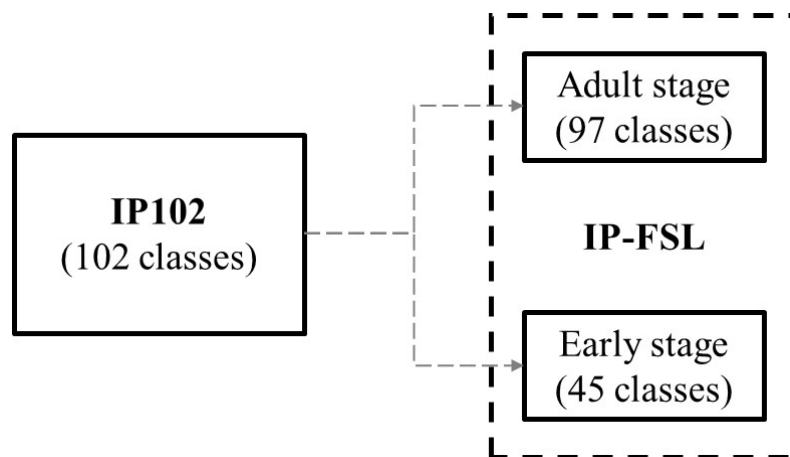


Figure 2. IP-FSL classes as derived and separated from IP102.

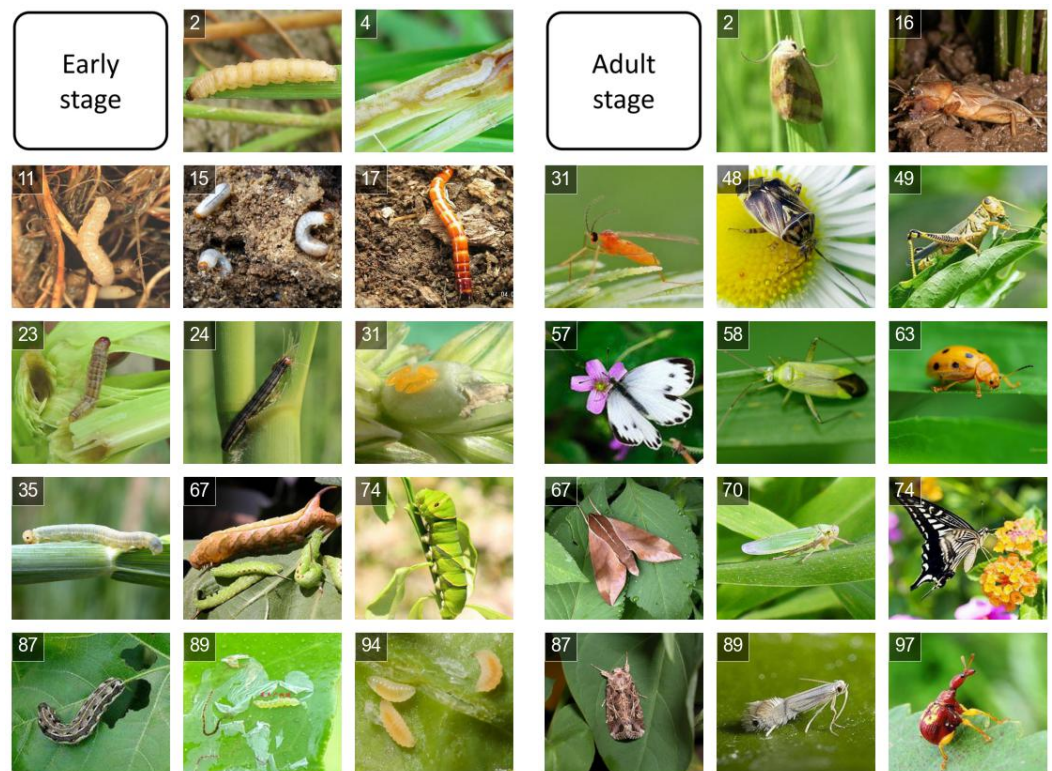


Figure 3. Some samples from the IP-FSL data set. Class labels refer to the numbers in Table 1. Some examples of classes with two maturity stages (adult and early, appearing in both subsets, classes 2, 31, 67, 74, 87, and 89).

Table 1. IP-FSL image data set information, derived from IP102 (Insect Pest 102) [15], assembled specifically for this few-shot learning research. The names of the insects were kept as published in the original source (IP102), and in the categories may contain common, as well as scientific, names.

