Few-Shot Learning for Plant Disease Classification Using ILP

Dany Varghese(✉), Uzma Patel, Paul Krause, and Alireza Tamaddoni-Nezhad

University of Surrey, Guildford GU2 7XH, UK
{dany.varghese,up00039,p.krause,a.tamaddoni-nezhad}@surrey.ac.uk

Abstract. Plant diseases are one of the main causes of crop loss in agriculture. Machine Learning, in particular statistical and neural nets (NNs) approaches, have been used to help farmers identify plant diseases. However, since new diseases continue to appear in agriculture due to climate change and other factors, we need more data-efficient approaches to identify and classify new diseases as early as possible. Even though statistical machine learning approaches and neural nets have demonstrated state-of-the-art results on many classification tasks, they usually require a large amount of training data. This may not be available for emergent plant diseases. So, data-efficient approaches are essential for an early and precise diagnosis of new plant diseases and necessary to prevent the disease's spread. This study explores a data-efficient Inductive Logic Programming (ILP) approach for plant disease classification. We compare some ILP algorithms (including our new implementation, PyGol) with several statistical and neural-net based machine learning algorithms on the task of tomato plant disease classification with varying sizes of training data set (6, 10, 50 and 100 training images per disease class). The results suggest that ILP outperforms other learning algorithms and this is more evident when fewer training data are available.

Keywords: Few-shot Learning · Data Efficient Machine Learning · ILP · Inverse Entailment · Plant Disease Classification

1 Introduction

Crop cultivation and production play a crucial role in the field of agriculture. The primary cause of agriculture losses is infected crops, which in turn reduces the production rate. Thus, plant diseases have become a significant threat to global food security. Also, sustainable farming can play a vital role in our climate and the earth's ecosystems. Agriculture needs biodiversity and vice versa. Livestock and other crops nourish themselves and originate from existing crops, whilst biodiversity maintains and provides environments necessary for the production of crops. Therefore, plant diseases pose a considerable threat to the global economy and the codependent relationship between agriculture and biodiversity [44]. Diseases can affect whole crops and cause 100% losses. It is essential to determine what is wrong as fast as possible before it has spread or ripened.

D. Garg et al. (Eds.): IACC 2022, CCIS 1781, pp. 321–336, 2023.
https://doi.org/10.1007/978-3-031-35641-4_26

Moreover, the lack of access to technology in some areas means that management of these issues may be poor, and farmers could lack access to knowledge on how to deal with the illnesses. Therefore, an effective plant disease detection system could cut the cost and time spent on the issue while providing enough knowledge to understand the impact on crops better. Several systems for plant disease detection have been introduced in recent years—the fast progress in machine vision and artificial intelligence speeds up the research interest in this area [8,31,32]. Deep Neural Networks (DNNs) have demonstrated state-of-the-art results in plant disease detection [47,53]. Also, several mobile-based platforms were introduced to help the farmers, such as Plantix [16]. However, existing learning algorithms require extensive sets of training examples, e.g. we need hundreds or thousands of images to train DNNs for image classification.

However, we might only have a small number or even only one training example in some applications. For example, an early and precise diagnosis of new plant diseases may be essential in preventing the spread of the disease. Machine learning algorithms that work with a few training examples (one-shot learning, few-shot learning) would make a significant contribution to risk mitigation for the industry. Also, according to Algorithmia's "2020 State of Enterprise Machine Learning" [2], 50% of respondents said it took 8–90 days to deploy one model, with only 14% saying they could deploy in less than a week [18]. Furthermore, a recent study on life cycle assessment of large AI models shows that the process can emit more than 626,000 pounds of carbon dioxide equivalent-nearly five times the lifetime emissions of the average American car [50]. It is thus imperative to reduce the effort needed to train models as their use becomes more prevalent.

Since the rules and the knowledge in traditional expert systems are defined and formulated by human experts, these rules and knowledge are easy for humans to understand and interpret. In this scenario, Inductive Logic Programming (ILP) has several advantages over most machine learning approaches. Because logic resembles natural language, it can be easily read by humans. Also, ILP systems can perform well with very small amounts of data [36], even succeeding with one-shot(single) data [55,56]. This efficiency is enhanced further by the inclusion of background knowledge in the form of logic rules. ILP [37] is a machine learning formalism that induces a hypothesis that generalizes examples. ILP uses a first-order logic program as data, whereas most forms of ML use vectors or tensors to represent data. ILP models are more data-efficient, explainable, and can incorporate human knowledge more easily compared to other forms of machine learning. In this paper, we introduce a data-efficient machine learning approach which can learn from small amounts of data. We illustrate its use on plant disease detection, and show that it can outperform more traditional algorithms; especially in cases where limited training data is available.

2 Related Work

Plenty of works have been devoted to detection and classification using image processing in history, and still, it continues to attract researchers to this field.

We present the literature study in three parts. Part one explains few-shot learning and its importance in the machine learning community, whereas, in part two, we reviewed plant disease classification. Part 3 will give an overview of different feature extraction techniques.

