Integrating Domain Knowledge for Enhanced Concept Model Explainability in Plant Disease Classification

Jihen Amara[1,2(✉)], Sheeba Samuel[1,2], and Birgitta König-Ries[1,2]

[1] Heinz Nixdorf Chair for Distributed Information Systems, Friedrich Schiller University Jena, Jena, Germany
{jihene.amara,sheeba.samuel,birgitta.koenig-ries}@uni-jena.de
[2] Michael Stifel Center Jena, Jena, Germany

Abstract. Deep learning-based plant disease detection has seen promising advancements, particularly in its remarkable ability to identify diseases through digital images. Nevertheless, these systems' opacity and lack of transparency, which often offer no human-interpretable explanations for their predictions, raise concerns with respect to their robustness and reliability. While many methods have attempted post-hoc model explainability, few have specifically targeted the integration and impact of domain knowledge. In this study, we propose a novel framework that combines a tomato disease ontology with the concept explainability method Testing with Concept Activation Vectors (TCAV). Unlike the original TCAV method, which required users to gather diverse image concepts manually, our approach automates the creation of images based on relevant concepts used by domain experts in plant disease identification. This not only simplifies the concept collection and labelling process but also reduces the burden on users with limited domain knowledge, ultimately mitigating potential biases in concept selection. Besides automating the concept image generation for the TCAV method, our framework gives insights into the significance of disease-related concepts identified through the ontology in the deep learning model decision-making process. Consequently, our approach enhances the efficiency and interpretability of the model's diagnostic capabilities, promising a more trustworthy and reliable disease detection model.

Keywords: Explainable Artificial Intelligence · Plant Disease Classification · Tomato Disease Ontology · Deep Neural Networks

1 Introduction

Addressing global hunger for a projected 9 billion people by 2050 is a crucial challenge [31]. However, obstacles like limited crop productivity, environmental concerns, and rising plant diseases hinder progress in agriculture. Hence, innovative solutions are needed. Artificial intelligence (AI) technologies promise to

A. Meroño Peñuela et al. (Eds.): ESWC 2024, LNCS 14664, pp. 289–306, 2024.
https://doi.org/10.1007/978-3-031-60626-7_16

provide such solutions. It offers unprecedented opportunities to enhance various facets of the field from precision agriculture to advanced automatized crop management [32]. One particularly promising path is the integration of deep learning, a subset of AI, in the identification of plant diseases through image data analysis. This not only enables fast and early disease detection but also paves the way for more effective and targeted intervention to minimize crop losses.

In recent years, we noticed a surge in the number of works successfully applying deep learning for plant disease image classification [2,3]. However, the reception of such models by plant scientists and farmers remains mixed due to their black-box nature. This uncertainty comes from the limited understanding of the internal process by which such models learn and encode plant disease traits and features. The absence of transparency throughout the decision-making process is a crucial concern in numerous critical application domains including plant disease diagnosis. Hence, explainability of deep learning models becomes a necessity for the swift realisation of AI practical applications in agriculture.

Different explainability methods have emerged to generate saliency heatmaps [26,30,33]. They rely mostly on the backpropagation of gradients to assess the impact of individual pixel changes on the model decision. However,compared to other fields of application of deep neural networks (DNN), plant disease classification carries an additional challenge. Plant diseases can have different symptoms such as discoloration, lesions, or abnormal growth patterns. These symptoms can be subtle and may vary depending on the disease stage making it hard to grasp without expert knowledge. Also, different plant diseases may exhibit similar symptoms. Therefore, common explanation methods such as saliency maps visualisation could not provide pertinent explanation on how much such visual concepts (i.e., color or symptom abnormalities) influence the model decision.

Hence different concept explanation methods [20,34] giving the attribution of concepts rather than pixels have been proposed. One of these methods is Testing with Concept Activation Vectors (TCAV) [4,20]. A concept represents an abstraction which could range from a simple color to an object or a complex idea [22]. Given any user-defined concept, TCAV detects if that concept is embedded within the latent feature space learned by the network [22]. Hence, in the original TCAV method [20], users were required to gather diverse image concepts manually. This posed a potential challenge, particularly for machine learning engineers lacking specialised knowledge of the specific domain in study. We propose to leverage semantic web methods to tackle this issue effectively. An ontology can provide relevant concepts experts use in identifying plant diseases and aids in automating the creation of images based on these concepts. This not only simplifies the concept collection and labelling process but also alleviates the burden on users with limited domain knowledge. It can also help avoid human bias, which may influence the choice of concepts to test since it could reflect the community's understanding. Moreover, the ontology could define abstract concepts that might not have direct visual representations but can be inferred from related concepts.

In essence, our proposed framework not only automates concept image generation for the TCAV method but also offers insights into how important these disease-related concepts identified by using the ontology are for the deep learning model's decision-making process. This approach enhances both the efficiency and interpretability of the diagnostic capabilities of the model. Hence, the aim of this paper is to provide a semantic aware concept explainability method for plant diseases based deep learning classification. We choose tomato diseases as a use case to understand what semantic concepts DNN learns. The tomato disease image dataset was extracted from the PlantVillage dataset [16].