Category Name	#Adult/#Early	Category Name	#Adult/#Early	Category Name	#Adult/#Early
1 rice leaf roller	50/50	35 wheat sawfly	50/50	69 <i>Xylotrechus</i>	50/-
2 rice leaf caterpillar	50/50	36 cerodonta denticornis	50/32	70 <i>Cicadella viridis</i>	50/-
3 paddy stem maggot	50/50	37 beet fly	50/-	71 <i>Miridae</i>	50/-
4 asiatic rice borer	50/50	38 flea beetle	50/-	72 <i>Trialeurodes vaporariorum</i>	50/-
5 yellow rice borer	50/50	39 cabbage army worm	50/50	73 <i>Erythroneura apicalis</i>	42/-
6 rice gall midge	50/31	40 beet army worm	50/50	74 <i>Papilio xuthus</i>	50/50
7 rice stemfly	50/47	41 Beet spot flies	50/50	75 <i>Panonychus citri</i> McGregor	50/-
8 brown plant hopper	50/17	42 meadow moth	50/25	76 <i>Phyllocoptes oleiverus ashmead</i>	-/50
9 white backed plant hopper	50/18	43 beet weevil	50/-	77 <i>Icerya purchasi</i> Maskell	50/-
10 small brown plant hopper	50/-	44 sericaorient alismots chulsky	50/-	78 <i>Unaspis yanonensis</i>	50/-
11 rice water weevil	50/50	45 alfalfa weevil	50/50	79 <i>Ceroplastes rubens</i>	50/-
12 rice leafhopper	50/-	46 flax budworm	50/50	80 <i>Chrysomphalus aonidum</i>	50/-
13 grain spreader thrips	50/-	47 alfalfa plant bug	50/-	81 <i>Parlatoria zizyphus</i> Lucas	44/-
14 rice shell pest	50/50	48 tarnished plant bug	50/-	82 <i>Nipaecoccus vastalor</i>	50/-
15 grub	-/50	49 <i>Locustoidea</i>	50/-	83 <i>Aleurocanthus spiniferus</i>	-/50
16 mole cricket	50/-	50 <i>lytta polita</i>	50/-	84 <i>Tetradacus c Bactrocera minax</i>	50/50
17 wireworm	50/50	51 legume blister beetle	50/-	85 <i>Dacus dorsalis</i> (Hendel)	50/40
18 white margined moth	26/50	52 blister beetle	50/-	86 <i>Bactrocera tsuneonis</i>	50/20
19 black cutworm	50/50	53 <i>therioaphis maculata</i> Buckton	50/-	87 <i>Prodenia litura</i>	50/50
20 large cutworm	50/50	54 <i>odontothrips loti</i>	50/-	88 <i>Adristyrannus</i>	50/40
21 yellow cutworm	50/50	55 Thrips	50/-	89 <i>Phyllocnistis citrella</i> Stainton	50/50
22 red spider	50/-	56 alfalfa seed chalcid	50/-	90 <i>Toxoptera citricidus</i>	50/-
23 corn borer	50/50	57 <i>Pieris canidia</i>	50/-	91 <i>Toxoptera aurantii</i>	50/-
24 army worm	35/50	58 <i>Apolygus lucorum</i>	50/-	92 <i>Aphis citricola</i> Vander Goot	50/-
25 aphids	50/-	59 <i>Limacodidae</i>	50/50	93 <i>Scirtothrips dorsalis</i> Hood	50/-
26 Potosiabre vitarsis	50/-	60 <i>Viteus vitifoliae</i>	-/50	94 <i>Dasineura</i> sp.	33/50
27 peach borer	50/50	61 <i>Colomerus vitis</i>	-/50	95 <i>Lawana imitata</i> Melichar	50/-
28 english grain aphid	50/-	62 <i>Brevipoalpus lewisii</i> McGregor	47/-	96 <i>Salurnis marginella</i> Guerr	50/-
29 green bug	50/-	63 <i>oides decempunctata</i>	50/-	97 <i>Deporaus marginatus</i> Pascoe	50/-
30 bird cherry-oataphid	50/-	64 <i>Polyphagotars onemus latus</i>	50/-	98 <i>Chlumetia transversa</i>	50/50
31 wheat blossom midge	50/50	65 <i>Pseudococcus comstocki</i> Kuwana	50/-	99 <i>Mango flat beak leafhopper</i>	50/-
32 <i>penthaleus major</i>	50/-	66 <i>parathrene regalis</i>	40/30	100 <i>Rhytidodera bowrinii</i> white	50/-
33 longlegged spider mite	50/-	67 <i>Ampelophaga</i>	50/50	101 <i>Sternochetus frigidus</i>	50/-
34 wheat phloeothrips	50/-	68 <i>Lycorma delicatula</i>	50/-	102 <i>Cicadellidae</i>	50/-
				TOTAL	4767/2050

2.2. Metric-Based Multi-Class Networks

FSL algorithms learn through tasks to adapt to new tasks, as shown in Figure 1. Matching [24] and Prototypical [25] networks are competitive few-shot metric-based multi-class approaches. Matching and prototypical networks originally use cosine and Euclidean distances as similarity measures. We propose here to leverage and evaluate those frameworks using other divergences, such as Mahalanobis, Kullback–Leibler, and Itakura–Saito. This group of divergences, also called Bregman divergences, measures differences between distributions, and as we are going to show in this research, can produce even better results in this FSL setting.

2.2.1. Matching Networks

Matching networks [24] are examples of multi-class classification. They consist of two embedding functions (f and g , being appropriate convolutional neural networks (CNN), and potentially $f = g$) for feature extraction. An attention mechanism uses the cosine similarity to compare a test sample \hat{x} with samples in the support set, where the class probability is obtained, as given in Equation (1):

$$P(\hat{y}|\hat{x}, S) = \sum_{i=1}^{N \times K} a(\hat{x}, x_i) y_i, \quad (1)$$

in which $a(\cdot, \cdot)$ is the attention mechanism described as Equation (2):

$$a(\hat{x}, x_i) = \frac{\exp(c(f(\hat{x}), g(x_i)))}{\sum_{j=1}^{N \times K} \exp(c(f(\hat{x}), g(x_j)))}, \quad (2)$$

$c(\cdot, \cdot)$ is the cosine similarity, and f and g are the embedding functions.

