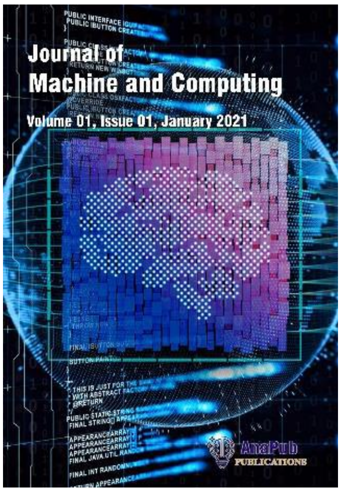
CauliCaps: Capsule Network for Pioneering Disease Detection in Cauliflower Cultivation with Mobile Application Development



Meenalochini M and Amudha P

DOI: 10.53759/7669/jmc202505170

Reference: JMC202505170

Journal: Journal of Machine and Computing.

Received 04 May 2025

Revised from 11 July 2025

Accepted 22 July 2025

Please cite this article as: Meenalochini M and Amudha P, "CauliCaps: Capsule Network for Pioneering Disease Detection in Cauliflower Cultivation with Mobile Application Development", Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505170.

CauliCaps: Capsule Network for Pioneering Disease detection in Cauliflower Cultivation with Mobile Application Development

[1]**M. Meenalochini**
Research Scholar, Department of Computer Science and Engineering, School of Engineering,
Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore,
Tamil Nadu, India
meenalochinicse@gmail.com

[2]P.Amudha
Professor, Department of Computer Science and Engineering, School of Engineering,
Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore,
Tamil Nadu, India
*amudha_cse@avinuty.ac.in

Abstract

Cauliflower cultivation is challenged by various diseases that can severely impact crop health and yield. Traditional disease detection methods are often labour-intensive and prone to errors, highlighting the need for automated and efficient prediction systems. In this study, we propose the use of Capsule Neural Networks (CapsNet) for disease prediction in cauliflower cultivation named as CauliCaps. CapsNet introduces dynamic routing units to capture spatial relationships and hierarchical structures within images more effectively than traditional Convolutional Neural Networks (CNNs). We assemble a comprehensive dataset of labelled cauliflower leaf images, preprocess them for optimal input, and train the CapsNet model using an appropriate loss function and optimization algorithm. Metrics including accuracy, precision, recall, and F1-score are used to compare the model's performance to state-of-the-art techniques. Additionally, we discuss the development of a mobile application based on the trained CapsNet model for real-time disease diagnosis in cauliflower cultivation. This research aims to advance disease prediction in cauliflower cultivation, enabling proactive management strategies and ultimately contributing to improved crop health and sustainability.

Keywords: Disease detection, Capsule Neural Networks, Convolutional Neural Networks, Cauliflower disease, Mobile Application.

Introduction

Cauliflower (Brassica oleracea L. botrytis) is a cultivar of the Brassica wild cabbage species, renowned globally for its nutritional richness, prolific yield, and substantial economic contributions to the agricultural sector. Renowned for its versatility in culinary applications, cauliflower has gained widespread popularity as a staple vegetable due to its dense concentration of essential nutrients, including vitamins, minerals, and dietary fiber. Its adaptability to various climates and soil conditions, coupled with its relatively low maintenance requirements, has positioned cauliflower as a preferred choice among farmers and consumers alike. Furthermore, its significant economic impact stems from its widespread cultivation, with cauliflower serving as a valuable commodity in domestic and international markets. As a result, the cultivation of cauliflower plays a vital role in global food security and economic sustainability, making it a cornerstone of modern agricultural practices (Li et al. 2022).

Numerous diseases, including black rot, downy mildew, and bacterial spot rot, present difficulties for the cauliflower agricultural industry. These diseases have the potential to severely affect the growth and productivity of cauliflower crops. It is imperative for farmers to promptly identify these diseases and employ appropriate methods to mitigate their impact. There are a number of ways to manage cauliflower diseases, including chemical control with using pesticides, biological inhibition with aqueous extracts and beneficial microorganisms like Bacillus species, and physical methods like sterilization with hot water and the utilization of nanoparticles (Liu et al., 2022).

The use of pesticides in cauliflower cultivation, while essential for protecting crops from pests and diseases, presents potential risks to human health and the environment. Pesticides contain chemical compounds designed to target and eliminate pests and pathogens that can damage cauliflower plants. However, prolonged exposure to pesticides can pose health risks to farmworkers, consumers, and surrounding ecosystems. Additionally, diseases caused by bacteria or fungi in cauliflower, if left untreated, can lead to various health problems, including allergic reactions, when consumed (Pathak et al. 2022). The prevalence of plant diseases in cauliflower cultivation significantly affects both the quantity and quality of cauliflower produced, leading to substantial losses in annual harvests.

Traditional methods for detecting cauliflower diseases in agriculture are fraught with limitations that hinder effective disease management. Primarily relying on subjective human visual inspection, these methods are prone to errors and inconsistencies. Manual inspection processes are labor-intensive and time-consuming, often resulting in delayed disease detection, which allows for the rapid spread of infections within cauliflower crops. Moreover, the costs associated with training and retaining agricultural experts for disease identification pose significant financial burdens, especially for farmers in remote areas with limited access to specialized expertise (Kanna et al. 2023).

Additionally, traditional methods frequently overlook early or asymptomatic infections, as they rely on visible symptoms that may not manifest until the disease has progressed substantially. Furthermore, these methods are heavily reliant on specific environmental conditions, making them less adaptable to diverse agricultural settings. Moreover, the lack of proper documentation and data collection further complicates the disease monitoring process, hindering efforts to develop effective management strategies. Ultimately, the scalability of traditional cauliflower disease detection methods is limited, impeding their widespread adoption and effectiveness in agricultural settings.

Significant progress has been achieved in the creation of mobile applications for the detection of plant diseases in recent years. These apps use machine learning algorithms to identify patterns and symptoms linked to different diseases. The algorithms were trained on massive databases of plant photos. By simply capturing a photo of the affected plant using their smartphone, farmers can receive instant feedback on the presence of diseases, along with recommendations for treatment and management strategies. The development of mobile applications for plant disease detection presents numerous opportunities to revolutionize agriculture and empower farmers with valuable tools for crop protection. However, challenges such as ensuring accuracy, scalability, and accessibility must be addressed to maximize the impact of these applications in real-world farming scenarios (Rimon et al. 2021).

There has been a notable increase in demand across several precision agriculture areas for the combination of optical imaging, machine vision, and artificial intelligence techniques in vegetable disease diagnosis and management (Teet and Hashim, 2023). These cutting-edge technologies improve agricultural productivity and sustainability by providing novel approaches to the detection, tracking, and mitigation of plant diseases. Plant health factors including leaf color, texture, and morphology may be thoroughly analyzed by using optical imaging systems, which are outfitted with sophisticated sensors and high-resolution cameras.

Machine vision algorithms process the captured images to extract relevant features and patterns indicative of disease presence or progression. Leveraging artificial intelligence, including deep learning models like Convolutional Neural Networks (CNNs) and Capsule Neural Networks (CapsNets), facilitates automated disease diagnosis with remarkable accuracy and efficiency. By harnessing the power of these cutting-edge technologies, farmers and agricultural professionals can make informed decisions in real-time, optimizing resource allocation, minimizing crop losses, and ultimately contributing to the advancement of modern agriculture. This study aims to leverage CapsNet for disease detection in cauliflower crops and develop a mobile application for real-time diagnosis. By harnessing the power of deep learning and mobile technology, we seek to enhance disease prediction accuracy, enable early detection, and facilitate timely intervention strategies to safeguard cauliflower production.

Contributions of the work

- By proposing CauliCaps, the use of CapsNet for disease prediction in cauliflower cultivation, the study introduces a novel deep learning architecture that surpasses traditional Convolutional Neural Networks (CNNs) in capturing spatial relationships and hierarchical structures within images. This advancement in neural network architecture holds promise for more accurate and efficient disease detection in cauliflower plants.
- In addition to model development and evaluation, the research extends its impact by discussing the development of a mobile application based on the trained CapsNet model. This application enables real-time disease diagnosis in cauliflower cultivation, empowering farmers with actionable insights for proactive management strategies.
- Through experiments, the study thoroughly evaluates the performance of the CapsNet model against state-of-the-art methods. Metrics such as accuracy, precision, recall, and F1-score are meticulously analyzed, providing insights into the efficacy and reliability of the proposed approach for disease prediction in cauliflower cultivation.