In summary, our contributions are:

- A new ontology to represent symptoms and abnormalities associated with tomato diseases.
- Mapping of concepts learnt by DNN within its latent space for plant disease classification to semantic concept descriptions of plant diseases within the ontology using CAVs.
- Automated concept labelling and generation such as color and symptoms for TCAV within the context of plant diseases.
- Analysis of contribution of various disease-related characteristics to the predictions made by a deep neural network. This provides valuable insights on the significance of different features in the decision-making process which could help in improving the accuracy and interpretability of plant disease predictions.

The remainder of the paper is organised as follows. Section 2 introduces the possible use cases. Section 3 discusses related work on the use of ontologies in explainability. Section 4 explains the methods and proposed approach and Sect. 5 provides details about the experiment and results. Finally, Sect. 6 provides the conclusion.

2 Use Cases

Our proposed framework of combining ontology and concept explainability for tomato disease classification with deep learning can offer several benefits and use cases. Some of those potential applications could be:

- Explanation of predictions: Our framework can provide explanations for the predictions made by the deep learning model. Users such as plant experts, agriculture policy makers, regulators and stakeholders can understand why a specific classification was made, which is crucial for building trust in the model.
- Domain-Specific understanding: By incorporating the ontology, the framework can leverage domain-specific knowledge about tomato diseases. This helps in transferring insights from experts to the model developers which enhances their understanding of the context and improves model accuracy.

– Identification of relevant features: The framework can highlight the specific concepts or features within the input data that contributed the most to a particular classification. This can be valuable for researchers and practitioners to identify key indicators of tomato diseases and improve their collected dataset.
– Error analysis and improvement: The framework can help in understanding semantic errors made by the model, indicating which parts of the input data might have led to a misclassification. This information can guide further model refinement and training.

3 Related Work

In recent years, there has been an increasing interest in understanding and explaining the prediction behavior of deep neural networks. One of the most popular methods is the saliency and attribution approaches [26,27,30,33], where the explanation for the DNN is given as an importance map highlighting the contribution of each feature in its decision. Even though these methods increase the explainability of the DNN, they are limited in their understandability, leaving it to the user to interpret such maps. For instance, the importance of a single pixel in the classification does not bring a meaningful explanation, and it is also contrived by the number of features [22].

Hence, methods such as TCAV [20] present the use of "human-interpretable concepts" for explaining DNN networks. Still, no information is provided concerning how these concepts are relevant to the output of the DNN. The user also needs to collect these concepts as images, making interpreting abstract concepts hard. Consequently, a lot of researchers argue that an effective explainability of deep learning models cannot be achieved without the use of domain knowledge through the integration of semantic web technologies [12].

In [25] the authors suggest employing ontologies as background knowledge framework to facilitate obtaining formulae that interpret the functioning of deep models. In their work, the network is trained to classify scene objects. Based on the classification output, they run a DL-Learner on the Suggested Upper Merged Ontology [24] to generate class expressions that act as explanations. However, their approach is constrained specifically in its need for labeled data with the required different concepts.

Similarly, in [12], the authors proposed a neuro-symbolic framework where the semantics in the knowledge base are aligned with the annotations in the dataset. The model to explain is a DNN model trained for multi-label image classification, and the explanation was generated in a logical language. The specific focus of their study is the classification of food recipes. A different approach was proposed in [7], where the authors introduced explainable classifiers using domain knowledge. Their approach involved creating synthetic images of Pizza for training the DNN based on the specifications outlined in the pizza ontology. Then, they proposed a method that integrates a DL model with a graph of tensors automatically generated from description logic assertions extracted from the relevant ontology.

Furthermore, in [28] the authors tried to provide better explanations by mapping the internal state of neural network (neuron activations) to the concepts of an ontology to find symbolic justification for the output of DNN. This mapping is achieved by training small neural networks to predict a single concept from the DNN neuron's activation. However, this method assumes the presence of concepts in the images. That's why they used synthetic image datasets of trains modeled accordingly to present the needed concepts.

In [8], the authors proposed combining ontology with deep learning cassava disease classification. However, the ontology was only used to infer diseases based on simulated sensor observations such as temperature and soil moisture and provide extra domain knowledge about the classified disease without explaining the trained model behavior.

While current methods integrating explainability and web semantics have shown potential, many rely on deep learning models trained on synthetic images featuring a predefined set of concepts, limiting their real-world applicability. Additionally, a real-world application is burdened with the need for multi-class annotation, such as the example in [25]. In our work, we address this limitation by focusing on explaining a deep neural network trained on real-world images of plant disease. Our approach involves automatically mapping semantic concepts to activations learned within the network. The proposed system integrates an ontology applicable to any images of tomato diseases, enhancing interpretability. This contributes to a more flexible and practical model. As far as we know, this is the first approach to automatically explore associating semantic concepts with visual concepts for plant disease classification.

4 Methods

The framework illustrated in Fig. 1 represents our workflow for enhancing and automatizing concept explainability using knowledge in the form of an ontology. In the preparatory phase (pre-runtime) we train the deep neural network on a set of tomato disease images and we create the ontology. Using only the image annotations (Target Classes), the ontology can provide necessary concepts, properties and axioms related to visual tomato disease identification. For example, a bacterial spot tomato disease can be described by its appearance on the leaf with symptoms such as black coloration and spots spreading. Hence, the bacterial spot target class can be defined using the ontology axiom as ($\exists$hasSymptom.BacterialSpotsOnLeaf $\sqcap$ $\exists$hasColor.Brown). Section 4.1 will give a more detailed explanation of the used ontology. In the following part, we briefly describe the steps in our framework. First, as described above, a disease concept ontology is employed to get concepts related to different tomato disease classes in our image dataset.