2.1 Few-Shot Learning

In order to learn from a limited number of examples with supervised information, a new machine learning paradigm called Few-Shot Learning (FSL) was proposed. The seminal work toward few-shot learning dates back to the early 2000 s with work by Li Fei-Fei et al. [29]. The authors developed a variational Bayesian framework for image classification, using the premise that previously learned classes could be leveraged to help forecast future ones when very few examples are available from a given class [12,30]. More recently, Lake et al. approached the problem of one-shot learning as an instance of few-shot, addressing one-shot character recognition with a method called Hierarchical Bayesian Program Learning (HBPL) [27]). A detailed study on FSL can be found in [58].

According to Mitchell [1], a learning approach can be considered as a program that can learn from experiences (E) related to the task (T) to improve performance(P). Following this definition, we can define FSL as a machine learning problem that takes the tuple $< T, P, E >$; E contains only a limited number of examples with supervised information for the target T. FSL can relieve the burden of large scale data collection and reduce the data gathering effort for data-intensive applications. Several learning approaches have been introduced for FSL, and they can be mainly classified into 4 categories:

1. Weakly supervised learning
   (a) Semi-supervised learning [21,42,59]
   (b) Active learning [13,39]
2. Transfer learning [60,61]
3. Meta-learning [45,61]
4. ILP/MIL based methods [10,55,56]

Focusing on the research concept of this paper, Li et al. [28] proposed a semi-supervised few-shot learning approach to solve the plant leaf disease recognition problem. They have used the transfer learning concepts and implemented them using deep learning methods. Extensive comparison experiments considering the domain split and few-shot parameters (N-way, k-shot) were carried out to validate the correctness and generalization of proposed semi-supervised few-shot methods.

Chen et al. [7] used local feature matching conditional neural adaptive processes (LFM-CNAPS) based on meta-learning that aims at detecting plant diseases. They have applied intense training on datasets like 20,000 training iterations. David et al. [3] presented the FSL approach to plant disease classification using transfer learning and the Siamese network.

2.2 Plant Disease Classification Methods

Sunil et al. [17] used multiple descriptors such as Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA) and Grey-Level Co-occurrence Matrix (GLCM) to extract informative features of leaves. Before feature extraction, they also applied K-means clustering and histogram equalization on 256 $\times$ 256 data items. They have done experiments on the village database of tomato leaves and compared it with Support Vector Machine (SVM), Convolutional Neural Network (CNN) and K-Nearest Neighbor (K-NN), reporting the accuracy of the resulting models as 88%, 99.6% and 97% respectively.

Kalpesh et al. [25] have experimented on tomato, apple, potato, and grape leaves from the Plant-Village dataset. They used GLCM to shape and texture features and applied feature engineering mechanisms to find highly correlated features. The database was divided into two sets during the experiment: the training set, which contained 70%, and the testing set, which comprised 30%. A 93% accuracy rate has been reported using the Random Forest model.

G. Saradhambal et al. [26] proposed an approach to produce a system for automatic plant disease detection. The research was done to predict the infected area of the leaves by applying a k-means clustering algorithm and Otsu's classifier. Both the shape and texture features were extracted in the proposed work. The shape-oriented features extracted in this work included area, colour axis length, eccentricity, solidity and perimeter. In contrast, the texture oriented features were contrast, correlation, energy, homogeneity, and mean. The classification in this research was done using a neural network.

Irfan et al. [41] have performed some experiments on rice plant disease images by using the Probabilistic Neural Network (PNN), one of the Artificial Neural Networks (ANN) models. First, the images were pre-processed with the median filtering method, and then the OTSU method was used for segmentation. Later, they applied GLCM for feature extraction. The accuracy rate of this system was 76.8%.

2.3 Feature Extraction

Accuracy is the main parameter used to calculate the performance of a model. The classifier's accuracy depends primarily on the extracted features. So, feature extraction plays a vital role in identifying disease and improves diagnostic accuracy. Hand-crafted feature engineering and deep learning feature extraction are the two main types of feature extraction. Hand-crafted feature engineering from images can be mainly divided into three types, shape, texture, and colour. In this paper, we only consider the hand-crafted feature engineering methods.

The shape of an object is an essential and fundamental visual feature for describing image content [5,19]. It can be considered a silhouette of the object, invariant to rotation, scale and translation. Shape features are less developed than their colour and texture counterparts because of the inherent complexity of representing shapes.

Among the visual features, colours are the most vital, reliable, and widely used features. The colour feature descriptor of images has been used widely, showing its robustness to background complication and independence over image size and orientation. Colour feature extraction methods broadly fall into global and local methods [40]. In global methods, the feature extraction process considers the complete image, including global colour histogram, intersection, and image bitmap. On the other hand, local methods consider a portion of the image, including local colour histogram, colour correlogram, and colour difference histogram.

Texture is the primary term used to define objects or concepts of a given image. Tactile texture directs the natural feel, and visual texture refers to seeing the image's shape or contents [4]. In image processing, texture can be defined as a function of spatial variation of the brightness intensity of the pixels. Texture analysis is vital in computer vision cases such as object recognition, surface defect detection, pattern recognition, and medical image analysis.