Related works

For the classification and identification of agricultural diseases, researchers have put forth a number of creative frameworks and approaches. Using the Complete Concatenated Block (CCB) as its central functional unit, Arun and Umamaheswari (2023) presented the Complete Concatenated Deep Learning (CCDL) framework. The CCDL framework efficiently restricts the increment of parameters in the model by arranging a point-wise convolution layer before each convolution layer. With a reduced model size, the PCCDL-PSCT method obtained an astounding accuracy of 98.14%. Chug et al. (2023) introduced a novel framework that uses 40 different Hybrid Deep Learning (HDL) models to combine the best aspects of deep learning with machine learning. Using the IARI-TomEBD dataset, these models showed high accuracy values ranging from 87.55% to 100%. Using a Fully Convolutional Network (FCN) algorithm for target crop image segmentation, Huang et al. (2023) suggested an FC-SNDPN-based automated method for crop leaf disease diagnosis. 97.59% identification accuracy was attained by the SNDPN method, which combines SNDPN for identifying diseases with FCN for foreground segmentation. In order to prevent rice crop disease, Haridasan et al. (2023) used a computer vision-centric strategy that included image processing, machine learning (ML), and deep learning (DL) approaches. Their approach, which combines CNN and SVM classifier, identified affected areas and classified particular paddy diseases based just on visual features, yielding a validation accuracy of 91.45%. Together, these results show that crop disease research has advanced significantly. detection and classification, leveraging a diverse range of methodologies and frameworks.

A CNN with transfer learning was proposed by Li et al. (2022) to identify surface flaws in fresh-cut cauliflower. With a 99.27% accuracy rate and a 99.24% F1 score, the model—which is based on MobileNet—performed magnificently after being refined using a dataset of 4,790 photos of cauliflower. In order to classify eggplant disorders based on photos taken with a smartphone, Saad et al. (2022) used convolutional neural networks (CNNs). With an astounding 99.06% success rate in disease detection, DenseNet201 demonstrated the highest accuracy out of multiple transfer learning models. Using deep learning models such as Inception V3 and VGG16, Bakr et al. (2022) presented a web-based tool for plant disease diagnosis. At 99.44% for training and 9[illegible].70% for validation, the suggested model produced impressive results. In order to improve plant disease detection, Sutaji et al. (2022) presented the combination of the Xception and MobileNetV2 models. Across multiple datasets, the ensemble model outperformed individual model results, achieving the best accuracy rate of 99.10%. EfficientNet was proposed by Chowdhury et al. (2021) to classify tomato illnesses based on segmented leaf pictures. Outstanding results were obtained by the Modified U-net segmentation model, which had an accuracy rate of 98.66%. EfficientNet-B7 fared better in the binary and six-class classifications, with accuracy rates of 99.95% and 99.12%, respectively. Together, these research show how effective deep learning techniques are at identifying and categorizing plant diseases, providing viable options for farming administration and crop condition surveillance.

In order to accurately identify common vegetable illnesses as tomato powdery mildew, cucumber downy mildew, and leaf mold, Xue Zhao et al. (2022) created a novel Deep Transfer Learning with Squeeze-and-Excitation ResNet50 (DTL-SE-ResNet50) model. Their model demonstrated impressive performance in both simple and complicated agricultural scenarios by utilizing Convolutional Neural Network (CNN) architecture. In comparable experimental conditions, DTL-SE-ResNet50 achieved an outstanding 97.24% recognition accuracy, demonstrating its superiority over other well-known models such as EfficientNet, AlexNet, VGG19, and InceptionV3. Zhao et al. (2021) introduced an innovative deep learning-based sorting system tailored for soybean seeds, aimed at accurately identifying and sorting high-quality seeds. Their system meticulously collected For accurate seed categorization, a deep learning model was utilized, along with an alternative rotation technique to obtain surface feature data. The sorting system demonstrated its effectiveness and dependability with an astounding accuracy rate of 98.87% and a sorting speed of 222 seeds per minute. An improved Mask R-CNN algorithm was presented by Long et al. (2021) for the purpose of segmenting tomato fruits at different stages of ripeness in greenhouse settings. The mean average precision of the refined model was an astounding 95.45% for green ripe, half ripe, and fully ripe tomato stages. After the model was integrated with a picking robot and put through field testing, it showed an impressive 90% correct recognition rate, highlighting its usefulness. The YOLOv3 algorithm is combined with manual features in an innovative method of image processing by Lu et al. (2021) to create a state-of-the-art computer vision-based automated grading box for winter jujubes. Their method, which shown its ability to improve agricultural output and efficiency, was developed through extensive experimentation and achieved an amazing 97.28% accuracy in grading winter jujubes, with a quick detection time of 1.39 seconds.

Methodology

The research methodology outlined in this section encompasses a comprehensive workflow designed to facilitate the detection and management of cauliflower diseases effectively. Fig. 1 illustrates the sequential flow of the research, which begins with the acquisition and characterization of the dataset. The dataset consists of a diverse collection of labelled cauliflower leaf images, encompassing both healthy specimens and those afflicted with various diseases. Subsequently, preprocessing techniques are applied to the dataset to enhance features.

Feature extraction is a critical step in the methodology, where relevant features are identified and extracted from the cauliflower leaf images using Hu moments. This process of disease detection involves leveraging advanced techniques, such as Capsule Neural Networks

(CapsNet), to capture spatial relationships and hierarchical structures within the images effectively.

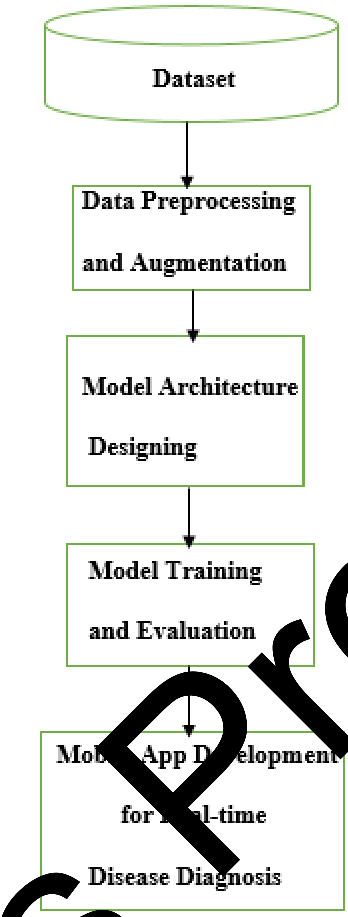

Dataset
Data Preprocessing and Augmentation
Model Architecture Designing
Model Training and Evaluation
Mobile App Development for Real-time Disease Diagnosis

Figure 1. Flow of the proposed work

Dataset

For this research, a comprehensive dataset comprising original and augmented cauliflower images was curated to facilitate the analysis of disease detection. The dataset encompasses three distinct classes representing different diseases afflicting cauliflower plants, along with images of disease-free cauliflower specimens for comparison. These images were meticulously compiled from the Manikganj region of Bangladesh, renowned for its vegetable production. Specifically, the dataset comprises a total of 656 original images and 7,360 augmented images, collectively representing a diverse range of disease states and healthy cauliflower specimens. Table 1 presents an overview of the total number of photos acquired for every class in the collection, illustrating the data distribution among various disease categories. This carefully selected information forms the basis for further phases of analysis and model building, allowing powerful machine learning algorithms to accurately characterize and forecast cauliflower diseases.

Table 1. Classes of the dataset

Classes	**Total number of images**
No disease	1770
Black Rot	1800
Downy Mildew	2060
Bacterial Spot Rot	1730

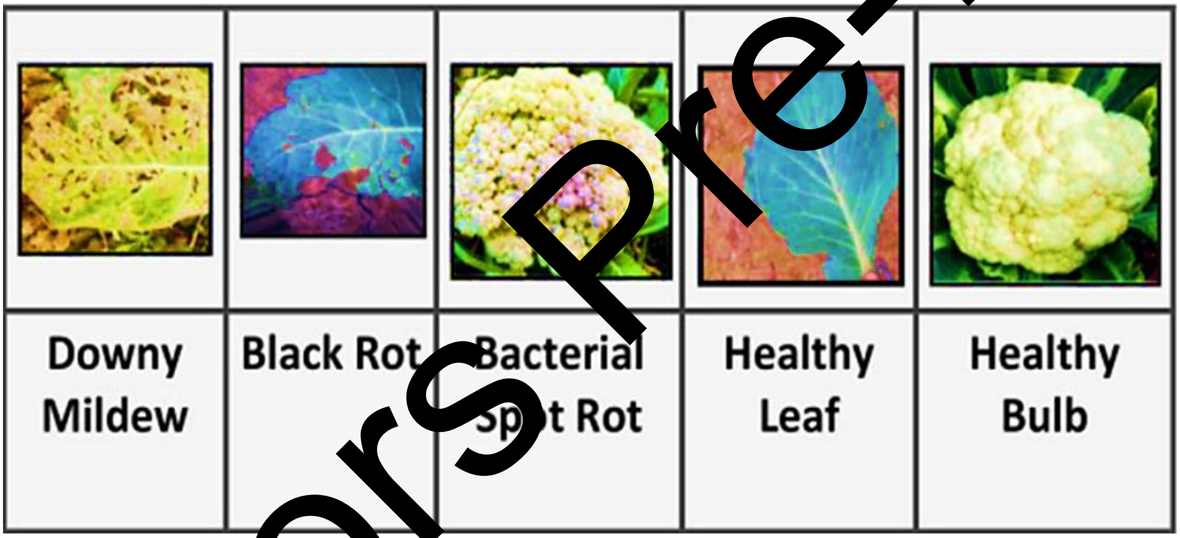


Figure 2. Images of the different classes of cauliflower

Preprocessing

The preprocessing of the dataset involves several steps to prepare the images for input into the Capsule Neural Network (CapsNet) model. To enhance model generalization and increase variability, these procedures usually involve standardizing pixel values, minimizing the photos to a compatible size, and increasing the dataset.