Based on the target class label, the ontology provides all important properties linked to the specified disease class that could be visible in the image. For example, some of these properties (concepts) could be color or symptom texture. The generated concept labels (i.e., color brown) are then used to automatically generate corresponding images (i.e., different shades of brown images). More details

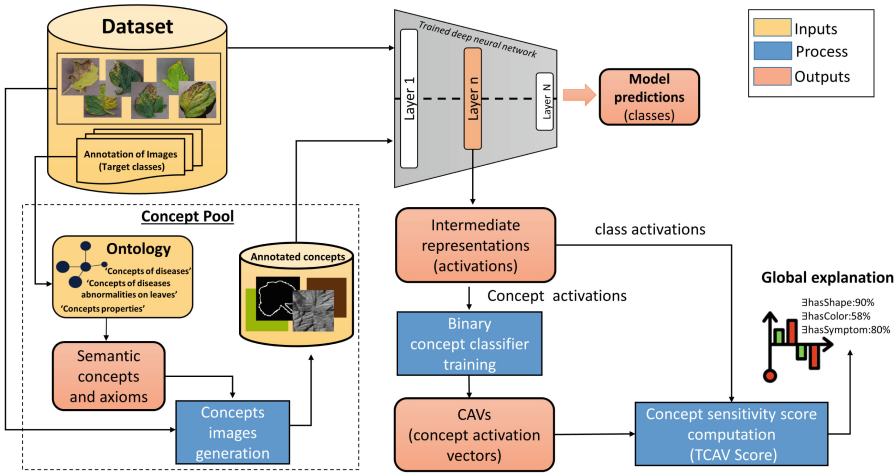

Fig. 1. The framework of the proposed method. Our modeled ontology is used to provide concepts related to the target class (disease label). These semantic concepts are then used for automatic image generation, and a list of annotated concepts such as (color, shape, and symptom) are created. Concept classifiers are then trained, and CAVs and TCAV scores are computed.

on the generation process will be described in Sect. 4.2. These generated concepts in the form of images with target class images and random images are subsequently employed to derive Concept Activation vectors (CAVs) [20] and compute the sensitivity score (TCAV score) (see Sect. 4.3). This helps us understand how sensitive the trained deep learning model is to these domain-specific concepts. For instance, we can quantify the influence of the concept ∃hasColor.Brown on the 'BacterialSpot' prediction as a single score. In the upcoming sections, we will describe various parts of our framework in detail. First, we will introduce our ontology and the modelling process (Sect. 4.1), then we will explain the process for generating images related to the concepts (Sect. 4.2). Finally, we will provide an overview of the TCAV algorithm (Sect. 4.3).

4.1 Ontology Development (Ontology Based Explanation)

This section describes the steps for developing the Tomato Disease Concepts (TomatoDCO) ontology. It uses OWL for modelling knowledge about classes, properties and axioms related to phenotype of various tomato diseases. We follow best practice recommendations on ontology engineering [23] to develop this ontology.

Ontology Requirements. The first step to developing any ontology is to define its scope, specifying the aspects it aims to model. In our work, the ontology is

designed to include the tomato plant diseases domain. The ontology should also provide the different appearances (visual concepts) related to tomato disease. We integrate the TCAV explainability method with an ontology to provide a more comprehensive understanding of the model's decision-making process. The ontology will serve as a structured knowledge base, describing and modelling each disease class's specific symptoms and abnormalities. These properties will be then used to generate associated concept images. Following the scope, the ontology should be able to answer the following competency questions (CQs):

- What are the diseases that tomato plants could have?
- What are the possible symptoms and appearances of tomato disease X?
- What are the diseases that are caused by bacteria, fungi, viruses, or insect damage?
- What are the diseases if a tomato plant has symptoms/appearance A,B,..?

Ontologies Reuse. To model our ontology, we follow the recommendation to start by checking existing ontologies focused on plant diseases. We reuse relevant elements to help achieve our goal of creating explanations of our trained neural network, particularly concerning the diseases existing in our image dataset. To develop our TomatoDCO ontology, we reused and followed the disease hierarchy from [18,19], presenting a rice disease ontology (RiceDO) that helps identify rice diseases from existing symptoms in the plant. It was evaluated and assessed by ontology experts and senior agronomists, where important criteria such as appropriateness, consistency, and ontology satisfaction were considered [19]. The other most pertinent ontologies for our case are Plant Protection Ontology (PPO) [5], Plant Disease Ontology (PDO) [17], and Phenotype and Trait Ontology (PATO) [15]. PATO defines various phenotypic traits across different species. These traits include characteristics like color (e.g., brown, black), temperatures (e.g., high, low), and symptoms (e.g., swelling) [18]. PDO defines diseases in maize, wheat, and rice, categorized into bacteria, fungi, and viruses. PPO classifies barley disorders into abiotic and biotic (with further subcategories for bacteria, fungi, and viruses). RiceDO used and extended PDO, PPO, and PATO ontologies under the domain of rice diseases. It classifies diseases into bacteria, fungi, and viruses. Even though these ontologies serve as a valuable reference for comprehending and categorizing plant diseases and disorders, they are developed to integrate them with a decision expert system, which differs from our goal. Hence, we reused concepts that help our aim of providing properties associated with each disease visual manifestation that could be exploited later as concepts for our explainability algorithm. These existing ontologies (i.e., RiceDO and PDO) also don't include information on tomato diseases. Therefore, we adopt their approach of classifying diseases in defining specific classes relevant to tomato disease.