Hu Moments. Moments and the related invariants have been extensively analyzed to characterize the patterns in images in a variety of applications. We use Hu Moments (or rather Hu moment invariants), a set of 7 numbers calculated using central moments invariant to image transformations [22]. The first six moments have been proved invariant to translation, scale, rotation, and reflection. While the seventh moment's sign changes for image reflection. Hu firstly introduces moment invariants. In [20], Hu derived six absolute orthogonal invariants and one skew orthogonal invariant based upon algebraic invariants, which are not only independent of position, size and orientation but also independent of parallel projection

Haralick Texture. Haralick texture features are estimated from a Grey Level Co-occurrence Matrix (GLCM), a matrix that counts the co-occurrence of neighbouring grey levels in the image [18]. The GLCM is a square matrix with the dimension of the number of grey levels N in the region of interest (ROI). The GLCM functions characterize an image's texture by calculating how often pairs of pixels with distinct values and specified spatial relationships appear in an image, composing a GLCM, and then pulling statistical estimates from this matrix.

3 Methodology

The basic methodology of the developed system is presented schematically in Fig. 1.

3.1 Image Data

The image data were collected from a well-known dataset, "PlantVillage" [23,43]; freely available in [35]. The dataset contains 50,000 RGB images of 14 crops.

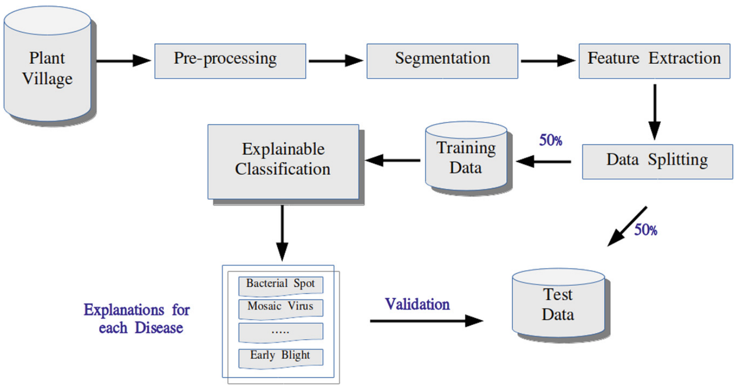


Fig. 1. System Description

The current paper focuses on tomato images which contain ten classes as explained in Table 1, including healthy images. To address the unbalance in the dataset, we created new instances which contain 50 images from each class.

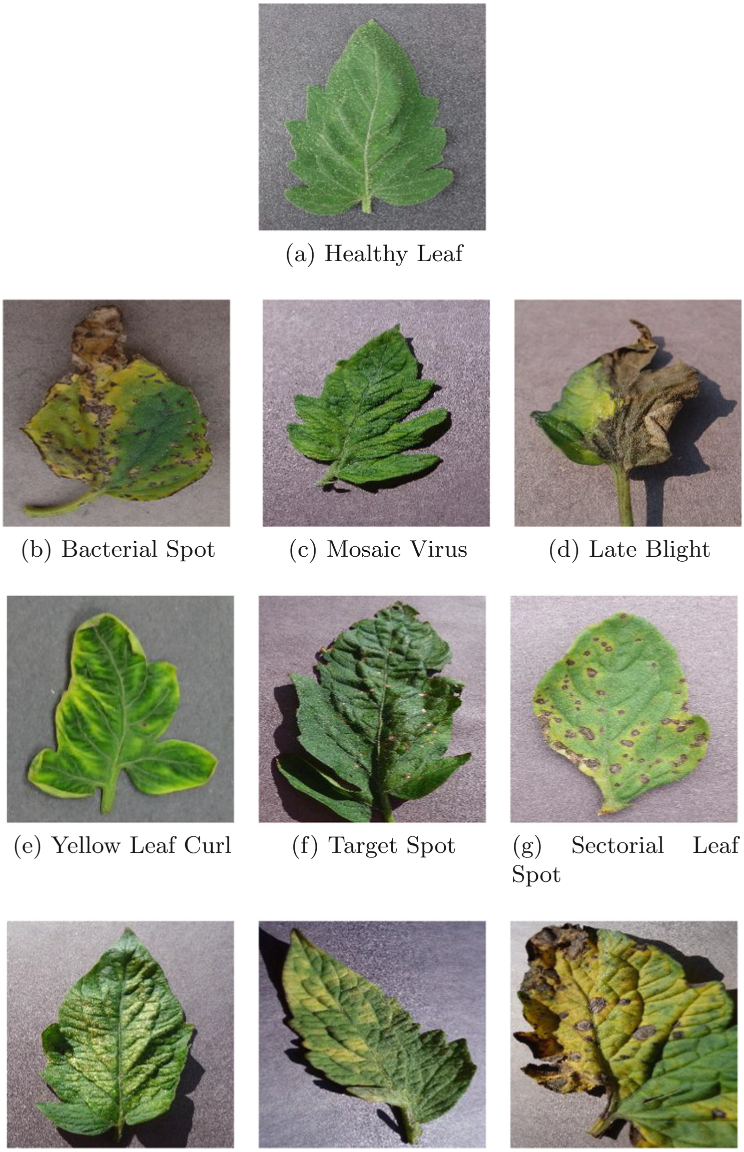
Leaf Diseases and Symptoms. Over a thousand different fruit, vegetable and herb species are cultivated worldwide. The growth process for them all is not the same, and neither are the diseases that affect them. In an ideal world, we would be able to quickly identify each of these species and all the diseases that affect them, to eliminate and reduce any possible problems. To reach this ideal, we must find the most efficient method to design an application or system to identify the diseases. Identification must be carried out at a high level of accuracy since this is the deciding factor on what to do next. We begin this process by gaining some background knowledge on each disease in our chosen dataset.

