Semantic-Search: A Knowledge-Driven Classification Method for Accurate Identification of Plant Diseases

Kiran K. Kethineni*, Saraju P. Mohanty* and Elias Kougianos†
*Department of Computer Science and Engineering, University of North Texas, USA
†Department of Electrical Engineering, University of North Texas, USA
Email: kirankumar.kethineni@unt.edu, saraju.mohanty@unt.edu, elias.kougianos@unt.edu

***Abstract*—As the global population is increasing and agricultural resources are diminishing, effective management of plant diseases become critical. Convolutional Neural Network (CNN)-based models have been widely adopted in Agriculture Cyber-Physical Systems (A-CPS) due to their high accuracy. However, these models require large, labeled datasets for training, limiting their practicality in plant disease detection where data is heterogeneous, scarce, and subjected to continual evolution. To address these challenges, we propose Semantic-Search, a knowledge-driven system that interprets plant diseases based on high-level semantic features like visual patterns, colors, and crop type, rather than relying solely on low-level image features. The system classifies diseases by querying a structured knowledge base using these semantics. "semantic search" here refers to classification based on semantic feature extraction and knowledge base querying. It enables greater adaptability, interpretability, and robustness to intra-class variability caused by differing environmental conditions. The system can incorporate new disease classes through simple knowledge base updates, eliminating the need for retraining. This facilitates efficient scaling to evolving set of diseases. We validated the proposed approach on a dataset of 11,000 images encompassing 21 diseases across 11 plant species, achieving an accuracy of 90%, thereby demonstrating its effectiveness, scalability.**

***Keywords*—Agriculture Cyber-Physical System (A-CPS); Convolutional Neural Networks (CNN); Natural Language Processing (NLP); Cognitive Computing; Semantic Understanding; Knowledge-driven; Database Query**

I. Introduction

Advancements in the Internet of Things (IoT) and imaging techniques have been integrated into Agriculture Cyber-Physical Systems (A-CPS) [1] to detect disease infestations at early stages and enable timely treatment. While imaging techniques can effectively detect whether crops are diseased, they are not robust enough to accurately classify the specific type of disease. To address this limitation, widespread development of CNN-based methods using Artificial Intelligence (AI) and Machine Learning (ML) started within the framework of precision agriculture [2].

All CNN-based classification and segmentation models consist of two primary components: a feature extractor and a classifier. The feature extractor uses a series of 2D or 3D kernels at various scales to capture the most significant features effectively. The classifier layer receives the extracted features and uses them to determine the unique combination of characteristics that define each class [3]. The feature extractor and classifier are trained using machine learning algorithms, eliminating the need for human intervention in curating them. They rely entirely on the data provided during training, learn the most effective features and classification rules. This modeling paradigm is termed Data-Driven Classification as depicted in Fig. 1. So, to add a new class to the model, it must be provided with a large number of images representing that class, and the model must be retrained. This process involves significant computational resources and human effort for labeling the images.

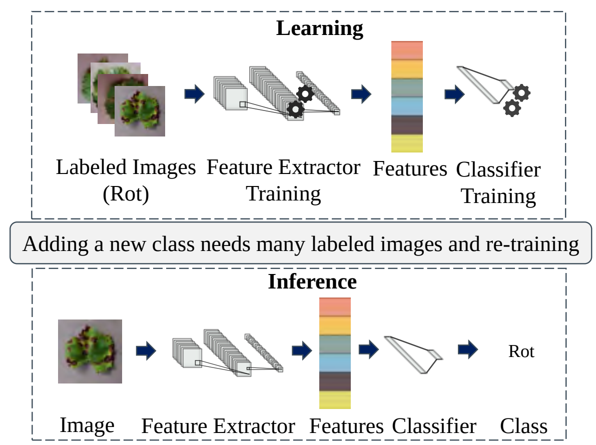


Figure 1: Illustration of the data-driven approach.

In contrast to data-driven methods, this article "Semantic-Search" introduces a novel approach that performs classification based on the knowledge embedded within the model. The proposed method semantically analyzes the image and utilizes the detected semantic features along with an associated knowledge base to achieve classification through a knowledge-driven approach [4].

The rest of the paper is organized as follows: Section II introduces our proposed solution and explains its new contributions in detail. In Section III, we provide a brief overview of previous work in this area, helping to set the context for our approach. Section V explains our proposed method in detail. In Section VI, we present the experimental results. Finally, Section VII concludes the article and offers suggestions for future research.

II. Novel Contributions of the Current Paper

A. Problem Addressed in the Current Paper

In agriculture, the vast diversity of plant species and the multitude of diseases affecting them pose significant challenges for image-based classification. A single disease may exhibit different visual features across varying geographic and environmental conditions, resulting in an exceedingly large number of disease classes. Consequently, acquiring a sufficiently comprehensive set of leaf images for every class to effectively train a CNN model is infeasible. This inherent data dependency of data-driven CNN-based AI models renders them unsuitable for scaling and adapting to new disease classes. Semantic-Search aims to overcome these challenges by classifying images based on their semantic characteristics, making it more adaptable to new classes and transparent in its decision-making process. It enables rapid development of localized models tailored to specific geographic regions and effectively handles intra-class variability caused by environmental factors.

B. Proposed Solution of the Current Paper

Table I: Brief overview of a few disease semantics.

Plant	Disease	Symptom	Semantics
Apple	Cedar rust	Orange or yellow patches encircled by a red band, that turn brown as the infection progresses.	Objects: Patches Colors: Brown, Red, Yellow, Orange
Apple	Powdery mildew	White, velvety patches appear on the underside of the leaves.	Texture: Velvety Colors: White
Corn	Rust	Brown flecks appear on the surface of the leaves.	Objects: Flecks Colors: Brown
Grape	Rot	Brown lesions appear on the leaves, eventually developing black.	Objects: Lesions Colors: Brown, Black
Tomato	Black mold	Presence of black or brown lesions.	Objects: Lesions Colors: Black, Brown

120 types of plant diseases were studied to analyze their semantics and visual features. Our examination revealed that while each disease related to a specific crop has distinctive characteristics, different diseases across various plants may share similar visual traits. This pattern is evident in diseases such as Blight disease in corn and Anthracnose disease in grape plants exhibit spots with gray or brown coloration. Powdery mildew disease appears as a white, powdery surface on grape plants, whereas in apples, it manifests as a velvety texture. Table I provides an overview of a few plant diseases descriptions from the Plant Village website [5] and the corresponding semantic characteristics they exhibit. Thus, despite the vast number of plant leaf diseases, each disease can be represented as a combination of a semantic feature and a few colors from a limited set of colors and semantic features. This structured representation significantly reduces the feature space of the classification model, enabling it to classify a large number of diseases more efficiently while improving its generalization ability.

While CNN-based AI models rely heavily on data for learning and classification, humans process images using their cognitive abilities and domain knowledge to make informed decisions. Human brains are embedded with cognitive skills necessary to detect various objects in an image and semantically understand it, as well as to identify entities and objects in a known language. These cognitive abilities enable the retrieval of the corresponding label for the image based on acquired knowledge [6]. However, since the necessary cognitive abilities are already present, adding a new class requires updating only the knowledge.