Concepts Identification. Since our image dataset is extracted from the PlantVillage dataset [16] available under an open licence [1], we use it as our primary knowledge source along with [6] to collect information about signs and

symptoms associated with the mentioned diseases for our ontology. The most important concepts we identified in this step were different types of diseases such as bacterial (i.e., bacterial spot), fungal (i.e., early blight, late blight, leaf mold, septoria leaf spot, and target spot), viral (i.e., mosaic virus and yellow leaf curl virus) and diseases due to insect damage such as two-spotted spider mites disease. Diseases symptoms could be visual abnormalities such as changes in color, for example, black, brown, and yellow, and changes in leaf shape. Also, the emergence of textural changes on the leaf, such as blight or spots.

Classes Definition and Classes Hierarchy. The structure of our ontology, TomatoDCO, is shown in Fig. 2 . We have two top-level classes: 'TomatoDisease' and 'Abnormality' and three object properties (hasColor, hasShape and hasSymptom).

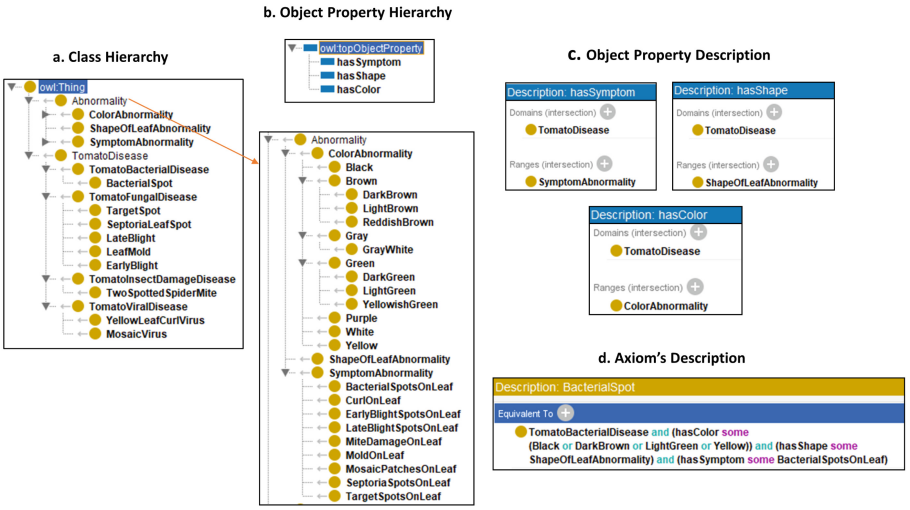

Fig. 2. The structure of TomatoDCO ontology is divided into three parts: (a) the class hierarchy of TomatoDCO; (b) the object properties of TomatoDCO; (c) the object properties descriptions (range and domain); and (d) an example of axioms representing concepts of abnormalities of the tomato bacterial spot disease.

A detailed description of these components is given in the following:

– Abnormality: We reuse this class from RiceDO ontology and PPO and extend it to meet our requirements. This class presents the kind of abnormalities visually noticed on a plant when it is affected by the disease. These abnormalities concepts are important to quantify how sensitive they are for our plant disease-trained model and to know to which extent our model is learning the true semantic representation of a disease. As shown in Fig. 2, they include:

  - ColorAbnormality: The different color changes that could emerge because of the disease (e.g., Brown). The terms of colors are mapped to the existing ones in RiceDO ontology by using owl:equivalentClass, which is also mapped to the existing ones in PATO.
  - ShapeOfLeafAbnormality: the abnormalities that happen to the shape of the leaf because of the disease.
  - SymptomAbnormality: the symptoms of abnormalities of a leaf affected by a disease can vary according to the specific pathogen causing the problem, such as the emergence of spots or patches on the leaf (e.g., 'having a bacterial spot symptom on leaves' can be defined by BacterialSpotsOnLeaf)
- TomatoDisease: This class classifies the tomato diseases into bacterial, fungal, and viral, like the PDO and RiceDO ontologies, and also adds the class for diseases caused by insect damage since this could occur in the real world. It is worth noting also that PDO also lacks information regarding tomato diseases.

Object Properties. In our use case, we define three object properties that are necessary to describe the appearance of each tomato disease and are useful for extracting the required semantic concepts for the TCAV method.

- hasColor: This property defines a relation from TomatoDisease to ColorAbnormality.
- hasShape: This property defines a relation from TomatoDisease to ShapeOfLeafAbnormality.
- hasSymptom: This property defines a relation from TomatoDisease to SymptomAbnormality.