In general, matching networks changes the way samples are embedded, matching the support set S to the support and query samples, through a full context embeddings (FCE) process. Query and support images go through f and g structures, respectively, for feature extraction. Matching nets predict the probability of query samples by measuring the cosine similarity between support and query embeddings.

2.2.2. Prototypical Networks

Prototypical nets (ProtoNet) architecture [25] consists of a CNN for image features extraction, and a classifier based on Euclidean distance. The main idea is that the centroid of support embeddings (prototypes) yields relevant class representatives. ProtoNet aims to learn a metric in the feature space that represents a similarity by distance for image predictions. Query images are labeled by finding the closest class prototype.

Each prototype corresponds to the average of the class embeddings, calculated according to Equation (3):

$$c_n = \frac{1}{K_n} \sum_{i=1}^{K_n} f_{\Phi}(X_i), \quad (3)$$

where c_n represents the centroid of the class n .

Query images are classified according to a probability distribution. Such probabilities are given by softmax over distances between prototypes and query embeddings, according to Equation (4):

$$p_{\phi}(y = n|X) = \frac{\exp(-d(f_{\Phi}(X), c_n))}{\sum_{n'} \exp(-d(f_{\Phi}(X), c_{n'}))}. \quad (4)$$

ProtoNet learning proceeds by minimizing the negative log-probability $J(\phi) = -\log_{\phi}(y = n|x)$ of the true class n via stochastic gradient descent (SGD).

The learning structure is an important factor in the metric models, but the performance depends on the chosen similarity metric [24,25]. In the next section, the concepts of other

divergences, not fully considered before for FSL frameworks, but used to quantify the similarity between distributions, are revised for further use in this proposal.

2.3. Leveraging FSL with Other Divergences

Bregman divergences have been applied to optimization, clustering, and machine learning problems [26–28], but not fully explored in FSL. This group of divergences establishes a generalized measure between distributions, defined in terms of a strictly convex function [29]. Therefore, given a continuously differentiable, strictly convex function, $F : S \rightarrow \mathbb{R}$, defined in a convex domain $S \subseteq \mathbb{R}^d$, a Bregman divergence between $\mathbf{x}, \mathbf{y} \in S$ induced by F , is defined as

$$D_F(\mathbf{x}, \mathbf{y}) = F(\mathbf{x}) - F(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla F(\mathbf{y}) \rangle, \quad (5)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, and $\nabla F(\mathbf{y})$ represents the gradient vector of F evaluated at \mathbf{y} .

Bregman divergences have pertinent properties, among them non-negativity $D_F(\mathbf{x}, \mathbf{y}) \geq 0$, in which $D_F(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$. Furthermore, with some exceptions, Bregman divergences are considered asymmetric, given that $D_F(\mathbf{x}, \mathbf{y}) \neq D_F(\mathbf{y}, \mathbf{x})$. The concepts of three main Bregman divergences for similarity measure are presented in the next sections.

2.3.1. Squared Mahalanobis Divergence

The Mahalanobis divergence, generated by the convex function $F(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$, is defined as a distance between a point and a distribution. For this reason, it takes into account the covariance between the variables. The Mahalanobis distance between a vector \mathbf{x} and a distribution \mathbf{y} can be calculated by Equation (6):

$$D_F(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{A} (\mathbf{x} - \mathbf{y}), \quad (6)$$

which is called Mahalanobis distance when \mathbf{A} is the inverse of the covariance matrix.

Equation (6) attempts to solve the Euclidean distance problem when the data have a linear correlation. It has the effect of transforming variables into uncorrelated variables, by scaling them through the covariance matrix. That way, the Equation (6) corresponds to computing the Euclidean distance with scaled data.

In this work, the low time cost for estimating the covariance matrix was prioritized. Therefore, we assumed that the covariance estimation based on task prototypes yields relevant results with low training time. That way, \mathbf{x} corresponds to the query embeddings set and \mathbf{y} the prototypes set of a task. This approach allows using K-shot ≥ 1 without paradigm shifts in the covariance estimation algorithm.

2.3.2. Kullback–Leibler Divergence

KL-divergence, or relative entropy, is generated by the convex function of negative entropy, for discrete distribution $\sum_{j=1}^d x_j \log_2 x_j$. It quantifies the difference between two probability distributions. Bregman divergence between two discrete probability distributions, which corresponds to the convex function generating the KL-divergence, is described as:

$$D_F(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^d x_j \log_2 \left(\frac{x_j}{y_j} \right) = KL(\mathbf{x} || \mathbf{y}). \quad (7)$$

For the experiments in this work, the embeddings and prototypes from the task were transformed into probability vectors for the KL-divergence computation. In other words, given a feature vector \mathbf{y}' , the new probability vector is calculated as $\mathbf{y} = \mathbf{y}' / \text{sum}(\mathbf{y}')$ such that $\sum_{j=1}^d y_j = 1$. A constant ϵ was added to the vectors before calculating the divergence to avoid infinite negative results ($\log 0$) or division by zero.

2.3.3. Itakura–Saito Divergence

The Itakura–Saito divergence, or IS-divergence, is an asymmetric measure widely used in signal processing. IS-divergence is generated by the function $F(x) = -\log x$, and it can be calculated as follows:

$$D_F(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^d \frac{x_j}{y_j} - \log\left(\frac{x_j}{y_j}\right) - 1 = IS(\mathbf{x}||\mathbf{y}). \tag{8}$$

As in the KL computation, probability vectors are generated and a constant ϵ added for IS-divergence computation.