By using CNN-based models to identify, localize the semantic patterns present in leaf diseases and applying color filters to detect the colors in the diseased area, meaningful textual-visual features can be extracted. When combined with the crop name, these features can effectively classify the disease by querying an associated database that serves as a knowledge base. On the other hand, integrating Natural Language Processing (NLP) into the system will enable it to effectively identify entities such as plant names, disease names, semantics, and colors from the presented text. This capability allows the model to extract disease-related information from text, accurately associate it with its semantic features, and create a record in the database, contributing to continuous knowledge updating.

Thus, this paper proposes mimicking human cognitive abilities where by CNNs are utilized for cognitive computing while NLP, and a Database serve as a knowledge base. This approach leverages a knowledge-driven mechanism, as illustrated in Fig. 2, enabling the classification model to learn a new class from a simple text input without requiring a large number of labeled images. As a result, it significantly reduces computational power requirements and minimizes human effort needed.

C. Novelty and Significance of the Proposed Solution

The novel contributions of Semantic-Search are as follows.

1) Semantic Understanding: The proposed approach emphasizes identifying patterns and objects within the diseased area, along with the semantic description of the disease, rather than rigidly coupling the model to specific low level features of individual diseases.
2) Knowledge-Driven Approach: Instead of relying solely on data, the classification method leverages a knowledge-based system, allowing the addition or removal of disease classes through simple text inputs and database updates.
3) Interpretability and Explainability: The Semantic-Search approach provides users with a description of the identified semantics, offering a transparent explanation for the classification decision, unlike black-box methods [7].
4) Scalability: The proposed knowledge update mechanism is simple and efficient, making it easy to scale to a larger number of diseases without requiring extensive computational resources.

III. Related Prior Works

Image classification and object recognition have attracted significant interest from researchers, resulting in the development of various classification methods. An image processing

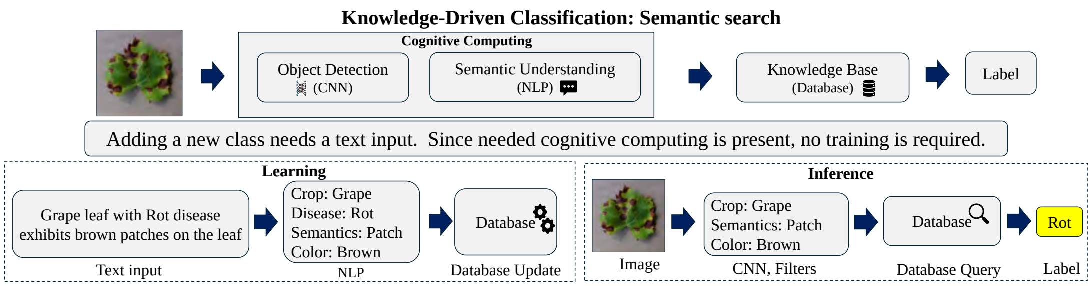


Figure 2: Proposed Semantic-Search (Knowledge driven approach).

Table II: Brief summary of relevant literature.

Research	Year	Method adopted	Remark	Comparison
Park et al. [8]	2004	Texture feature classification by neural network.	Lacks semantic understanding and requires retraining for new classes.	Less adaptive than Semantic-Search, which avoids retraining and incorporates new classes via text inputs.
Agrawal et al. [9]	2011	Image histogram comparison using an SVM classifier.	Does not consider shape or texture, leading to potential misclassification.	Semantic-Search leverages multiple semantic cues (texture, color, crop type), reducing misclassification risk.
Kai. [10]	2019	Classification using color, texture, and shape features with an SVM model.	Requires retraining when adding new classes.	Semantic-Search eliminates retraining and scales efficiently with knowledge base updates.
Vailaya et al. [11]	2001 1999	Low-level feature extraction combined with Bayesian networks for hierarchical classification.	Requires retraining for new classes.	Semantic-Search directly integrates high-level semantics, offering more interpretability and flexibility than Bayesian classification of low-level features.
Yang et al. [12]	2007	Representing key points as text using Bag of Visual Words and classifying the word histogram with SVM.	Lacks explicit semantic meaning in classification.	Semantic-Search employs explicit semantic features in knowledge base, leading to better interpretability and adaptability.
Su et al. [13]	2012	Bag of visual words and semantic attributes.	Improves interpretability but still relies on feature vector classification.	Semantic-Search extends semantic attributes with structured knowledge querying, enabling direct disease class addition without retraining.
Marino et al. [14] Menglong et al. [15]	2017 2018	Knowledge maps are built to model relationships between objects in images for classification.	Requires complex knowledge map construction and traversal.	Semantic-Search simplifies knowledge modeling via direct semantic-keyword matching, making updates easier and less resource-intensive.
Jearanaiwongkul et al. [16]	2018	Ontology-based classification of plant diseases using structured knowledge from farmers.	Requires a structured ontology, farmers have to input their findings to classify.	Semantic-Search generalizes ontology-based methods by using visual semantics and knowledge bases, reducing reliance on manual farmer input.
Semantic-Search	2025	Semantic understanding with knowledge base search.	Does not require retraining, new classes can be added efficiently via text inputs.	NA

algorithm extracting texture features was proposed in [8] to classify images by feeding the texture feature vector into a neural network, rather than relying on fine-grained details as in CNNs. To interpret image content, the authors of [9] computed the image histogram and compared it with histograms of images from known classes using an SVM. However, since this approach does not account for shapes, the article [10] incorporated color, texture, and shape features for classification. In [11], the authors aimed to mimic human perception of images by computing low-level features using color and edge information. These features were analyzed hierarchically through Bayesian networks, requiring retraining to incorporate a new class.

While the above methods used content and semantics in the image to classify them, the following methods try to represent the findings in the image as text and classify them. Bag of visual words has been used to classify scene in [12]. They represented findings in the image as histogram of words and is presented to a SVM classifier for classification. Authors of [13] proposed usage of semantic attributes to overcome the limitation of lacking explicit meanings in bag of words. An image may contain multiple objects, and the relationships between these objects can play a crucial role in classification. This key characteristic of images has been explored in [14] and [15], where knowledge maps were developed to capture relationships between real-world objects. These maps were then traversed using identified objects to aid in image classification. Similarly, [16] explored an ontology-based approach for classifying plant diseases, leveraging insights gathered from farmers. These methods either require retraining of the classifier or rely on intricate knowledge maps, which are complex to construct and traverse. To overcome these challenges, we propose a semantic understanding framework wherein image features are interpreted meaningfully and matched against a relational database serving as a knowledge base. This approach enables the addition of new disease classes without retraining the model and with minimal human intervention. A brief overview of the related works, along with a comparison to the proposed method, is presented in Table II.

IV. Knowledge-Driven Feature Engineering

From Table I, we observe that each plant leaf disease can be represented as a combination of an object and specific colors. Table III provides a consolidated list of the semantics derived from the plant diseases studied. Colors with very similar hues, where human perception may not reliably distinguish, were merged. For example, orange was merged into brown due to their close visual similarity.