These properties will be then used to axiomatize the various visual abnormalities that occur on a leaf when affected by a certain disease.

Concept Definition. The appearance of each tomato disease is described in class description by using equivalent-to relation. For example, a tomato bacterial spot disease can cause the emergence of bacterial spot lesions that develop randomly on the leaflets, and they turn brown or black and sometimes have a yellow hallo. In some cases, entire leaves can turn yellow and wilt [6]. Since in our dataset images of the disease come from different stages, we make sure to integrate all the possible colours of abnormality. Hence as shown in Fig. 2.d, these could be described as:

$$\begin{aligned} \text{BacterialSpot} \equiv & \text{TomatoBacterialDisease} \sqcap \\ & (\exists \text{hasSymptom.BacterialSpotsOnLeaf}) \sqcap \\ & (\exists \text{hasColor.}(\text{Black} \sqcup \text{Yellow} \sqcup \text{DarkBrown} \sqcup \text{LightGreen})) \sqcap \\ & (\exists \text{hasShape.ShapeOfLeafAbnormality}) \end{aligned}$$

Hence, the TomatoDCO ontology is used to help the mapping between the visual level (target class image, i.e., bacterial spot) and the semantic level (what is the

disease corresponding concepts (i.e., color, symptom, and shape)). In the following section, we describe how these concepts extracted thanks to the ontology could be defined visually as images.

4.2 Synthetic Concepts Images Generation

The texture of a leaf can provide valuable insights into the health of a plant, as changes in texture are frequently associated with specific diseases. Symptoms such as wilting, discoloration, or lesions may manifest, affecting the uniformity of the leaf surface. Hence, we propose visually representing the disease symptom (hasSymptom) by getting the texture details from the leaf image while excluding shape and color information. We design three different visual concept generation methods for texture (hasSymptom), shape (hasShape), and color (hasColor) separately.

Texture Generation Method. For texture generation, we follow the method described by Ge et al. [13]. The method is based on initially segmenting the leaf images from the background. Then, in order to eliminate color information, the segmented leaf is converted into a grayscale image. Subsequently, the grayscale image is divided into multiple square patches using an adaptive strategy where the patch size and location adjust according to the leaf size to include a broader range of texture information. If the overlap ratio between a patch and the original leaf segment exceeds a specified threshold τ (set to 0.99 in our experiments, indicating that over 99% of the patch area belongs to the object) the patch will be included in the patch pool. Four patches are randomly selected from the pool and then concatenated into a new texture image to capture both local (individual patch) and global (entire image) texture characteristics. This generated texture image corresponds to the target class disease symptom defined in our ontology. The segmentation step is omitted since we already have a segmented version of our image dataset. Figure 3 visualises the used method.

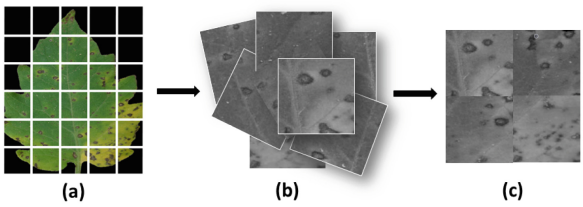

Fig. 3. The process for extracting texture. (a) Crop images and compute the overlap ratio between the 2D mask and patches. Patches with overlap > 0.99 are shown in a green shade. (b) add the valid patches to a patch pool. (c) is the final texture feature, the concatenation of k randomly selected patches from the patches pool. (Color figure online)

Color Generation Method. We extract the color concepts associated with the target class (e.g., bacterial spot) using the TomatoDCO ontology that specifies distinct colors linked to each tomato disease class. Then, we automate the generation of different images of the same color with varying intensities through the random perturbation of RGB values within the predefined color spectrum. These images will be used later with the TCAV method to quantify the model's sensitivity to the corresponding color when classifying a particular class.

Shape Generation Method. When a leaf is affected by a disease, its shape witnesses different changes. Some diseases cause damage along the edges of the leaves, resulting in distortion and curling. Hence, we visualize the shape abnormality concept by extracting only the shape edge using binary segmentation. An example is presented in Fig. 4.

Fig. 4. Example of extracted shape contours

4.3 Testing with Concept Activation Vectors (TCAV)

In this section, we provide a brief overview of CAVs and outline the approach employed in this study to compute TCAV scores. These scores measure the influence of a semantic concept on the predictions made by DNN. The TCAV method was proposed by kim et al. [20] to explain deep neural models without any retraining. A key component within TCAV is the concept activation vector (CAV) v_c^l, a vector representation of a concept within a specified convolutional layer l of DNN. To identify CAV in layer l, a set of positive and negative examples representing concept and non-concept (i.e., random) instances is needed. These examples are represented in the form of images and a binary linear classifier is trained to distinguish between them. The vector orthogonal to the decision boundary separating the two classes, i.e., the vector pointing in the direction of the representations of the concept images, is the CAV. To assess the impact of a CAV on a class of input images, the authors proposed the TCAV score metric. It uses directional derivatives, denoted as $S_{C,k,l}(x)$, , to gauge the contextual sensitivity of a concept across an entire input class, offering comprehensive explanations. The formula for calculating the TCAV score is as follows:

$$TCAV_{Q_{C,k,l}} = \frac{|x \in X_k; S_{C,k,l}(x) > 0|}{|X_k|} \tag{1}$$

where k denotes the class labels, X_k represents all inputs, and $S_{C,k,l}(x)$ is the directional derivative of a sample's activation x from layer l concerning class k and concept C. The TCAV score calculates the ratio of the class k's inputs positively influenced by concept C. To make sure that only meaningful CAVs are taken into account, a statistical significance two-sided t-test is performed [20].