In this study, we have taken the following nine plant diseases, illustrated in Fig. 2, for the experiments, where a summary of the symptoms of each disease can be found in Table 1:

1. Bacterial Spot
2. Mosaic Virus
3. Late Blight
4. Yellow Leaf Curl
5. Target Spot
6. Sectorial Leaf Spot
7. Spider Mites
8. Leaf Mould
9. Early Blight

3.2 Pre-processing

There are 3 main stages during pre-processing:



(a) Healthy Leaf
(b) Bacterial Spot
(c) Mosaic Virus
(d) Late Blight
(e) Yellow Leaf Curl
(f) Target Spot
(g) Sectorial Leaf Spot
(h) Spider Mites
(i) Leaf Mould
(j) Early Blight

Fig. 2. Healthy & Disease Effected Leaves

1. Resizing
2. Image Segmentation
3. Colour Space Conversion

Table 1. Summary of Symptoms of Each Disease

Disease	Symptoms	Ref.
Bacterial Spot	Small moist circular areas appear on the leaves, starting off as a yellow-green colour which can darken to brown-red	[34]
Mosaic Virus	Leaves are mottled with yellow, white, and green blister-like spots and plant growth is often stunted. Leaves can sometimes be curled and crimpled	[24]
Late Blight	Begins as pale-green or olive-green areas that quickly enlarge to become brown-black, water-soaked, and oily-looking	[15]
Yellow Leaf Curl	small, crumpled and curl upwards. Leaves are also marginal yellowing	[11]
Target Spot	small, light brown lesions, with concentric patterns and a yellow halo form on the leaves	[33]
Septorial Leaf Spot	Circular spots appear on the underside of the older leaves, with a yellow halo surround the spots. Unlike early blight, Septoria leaf spots have a brown margin and lighter grey-tan centres	[14]
Spider Mites	leaf to become speckled, dull and botchy with pale yellow and reddish-brown spots	[48]
Leaf Mould	Pale green spots can be found near the tips of the tomato plants leaves which eventually enlarge and turn from green to brown to a purplish black	[46]
Early Blight	Start towards the bottom of the plant where dark brown spots can form with yellow concentric halo rings on leaves and the stems of the plant	[6]

Firstly, we resized all images to 256 $\times$ 256 pixels. Then, segmentation is performed to separate the image of the leaf from the background. The colour of the leaf is extracted from the image. Colour space conversion is essential since R, G, and B in RGB are all co-related to the colour luminance, i.e., We cannot separate colour information from luminance. HSV or Hue Saturation Value separates image luminance from colour information. This makes it easier when we are working on or need the luminance of the image/frame. Also, using only the Hue component makes the algorithm less sensitive to lighting variations.

3.3 Feature Extraction and Scaling

Feature extraction is the process of reducing dimensionality in the dataset by using various methods to combine existing features to create new features. The new features summarize what the original features tell us without losing essential information or knowledge. Overfitting can occur when a dataset's dimensionality is high, an issue we look to avoid through feature extraction. Other benefits include better accuracy, a reduction in training time, better data visualization and simpler explanations of our model. Feature extraction also allows our model to understand what it is looking for and subsequently indexes and retrieves relevant information. This information is then used to mathematically describe various attributes, such as colours, textures, or shapes. As explained in Sect. 2.3, we mainly extract 13 Haralick features using Grey Level Co-occurrence Matrix.

Feature scaling is an essential step in standardizing the independent features present in the data to a common range. In this stage, we normalise the highly varying magnitudes or values or units. If feature scaling is not performed, then a machine learning algorithm will be biased by features with higher numerical values and reduce the influence of features with smaller values, regardless of the unit of the values. As part of the normalisation process, we have rounded all the feature values to 5 decimal points.

4 Empirical Evaluation

This section empirically evaluates ILP systems against other machine learning approaches using four differently sized datasets.