Similarly, to add a new class to the model, a text input describing the disease is provided, and the corresponding entities are extracted using NLP methods. Table IV presents the different types of entities to be identified in the given text.

Table III: Overview of engineered semantics.

Semantics	Instances
Shape	Spot (Spots, Lesions, Patches), Flecks, Curls, Stripes
Color	Yellow, Black, Brown, White, Red, Gray, Pink
Texture	Powdery, Mosaic, Velvety

Table IV: Types of entities to identify using NLP.

Entity Type	Examples
Plant	Apple, Tomato, Corn
Disease	Powdery Mildew, Rust, Blight
Color	Yellow, Brown, Dark Green
Semantic	Spots, Stripes, Patches, Powdery

V. The Proposed Method

The proposed knowledge-driven method consists of two distinct paths: one for learning new classes and the other for performing inference, both utilizing a common database. The detailed construction and functionality of the model are presented in the following subsections.

A. Learning

To emulate human learning, we propose training the model using textual input from the user, where the disease is described through semantic attributes and visual features. The following subsections outline the methods employed for learning, as illustrated in Fig. 3.

1) Natural Language Processing: Analyzing human language to extract meaning or context from a given text is known as Natural Language Processing (NLP). Since our goal is to recognize the entities listed in Table IV from a given text, we incorporate Named Entity Recognition (NER) into our model. The first step in Named Entity Recognition (NER) is tokenization, where the input text is segmented into individual words or subwords. After tokenization, embeddings for these tokens are computed using spaCy's Tok2Vec pipeline, which captures contextual relationships between words. These contextual embeddings, are then processed by a neural network-based

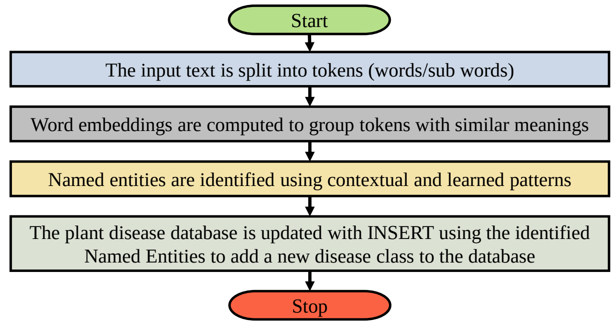


Figure 3: Learning workflow of the proposed method.

sequence labeling model. In this process, spaCy's transition-based approach [17] leverages a CNN feature extractor to analyze the embeddings in the surrounding context and assign each token to a predefined entity category, enabling the identification of relevant entities within the text. Thus, when a user provides a textual description of a disease, the NER model can accurately identify key entities such as the plant name, disease name, disease's semantic features, and associated colors.

2) Knowledge Updating: A database is a system that enables structured storage, updating, and retrieval of information. Among various types of databases, the Relational Database Management System (RDBMS) [18] is the most widely used. In an RDBMS, data is organized in the form of tables, where each entry is referred to as a record. However, RDBMS does not support storing lists directly, meaning multiple colors related to a disease cannot be stored in a single column efficiently. Additionally, spelling mistakes or extra spaces can cause inconsistencies in data retrieval. To address this, colors and semantic features are stored in separate tables, and these child tables are linked to the parent plant diseases table using foreign keys. Instead of storing color names directly, each disease record references the corresponding color by its unique ID, ensuring data consistency, normalization, and efficient querying as illustrated in the Entity-Relationship Diagram (ERD) in Fig. 4.

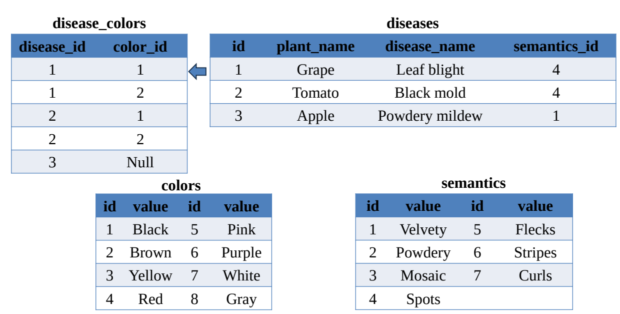


Figure 4: Entity-Relationship Diagram of the database.

The named entities identified by the NER model, as described in Section V-A1, are used to construct an INSERT query in Structured Query Language (SQL) as detailed in Algorithm 1, facilitating the storage and persistence of the disease's semantic features in the database. This approach enables incremental knowledge updates with minimal effort.

Algorithm 1: Insert Disease Record

```
Require: plant name plant_name, disease name disease_name,
    semantic label semantics, color list colors_list
Ensure: A new disease entry with linked colors is added
Execute SQL: SELECT id FROM semantics WHERE value =
  semantics
if no row returned then
    Raise error: Semantic not found
end if
Let semantics_id ← retrieved id
Execute SQL: INSERT INTO diseases (plant_name, disease_name,
  semantics_id)
    VALUES (plant_name, disease_name, semantics_id)
Let disease_id ← last inserted row ID
for each color in colors_list do
    Execute SQL: SELECT id FROM colors WHERE value = color
    if no row returned then
        Raise error: Color not found
    end if
    Let color_id ← retrieved id
    Execute SQL: INSERT OR IGNORE INTO disease_colors
    (disease_id, color_id)
      VALUES (disease_id, color_id)
end for
Commit database changes
```

Algorithm 2: Get Best Match for Disease

```
Require: plant, semantics, colors_list
Ensure: Best matching disease name or "No Disease Found"
Retrieve records with input plant and semantics
    SELECT d.id, d.plant_name, d.disease_name,
    s.value AS semantics_value, GROUP_CONCAT(c.value, ',
    ') AS colors
    FROM diseases d
    JOIN semantics s ON d.semantics_id = s.id
    LEFT JOIN disease_colors dc ON d.id = dc.disease_id
    LEFT JOIN colors c ON dc.color_id = c.id
    WHERE d.plant_name = plant AND s.value = semantics
    GROUP BY d.id
Assign weights to colors based on their order:
for i = 0 to |colors_list| − 1 do
    Let color ← colors_list[i]
    weight_dict[color] ← max(0.1, 1.0 − 0.2 × i)
end for
Initialize best_score ← −1, best_record ← None
for each record in SQL results do
    Extract record_colors from record[4]
    Compute union ← colors_list ∪ record_colors
    Compute weighted_intersection:

        weighted_intersection ← ∑ weight_dict[color]
            for color ∈ record_colors ∩ colors_list

    Compute:
        score ← weighted_intersection / |union|

    if score > best_score then
        Update best_score and best_record
    end if
end for
if best_record is None then
    return "No Disease Found"
else
    return best_record[2] {Disease name}
end if
```

B. Inference

To perform inference on an image in a manner similar to human perception, the method first identifies key semantic features and color attributes, these semantics serve as the basis for classification. Fig. 5 and the following subsections provide a detailed explanation of the processes involved in semantic identification and classification.