3. Experiments

The experiments were conducted using two main few-shot models, prototypical and matching networks. We organized the experiments in three scenarios: (I) classification of mini-imagenet data set as a baseline experiment for choosing the best network, (II) insect classification at the adult maturity stage, and (III) insect classification at the early stage.

3.1. Episode Training Process

3.1.1. Prototypical Nets

The episode training of the prototypical net starts randomly selecting N classes from the source set. Figure 4 presents the framework with a three-way task for demonstration. After that, the data for the respective task is divided into support set S, and query set Q, according to the parameters N-way, K-shot, and q, previously assigned, and the CNN embeds all images to generate support and query embeddings. Prototypes are computed from the support set as class representatives. The divergences are then computed between prototypes and query embeddings to classify Q images according to a probability distribution over divergences.

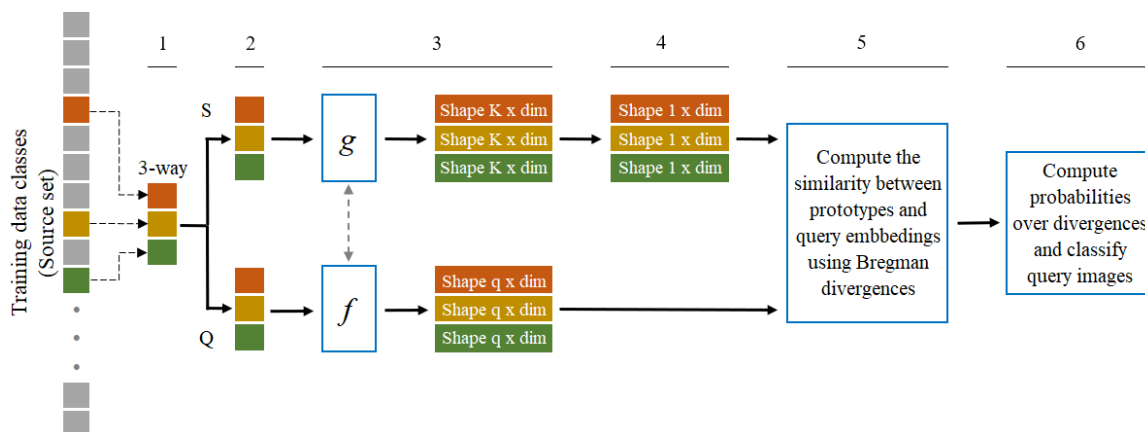


Figure 4. A generalized episode of prototypical networks, where ‘dim’ represents the dimension of the vectors. (1) Random selection of classes. (2) Samples division into support and query sets. (3) Embeddings generation. (4) Prototypes generation.

3.1.2. Matching Nets

The matching net episode training (Figure 5), differ to the prototypical episode in two aspects: (1) it uses a mechanism to generate full context embeddings (FCE), and (2) similarities are computed between query and support embeddings, instead of query and prototypes.

A test episode is similar to a training episode for both networks, except that the parameters of the models are frozen during testing time. Furthermore, the target set is used instead of the source set. Finally, the test episode ends with the classification of Q test images, where the accuracy of the model is computed.

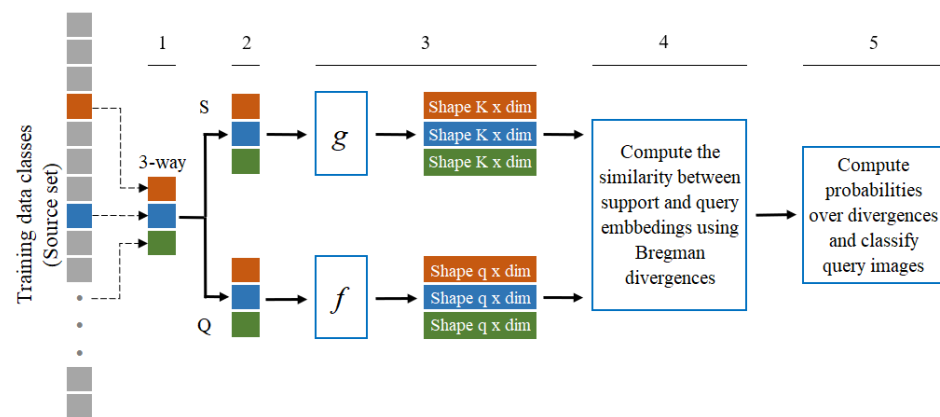


Figure 5. A generalized episode of Matching networks, where ‘dim’ represents the dimension of the vectors. (1) Random selection of classes. (2) Samples division into support and query sets. (3) Full context embeddings generation.

3.2. Experiment I

The first experiment evaluates prototypical and matching networks, along with different divergences, in a benchmark public data set. The goal is to choose the best model for insect classification, based on experimental results in a consolidated benchmark data set. Mini-ImageNet [24] is a widely used benchmark for few-shot classification. This data set consists of 100 classes, each containing 600 images. Here, the classes were divided into 80% for models training (source set) and 20% for testing (target set), following a conventional division training/testing [16]. We evaluated the models in one-shot and five-shot settings. Moreover, we used five-way tasks with $q = 15$.

3.3. Experiment II

This experiment aims to use the model with the highest accuracy provided by Experiment I. The training steps of the respective model, as presented in Section 3.1, were used to classify insects only in adult maturity stage. We divided the adult stage subset classes at a rate of 80:20 for training and testing [16], respectively. Therefore, the model was tested on classes unseen in training tasks.