4.1 Materials and Methods

We have created four chunks of images from the original dataset. The first two chunks contain 100 and 50 images per class. The third and fourth chunks contain 10 and 6 images per class. During the experiments, we followed the hold-out learning strategy. We have split the data in the ratio of 1:1. Since an ILP architecture always considers the learning problem as a binary classification, we have followed a new execution strategy to have a fair comparison of both rule-based and machine learning approaches. We have run the experiment "10" times (equal to the number of classes; nine diseased plus one healthy), and in each run, examples from one class will be considered as a positive example/class and all others as a negative example/class.

It is vital to generate background knowledge using first-order logic rules since we are doing experiments using ILP. In [55], it is explained that the categorical value method is an excellent mechanism for dealing with numerical data. We have divided each feature into three categorical classes/ranges: low, medium, and high, according to the local population of feature points. We will use the feature names as predicates in the background knowledge.

We have developed our experiment using a novel ILP system, PyGol[1], based on meta inverse entailment(MIE), which is motivated by Mode-directed inverse entailment [36] but never uses mode to generate the bottom clause. The significant merit of this system is that the training phase is fully implemented in Python, and to have a fair comparison during the test phase, PyGol uses a Prolog interpreter. During the test phase, PyGol used a python-based approach

[1] Available from: https://github.com/danyvarghese/PyGol.

from the system PyILP [57], which uses PySwip [52] in the back-end. PyGol can automatically generate the first-order logic database, i.e. background knowledge, from relational data. A schematic diagram of generating the logic rules from the relational dataset is shown in Fig. 3.

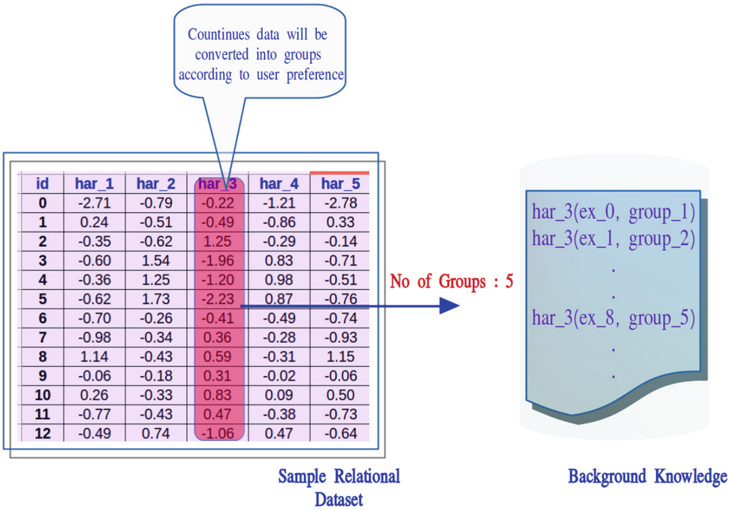


Fig. 3. Automatic generation of logical rules from relational dataset

PyILP is a novel, user-friendly Python interface for Inductive Logic programming(ILP) systems for teaching relational machine learning and comparing different algorithms, whereas PySwip act as the bridge between python and Prolog. PyGol enables us to do every experiment in a single platform called Jupyter Notebook. We have compared PyGol with two other state-of-the-art ILP systems, seven statistical machine learning approaches and two neural network approaches.

Aleph [49] is a state-of-the-art ILP algorithm that has been used for several real-time applications and is one of the ILP systems we are considering for empirical evaluation. Like PyGol, Aleph also uses the advantage of inverse entailment, but mode-directed inverse entailment.

The second system for our consideration is Metagol [9] which is developed based on the concept of meta-interpretive learning (MIL) [38] and has been successfully trialled in several applications [36, 10]. Learning recursive rules and predicate invention is the major contribution of MIL to the ILP community, but they cannot deal with noisy input, a significant disadvantage. In order to overcome this issue, a noise-tolerant version of metagol is introduced as metagol_nt [36,51]. During the empirical evaluation, we use the python version of metagol_nt, which is available from [54]. Other approaches used for evaluation are listed below;

1. Statistical Machine Learning
   (a) Decision Tree (DT)
   (b) Naive Bayes (NB)
   (c) Support Vector Machines (SVM)
   (d) Logical Regression (LR)
   (e) Latent Dirichlet Allocation (LDA)
   (f) K-Nearest Neighbors (KNN)
   (g) Random Forest (RF)
2. Artificial Neural Network Learning
   (a) Perceptron (Per.)
   (b) Multi-Layer Perceptron (MLP)

5 Results and Discussion

Figures 4, 5, 6 and 7 will provide an overall idea about each model on four different datasets. We have plotted overall performance and average F1-Score obtained on each run and an execution time comparison on different ILP systems. In each figure, the subplot 'a' indicates the system's overall performance in each run. The sub-plot 'b' indicates the F1 score during the classification. The F1 score is more important than accuracy in this scenario, since each run can be considered an imbalanced classification and will also sum up the predictive performance of a model.

While analysing each box plot, it is clear that machine learning approaches other than ILP systems respond differently with each dataset, but the ILP system responds almost consistently. In almost all cases, the accuracy of the ILP system lies between 80%–100%, which shows its efficiency compared to other models for classification. Also, from the box plot, it is clear that the median of accuracies from each run for typical machine learning approaches is always around 90% which is the default accuracy. From the plots related to the F1 score, other than the decision tree, none of them could achieve an average value of 0.5, meaning the predictive performance of those systems is worse than random.