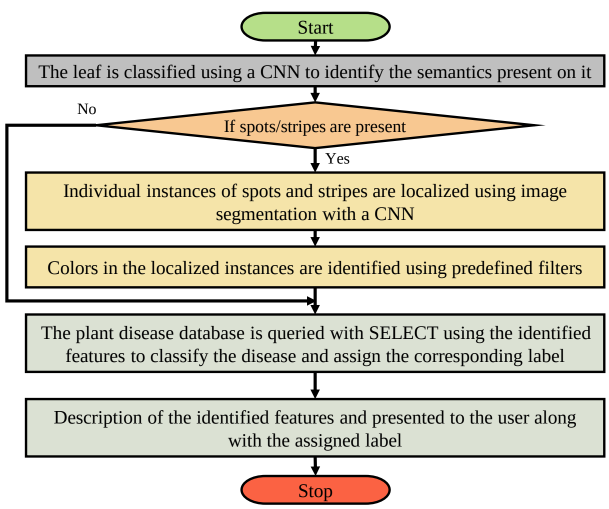


Figure 5: Inference workflow of the proposed method.

1) Semantics Identification: As discussed earlier, spots and stripes require further analysis beyond their initial identification to ensure accurate classification. Therefore, for semantics that do not require further analysis, we propose using a classification model, while for those requiring more detailed examination, we suggest segmentation in addition to classification.

The image of the diseased plant leaf is fed into a shallow CNN classification model [19] summarized in Table V, which consists of a convolutional feature extractor and a fully connected classifier layer. This model predicts the features at global level and outputs the label in text form. If the feature identified needs segmentation, the image is fed into a fully convolutional segmentation model presented in Table VII, which generates masks to localize the relevant semantics in the image. Predefined HSV color space filers presented in Table VI corresponding to the colors listed in Table III are applied to the identified semantic regions in the image. The top 3 identified colors are listed in the descending order of their occurrences in the region so that majority color is at the top of the list.

2) Classification: The semantic features and colors identified in Section V-B1, along with the plant name provided by the user, are used to retrieve the corresponding disease name from the database as described in the Algorithm 2 using SELECT query and Weighted Jaccard similarity. Thus, the proposed method leverages a database as a knowledge base, enabling dynamic knowledge updates and classification based on semantics through simple queries in a knowledge-driven approach.

VI. Experimental Results and Discussion

A. Results

The proposed solution was implemented and validated on 11,000 images covering 21 different diseases across 11 plant species, sourced from the PlantVillage [20] and PlantDoc [21] datasets. A NER model, depicted in Fig. 6, was developed using SpaCy's tok2Vec and NER to identify named entities, as presented in Table IV. The descriptions of 120 plant diseases

Table V: Summary of classification model.

Layer (Act: relu, Kernel:3x3)	**Output Shape**	**Parameters**
InputLayer	(None, 256, 256, 3)	0
Conv2D	(None, 256, 256, 32)	896
MaxPooling2D	(None, 128, 128, 32)	0
SeparableConv2D	(None, 128, 128, 64)	2400
MaxPooling2D	(None, 64, 64, 64)	0
SeparableConv2D	(None, 64, 64, 128)	8896
MaxPooling2D	(None, 32, 32, 128)	0
SeparableConv2D	(None, 32, 32, 128)	17664
MaxPooling2D	(None, 16, 16, 128)	0
SeparableConv2D	(None, 16, 16, 128)	17664
MaxPooling2D	(None, 8, 8, 128)	0
SeparableConv2D	(None, 8, 8, 128)	17664
MaxPooling2D	(None, 4, 4, 128)	0
SeparableConv2D	(None, 4, 4, 128)	17664
MaxPooling2D	(None, 2, 2, 128)	0
GlobalAverage Pooling2D	(None, 128)	0
Dense	(None, 32)	4128
Dense (softmax)	(None, 7)	231
Total Parameters		**87,207**
Trainable Parameters		87,207
Non-trainable Parameters		0
Optimizer: `adam`		
Loss Function: `sparse_categorical_crossentropy`		

Table VI: HSV filters for color detection.

Color	**cv2.COLOR_RGB2HSV**	
	Min	**Max**
Red	[0, 80, 80]	[5, 255, 255]
Brown	[6, 80, 80]	[20, 255, 255]
Yellow	[25, 80, 80]	[30, 255, 255]
Pink	[130, 80, 80]	[160, 255, 255]
Gray	[40, 30, 30]	[55, 120, 120]
Black	[0, 0, 0]	[179, 255, 60]
White	[0, 0, 210]	[179, 30, 255]

Table VII: Summary of segmentation model.

Layer (Act: relu, Kernel:3x3)	**Output Shape**	**Params #**	**Connected To**
InputLayer	(None, 256, 256, 3)	0	-
Conv2D	(None, 256, 256, 64)	1,792	input_layer
Conv2D_1	(None, 256, 256, 64)	36,928	conv2d
MaxPooling2D	(None, 128, 128, 64)	0	conv2d_1
SeparableConv2D	(None, 128, 128, 128)	8,896	max_pooling2d
SeparableConv2D_1	(None, 128, 128, 128)	17,664	separable_conv2d
MaxPooling2D_1	(None, 64, 64, 128)	0	separable_conv2d_1
SeparableConv2D_2	(None, 64, 64, 128)	17,664	max_pooling2d_1
SeparableConv2D_3	(None, 64, 64, 128)	17,664	separable_conv2d_2
MaxPooling2D_2	(None, 32, 32, 128)	0	separable_conv2d_3
SeparableConv2D_4	(None, 32, 32, 256)	34,176	max_pooling2d_2
SeparableConv2D_5	(None, 32, 32, 128)	35,200	separable_conv2d_4
MaxPooling2D_3	(None, 16, 16, 128)	0	separable_conv2d_5
SeparableConv2D_6	(None, 16, 16, 256)	34,176	max_pooling2d_3
SeparableConv2D_7	(None, 16, 16, 128)	35,200	separable_conv2d_6
MaxPooling2D_4	(None, 8, 8, 128)	0	separable_conv2d_7
SeparableConv2D_8	(None, 8, 8, 512)	67,200	max_pooling2d_4
SeparableConv2D_9	(None, 8, 8, 128)	70,272	separable_conv2d_8
MaxPooling2D_5	(None, 4, 4, 128)	0	separable_conv2d_9
SeparableConv2D_10	(None, 4, 4, 512)	67,200	max_pooling2d_5
SeparableConv2D_11	(None, 4, 4, 128)	70,272	separable_conv2d_10
Conv2DTranspose_1	(None, 8, 8, 128)	65,664	separable_conv2d_11
Concatenate_1	(None, 8, 8, 256)	0	conv2d_transpose_1, separable_conv2d_9
SeparableConv2D_16	(None, 8, 8, 256)	68,096	concatenate_1
SeparableConv2D_17	(None, 8, 8, 256)	68,096	separable_conv2d_16
Conv2DTranspose_2	(None, 16, 16, 128)	131,200	separable_conv2d_17
Concatenate_2	(None, 16, 16, 256)	0	conv2d_transpose_2, separable_conv2d_7
SeparableConv2D_18	(None, 16, 16, 256)	68,096	concatenate_2
SeparableConv2D_19	(None, 16, 16, 256)	68,096	separable_conv2d_18
Conv2DTranspose_3	(None, 32, 32, 128)	131,200	separable_conv2d_19
Concatenate_3	(None, 32, 32, 256)	0	conv2d_transpose_3, separable_conv2d_5
SeparableConv2D_20	(None, 32, 32, 128)	35,200	concatenate_3
SeparableConv2D_21	(None, 32, 32, 128)	17,664	separable_conv2d_20
Conv2DTranspose_4	(None, 64, 64, 128)	65,664	separable_conv2d_21
Concatenate_4	(None, 64, 64, 256)	0	conv2d_transpose_4, separable_conv2d_3
SeparableConv2D_22	(None, 64, 64, 128)	35,200	concatenate_4
SeparableConv2D_23	(None, 64, 64, 128)	17,664	separable_conv2d_22
Conv2DTranspose_5	(None, 128, 128, 128)	65,664	separable_conv2d_23
Concatenate_5	(None, 128, 128, 256)	0	conv2d_transpose_5, separable_conv2d_1
SeparableConv2D_24	(None, 128, 128, 128)	35,200	concatenate_5
SeparableConv2D_25	(None, 128, 128, 128)	17,664	separable_conv2d_24
Conv2DTranspose_6	(None, 256, 256, 64)	32,832	separable_conv2d_25
Concatenate_6	(None, 256, 256, 128)	0	conv2d_transpose_6, conv2d_1
SeparableConv2D_26	(None, 256, 256, 64)	9,408	concatenate_6
SeparableConv2D_27	(None, 256, 256, 64)	4,736	separable_conv2d_26
Conv2D_2 (1,1) (softmax)	(None, 256, 256, 2)	130	separable_conv2d_27
Total Parameters		**1,451,778**	
Trainable Params		1,451,778	
Non-trainable Params		0	
Optimizer: `adam`			
Loss Function: `sparse_categorical_crossentropy`			