Different divergences were evaluated as a few-shot similarity function in n -way k -shot parameters settings to obtain the best model performance. For all experiments, therefore, we analyzed one-shot and five-shot in three-way and five-way tasks. In addition, $q = 5$ was fixed for all experiments.

3.4. Experiment III

The third experiment consists of insect classification only at the early maturity stage. The procedures match with those in Experiment II, including the network used. We divided early stage subset classes at a rate of 80:20 for training and testing [16], and then evaluated the classification tasks for one-shot and five-shot settings related to three-way and five-way tasks. Moreover, we set $q = 5$.

3.5. Experiments Setups

All training and testing setups were equally performed for insect classification into two maturity stages. The model inputs color images (RGB) without any image preprocessing. However, some transformations were carried out to standardize and increase the number of classes. Initially, all the images in IP-FSL were resized to $96 \times 96 \times 3$ format, and rotated in multiples of 90° to create new augmented classes. Thus, after multiple rotations up to 270° , each subset ended with four-fold the initial number of classes, keeping the same number of samples in each class.

The experiments were conducted through the Google Colaboratory platform. The Pytorch library version 1.9.0 was used for writing and training the models. The model was

trained using the source set for 10 epochs, with 2000 episodes/epoch. We carried out 20,000 training episodes in each combination setting of N-way, K-shot, and divergence. The initial learning rate of 10^{-3} falls in half after each epoch.

In few-shot learning, the results are commonly presented as the average of several testing tasks. In this work, the average accuracy of 1000 testing episodes for each experiment is computed and shown.

4. Results

The models learn image features to differentiate classes through a set of divergences. In addition to the Euclidean, we investigated the results of the Mahalanobis, Kullback–Leibler, and Itakura–Saito divergences. Our implementation, therefore, integrates these divergences as dissimilarity measures for Mini-ImageNet data set classification in Experiment I, and for insect classification of adult and early life cycle, through Experiments II and III, respectively.

4.1. Experiment I: Mini-ImageNet Classification

The Experiment I was carried out to evaluate the Prototypical and Matching networks performance on the Mini-ImageNet data set, with the proposed divergences in order to choose the most appropriate. Table 2 presents these results, in which bold numbers indicate the best accuracy for each model and related K-shot setting.

Table 2. Results for Experiment I (Mini-ImageNet). ED: Euclidean distance, MD: Mahalanobis distance, KL: KL-divergence, IS: IS-divergence.

Model	One-Shot				Five-Shot			
	ED	MD	KL	IS	ED	MD	KL	IS
Prototypical networks	0.4979	0.4389	0.5179	0.5008	0.6986	0.6270	0.7097	0.6984
Matching networks	0.4900	0.5280	0.5260	0.5360	0.6480	0.6300	0.6620	0.6104

From the results in Table 2, both networks show close accuracy to each other. However, prototypical networks achieved the highest accuracy of 0.7097. Because of this, further experiments were performed with prototypical networks on the IP-FSL data set.

4.2. Experiment II: Adult Stage Insect Classification

For the adult stage subset, prototypical networks training and testing were carried out in a meta-learning procedure, that is, performed on source and target sets, respectively, without classes overlap. A chain of 16 experiment types was carried out for evaluation. For each of the divergences investigated, we evaluated tasks of three-way, five-way, and one-shot, five-shot settings, and results are presented in Table 3.

Table 3. Results for Experiment II: prototypical networks on adult stage subset.

N-Way	One-Shot				Five-Shot			
	ED	MD	KL	IS	ED	MD	KL	IS
Three-way	0.7434	0.7568	0.7797	0.7595	0.8435	0.8527	0.8633	0.8471
Five-way	0.6216	0.6321	0.6694	0.6580	0.7615	0.7476	0.7743	0.7768

4.3. Experiment III: Early Stage Insect Classification

For the early stage experiments, model training and testing were also performed by meta-learning paradigm using source and target sets, respectively, without class relationship between both. Table 4 gives the results.

Table 4. Results for Experiment III: prototypical networks on early stage subset.

N-Way	One-Shot				Five-Shot			
	ED	MD	KL	IS	ED	MD	KL	IS
Three-way	0.8045	0.7920	0.8167	0.8107	0.8621	0.8670	0.8791	0.8778
Five-way	0.6758	0.6786	0.6906	0.6859	0.7722	0.7780	0.8072	0.8044

5. Discussion

In this study we have addressed the important problem of insect pest image recognition, adult and early stage maturity categories, using few samples and a leveraged prototypical network learning approach. Since insects at different stages can damage crops to different levels, recognizing specific stages can mitigate the spread and the impact of further damage on crops. For this reason, we assembled the IP-FSL based on two life cycles, and we designed a few-shot experimental approach to differentiate insects by leveraging state-of-art models with other divergences for similarity measurement, and compared their performance. Encouraging general results were obtained with respect to the two maturity stages, comparable with literature results but using fewer samples. We have achieved high accuracy in both categories, adult and early, of 86.33% and 87.91%, respectively.

Insect pest recognition is a challenging and relevant issue in agriculture, and entomology in general [4]. Practical applications require rapid and accurate visual recognition to control infestations in crops. Two of the ways to address it are using few-samples machine learning algorithms, and learn them in a specific maturity stage of the pests, separately. In our approach we propose to have insect maturity stages categories addressed separately, since visually they are far distinct.