Focusing on the ILP systems, as the number of examples in each class increases, Aleph's performance improves. However, for metagol_nt, it deteriorates. Both PyGol and Aleph outperform all the statistical approaches, with a large-scale difference. We have done some experiments with CNN, but it couldn't perform well (just giving predictive accuracy always) since our dataset is imbalanced w.r.t. to the concept of a binary classification problem. That is the reason why we did not report the results from CNN. We also noticed that the average response time of PyGol is just 7 s.

Aleph's learning procedure depends on the order of examples given and also follows a greedy approach while learning. During the learning cycle, a seed sample will be chosen then a candidate hypothesis will be generated from the bottom clause, and examples covered by the theory are removed, and the process continues. But PyGol follows a global theory generation procedure in which the system generates all the possible candidate hypotheses from all the examples.

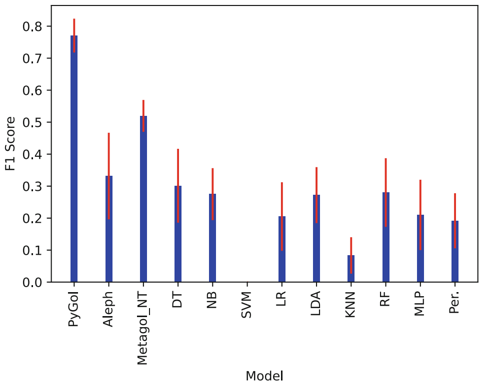


Fig. 4. Performance comparison on Dataset with 6 images per class

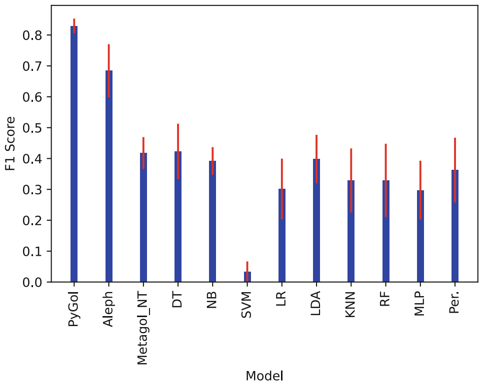


Fig. 5. Performance comparison on Dataset with 10 images per class

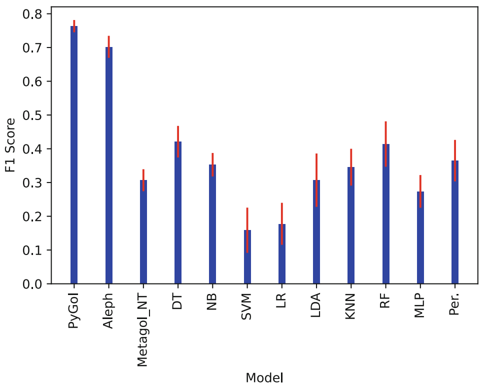


Fig. 6. Performance comparison on Dataset with 50 images per class

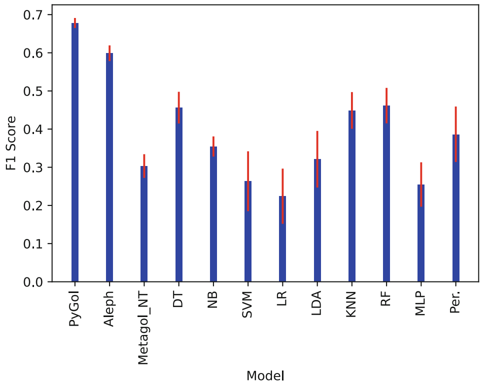


Fig. 7. Performance comparison on Dataset with 100 images per class

6 Conclusion

The objective of this study is to introduce a few-shot model using ILP for plant disease classification. In order to evaluate the data efficiency of each system, we have divided the image dataset into four chunks containing 6, 10, 50 and 100 images per class and experiments on individual chunks. As discussed, the ILP approaches significantly outperform all the machine learning approaches selected, even a convolutional neural network. It is also evident from the performance evaluation that machine learning approaches struggled to learn from the small datasets of 6 or 10 images per class. However, ILP approaches perform well in this scenario too. This shows that the proposed approach of using ILP is data-efficient with respect to other statistical machine learning approaches. Furthermore, the proposed PyGol approach can reduce user interaction compared to other ILP systems since it does not use mode declarations and meta-rules as in Aleph and MIL, respectively. It shows the potential of PyGol for automated data science.