were annotated for named entities to create a base dataset. The disease descriptions were paraphrased, rephrased, and jumbled, then corrected using ChatGPT APIs to generate multiple variations of the same disease descriptions, resulting in a dataset of 36,000 records. The developed NER model achieved F1 score of 98%. The F1 scores for each entity are shown in Fig. 7.

The proposed CNN classification model was trained on 7,000 images to identify the semantic shapes described in Table III, achieving the accuracies presented in Fig. 8a. Meanwhile, the CNN segmentation model was trained on 3,000 images from a different dataset [22] having images of leaves from farm instead of the images taken under controlled conditions, with results shown in Fig. 8b to test for the generalizability of the model on unseen data. Color filters were defined in HSV space for the color engineered as per Table VI.

All these models operate behind the UI, with the *Home* page (shown in Fig. 9) allowing users to input text to add a new disease class and classify an image of a diseased leaf. The *Add* screen (shown in Fig. 10) includes a guide to help the users understand the semantics and colors the way the developed model interprets, it accepts textual description of the disease as input from users. Upon submission, the text is processed by the NER model, which identifies the entities and sends them to the *Confirm* screen for user confirmation, as shown in Fig.

```
nlp.pipeline
```

```
[('tok2vec', <spacy.pipeline.tok2vec.Tok2Vec at 0x7806648e40b0>),
 ('ner', <spacy.pipeline.ner.EntityRecognizer at 0x7806649b80b0>)]
```

```
nlp.pipe_labels['ner']
```

```
['COLOR', 'DISEASE', 'PLANT', 'SEMANTIC']
```

Figure 6: Pipelines and Named Entities of the spaCy model.

```
Per-Entity F1 Scores:
PLANT: 0.9977
DISEASE: 0.9935
COLOR: 0.9834
SEMANTIC: 0.9927

Overall Model F1 Score:
ENTS_F: 0.9898
```

Figure 7: Metrics of the spaCy NER model.

Class Integer	Class Label	F1 Score
0	Curls	0.936170
1	Flecks	0.996785
2	Mosaic	0.945338
3	Powdery	0.969697
4	Spots	0.854599
5	Stripes	0.990596
6	Velvety	0.889632
Total	Accuracy	0.930427

(a) Classification model.

```
94/94 ━━━━━━━━━━━━━━━━━━━━ 2s
Validation Loss: 0.2196
Validation Accuracy: 0.9093
94/94 ━━━━━━━━━━━━━━━━━━━━ 3s
Precision: 0.9074
Recall: 0.9093
F1 Score: 0.9082
Mean IoU: 0.7336
```

(b) Segmentation model.

Figure 8: Performance metrics of the CNN models.

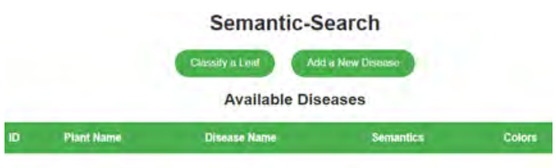


Figure 9: *Home* page of the UI developed for Semantic-Search.

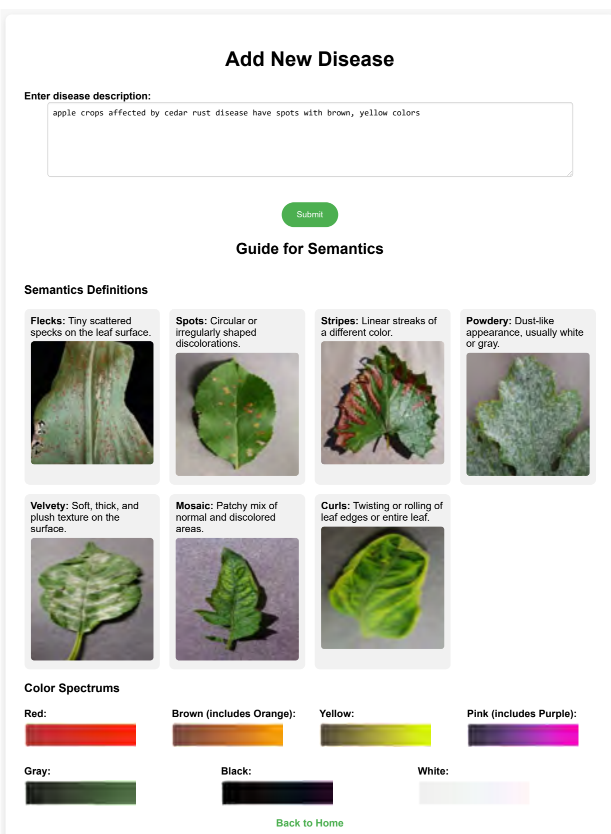


Figure 10: *Add* page with guide and user's text input.