5 Experiments and Results

5.1 Dataset and Trained Model

All experiments were performed using the Inception-V3 [4,29] model fine tuned on the tomato images from the PlantVillage dataset [16]. The model was created and loaded with pretrained weights on the ImageNet dataset [11] and top new layers were added. They consist of three dense layers with corresponding dropout layers. For training and optimizing the weights on the tomato disease dataset, we froze the first 51 convolutional layers and made the rest trainable for InceptionV3. Training optimization was carried out via a stochastic gradient descent optimizer with a learning rate of 0.0001 and momentum of 0.9. We used a batch size of 20 and 20 epochs for training. We use data augmentation techniques to increase the dataset size in training and solve the class imbalance while including different variations. These variations consist of transformations such as random rotations, zooms, translations, shears, and flips to the training data as we train. The performance of the trained models is evaluated using recall, precision, and accuracy metrics [10].

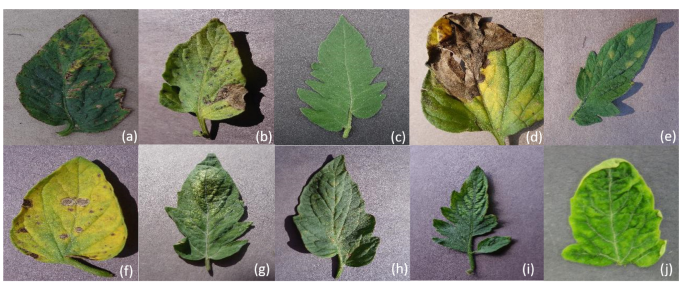

Fig. 5. Sample images from the PlantVillage Dataset. (a) Bacterial Spot, (b) Early Blight, (c) Healthy, (d) Late Blight, (e) Leaf Mold, (f) Septoria Leaf Spot, (g) Two-spotted Spider Mites, (h) Target Spot, (i) Mosaic Virus, and (j) Yellow Leaf Curl Virus. (Color figure online)

The total number of images is 18,160, divided into ten classes (nine diseases and a healthy class). The data was separated into three sets, containing 80% of the data in the training set; the remaining 20% were divided between the testing and validation sets. Figure 5 presents one example of each disease class. The

Table 1. Per class precision, recall, F1-Score for the test set Class.

Class Label	Sample Count	Precision	Recall	F1-score
Bacterial Spot	191	1	0.64	0.78
Early Blight	119	0.99	0.62	0.76
Late Blight	178	0.89	0.99	0.94
Leaf Mold	77	0.96	0.92	0.94
Septoria Leaf Spot	198	0.9	0.96	0.93
Two-spotted Spider Mite	177	0.87	0.99	0.93
Target Spot	142	0.7	0.96	0.81
Yellow Leaf Curl Virus	534	0.99	1	0.99
Mosaic Virus	37	0.97	0.89	0.93
Healthy	163	0.98	0.99	0.98

trained model achieved the following training, validation and testing accuracies, respectively: 0.98, 0.92 and 0.92.

Table 1 shows the precision, recall, and F1-score for each class. The model was implemented using Keras [9], and was saved for subsequent interpretability analysis. We experimented on a server with a GPU that consists of two NVIDIA Tesla V100 with 128 GB of RAM.

5.2 Experimental Setup

In this work, our aim is to study the correlation between concepts derived from a domain ontology modelling the knowledge about diseases and those learned within the activation of the neural networks. For example, if a neural network is trained to identify late blight disease, then ontological concepts representing the disease like ∃hasColor.Black and ∃hasSymptom.Blight should be important for the decision. It is worth noting that none of these concepts were part of the predefined class labels of the network; rather they were all derived through ontology reasoning. Hence, the first step of our approach is exploiting the ontology to identify and automatically generate concepts specific to each disease class, as described in Sect. 4.2.

The subsequent step involves utilising the generated images that represent each concept for training the concept activation vectors (CAVs). To train CAVs, a set of 30 images per concept was generated. The selection of this number aligns with the recommendation in the original TCAV paper [20], where it is asserted that such a number suffices to learn CAVs. For the target classes, we randomly chose 30 images for each from the training set. Images for creating ∃hasSymptom concepts were selected randomly from the test set which the model was not trained on. We used the "mixed_8" bottleneck layer of the InceptionV3 model for these experiments. As demonstrated in [14,20], initial layers are better at capturing textures and colors while later ones are better at recognizing objects;

the choice of the "mixed_8" layer balances between these considerations. We used images of healthy tomatoes without any presence of disease as random (i.e., non concept) images. We believe this choice allows training CAVs with a better fine-grained recognition. The TCAV score is used to evaluate the concept's importance to a specific target class. To check statistical significance of learned concepts, we trained an additional 70 random CAVs. The distribution of random concept TCAV scores and actual concept TCAV scores was then compared by conducting a two-sided t-test [21] with ($\alpha = 0.05$) to assure significance of the found CAVs. In the results section, statistical insignificance is represented by stars. Our code and ontology can be accessed on GitHub[1].