Figure 6 presents a sample classification task performed in Experiment II, for the adult insect recognition, in which 15 query images are labeled according to three-way classes. The performance for the three-way in Experiment II, given in Table 3, showed that our approach recognized insects with better accuracy of 77.97% in one-shot and 86.33% in five-shot using KL-divergence. In the five-way tasks, the best performance for one-shot achieved 66.4% using KL, and 77.68% in five-shot using IS-divergence, although KL came very close with 77.43% in five-shot. While IS outperforms KL in five-shot, the difference is very low (0.25%). We assume that there is an advantage of KL-divergence for the adult stage insect classification, since it was the most accurate by a larger margin in the three-way case. In contrast, Euclidean and Mahalanobis distances yielded considerably lower accuracy. KL and IS were shown to be promising approaches to measure the dissimilarity between adult insects, with the best accuracy achieved by KL, which improved the final performance of the few-shot model.

In the early stage insect classification (Experiment III), KL and IS also yielded better accuracy, with KL achieving best results in all settings. The three-way setting procedure is presented in Figure 7, where 15 query images are labeled according to three task classes. In this situation, KL-divergence achieved the better accuracy of 81.67% in one-shot, and 87.91% in five-shot. In five-way, KL is also the best similarity approach, achieving 69.06% and 80.72% for one-shot and five-shot, respectively.

As seen in the Experiments I, II, and III, accuracy increases as K-shot increases. K-shot represents the number of support images in each insect class. It is presumptive to say that there is more information learned by the network when a greater number of class images are explored, possibly it can be enhanced in 10-shot. But in a few-shot context, it is important that K is not high.

In contrast to K-shot, accuracy increases when N-way is smaller. N-way represents the number of classes within a classification task, for which the model needs to label the query images. It is also presumptive to say that a greater range of classes to label query images results in greater difficulty in correctly classifying them.

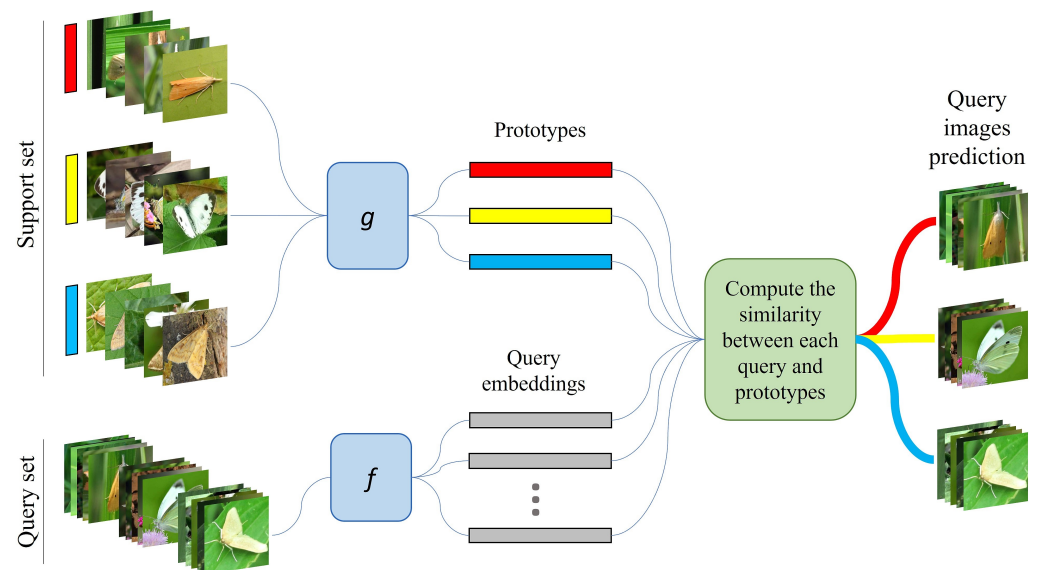


Figure 6. Instance of Adult stage insect classification in 3-way, 5-shot and $q = 5$.

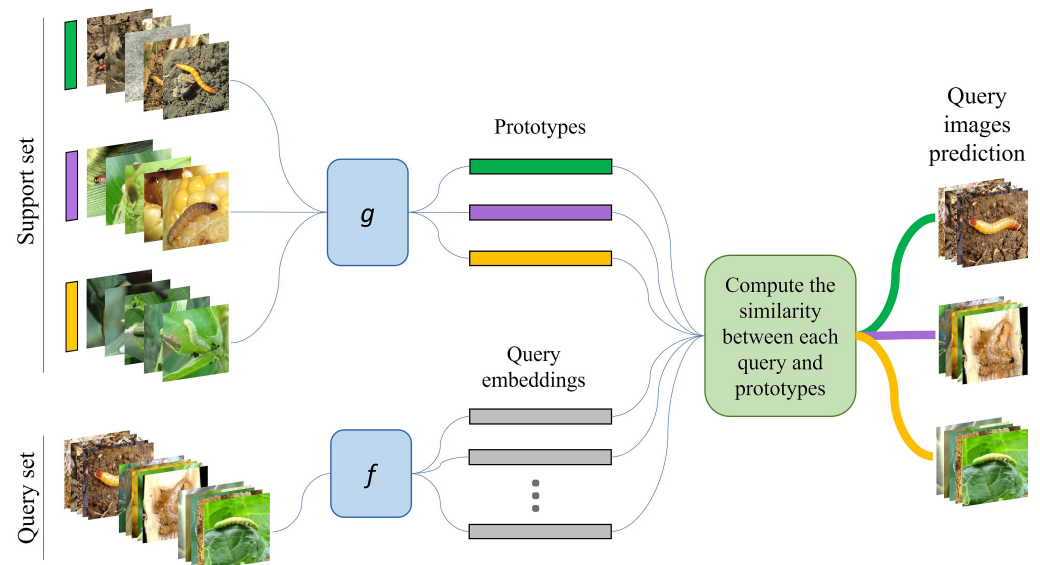


Figure 7. Instance of early stage insect classification in 3-way, 5-shot and $q = 5$.