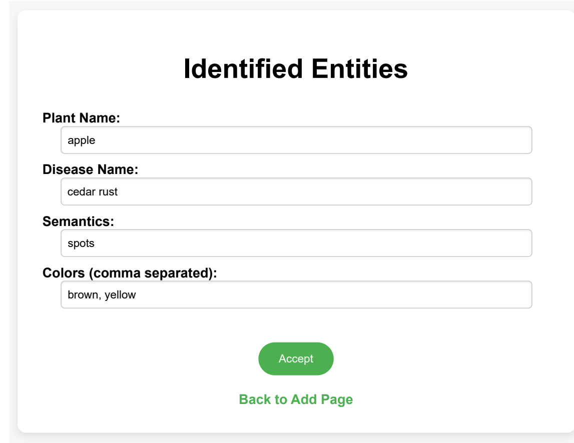


Figure 11: *Confirm* page with NER results for user's text input.

```
INFO:werkzeug:127.0.0.1 - - [24/May/2025 21:41:35] "GET /static/images/yellow.png HTTP/1.1" 304 -
INFO:werkzeug:127.0.0.1 - - [24/May/2025 21:42:14] "POST /entities HTTP/1.1" 200 -
Raw Text: apple crops affected by cedar rust disease have spots with brown, yellow colors
Identified entity, Label: PLANT, Entity: apple
Identified entity, Label: DISEASE, Entity: cedar rust
Identified entity, Label: SEMANTIC, Entity: spots
Identified entity, Label: COLOR, Entity: brown
Identified entity, Label: COLOR, Entity: yellow
INFO:werkzeug:127.0.0.1 - - [24/May/2025 21:45:11] "POST /confirm HTTP/1.1" 302 -
INFO:werkzeug:127.0.0.1 - - [24/May/2025 21:45:11] "GET / HTTP/1.1" 200 -
   Disease 'cedar rust' added successfully.
INFO:werkzeug:127.0.0.1 - - [24/May/2025 21:45:11] "GET /static/images/flecks.JPG HTTP/1.1" 304 -
```

Figure 12: Server-side logs for adding a disease by a text input.

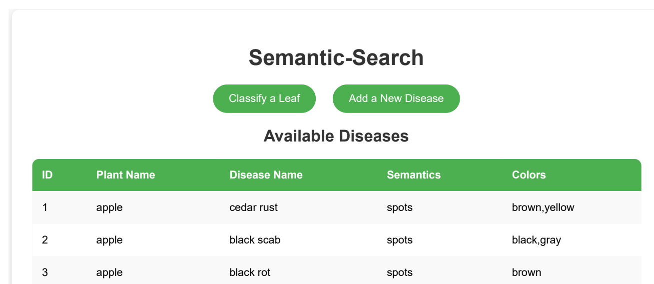


Figure 13: *Home* screen showing list of added diseases.

11. Upon user acceptance, a record is created in the database with the identified entities, as shown in Fig. 12, updating the system's knowledge. The system then returns to the *Home* page, displaying the list of added diseases, as represented in Fig. 13. Since black spots on leaves may not be pure black but rather a fusion of leaf colors, they can sometimes be inferred as gray. Therefore, when black or gray is mentioned in the text, both colors are added to the disease record.

Choosing to classify a leaf directs the user to the *Classify* page, where they select the crop type from a list of available plants retrieved from the database and upload an image of diseased leaf, as shown in Fig. 14. Upon clicking classify, the image is first analyzed to identify semantics and, if necessary, undergoes segmentation to localize disease-affected areas and detect colors. The extracted semantics, top 3 colors present in the semantics are then used to query the database and classify, as illustrated in Fig. 15, enabling classification based on the stored knowledge. The developed model achieved an accuracy of 90% in classifying diseased leaves. Confusion matrix of the model is presented in Fig. 16 and detailed classification metrics are shown in Fig. 17.

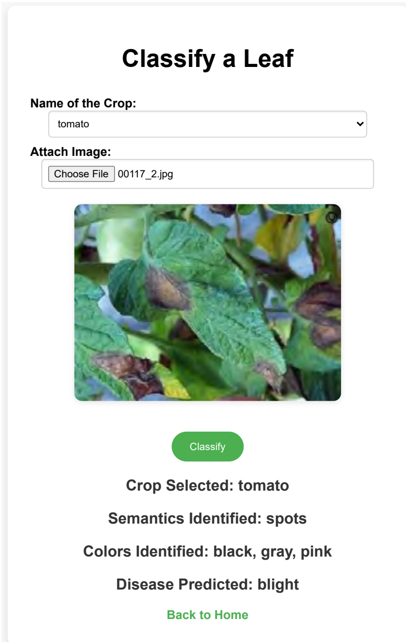


Figure 14: *Classify* page with classification results.

```
Starting classification for crop: tomato
Using image: temp_image.jpg
Running classification model...
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
Prediction: [1.6414066e-13 4.0870262e-05 7.4926800e-14 9.0081759e-10 9.9957711e-01
 3.8203711e-04 8.8334878e-10]
Running segmentation for color extraction...
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 368ms/step
INFO:werkzeug:127.0.0.1 - - [24/May/2025 23:50:35] "POST /api/classify HTTP/1.1" 200 -
Color counts: {'red': 3, 'brown': 83, 'yellow': 56, 'pink': 146, 'gray': 502, 'black': 1562, 'white': 1}
Extracted colors (sorted by count): black, gray, pink
Sorted prediction indices: [4 5 1 3 6 0 2]
Plant name:  tomato
Semantics:  spots
colors list: ['black', 'gray', 'pink']
weighted dictionary : {'black': 1.0, 'gray': 0.8, 'pink': 0.6}
normalized colors : ['black', 'gray', 'pink']
Matching disease record found:
  Plant Name: tomato
  Semantics: spots
  Disease Predicted: blight
  Record Colors: black, brown
```

Figure 15: Server-side logs for the classification of an image.

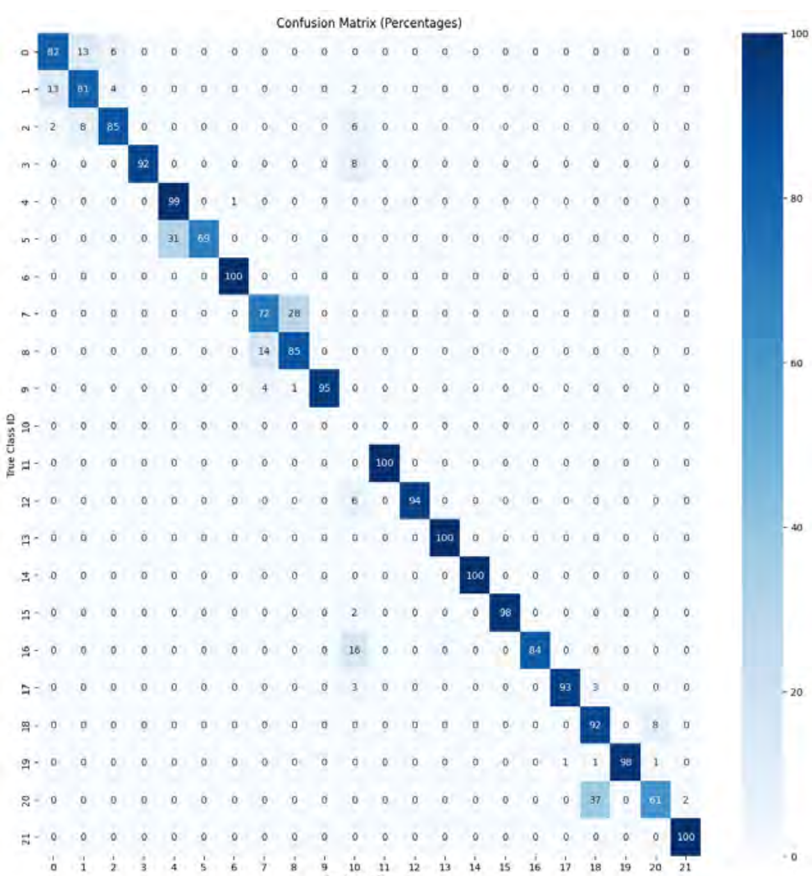


Figure 16: Confusion matrix of the developed Semantic-Search.