5.3 Findings and Analyses

To evaluate our approach, we will concentrate on the quantitative evaluation of TCAV scores. Figure 6 presents the different sensitivity score (TCAV) highlighting the contribution of the semantic concepts to their relevant corresponding neural classification. Essentially, TCAV quantifies the impact of a given concept on a specific target class. For instance, in the case of Tomato Mosaic Virus, the results show a high TCAV score for all the semantic concepts such as ∃hasColor.Yellow (0.97) and ∃hasSymptom.MosaicPatchesOnLeaf (0.9). Contrarily, for the disease Target Spot the model did not learn the concept ∃hasSymptom.TargetSpotsOnLeaf, which may explain the low precision of the class as shown in Table 1. This suggests that the model may not be optimal for robustly detecting the TargetSpot disease and that gathering more training data with clear symptom texture existence could enhance the results. In contrast, for Septoria Leaf Spot, even though the symptom concept was not important, the abnormalities in concepts like hasColor and hasShape were sufficient for the model to identify the class. Additionally, for the class 'Two-spotted spider mite', none of the disease's semantic concepts made a significant contribution. This suggests that the model may not have effectively learned these important semantic concepts associated with this disease class. However, the class achieved a precision of 0.8, indicating that the model is learning to identify this class through another bias in the dataset. This insight highlights the need for a closer examination of the class and the model.

The results further highlight the significance of color concepts for different disease classes, supporting the findings of [16] where they observed an accuracy drop when the model was trained on grayscale images. In summary, these findings provide insights into the contribution of different semantic concepts in the decision of the model which shows to which extent the model is consistent with domain knowledge. Our approach not only relates the classified diseases to their symptoms and signs but also tries to quantify the contribution of these symptoms to the model decision.

[1] https://github.com/fusion-jena/XAI_TCAV_ONTO.

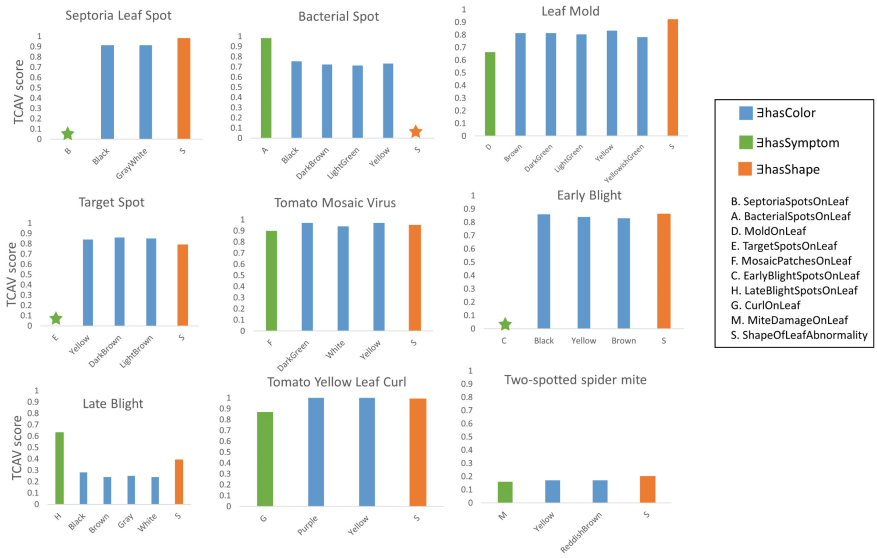

Fig. 6. Conceptual importance (TCAV scores) for the different disease semantic concepts for each class

6 Conclusion

With this work, we contribute to improving the explainability, dependability and trustworthiness of deep learning models by adding expert knowledge through an ontology. For our implementation, we focus on the identification of plant diseases as a use case. Our novel approach automatically generates concepts related to observable disease features using the ontology. This lets users peek into the model and see how its results depend on these concepts, all without needing to manually collect concepts. Through experimental evaluation, we showed the sensitivity of the model to these concepts. By formalizing expert knowledge in an ontology, we can enhance our comprehension of the relationships between various concepts within a model and also make the examination and correction of misclassifications and biases easier. We believe that our approach could be easily extended to other domains due to different points. First, our ontology is built upon a conceptual framework that involves color, symptom (texture), and shape abnormalities. This framework is not specific to tomato diseases and can be adapted to cover characteristics in other plant diseases or domains. Second, our ontology design is modular and flexible. Separating disease characteristics and types into distinct modules made the inclusion of new diseases and their corresponding specific concepts easier. Third, the most important features when describing images such as leaves or other objects are shape, color, and texture. Our proposed approach for generating images for such concepts is domain-independent, which shows its adaptability behind the current use case. Despite our findings, we acknowledge some challenges, like the difficulty of capturing all expert knowledge in an ontol-

ogy. Also, further detailed tests with a high-quality dataset are needed for more comprehensive interpretation of the TCAV scores for this particular use case.