1. **Resizing Images**: Resize all images to a uniform size to ensure consistency in input dimensions. Let W_{new} and H_{new} represent the new width and height, respectively. For each image I_{original}, the resizing operation can be defined as:

$$I_{resized} = resize(I_{original}, W_{new}, H_{new}) \quad (1)$$

2. **Normalization**: Normalize pixel values to a common scale to facilitate model training. In order to prepare pixel values for neural network input, they must be scaled from their native range—for example, 0-255 for RGB images—to a new range, such as 0-1 or -1 to 1. The normalization equation for each pixel $p_{\text{normalized}}$ can be expressed as:

$$p_{normalized} = \frac{p_{original}}{255} \quad (if\ scaling\ to\ range\ 0-1) \quad (2)$$

$$p_{normalized} = \frac{p_{original} - mean}{std} \quad (if\ scaling\ to\ zero\ mean\ and\ unit\ variance) \quad (3)$$

3. **Data Augmentation**: To boost variability and strengthen model robustness, enhance the dataset by using transformations like flipping, rotation, and translation. Let I_{original} represent the original image and $I_{\text{augmented}}$ denote the augmented image obtained after applying transformations. The augmentation process can be represented as:

$$I_{augmented} = augment(I_{original}) \quad (4)$$

where augment(·) is a function that applies random transformations.

4. **Splitting Dataset**: To evaluate the performance of the model, divide the dataset into test, validation, and training sets. Typically, 10% of the dataset is used for testing, 10% for validation, and 80% for training.

These preprocessing steps ensure that the dataset is appropriately formatted and augmented to train the CapsNet model effectively for cauliflower disease detection.

Feature extraction

Feature extraction from the dataset involves extracting relevant information or features from the images that can be used to train machine learning models. In the context of image data, common techniques include extracting texture, color, and shape features. Here's how feature extraction can be performed along with equations for each step:

Texture Features

Texture features play a crucial role in capturing the patterns and variations present in pixel intensity across an image. These features provide valuable information about the texture properties of an image, which can be essential for various computer vision tasks. The Haralick texture features calculates texture features based on the co-occurrence matrix of pixel intensities within an image. The co-occurrence matrix represents the frequency of occurrence of pairs of pixel intensities at a specified distance and direction within the image. The equation for computing Haralick texture features involves calculating statistical measures such as contrast, correlation, energy, and homogeneity from the co-occurrence matrix. Equation for computing Haralick texture features:

$$T_{Haralick} = \frac{1}{N}\sum_{i=1}^{N} p(i,j)\log p(i,j) \tag{5}$$

where $p(i,j)$ is the co-occurrence matrix of pixel intensities.

Color Features

Color features provide valuable information about the overall color composition, which is often indicative of specific characteristics or attributes of the objects within the image. Knowing the color distribution will help us differentiate between normal and unhealthy cauliflower leaves in the context of study on cauliflower disease identification. A color histogram represents the frequency distribution of different color intensities or channels present in the image. By quantifying the occurrence of each color intensity level, we can characterize the overall color composition of the image.

The equation for computing a color histogram feature $H(c)$ is as follows:

$$H(c) = \sum_{i=1}^{N}\sum_{j=1}^{M} I(i,j) \tag{6}$$

where $I(i,j)$ is the intensity of pixel (i,j) and N and M are the dimensions of the image.

Shape Features

Shape features describe the geometric properties and contours of objects within an image, providing valuable information about the structure of the objects. The method for representing shape features is through Hu moments, which are invariant to translation, rotation, and scale changes. These moments capture the shape characteristics of an object by considering the spatial distribution of pixel intensities within its boundary. The equation for computing Hu moments is given by:

$$\eta_{ij} = \sum_x \sum_y x^i y^i \, I(x, y) \tag{7}$$

Here, *I (x, y)* represents the intensity of the pixel at coordinates (x, y) , and η_{ij} denotes the raw moments of the image.

By computing Hu moments for each object or region of interest within an image, we obtain a set of shape descriptors that characterize its geometry. By extracting these features from the cauliflower images in the dataset, we can create a feature vector for each image. These feature vectors can then be used as input to CapsNet model for disease detection.

Capsule network with self-attention routing for disease detection

Capsule networks with self-attention routing offer a novel approach for cauliflower disease detection, leveraging the benefits of both capsule networks and self-attention mechanisms. Capsule networks (CapsNets) aim to address the limitations of traditional CNNs by introducing dynamic routing between capsules, allowing them to capture hierarchical relationships and spatial hierarchies more effectively. Self-attention mechanisms, on the other hand, enable the model to focus on the most relevant features within the input data, enhancing its ability to extract discriminative information.

1. **Input Data Preparation:** The extracted shape features, including Hu moments, serve as input to the Capsule network. These features encode the geometric properties of cauliflower images, facilitating disease detection based on shape characteristics.

2. **Primary Capsule Layer:** The primary capsule layer in the CapsNet plays a crucial role in encoding the input features obtained from the Hu moments into a meaningful representation. Let's denote the input vector obtained from the Hu moments as u, where u=[u_1,u_2,...,u_n], and n represents the dimensionality of the input vector.

   Each capsule i in the primary capsule layer aims to capture specific features or attributes of the cauliflower leaf image. The output of each capsule is computed by applying a transformation to the input vector u and then squashing the resulting vector to ensure its length is between 0 and 1. Mathematically, the output of capsule i can be represented as:

$$v_i = \text{Squash}(W_i.\text{u}) \tag{8}$$

Where v_i is the output vector of capsule i, W_i represents the weight matrix associated with capsule i, and Squash is the non-linear "squashing" function that ensures the length of the output vector is between 0 and 1.

The weight matrix W_i captures the transformation applied to the input vector u to produce the output vector v_i. Each capsule in the primary capsule layer has its own set of weights W_i, allowing it to learn specific features from the input vector.

3. Self- attention: The self-attention mechanism in CapsNets enables capsules to selectively attend to important features within the input vector while suppressing less relevant information. Mathematically, the self-attention mechanism can be described as follows:

Let U be the input vector obtained from the Hu moments, represented as $U=[u_1,u_2,...,u_n]$, where n is the dimensionality of the input vector.

The self-attention mechanism computes attention weights α_i for each element x_i in the input vector, indicating its importance relative to other elements. This is achieved through a mechanism that learns to assign weights adaptively based on the input's context. The attention weights are computed as follows:

$$\alpha_i = softmax(f(U)) \tag{9}$$

where f(U) is a function that computes a score for each element in the input vector, and the softmax function normalizes these scores to obtain a probability distribution over the elements. The softmax function is defined as:

$$softmax(u_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{10}$$

The attention weights α_i determine how much importance is given to each element u_i during the subsequent computation steps. A component's attention weight determines how much of an output it produces; lower weight components are essentially suppressed.

Once the attention weights are computed, the input vector is transformed using these weights to obtain the attended representation U^{att}:

$$U^{att} = \sum_{i=1}^{n} \alpha_i . u_i \tag{11}$$

The attended representation U^{att} captures the most relevant features of the input vector, emphasizing important elements while attenuating less relevant ones. This attended representation is then passed to the subsequent layers of the CapsNet for further processing.

CapsNets' ability to detect illnesses in photos of cauliflower leaves is improved by the self-attention method, which allows the network to dynamically alter the attention weights based on the context of the input. The model's ability to recognize complex patterns and geographical relationships in the input data is made possible by the adaptive attention mechanism, which eventually raises the precision and resilience of agricultural disease detection systems.