Class	Id	Accuracy	F1 Score	Precision	Recall
Apple black rot	0	81.62	85.96	90.80	81.62
Apple cedar rust	1	81.04	72.31	65.27	81.04
Apple scab	2	84.81	88.35	92.19	84.81
Cherry mildew	3	91.67	95.65	100	91.67
Corn blight	4	99.40	86.43	76.46	99.40
Corn leaf spot	5	69.20	81.79	100	69.20
Corn rust	6	100	99.67	99.34	100
Grape black rot	7	72.20	74.82	77.63	72.20
Grape blight	8	85.40	79.59	74.52	85.40
Grape esca	9	95.14	97.51	100	95.14
Disease not found	10	NA	NA	NA	NA
Orange citrus	11	100	100	100	100
Peach bacterial spot	12	93.80	96.80	100	93.80
Pepper bacterial spot	13	100	100	100	100
Potato blight	14	99.60	99.79	100	99.60
Squash mildew	15	98.00	98.99	100	98.00
Strawberry scorch	16	84.23	91.43	100	84.22
Tomato bacterial spot	17	93.40	96.19	91.51	99.40
Tomato blight	18	91.93	78.65	68.73	91.93
Tomato curl virus	19	97.80	96.87	99.59	97.20
Tomato mold	20	61.04	72.04	87.86	61.04
Tomato mosaic virus	21	100	99.01	98.04	100
Average		90	91	93	90

Figure 17: Per-class classification metrics.

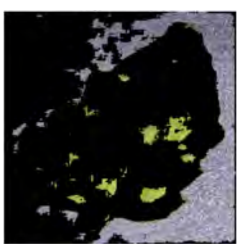
Table VIII: Comparison of the proposed model with existing approaches.

Approach	Semantic Understanding	Need for Retraining	Scalability	Retraining Cost	Accuracy
CNN-based models	Limited (pixel-level)	Required for new classes	Low	High	97%
Histogram-based (SVM) models	No	Required	Low	Moderate	80%
Bag of Visual Words	No	Required	Low	Moderate	82%
Knowledge Graph-based methods	Yes	No, but complex graph construction	Moderate	Low	83%
Proposed model	Yes (Object detection + Color filters + NLP)	No (Database update instead)	High	Very Low	90%

B. Limitations

Since the proposed method classifies diseases based on color and semantic features, precise localization of the diseased region is critical. Inclusion of background or healthy parts of the leaf around the diseased area as depicted in Fig. 18a introduces additional colors, leading to incorrect labeling.Such discrepancies arising from segmentation errors led to misclassifications in cases such as tomato mold, thereby reducing the model's accuracy.

Furthermore, as the the segmentation and disease classification pipelines are dependent on the label corresponding to the detected semantics, which is the label with highest probability, leaves exhibiting multiple diseases are labeled according to the dominant disease as in case of image of corn leaf in Fig. 18b. Corn leaf exhibits a dominant stripe pattern in addition to spots, which can lead to misclassification as blight disease. These limitations collectively contributed to a 10% reduction in the overall accuracy of the proposed model.



(a) Improper segmentation.

(b) Multiple semantics.

Figure 18: Scenarios resulting in misclassifications.

C. Comparative Analysis

The proposed framework achieved a classification time of 1.8 seconds on an Intel Xeon CPU and 132 milliseconds on an NVIDIA L4 GPU. In comparison, the earlier version presented in [23] required 8.7 seconds on the Intel Xeon CPU and 2.1 seconds on the NVIDIA L4 GPU. The prior approach employed a complex pipeline that involved contour fitting around leaf edges for background removal, separate models for texture identification, segmentation of the image into five parts for classification, followed by localization and color filtering prior to SQL-based classification. In contrast, the current method utilizes a single model to detect all semantic features and employs a deeper segmentation model for precise localization.

By embedding the necessary cognitive abilities into the model, new diseases can be incorporated through simple textual updates, unless the disease exhibits rare or unseen colors. This approach also enables rapid development of localized classification models to address feature variability arising from environmental conditions, rather than relying on a single global model that may struggle to accommodate diverse feature representations of the same disease.

To provide a fair comparison, we broadly categorized prior works (Table II) into four representative methodological families: CNN-based, histogram-based, bag-of-visual-words, and knowledge-graph approaches. Since the original codes for these methods were not publicly available, we re-implemented representative baseline models following the principles described in the respective literature. Each baseline was trained and evaluated on the same dataset and under identical experimental settings as our proposed method. Table VIII summarizes the results, highlighting the novelty and effectiveness of the proposed approach in terms of scalability and retraining.

VII. Conclusion

This article presents a knowledge-driven classification model that leverages a database as a knowledge base to semantically interpret and classify plant diseases, moving beyond the reliance on low-level features typically used in convolutional neural networks (CNNs). As the framework employs multiple models operating in sequence, the accuracy and precision of each component directly impact the final predicted label and the overall model performance. This model demonstrates its practicality in disease management within ACPS, where classification aids in identifying the appropriate diseases and pesticides for spraying systems [24], [25]. While the proposed framework integrates NLP and database capabilities, it can also be incorporated into existing automation systems to autonomously learn new diseases from large language models (LLMs) and leverage available database resources. Although the model is scalable and efficient, it requires manual feature engineering and database creation, and the user must select the crop type, making it not fully autonomous. Models that can map image features and text embeddings to the same feature space, while incorporating color information, could enable learning of new diseases and scalability without the need for manual feature engineering, making them a promising direction for future research.

Acknowledgment

This journal paper is based on a preliminary conference presentation [23].

References

[1] A. Mitra, S. L. Vangipuram, A. K. Bapatla, V. K. Bathalapalli, S. P. Mohanty, E. Kougianos, and C. Ray, "Smart agriculture: A comprehensive overview," *SN Computer Science*, vol. 5, no. 8, p. 969, 2024.

[2] A. Upadhyay, N. S. Chandel, K. P. Singh, S. K. Chakraborty, B. M. Nandede, M. Kumar, A. Subeesh, K. Upendar, A. Salem, and A. Elbeltagi, "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture," *Artificial Intelligence Review*, vol. 58, no. 3, pp. 1–64, 2025.