In future work, we aim to test our approach on more challenging plant diseases datasets where leaves could be infected by more than one disease. We plan also to consider how combining neural and semantic representation via knowledge graphs can be generalised to other problems such as object detection and image classification. The explainability framework can be integrated into a decision support system, providing actionable insights to farmers and stakeholders for disease management and crop protection.

Acknowledgement. Supported by the Carl Zeiss Foundation (project 'A Virtual Werkstatt for Digitization in the Sciences (K3)' within the scope of the programline 'Breakthroughs: Exploring Intelligent Systems for Digitization - explore the basics, use applications').

References

1. Plantvillage. www.plantvillage.psu.edu. Accessed 13 Nov 2023
2. Ahmad, A., Saraswat, D., El Gamal, A.: A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. Smart Agric. Technol. **3**, 100083 (2023)
3. Amara, J., Bouaziz, B., Algergawy, A.: A deep learning-based approach for banana leaf diseases classification. Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband (2017)
4. Amara, J., König-Ries, B., Samuel, S.: Concept explainability for plant diseases classification. In: Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP, pp. 246–253 (2023)
5. Ammar, H.: Ontology for plant protection. https://sites.google.com/site/ppontology/home (2009)
6. Blancard, D.: Tomato Diseases: Identification, Biology and Control: A Colour Handbook. CRC Press, Boca Raton (2012)
7. Bourguin, G., Lewandowski, A., Bouneffa, M., Ahmad, A.: Towards ontologically explainable classifiers. In: Farkaš, I., Masulli, P., Otte, S., Wermter, S. (eds.) ICANN 2021. LNCS, vol. 12892, pp. 472–484. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86340-1_38
8. Chhetri, T.R., Hohenegger, A., Fensel, A., Kasali, M.A., Adekunle, A.A.: Towards improving prediction accuracy and user-level explainability using deep learning and knowledge graphs: a study on cassava disease. Expert Syst. Appl. **233**, 120955 (2023)
9. Chollet, F.: Deep learning with Python. Simon and Schuster (2021)
10. Davis, J., Goadrich, M.: The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240 (2006)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)

12. Donadello, I., Dragoni, M.: SeXAI: introducing concepts into black boxes for explainable artificial intelligence. In: Proceedings of the Italian Workshop on Explainable Artificial Intelligence co-located with 19th International Conference of the Italian Association for Artificial Intelligence, XAI. it@ AIxIA 2020, Online Event, 25–26 November 2020, vol. 2742, pp. 41–54. CEUR-WS (2020)
13. Ge, Y., Xiao, Y., Xu, Z., Wang, X., Itti, L.: Contributions of shape, texture, and color in visual recognition. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13672, pp. 369–386. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19775-8_22
14. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
15. Gkoutos, G.V., Green, E.C., Mallon, A.M., Hancock, J.M., Davidson, D.: Using ontologies to describe mouse phenotypes. Genome Biol. **6**, 1–10 (2005)
16. Hughes, D., et al.: An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060 (2015)
17. Jaiswal, P., et al.: Planteome: a resource for common reference ontologies and applications for plant biology (2017)
18. Jearanaiwongkul, W., Anutariya, C., Andres, F.: An ontology-based approach to plant disease identification system. In: Proceedings of the 10th International Conference on Advances in Information Technology, pp. 1–8 (2018)
19. Jearanaiwongkul, W., Anutariya, C., Racharak, T., Andres, F.: An ontology-based expert system for rice disease identification and control recommendation. Appl. Sci. **11**(21), 10450 (2021)
20. Kim, B., et al.: Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In: International Conference on Machine Learning, pp. 2668–2677. PMLR (2018)
21. Koch, G.G.: One-sided and two-sided tests and ρ values. J. Biopharm. Stat. **1**(1), 161–170 (1991)
22. Molnar, C.: Interpretable machine learning. Lulu.com (2020)
23. Noy, N.F., et al.: Ontology development 101: a guide to creating your first ontology (2001)
24. Pease, A., Niles, I., Li, J.: The suggested upper merged ontology: a large ontology for the semantic web and its applications. In: Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, vol. 28, pp. 7–10 (2002)
25. Sarker, M.K., Xie, N., Doran, D., Raymer, M., Hitzler, P.: Explaining trained neural networks with semantic web technologies: first steps. arXiv preprint arXiv:1710.04324 (2017)
26. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
27. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: SmoothGrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
28. de Sousa Ribeiro, M., Leite, J.: Aligning artificial neural networks and ontologies towards explainable AI. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4932–4940 (2021)
29. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)

30. Tjoa, E., Khok, H.J., Chouhan, T., Guan, C.: Enhancing the confidence of deep learning classifiers via interpretable saliency maps. Neurocomputing **562**, 126825 (2023)
31. Van Dijk, M., Morley, T., Rau, M.L., Saghai, Y.: A meta-analysis of projected global food demand and population at risk of hunger for the period 2010–2050. Nat. Food **2**(7), 494–501 (2021)
32. Wakchaure, M., Patle, B., Mahindrakar, A.: Application of AI techniques and robotics in agriculture: a review. Artif. Intell. Life Sci. 100057 (2023)
33. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
34. Zhou, B., Sun, Y., Bau, D., Torralba, A.: Interpretable basis decomposition for visual explanation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 119–134 (2018)