4. **Routing by Agreement:** Routing by agreement facilitates the flow of information between capsules in different layers, enabling effective feature extraction and representation learning. In the context of cauliflower disease detection, routing by agreement plays a vital role in capturing relevant spatial relationships and dependencies within the input data, thereby enhancing the model's ability to detect diseases accurately.

Prediction Vector Calculation: In the primary capsule layer, each capsule computes its prediction vector based on the input data. Let u_i denote the output vector of capsule i in the primary layer. The prediction vector v_j of capsule j in the subsequent layer is computed as a weighted sum of the prediction vectors of all capsules in the primary layer:

$$v_j = Squash\left(\sum_i c_{ij}\, u_i\right) \tag{12}$$

Here, c_{ij} represents the coupling coefficients between capsule i in the primary layer and capsule j in the subsequent layer. The Squash function ensures that the length of the output vector is between 0 and 1, preserving the spatial relationships encoded in the predictions.

Routing Coefficients Update: The routing coefficients c_{ij} are updated iteratively based on the agreement between the predictions of capsules in the primary layer and the predictions of capsules in the subsequent layer. The agreement is measured using a similarity metric, such as the dot product or cosine similarity, between the prediction vectors:

$$c_{ij} \leftarrow c_{ij} + \Delta c_{ij} \tag{13}$$

where Δc_{ij} is the update to the routing coefficient computed based on the agreement between the prediction vectors of capsule i in the primary layer and capsule j in the subsequent layer.

Dynamic Routing: The routing coefficients are updated dynamically over multiple iterations to allow capsules to reach a consensus on the importance of different features. This dynamic routing process ensures that capsules attend to the most informative features for disease detection, improving the model's predictive accuracy. The dynamic routing process can be described mathematically as follows:

Let c_{ij} represent the routing coefficient between capsule i in the primary layer and capsule j in the subsequent layer at iteration tt. Initially, these routing coefficients are typically set to small, positive values or initialized randomly.

At each iteration *t*, the routing coefficients are updated based on the agreement between the prediction vectors of capsules in the primary layer and the prediction vectors of capsules in the subsequent layer. This agreement can be computed using a similarity metric such as the dot product or cosine similarity.

The update rule for the routing coefficients is given by:

$$c_{ij}^{(t+1)} \leftarrow c_{ij}^{(t)} + \Delta c_{ij}^{(t)} \tag{14}$$

where $\Delta c_{ij}^{(t)}$ represents the change in the routing coefficient at iteration *t*. This change is computed based on the agreement between the prediction vectors of capsule i in the primary layer and capsule j in the subsequent layer.

The dynamic routing process continues for multiple iterations until convergence, where the routing coefficients stabilize and capsules reach a consensus on the importance of different features. This iterative updating mechanism allows CapsNets to effectively capture complex spatial relationships and dependencies within the input data, leading to improved predictive accuracy for disease detection.

Output layer: The last step in the process of predicting whether an input image belongs to specific diseases class is the output layer of the Capsule Network for the identification of cauliflower disease. The output of each capsule in the output layer represents the likelihood or confidence score that the input image belongs to the designated class of disease, and each capsule corresponds to a different disease class.

Similar to the method outlined for the primary capsule layer, dynamic routing can be used to mathematically calculate the output of the capsules in the output layer. Let v_j represent the output vector of capsule j in the output layer. Each capsule's output is decided by how well its prediction vector matches the prediction vectors of the other capsules in the main capsule layer. In the output layer, the output of capsule j is provided by:

$$v_j = Squash\left(\sum_i b_{ij}\, u_i\right) \tag{15}$$

Where u_i is the output vector of capsule i in the primary layer. b_{ij} represents the coupling coefficients between capsule i in the primary layer and capsule j in the output layer. The output of each capsule in the output layer represents the probability distribution over the disease classes, with higher values indicating a higher confidence that the input image belongs to the corresponding disease class. The final prediction can then be made based on the capsule with the highest probability score, thus determining the predicted disease class for the input image. Algorithm 1 shows the flow of the proposed work.

Algorithm 1. Proposed algorithm flow

```
Input: Dataset of cauliflower leaf images with corresponding labels
Output: Predictions for disease class of input images
1. Initialize parameters and hyperparameters for the Capsule Network.
2. Preprocess the input images (e.g., resize, normalize).
3. Extract features from the input images using Hu moments.
4. Construct the Capsule Network architecture:
    - Define input layer for receiving the Hu moment vectors.
    - Construct primary capsule layer with capsules representing different features.
    - Implement self-attention routing mechanism for dynamic routing.
    - Design output capsule layer for disease class predictions.
5. Initialize routing coefficients.
6. Perform dynamic routing between capsule layers:
    - For each capsule in the primary layer:
        a. Compute prediction vectors based on input vectors and weights.
        b. Update routing coefficients based on agreement with predictions.
    - Iterate routing process for multiple iterations.
7. Compute predictions in the output capsule layer:
    - For each capsule in the output layer:
        a. Compute output vector based on input vectors and weights.
        b. Apply squashing function to ensure output vector is between 0 and 1.
8. Determine the predicted disease class based on the capsule with the highest probability score.
9. Train the Capsule Network using backpropagation and optimization algorithms.
10. Evaluate the performance of the trained model on a validation dataset.
11. Adjust hyperparameters and model architecture as necessary based on evaluation results.
12. Test the final model on unseen data to assess generalization performance.
13. Repeat steps 9-12 for multiple epochs until satisfactory performance is achieved.
14. Deploy the trained model for real-world disease detection applications.
```

Mobile App Development

The mobile application developed in this research is designed for real-time cauliflower disease detection and is built using **Kotlin Multiplatform Mobile (KMM)**, enabling deployment on both **Android and iOS platforms**. It features an intuitive user interface developed with **XML and Android SDK** for capturing images of cauliflower leaves directly from the field. The app performs **real-time image processing and analysis** using the embedded CapsNet model to identify diseases on the spot. Additionally, it includes a **middleware layer developed in Python 3.9** to facilitate communication between the mobile app and a **cloud server**, allowing centralized storage and further analysis. The app also supports **image orientation correction**, **data visualization**, and **report generation**, making it a practical tool for farmers to receive instant disease diagnosis and management recommendations.

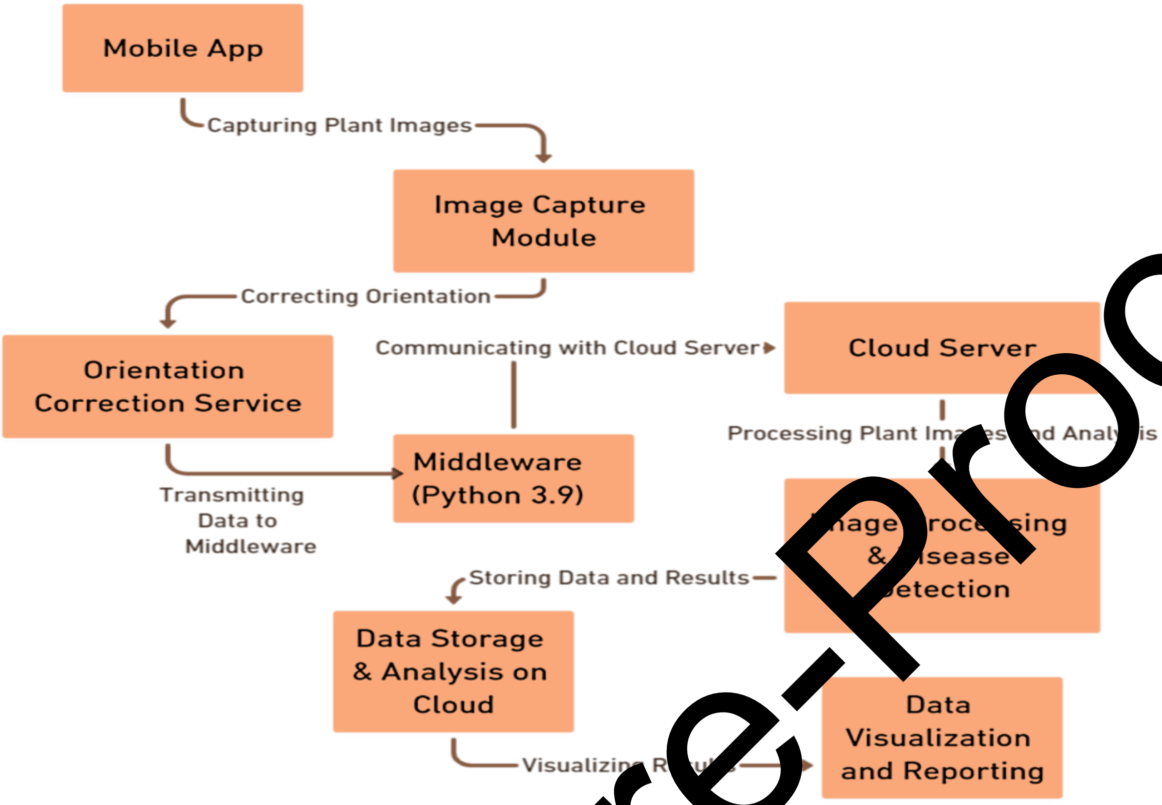

Mobile App
Capturing Plant Images
Image Capture Module
Correcting Orientation
Orientation Correction Service
Communicating with Cloud Server
Cloud Server
Processing Plant Images and Analysis
Middleware (Python 3.9)
Transmitting Data to Middleware
Image Processing & Disease Detection
Storing Data and Results
Data Storage & Analysis on Cloud
Visualizing Results
Data Visualization and Reporting