We observed that the insects in the early maturity stages are more accurately classified in IP-FSL. A possible reason is that adult insects have a higher visual similarity, which makes it difficult to label images correctly. This may explain the importance of identifying a more suitable similarity metric in few-shot learning, such as KL-divergence as shown.

To the best of our knowledge no other work has approached this proposed split into maturity categories (early and adult) yet. Regarding the whole IP102 data set, Wu et al. [15] reported accuracy rates of 49.5%, in [30] 55.2% was achieved, and more recently, Nanni et al. [31] showed accuracy results of 74.11%. Xie et al. [32] have proposed an insect image data set with 4508 images divided into 40 classes, and they designed a multi-level learning features procedure to represent the categories and then approached classification. Their accuracy was 89.30% in their data set. In smaller insect data sets, Ayan et al. [33] have experimented an ensemble procedure to combine CNN models based on a genetic algorithm to weight the results in the classifier. They have tested on a data set by [34] with 562 images, 10 classes, and achieved best accuracy of 95.16%. Deng et al. [34] on their proposed data set have obtained accuracy results of 85.50%.

As compared with the other works from the literature, our work here has brought in the following novelties:

- The IP-FSL data set with 6817 insect pest images, divided into species maturity stages (97 of adults, 45 of early stage samples);
- A few-shot leveraged prototypical network for classification, which achieved 86.33%, and 87.91% accuracy for adults and early categories, respectively.

These results are relevant for the classification of insect pests using few samples. We see it as a promising approach for practical field applications, especially if crop based focusing on the the most damaging species for a particular crop. Previous works did not focus on specific maturity stages to classify insects on image databases, and as discussed the accuracy rates reported are competitive with the results presented here.

6. Conclusions

We have approached insect pest image recognition with few samples, and also separating maturity stages, in this work, by an improved few-shot learning network. We have proposed a data set, IP-FSL, with 6817 samples of adults (97 classes), and early stages (45 classes) of insect pests, derived from IP102, and properly organized for this problem. We proposed to evaluate other divergences along with state-of-art FSL matching and prototypical networks, and we have shown that a leveraged prototypical network with KL divergence is the most promising for this setting.

Our results on adult, and early stages of the insect pests achieved 86.33% and 87.91% accuracy for three-way and five-shot experiments, respectively, which are high figures even if compared to other approaches with only adult classes.

Future directions to be explored include studies on cross-domain shifts in insect pest recognition, focus on specific crops and related insect ecosystems, and deploying mobile applications to help agronomists on detecting and identifying potential insect infestations on crops.

Author Contributions: Conceptualization, J.C.G. and D.L.B.; methodology, J.C.G. and D.L.B.; software, J.C.G.; validation, J.C.G. and D.L.B.; formal analysis, J.C.G. and D.L.B.; investigation, J.C.G. and D.L.B.; data curation, J.C.G.; writing—review and editing, J.C.G. and D.L.B.; supervision, D.L.B.; project administration, D.L.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data and code used in this research is freely available at <https://github.com/Jacocirino/FSLInsectImageRecognition.git> (accessed on 1 June 2022).

Acknowledgments: We would like to thank and acknowledge partial support with grants for this research from DPI/UnB, and DPG/UnB (University of Brasília).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN Convolutional neural networks
FSL Few-shot learning

References

1. Deutsch, C.A.; Tewksbury, J.J.; Tigchelaar, M.; Battisti, D.S.; Merrill, S.C.; Huey, R.B.; Naylor, R.L. Increase in crop losses to insect pests in a warming climate. *Science* **2018**, *361*, 916–919. [[CrossRef](#)] [[PubMed](#)]
2. Dent, D.; Binks, R.H. *Insect Pest Management*; Cabi: Wallingford, UK, 2020.
3. Barzman, M.; Bärberi, P.; Birch, A.N.E.; Boonekamp, P.; Dachbrodt-Saaydeh, S.; Graf, B.; Hommel, B.; Jensen, J.E.; Kiss, J.; Kudsk, P.; et al. Eight principles of integrated pest management. *Agron. Sustain. Dev.* **2015**, *35*, 1199–1215. [[CrossRef](#)]
4. Høye, T.T.; Årje, J.; Bjerge, K.; Hansen, O.L.; Iosifidis, A.; Leese, F.; Mann, H.M.; Meissner, K.; Melvad, C.; Raitoharju, J. Deep learning and computer vision will transform entomology. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2002545117. [[CrossRef](#)] [[PubMed](#)]
5. Lima, M.C.F.; de Almeida Leandro, M.E.D.; Valero, C.; Coronel, L.C.P.; Bazzo, C.O.G. Automatic detection and monitoring of insect pests—A review. *Agriculture* **2020**, *10*, 161. [[CrossRef](#)]