References

1. Machine Learning. McGraw Hill (1997)
2. Algorithmia': 2020 state of enterprise machine learning. https://algorithmia.com/state-of-ml
3. Argüeso, D., et al.: Few-shot learning approach for plant disease classification using images taken in the field. Comput. Electron. Agric. **175**, 105542 (2020)
4. Armi, L., Fekri-Ershad, S.: Texture image analysis and texture classification methods - a review (2019). https://doi.org/10.48550/ARXIV.1904.06554. https://arxiv.org/abs/1904.06554
5. Babapour Mofrad, F., Valizadeh, G.: A comprehensive survey on two and three-dimensional Fourier shape descriptors: biomedical applications. Arch. Comput. Methods Eng. **29**(7), 4643–4681 (2022)

6. Brian Hudelson, U.: Early blight. https://hort.extension.wisc.edu/articles/early-blight/
7. Chen, L., Cui, X., Li, W.: Meta-learning for few-shot plant disease detection. Foods **10**, 2441 (2021). https://doi.org/10.3390/foods10102441
8. Chen, Z., et al.: Plant disease recognition model based on improved yolov5. Agronomy **12**(2), 365 (2022)
9. Cropper, A.: Metgol - an ILP system based on meta-iterpretive learning (2016). https://github.com/metagol/metagol
10. Dai, W.Z., Muggleton, S., Wen, J., Tamaddoni-Nezhad, A., Zhou, Z.H.: Logical vision: one-shot meta-interpretive learning from real images. In: ILP (2017)
11. Davis, R., Miyao, G., Subbarao, K., Stapleton, J., Aegerter, B.: Tomato yellow leaf curl. https://www2.ipm.ucanr.edu/agriculture/tomato/Tomato-Yellow-Leaf-Curl/
12. Fink, M.: Object classification from a single example utilizing class relevance metrics. In: Advances in Neural Information Processing Systems, vol. 17 (2004)
13. Garcia, V., Bruna, J.: Few-shot learning with graph neural networks. arXiv preprint arXiv:1711.04043 (2017)
14. Garden, M.B.: Septoria leaf spot of tomato. https://www.missouribotanicalgarden.org/gardens-gardening/your-garden/help-for-the-home-gardener/advice-tips-resources/pests-and-problems/diseases/fungal-spots/septoria-leaf-spot-of-tomato.aspx
15. Gevens, A., Seidl, A., Brian Hudelson, U.: Late blight. https://hort.extension.wisc.edu/articles/late-blight/
16. GmbH, P.: Plantix : a mobile crop advisory app for farmers, extension workers and gardeners (2015). https://plantix.net/en/
17. Harakannanavar, S.S., Rudagi, J.M., Puranikmath, V.I., Siddiqua, A., Pramodhini, R.: Plant leaf disease detection using computer vision and machine learning algorithms. Glob. Trans. Proc. **3**(1), 305–310 (2022)
18. Hinek, J.P.: How long does it take to build an ml model?. https://m.mage.ai/how-long-does-it-take-to-build-an-ml-model-d68b8afa50a5
19. Hong, T.C.K., Economou, A.: What shape grammars do that cad should: the 14 cases of shape embedding. Artif. Intell. Eng. Des. Anal. Manuf. **36**, e4 (2022). https://doi.org/10.1017/S0890060421000263
20. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory **8**(2), 179–187 (1962)
21. Huang, H.P., Puvvada, K.C., Sun, M., Wang, C.: Unsupervised and semi-supervised few-shot acoustic event classification. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 331–335. IEEE (2021)
22. Huang, Z., Leng, J.: Analysis of Hu's moment invariants on image scaling and rotation, vol. 7, pp. V7–476 (2010)
23. Hughes, D.P., Salathe, M.: An open access repository of images on plant health to enable the development of mobile disease diagnostics (2015)
24. Iseli, M.: Mosaic virus - is my plant infected? identification & treatment. https://plantophiles.com/plant-care/mosaic-virus-symptoms-and-treatments/
25. Joshi, K., Awale, R., Ahmad, S., Patil, S., Pisal, V.: Plant leaf disease detection using computer vision techniques and machine learnings. In: ITM Web of Conferences (2022)
26. KSaradhambal, G., D.R.L.S.R.R.: Plant disease detection and its solution using image classification. Int. J. Pure Appl. Math. **119** (2018). https://doi.org/10.1051/itmconf/20224403002