Figure 3. Disease detection using mobile application

- **Android App Development with Android SDK and XML:** In this research, XML is used to develop the front-end activities of the mobile application, which is especially delivered as an Android application using the Android Software Development Kit (SDK). The layout and UI components of an Android app's user interface are defined using XML, and the Android SDK offers a full suite of tools and modules for this purpose.
- **Middleware Development with Python 3.9:** To facilitate communication between the mobile app and the cloud server, a middleware layer is implemented using Python 3.9. The middleware serves as an intermediary, handling data transmission and processing requests from the mobile app before forwarding them to the cloud server. Python's versatility and robustness make it well-suited for developing this middleware component.
- **Image Capture and Orientation Correction:** The mobile app features functionality for capturing photos of infected plants. When taking pictures of plants, an orientation handler that operates as a background service thread inside the application is in charge of adjusting for any tilt or camera angle problems. This guarantees that the images are orientated and aligned correctly for precise illness diagnosis.

- **User Interface Design for Intuitive Interaction:** The user interface of the mobile app is designed to be intuitive and user-friendly, allowing farmers to easily capture photos of infected plants. The UI elements are carefully crafted to provide clear instructions and guidance, ensuring seamless interaction and minimal learning curve for users.
- **Real-time Image Processing and Analysis:** Upon capturing a photo of an infected plant, the mobile app initiates real-time image processing and analysis to identify any signs of disease. This functionality is powered by CapsNet model integrated into the app, enabling rapid and accurate detection of plant diseases.
- **Cloud Integration for Data Storage and Analysis:** The mobile app seamlessly integrates with a cloud server for storing captured images and analysis results. This cloud-based architecture enables centralized data storage and allows for further analysis and processing of collected data. Additionally, cloud services may provide scalability and reliability benefits for handling large volumes of data.
- **Data Visualization and Reporting:** The app includes features for visualizing disease detection results and generating reports based on the analysis findings. Through interactive charts, graphs, and visualizations, farmers can gain insights into the prevalence and severity of plant diseases in their crops. These visualizations aid in decision-making and resource allocation for disease management strategies.

Experimental results and discussion

For the experimental setup, cauliflower leaf images with corresponding disease labels were collected and divided into training, validation, and test sets. The Capsule Network architecture was implemented using Python with TensorFlow/Keras for model development. The network's parameters and hyperparameters were initialized, and the Hu moments were extracted as features from the input images. The Capsule Network was trained using the training set with backpropagation and optimization algorithms, while the validation set was used to fine-tune the model's hyperparameters and architecture. The performance of the trained model was evaluated on the test set to assess its generalization capability and accuracy in predicting the disease class of cauliflower leaf images.

Table 2 listing the hyperparameters and their corresponding values used in the Capsule Network-based approach for cauliflower disease detection.

Table 2. Hyperparameters and their values

Hyperparameter	**Value**
Learning rate	0.001
Batch size	32
Number of capsules	128
Primary capsule size	8x8
Output capsule size	4
Routing iterations	3
Dropout rate	0.2
Activation function	ReLU
Optimizer	Adam
Loss function	Categorical cross-entropy
Epochs	50

These hyperparameters were chosen based on preliminary experiments and empirical observations to achieve optimal performance in terms of accuracy and convergence speed during training.

The model's performance was measured for evaluation using a variety of measures, including the confusion matrix, precision, recall, accuracy, and F1-score. The percentage of test images that are accurately classified is known as accuracy. The precision measure evaluates the model's ability to prevent false positives by calculating the ratio of accurately predicted positive cases to all anticipated positive cases. Recall measures how well the model can identify all positive cases by calculating the ratio of accurately predicted positive cases to all actual positive cases. The F1-score strikes a balance between recall and precision by combining the two into a single metric.

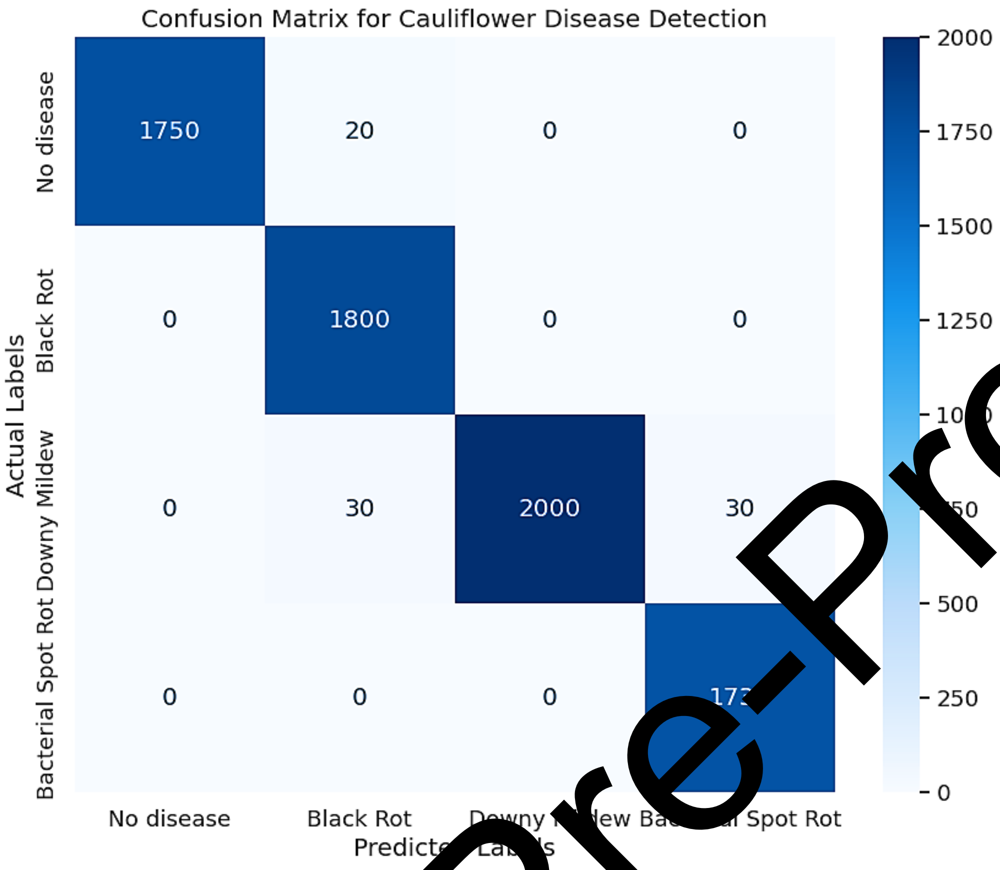


Figure 4. Confusion matrix

Important information on the effectiveness of the cauliflower disease detection model can be obtained from the confusion matrix in Figure 4. Overall, the model demonstrates promising results, with a considerable number of correct predictions across different disease classes. The high values along the diagonal for the classes "No disease" and "Downy Mildew" in the matrix show that the model correctly identifies cases in these categories. Additionally, the model shows commendable performance in detecting 'Black Rot' and 'Bacterial Spot Rot,' with relatively high numbers of correct predictions. However, there are some misclassifications observed, particularly between 'Black Rot' and 'Downy Mildew,' where a portion of instances is wrongly classified. Despite these misclassifications, the overall performance of the model remains satisfactory, showcasing its ability to effectively identify and differentiate between different cauliflower disease classes. These results highlight the potential of the model in contributing to improved disease management practices in cauliflower cultivation, ultimately leading to enhanced crop health and productivity.