[3] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, pp. 1–74, 2021.

[4] A. S. Gillis, "What are knowledge-based systems (KBSes)?" Mar. 2023, last accessed January 25 2025. [Online]. Available: https://www.techtarget.com/searchcio/definition/knowledge-based-systems-KBS

[5] "Crops," last accessed February 5 2025. [Online]. Available: https://plantvillage.psu.edu/plants

[6] I. Biederman, "Recognition-by-components: a theory of human image understanding." *Psychological review*, vol. 94, no. 2, p. 115, 1987.

[7] K. Wei, B. Chen, J. Zhang, S. Fan, K. Wu, G. Liu, and D. Chen, "Explainable Deep Learning Study for Leaf Disease Classification," *Agronomy*, vol. 12, no. 5, 2022.

[8] S. B. Park, J. W. Lee, and S. K. Kim, "Content-based image classification using a neural network," *Pattern Recognition Letters*, vol. 25, no. 3, pp. 287–300, 2004.

[9] S. Agrawal, N. K. Verma, P. Tamrakar, and P. Sircar, "Content based color image classification using SVM," in *Proc. Eighth International Conference on Information Technology: New Generations*, 2011, pp. 1090–1094.

[10] Q. Kai, "A painting image retrieval approach based on visual features and semantic classification," in *Proc. International Conference on Smart Grid and Electrical Automation (ICSGEA)*, 2019, pp. 195–198.

[11] A. Vailaya, M. Figueiredo, A. Jain, and H. J. Zhang, "Image classification for content-based indexing," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 117–130, 2001.

[12] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proc. International Workshop on Workshop on Multimedia Information Retrieval*, 2007, p. 197–206.

[13] Y. Su and F. Jurie, "Improving image classification using semantic attributes," *International journal of computer vision*, vol. 100, pp. 59–77, 2012.

[14] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," 2017. [Online]. Available: https://arxiv.org/abs/1612.04844

[15] C. Menglong, J. Detao, Z. Ting, Z. Dehai, X. Cheng, C. Zhibo, and X. Xiaoqiang, "Image classification based on image knowledge graph and semantics," in *Proc. IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2019, pp. 81–86.

[16] W. Jearanaiwongkul, C. Anutariya, and F. Andres, "An ontology-based approach to plant disease identification system," in *Proc. 10th international conference on advances in information technology*, 2018, pp. 1–8.

[17] M. Honnibal and M. Johnson, "An improved non-monotonic transition system for dependency parsing," in *Proc. Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. Association for Computational Linguistics (ACL), 2015, pp. 1373–1378.

[18] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*. McGraw-Hill Education, 2011.

[19] Y. Li, J. Nie, and X. Chao, "Do we really need deep cnn for plant diseases identification?" *Computers and Electronics in Agriculture*, vol. 178, p. 105803, 2020.

[20] A. Ali, "Plantvillage dataset," 2019, last accessed April 25 2024. [Online]. Available: https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset

[21] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "Plantdoc: A dataset for visual plant disease detection," in *Proc. 7th ACM IKDD CoDS and 25th COMAD*, 2020, p. 249–253.

[22] S. R. Rath, "Leaf disease segmentation," 2023, last accessed January 25 2025. [Online]. Available: https://www.kaggle.com/datasets/sovitrath/leaf-disease-segmentation-with-trainvalid-split

[23] K. K. Kethineni, S. P. Mohanty, and E. Kougianos, "Semantic-search: A knowledge-driven classification method for plant diseases," in *Proc. IEEE 2nd Conference on AgriFood Electronics (CAFE)*, 2024, pp. 95–99.

[24] K. K. Kethineni, S. P. Mohanty, E. Kougianos, S. Bhowmick, and L. Rachakonda, "Spraycraft: Graph-based route optimization for variable rate precision spraying," 2024. [Online]. Available: https://arxiv.org/abs/2412.12176

[25] A. Mitra, S. P. Mohanty, and E. Kougianos, "aGROdet 2.0: An Automated Real Time Approach for Multiclass Plant Disease Detection," *SN Computer Science*, vol. 4, no. 5, p. 657, 2023.

Kiran K. Kethineni (Student Member, IEEE) received his B.Tech. degree in Electronics and Communication Engineering from Vignan's University, Guntur, India, in 2016, and his M.S. degree in Computer Engineering from the University of North Texas, Denton, TX, USA, in 2018. He is currently pursuing a Ph.D. in Computer Science and Engineering at the University of North Texas, where he is a member of the Smart Electronics Systems Laboratory research group. His research interests include smart agriculture, as well as exploring the synergy between human and artificial intelligence for applications in the Internet of Things (IoT).

Saraju P. Mohanty (Senior Member, IEEE) received the bachelor's degree (Honors) in electrical engineering from the Orissa University of Agriculture and Technology, Bhubaneswar, in 1995, the master's degree in Systems Science and Automation from the Indian Institute of Science, Bengaluru, in 1999, and the Ph.D. degree in Computer Science and Engineering from the University of South Florida, Tampa, in 2003. He is a Professor with the University of North Texas. His research is in "Smart Electronic Systems" which has been funded by National Science Foundations (NSF), Semiconductor Research Corporation (SRC), U.S. Air Force, NIDILRR, IUSSTF, and Mission Innovation. He has authored 550 research articles, 5 books, and invented 10 granted/pending patents. His Google Scholar h-index is 62 and i10-index is 299 with 17,000 citations. He is a recipient of 21 best paper awards, Fulbright Specialist Award in 2021, IEEE Consumer Electronics Society Outstanding Service Award in 2020, the IEEE-CS-TCVLSI Distinguished Leadership Award in 2018, and the PROSE Award for Best Textbook in Physical Sciences and Mathematics category in 2016. He has delivered 31 keynotes and served on 15 panels at various International Conferences. He has been the Editor-in-Chief of the IEEE Consumer Electronics Magazine during 2016-2021 and currently serves on the editorial board of 8 journals/transactions. He has supervised 3 post-doctoral researchers, 19 Ph.D. dissertations, 29 M.S. theses, and 41 undergraduate projects.

Elias Kougianos received a BSEE from the University of Patras, Greece in 1985 and an MSEE in 1987, an MS in Physics in 1988 and a Ph.D. in EE in 1997, all from Louisiana State University. From 1988 through 1998 he was with Texas Instruments, Inc., in Houston and Dallas, TX. In 1998 he joined Avant! Corp. (now Synopsys) in Phoenix, AZ as a Senior Applications engineer and in 2000 he joined Cadence Design Systems, Inc., in Dallas, TX as a Senior Architect in Analog/Mixed-Signal Custom IC design. He has been at UNT since 2004. He is a Professor in the Department of Electrical Engineering, at the University of North Texas (UNT), Denton, TX. His research interests are in the area of Analog/Mixed-Signal/RF IC design and simulation and in the development of VLSI architectures for multimedia applications. He is an author of over 200 peer-reviewed journal and conference publications.