27. Lake, B., Salakhutdinov, R., Tenenbaum, J.: One-shot learning by inverting a compositional causal process. In: Advances in Neural Information Processing Systems (2013), 27th Annual Conference on Neural Information Processing Systems, NIPS 2013, 05–10-December-2013 (2013)
28. Li, Y., Chao, X.: Semi-supervised few-shot learning approach for plant diseases recognition. Plant Methods **17**, 1–10 (2021)
29. Li Fe-Fei, Fergus, Perona: A bayesian approach to unsupervised one-shot learning of object categories. In: Proceedings Ninth IEEE International Conference on Computer Vision, vol. 2, pp. 1134–1141 (2003)
30. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. **28**(4), 594–611 (2006)
31. Lu, J., Ehsani, R., Shi, Y., De Castro, A., Wang, S.: Detection of multi-tomato leaf diseases (late blight, target and bacterial spots) in different stages by using a spectral-based sensor. Sci. Rep. **8**, 2793 (2018)
32. Lu, J., Zhou, M., Gao, Y., Jiang, H.: Using hyperspectral imaging to discriminate yellow leaf curl disease in tomato leaves. Precis. Agric. **19**, 1–16 (2018)
33. McAvoy, G.: Take the right aim to tame target spot of tomato. https://www.growingproduce.com/vegetables/take-the-right-aim-to-tame-target-spot-of-tomato/
34. Michelle Marks, U.: Bacterial spot of tomato. https://hort.extension.wisc.edu/articles/bacterial-spot-of-tomato/
35. Mohanty, S.: Plantvillage-dataset. https://github.com/spMohanty/PlantVillage-Dataset
36. Muggleton, S., Dai, W.Z., Sammut, C., Tamaddoni-Nezhad, A.: Meta-interpretive learning from noisy images. Mach. Learn. **107**, 1097–1118 (2018)
37. Muggleton, S., de Raedt, L.: Inductive logic programming: theory and methods. J. Logic Program. 19–20, 629–679 (1994). Special Issue: Ten Years of Logic Programming
38. Muggleton, S., Lin, D., Tamaddoni-Nezhad, A.: Meta-interpretive learning of higher-order dyadic datalog: predicate invention revisited. Mach. Learn. **100**, 49–73 (2015)
39. Müller, T., Pérez-Torró, G., Basile, A., Franco-Salvador, M.: Active few-shot learning with fasl. arXiv preprint arXiv:2204.09347 (2022)
40. Narwade Manoorkar, J., Kumar, D.B.: Local and Global Color Histogram Feature for Color Content-Based Image Retrieval System, pp. 293–300 (2016)
41. Ökten, İ, Yüzgeç, U.: Rice plant disease detection using image processing and probabilistic neural network. In: Seyman, M.N. (ed.) Electrical and Computer Engineering, pp. 82–94. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-01984-5_7
42. Qi, A., Gryaditskaya, Y., Xiang, T., Song, Y.Z.: One sketch for all: one-shot personalized sketch segmentation. IEEE Trans. Image Process. **31**, 2673–2682 (2022)
43. Radovanović, D., Ukanovic, S.: Image-based plant disease detection: a comparison of deep learning and classical machine learning algorithms. In: 2020 24th International Conference on Information Technology (IT), pp. 1–4 (2020)
44. Reganold, J.P., Papendick, R.I., Parr, J.F.: Sustainable agriculture. Sci. Am. **262**(6), 112–121 (1990)
45. Ren, M., et al.: Meta-learning for semi-supervised few-shot classification. arXiv preprint arXiv:1803.00676 (2018)
46. RHS: Tomato leaf mould. https://www.rhs.org.uk/disease/tomato-leaf-mould

47. Sharma, P., Berwal, Y., Ghai, W.: Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. Inf. Process. Agric. **7**, 566–574 (2019)
48. Spengler, T.: What are two-spotted spider mites - two-spotted mite damage and control insects. https://www.gardeningknowhow.com/plant-problems/pests/insects/two-spotted-spider-mite-control.htm
49. Srinivasan, A.: A learning engine for proposing hypotheses (aleph) (2001). https://www.cs.ox.ac.uk/activities/programinduction/Aleph/aleph.html
50. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in NLP. CoRR (2019)
51. Tamaddoni-Nezhad, A.: Metagol_nt. https://github.com/atnezhad/Metagol_NT
52. Tekol, Y., contributors: PySwip v0.2.10 (2020). https://github.com/yuce/pyswip
53. Türkoğlu, M., Yanikoglu, B., Hanbay, D.: Plantdiseasenet: convolutional neural network ensemble for plant disease and pest detection. Signal Image Video Process. **16**(2), 301–309 (2022)
54. Varghese, D.: Metagol_nt. https://github.com/danyvarghese/Metagol_NT
55. Varghese, D., Bauer, R., Baxter-Beard, D., Muggleton, S., Tamaddoni-Nezhad, A.: Human-like rule learning from images using one-shot hypothesis derivation. In: Katzouris, N., Artikis, A. (eds.) Inductive Logic Programming, pp. 234–250. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-97454-1_17
56. Varghese, D., Tamaddoni-Nezhad, A.: One-shot rule learning for challenging character recognition. In: Proceedings of the 14th International Rule Challenge, Oslo, Norway, vol. 2644, pp. 10–27 (2020)
57. Varghese, D., Tamaddoni-Nezhad, A.: Pyilp (2022). https://github.com/danyvarghese/PyILP/
58. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: a survey on few-shot learning. ACM Comput. Surv. **53**(3), 1–34 (2020)
59. Ying, Z.: Weakly-supervised diagnosis with attention models (2022)
60. Yu, Z., Chen, L., Cheng, Z., Luo, J.: Transmatch: a transfer-learning scheme for semi-supervised few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12856–12864 (2020)
61. Zhu, P., Zhu, Z., Wang, Y., Zhang, J., Zhao, S.: Multi-granularity episodic contrastive learning for few-shot learning. Pattern Recognit. **131**, 108820 (2022)