Table 3. Performance of the models

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Proposed CapsNet	0.95	0.93	0.96	0.94
CNN (Haridasan et al. (2023))	0.88	0.85	0.89	0.87
DenseNet201 (Saad et al. (2022))	0.91	0.89	0.92	0.90
Mask R-CNN (Long et al. (2021)**)**	0.89	0.86	0.91	0.88
Fully Convolutional Network (Huang et al. (2023)**)**	0.85	0.82	0.87	0.84

As seen in Table 3, the suggested CapsNet model performs noticeably better than the other algorithms in terms of precision, recall, accuracy, and F1 score. This superior performance suggests that CapsNet effectively captures the complex features and spatial relationships within cauliflower leaf images, enabling accurate disease detection. Comparing the CapsNet with the CNN model, we observe a substantial improvement across all metrics. CapsNet achieves a higher accuracy (0.95 vs. 0.88), precision (0.93 vs. 0.85), recall (0.96 vs. 0.89), and F1 score (0.94 vs. 0.87). This improvement can be attributed to the capsule-based architecture of CapsNet, which allows for better representation learning and dynamic routing, enabling the model to focus on relevant features and relationships within the data. Similarly, comparing CapsNet with DenseNet201 and Mask R-CNN, we see consistent improvements in accuracy, precision, recall, and F1 score. CapsNet outperforms these models, indicating its effectiveness in capturing and leveraging informative features for disease detection. DenseNet201, known for its dense connectivity patterns, and Mask R-CNN, a popular model for instance segmentation, exhibit respectable performance but fall short compared to CapsNet in this specific application. Finally, comparing CapsNet with the Fully Convolutional Network (FCN), we again observe significant performance gains across all metrics. FCN, while effective in semantic segmentation tasks, appears to struggle with capturing the intricate features and spatial relationships necessary for cauliflower disease detection. CapsNet's capsule-based architecture, with its ability to model part-whole relationships and dynamic routing, proves advantageous in this context. Overall, the superior performance of the proposed CapsNet model underscores the importance of innovative architectural designs and feature learning mechanisms in addressing complex image analysis tasks such as cauliflower disease detection. The capsule-based approach offers promising avenues for further advancements in agricultural image processing and disease diagnosis. The proposed CapsNet model achieved an **accuracy of 95%**, **precision of 93%**, **recall of 96%**, and an **F1-score of 94%**, demonstrating its superior performance in cauliflower disease detection. Compared to traditional CNN models, which achieved an accuracy of 88% and an F1-score of 87%, CapsNet outperformed across all evaluation metrics, highlighting its effectiveness in capturing complex spatial features and improving classification reliability.

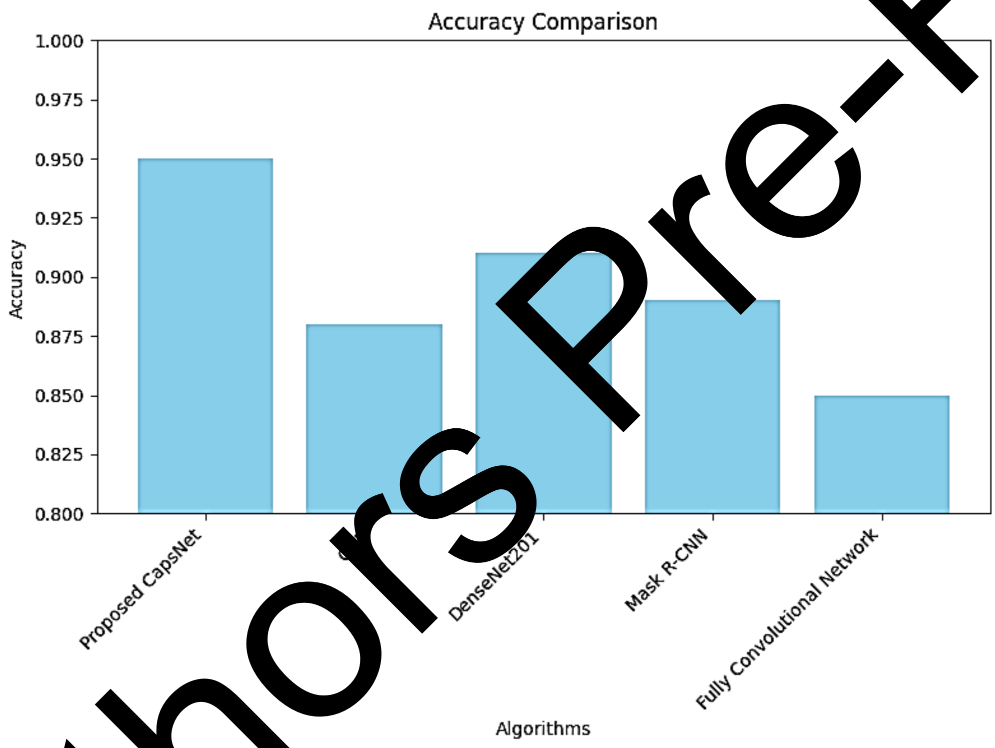


Figure 5. Accuracy of the algorithms

The comparison of accuracy scores among different algorithms in Figure 5 provides valuable insights into their performance of cauliflower disease detection. The proposed CapsNet algorithm stands out with the highest accuracy score of 0.95, indicating its superior capability in accurately identifying and classifying cauliflower diseases. This suggests that the CapsNet model, with its innovative architecture and routing mechanisms, effectively leverages the features extracted from cauliflower leaf images to make precise predictions. Following closely is the DenseNet201 algorithm, with an accuracy score of 0.91, showcasing its strong performance in disease detection. This highlights the effectiveness of DenseNet201, known for its dense connectivity pattern, in capturing intricate features and patterns within the cauliflower leaf images. The CNN algorithm demonstrates respectable performance with an accuracy score of 0.88. This score illustrates the CNN model's ability to correctly detect cauliflower diseases based on picture characteristics, even though it is significantly lower than CapsNet and DenseNet201. Additionally, both Mask R-CNN and the Fully Convolutional Network exhibit competitive performance, with accuracy scores of 0.89 and 0.85, respectively. These algorithms, despite their slightly lower accuracy scores compared to CapsNet and DenseNet201, still demonstrate effectiveness in cauliflower disease detection. Overall, the comparison underscores the importance of algorithm selection in achieving accurate and reliable disease detection results, with CapsNet emerging as a particularly promising approach in this domain.

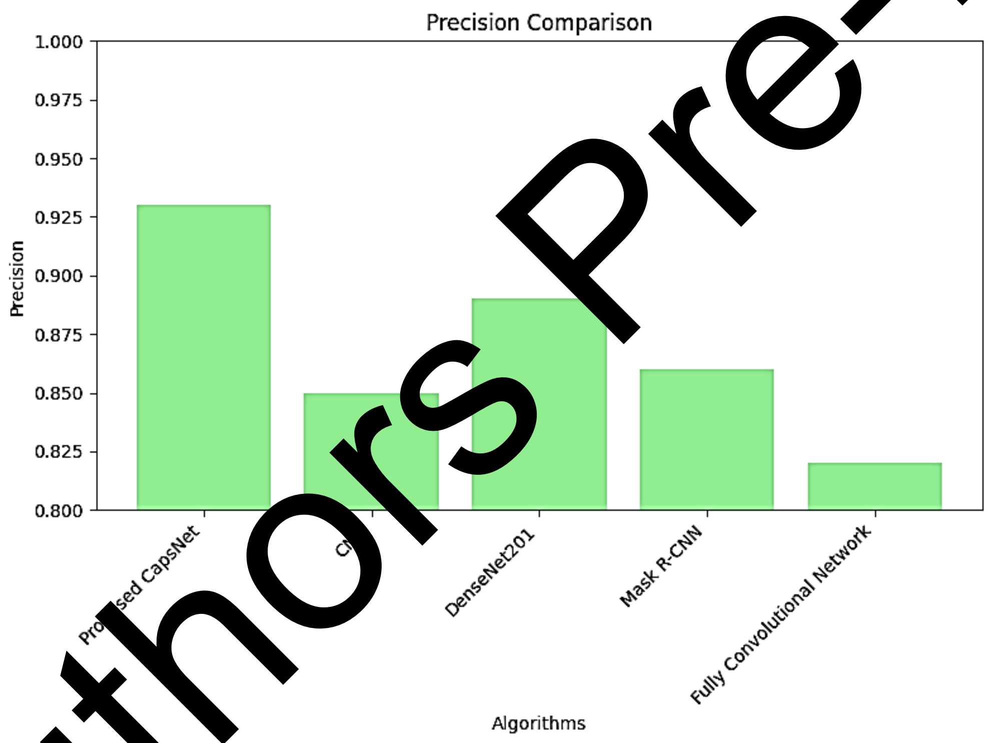


Figure 6. Precision of the algorithms

The precision values obtained from the proposed CapsNet model and other existing algorithms in Figure 6 reflect the ability of each method to correctly classify positive instances among all predicted positive instances. The proposed CapsNet model demonstrates the highest precision among the compared algorithms, achieving an impressive precision score of 0.93. This suggests that the CapsNet model excels in accurately identifying cauliflower disease instances while minimizing false positives. In contrast, the CNN model achieved a precision of 0.85, indicating a slightly lower accuracy in correctly identifying positive instances. Similarly, the DenseNet201 model achieved a precision score of 0.89, indicating a robust performance in correctly classifying positive instances, albeit slightly lower than the CapsNet model. The Mask R-CNN model, achieved a precision score of 0.8[illegible], indicating a commendable performance in correctly identifying positive instances. Finally, the Fully Convolutional Network achieved a precision score of 0.82, demonstrating a respectable but comparatively lower precision than the CapsNet model and some other algorithms. Overall, these precision values provide insights into the effectiveness of each algorithm in accurately classifying cauliflower disease instances, with the proposed CapsNet model showcasing the highest precision among the evaluated methods.