6. Li, W.; Zheng, T.; Yang, Z.; Li, M.; Sun, C.; Yang, X. Classification and detection of insects from field images using deep learning for smart pest management: A systematic review. *Ecol. Inform.* **2021**, *66*, 101460. [[CrossRef](#)]
7. Li, Y.; Wang, H.; Dang, L.M.; Sadeghi-Niaraki, A.; Moon, H. Crop pest recognition in natural scenes using convolutional neural networks. *Comput. Electron. Agric.* **2020**, *169*, 105174. [[CrossRef](#)]
8. Alves, A.N.; Souza, W.S.; Borges, D.L. Cotton pests classification in field-based images using deep residual networks. *Comput. Electron. Agric.* **2020**, *174*, 105488. [[CrossRef](#)]
9. Liu, J.; Wang, X. Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network. *Front. Plant Sci.* **2020**, *11*, 898. [[CrossRef](#)]
10. Kasinathan, T.; Singaraju, D.; Uyyala, S.R. Insect classification and detection in field crops using modern machine learning techniques. *Inf. Process. Agric.* **2021**, *8*, 446–457. [[CrossRef](#)]
11. Stork, N.E. World of insects. *Nature* **2007**, *448*, 657–658. [[CrossRef](#)]
12. Karar, M.E.; Alsunaydi, F.; Albusaymi, S.; Alotaibi, S. A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alex. Eng. J.* **2021**, *60*, 4423–4432. [[CrossRef](#)]
13. Chen, C.J.; Huang, Y.Y.; Li, Y.S.; Chen, Y.C.; Chang, C.Y.; Huang, Y.M. Identification of Fruit Tree Pests With Deep Learning on Embedded Drone to Achieve Accurate Pesticide Spraying. *IEEE Access* **2021**, *9*, 21986–21997. [[CrossRef](#)]
14. Thenmozhi, K.; Reddy, U.S. Crop pest classification based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* **2019**, *164*, 104906. [[CrossRef](#)]
15. Wu, X.; Zhan, C.; Lai, Y.K.; Cheng, M.M.; Yang, J. Ip102: A large-scale benchmark dataset for insect pest recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 8787–8796.
16. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*; MIT Press: Cambridge, MA, USA, 2018.
17. Fei-Fei, L.; Fergus, R.; Perona, P. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 594–611. [[CrossRef](#)] [[PubMed](#)]
18. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–34. [[CrossRef](#)]
19. Li, Y.; Yang, J. Meta-learning baselines and database for few-shot classification in agriculture. *Comput. Electron. Agric.* **2021**, *182*, 106055. [[CrossRef](#)]
20. Dhillon, G.S.; Chaudhari, P.; Ravichandran, A.; Soatto, S. A Baseline for Few-Shot Image Classification. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020; p. 20.
21. Kang, B.; Liu, Z.; Wang, X.; Yu, F.; Feng, J.; Darrell, T. Few-shot object detection via feature reweighting. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 8420–8429.
22. Li, Y.; Yang, J. Few-shot cotton pest recognition and terminal realization. *Comput. Electron. Agric.* **2020**, *169*, 105240. [[CrossRef](#)]
23. Yang, Z.; Yang, X.; Li, M.; Li, W. Small-sample learning with salient-region detection and center neighbor loss for insect recognition in real-world complex scenarios. *Comput. Electron. Agric.* **2021**, *185*, 106122. [[CrossRef](#)]
24. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Barcelona, Spain, 2016; Volume 29, pp. 3630–3638.
25. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2017; Volume 30, pp. 4077–4087.
26. Siahkamari, A.; Xia, X.; Saligrama, V.; Castañón, D.; Kulis, B. Learning to approximate a Bregman divergence. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; Volume 33, pp. 3603–3612.
27. Bréchet, C.; Fischer, A.; Levrard, C. Robust bregman clustering. *Ann. Stat.* **2021**, *49*, 1679–1701. [[CrossRef](#)]
28. Cilingir, H.K.; Manzelli, R.; Kulis, B. Deep Divergence Learning. In Proceedings of the 37th International Conference on Machine Learning, Virtual Conference, 13–18 July 2020; Proceedings of Machine Learning Research; Volume 119, pp. 2027–2037.
29. Banerjee, A.; Merugu, S.; Dhillon, I.S.; Ghosh, J.; Lafferty, J. Clustering with Bregman divergences. *J. Mach. Learn. Res.* **2005**, *6*, 1705–1749.
30. Ren, F.; Liu, W.; Wu, G. Feature reuse residual networks for insect pest recognition. *IEEE Access* **2019**, *7*, 122758–122768. [[CrossRef](#)]
31. Nanni, L.; Manfè, A.; Maguolo, G.; Lumini, A.; Brahmam, S. High performing ensemble of convolutional neural networks for insect pest image detection. *Ecol. Inform.* **2022**, *67*, 101515. [[CrossRef](#)]
32. Xie, C.; Wang, R.; Zhang, J.; Chen, P.; Dong, W.; Li, R.; Chen, T.; Chen, H. Multi-level learning features for automatic classification of field crop pests. *Comput. Electron. Agric.* **2018**, *152*, 233–241. [[CrossRef](#)]
33. Ayan, E.; Erbay, H.; Varçın, F. Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. *Comput. Electron. Agric.* **2020**, *179*, 105809. [[CrossRef](#)]
34. Deng, L.; Wang, Y.; Han, Z.; Yu, R. Research on insect pest image detection and recognition based on bio-inspired methods. *Biosyst. Eng.* **2018**, *169*, 139–148. [[CrossRef](#)]