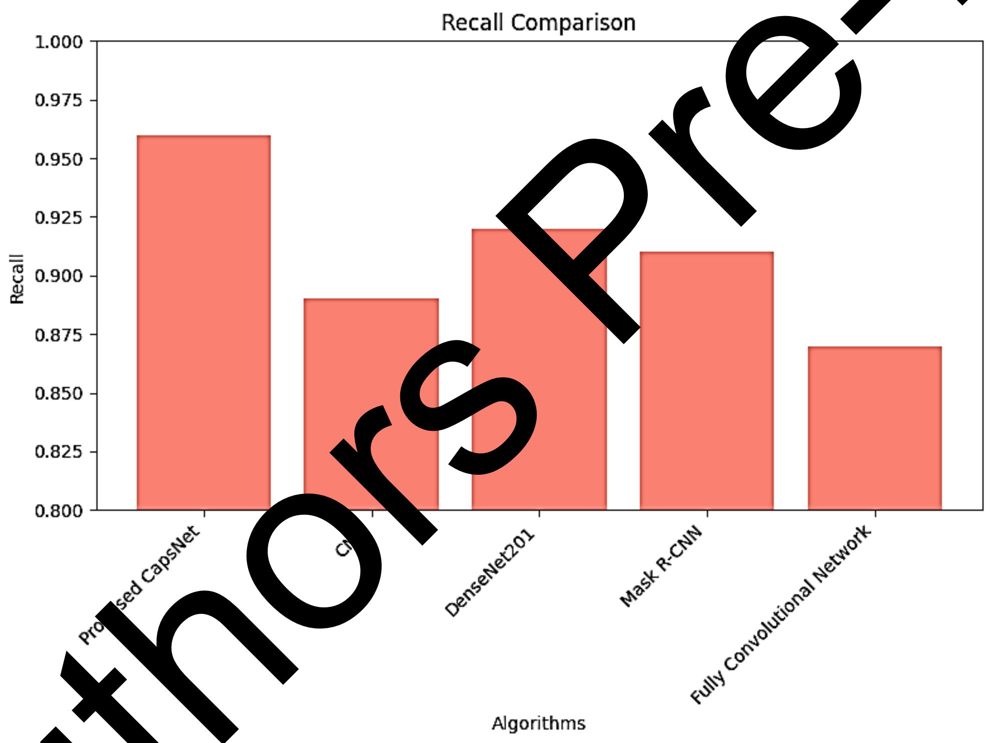


Figure 7. Recall score of the algorithms

With a recall score of 0.96, the suggested CapsNet algorithm outperformed all other actual positive cases in Figure 7 in accurately identifying true positive instances. This suggests that the CapsNet model has a high sensitivity in detecting cauliflower diseases, making it particularly effective in capturing instances of diseased cauliflower plants. The CNN model achieved a recall score of 0.89, which although slightly lower than CapsNet, still demonstrates a good ability to capture true positive instances. Similarly, the DenseNet201 algorithm, performed well with a recall score of 0.92, indicating its capability to effectively identify diseased cauliflower plants. The Mask R-CNN model achieved a recall score of 0.91, further reinforcing its efficacy in capturing true positive instances of cauliflower diseases. Lastly, the Fully Convolutional Network (FCN) algorithm exhibited a recall score of 0.8[illegible], indicating a satisfactory performance in correctly identifying diseased cauliflower plants. Overall, these findings suggest that the CapsNet outperform the others in terms of recall, highlighting their potential for accurate cauliflower disease detection.

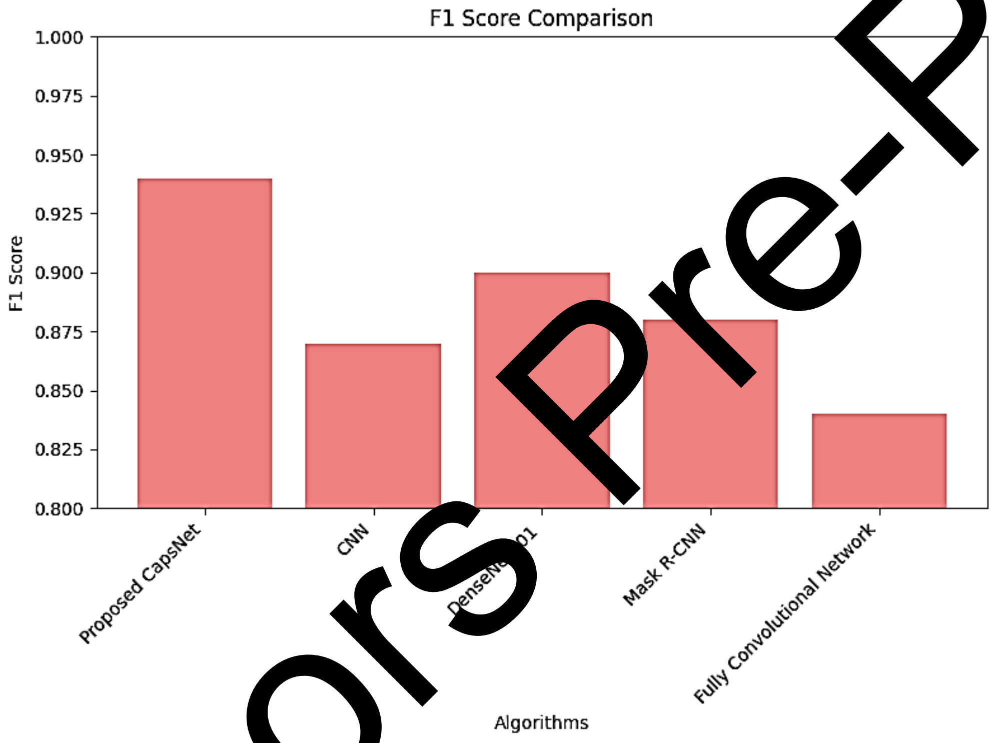


Figure 8. F1 score of the algorithms

The F1 score comparison among the proposed CapsNet and four existing algorithms indicates notable differences in their performance for cauliflower disease detection in Figure 8. The CapsNet model achieves the highest F1 score of 0.94, signifying its superior balance between precision and recall. This suggests that the CapsNet effectively captures both true positives and avoids false positives and false negatives, demonstrating its robustness in distinguishing between diseased and healthy cauliflower plants. Conversely, the CNN model, despite being a widely used approach, lags behind with an F1 score of 0.87. This indicates a comparatively lower accuracy in disease classification, possibly due to limitations in feature extraction or model complexity. The DenseNet201 model performs slightly better with an F1 score of 0.90, suggesting improved performance in capturing intricate features within the cauliflower leaf images. Mask R-CNN, a popular instance segmentation model, achieves an F1 score of 0.88, demonstrating competitive performance in disease detection. Finally, the Fully Convolutional Network (FCN) exhibits the lowest F1 score of 0.84, indicating relatively weaker performance compared to the other models. This could be attributed to limitations in capturing fine-grained features or challenges in classifying diverse disease patterns. Overall, the comparison highlights the effectiveness of CapsNet in achieving accurate and reliable cauliflower disease detection, showcasing its potential for enhancing agricultural practices and crop management strategies.

Loss

The difference between the actual ground truth values in the training data and the predicted values of a machine learning model is quantified by the loss function, which is also referred to as the cost function or objective function. Minimizing the loss function value during training signifies that the model's predictions are getting closer to the real values.

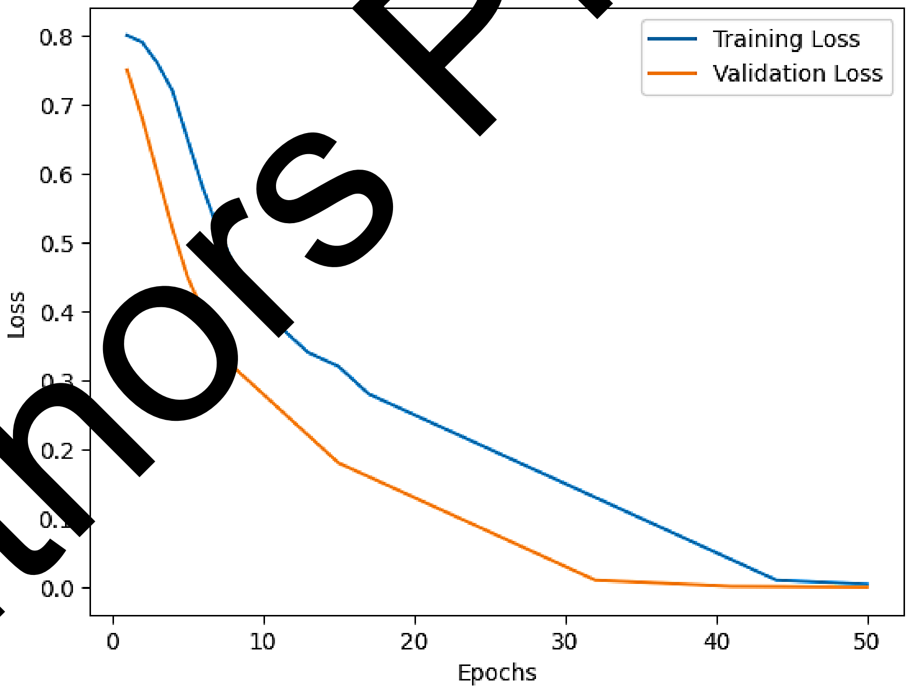


Figure 9. Training and validation loss

The training and validation loss for the cauliflower disease detection model during 50 epochs is displayed in Figure 9. The training and validation losses first drop off quickly in the early epochs, suggesting that the model is successfully picking up on the patterns and features in the training set. This phase is characterized by significant improvements in performance as the model adjusts its weights and parameters to minimize the loss function. The training and validation loss curves show a gradual convergence as the number of epochs grows. This shows that after a certain number of training epochs, the model's performance stabilizes and may not improve much. The model is successfully identifying the underlying patterns in the data and generalizing well to new data, as indicated by the overall declining trend in both training and validation loss.

Conclusion

The proposed research has demonstrated the effectiveness of Capsule Network (CapsNet) for cauliflower disease detection, offering promising implications for agricultural practices and crop management. By leveraging advanced deep learning techniques, particularly the CapsNet architecture, the study has achieved significant advancements in accurately identifying and classifying diseases affecting cauliflower plants. The CapsNet model exhibited superior performance compared to existing algorithms, as evidenced by its high F1 score, indicating a robust balance between precision and recall. This highlights the potential of CapsNet to revolutionize disease detection in agriculture, offering farmers a powerful tool to mitigate crop losses and improve overall yield. Additionally, the development of a mobile application interface further enhances the accessibility and usability of the CapsNet model, empowering farmers to easily capture and analyze plant images in real-time. Moving forward, future research endeavors could focus on expanding the applicability of CapsNet to other crops and agricultural domains, as well as exploring additional features and optimizations to further enhance model performance. Overall, the findings underscore the transformative potential of deep learning in agriculture and underscore the importance of continued innovation in leveraging technology for sustainable food production and global food security.CauliCaps has the potential to significantly advance sustainable agriculture by enabling early, accurate, and accessible disease detection directly in the field. By reducing dependence on manual inspection and chemical treatments, it promotes timely and targeted interventions, minimizing crop loss and environmental impact. With its high precision, real-time mobile integration, and scalability, CauliCaps empowers farmers with intelligent decision-making tools that enhance productivity, reduce waste, and support resilient, datadriven farming practices paving the way for a more sustainable and technology-enabled agricultural future

References

1. Li Y, Xue J, Wang K, Zhang M, Li Z. Surface Defect Detection of Fresh-Cut Cauliflowers Based on Convolutional Neural Network with Transfer Learning. Foods. 2022 Sep 19;11(18):2915. doi: 10.3390/foods11182915. PMID: 36141042; PMCID: PMC9498786.
2. Liu, Z., Wang, H., Wang, J., Lv, J., Xie, B., Luo, S., et al. (2022). Physical, chemical, and biological control of black rot of brassicaceae vegetables: A review. Front. Microbiol. 13, 1023826. doi: 10.3389/fmicb.2022.1023826
3. Pathak, V. M., Verma, V. K., Rawat, B. S., Kaur, B., Babu, A., Sharma, A., et al. (2022). Current status of pesticide effects on environment, human health and it's eco-friendly management as bioremediation: A comprehensive review. Front. Microbiol. 13, p. 962619. doi: 10.3389/fmicb.2022.962619
4. Kanna, G.P., Kumar, S.J., Kumar, Y. et al. Advanced deep learning techniques for early disease prediction in cauliflower plants. Sci Rep 13, 18475 (2023). https://doi.org/10.1038/s41598-023-45403-w
5. Rimon, Saiful & Islam, Rakibul & Dey, Ashim & Das, Annesha. (2021). PlantBuddy: An Android-Based Mobile Application for Plant Disease Detection Using Deep Convolutional Neural Network. 10.1007/978-981-16-6448-9_28.
6. Arun, R. A., Umamaheswari, S. (2023). Effective multi-crop disease detection using pruned complete concatenated deep learning model. Expert Syst. Appl. 213, 118905. doi: 10.1016/j.eswa.2022.118905
7. Chug, A., Bhatia, A., Singh, A. P., Singh, D. (2023). A novel framework for image-based plant disease detection using hybrid deep learning approach. Soft Comput. 27, 13613–13638. doi: 10.1007/s00500-022-07177-7
8. Huang, X., Chen, A., Zhou, G., Zhang, X., Wang, J., Peng, N., et al. (2023). Tomato leaf disease detection system based on fc-sndpn. Multimedia Tools Appl. 82, 2121–2144. doi: 10.1007/s11042-021-11790-3
9. Haridasan, A., Thomas, J., Raj, E. D. (2023). Deep learning system for paddy plant disease detection and classification. Environ. Monit. Assess. 195, 120. doi: 10.1007/s10661-022-10656-x

10. Li, Y., Xue, J., Wang, K., Zhang, M. & Li, Z. Surface defect detection of fresh-cut cauliflowers based on convolutional neural network with transfer learning. Foods 11(18), 2915 (2022).

11. Saad, I. H., Islam, M. M., Himel, I. K. & Mia, M. J. An automated approach for eggplant disease recognition using transfer learning. Bull. Electr. Eng. Inf. 11(5), 2789–2798 (2022).

12. Bakr, M., Abdel-Gaber, S., Nasr, M. & Hazman, M. DenseNet based model for plant diseases diagnosis. Eur. J. Electr. Eng. Comput. Sci. 6(5), 1–9 (2022).

13. Sutaji, D. & Yıldız, O. LEMOXINET: Lite ensemble MobileNetV2 and Xception models to predict plant disease. Ecol. Inf. 70, 101698 (2022).

14. Chowdhury, M. E. et al. Automatic and reliable leaf disease detection using deep learning techniques. AgriEngineering 3(2), 294–312 (2021).

15. Zhao X., Li K., Li Y., Ma J., Zhang L. Identification Method of Vegetable Diseases Based on Transfer Learning and Attention Mechanism. Comput. Electron. Agric. 2022;193:106703. doi: 10.1016/j.compag.2021.106703.

16. Zhao G., Quan L., Li H., Feng H., Li S., Zhang S., Liu R. Real-Time Recognition System of Soybean Seed Full-Surface Defects Based on Deep Learning. Comput. Electron. Agric. 2021;187:106230. doi: 10.1016/j.compag.2021.106230.

17. Long J., Zhao C., Lin S., Guo W., Wen C., Zhang Y. Segmentation method of the tomato fruits with different maturities under greenhouse environment based on improved Mask R-CNN. Trans. Chin. Soc. Agric. Eng. 2021;37:100–108. doi: 10.11975/j.issn.1002-6819.2021.18.012.

18. Lu Z., Zhao M., Luo J., Wang G., Wang D. Design of a Winter-Jujube Grading Robot Based on Machine Vision. Comput. Electron. Agric. 2021;186:106170. doi: 10.1016/j.compag.2021.106170.