

A DUAL-HEAD CNN-LSTM FRAMEWORK FOR SIMULATING AND ANALYZING PLANT DISEASE PROGRESSION USING SYNTHETIC TEMPORAL SEQUENCES

NALLAMOTHU RAGHU^{1*}, PARDEEP KUMAR²

¹Research Scholar, Dept of Artificial Intelligence, School of Engineering, Anurag University, India

²Associate Professor, Dept of Artificial Intelligence, School of Engineering, Anurag University, India

E-mail: *¹nallamothuraghus@gmail.com, ²pardeesep2@gmail.com

ABSTRACT

In this study, a novel deep learning framework for plant disease analysis that simulates disease progression using synthetic temporal sequences generated from static leaf images is proposed. Traditional image-based disease classifiers ignore the temporal nature of plant pathology, which limits interpretability and prediction accuracy. To address this, the sequences are synthesised by interpolating between a pseudo-healthy image and a diseased sample using alpha blending. These sequences mimic gradual progression across four frames. The design of a dual-head CNN-LSTM model that takes these sequences as input, leveraging convolutional layers to extract spatial features and LSTMs to capture temporal dynamics. The architecture branches into two heads: one for disease classification via softmax and another for severity estimation through regression. Further integrated a frame-wise attention mechanism and temporal consistency regularisation to improve interpretability and stability. Evaluation on the corn leaf disease dataset demonstrates strong performance, achieving 92.16% classification accuracy and a severity MSE of 0.0001. The comparison of the proposed model against traditional classifiers like Random Forests and Logistic Regression shows a 7–10% improvement in F1-score and significantly better severity awareness. Our approach offers a temporally aware, explainable AI solution for agricultural monitoring. Severity curves and attention maps enable not only post-hoc analysis but also facilitate real-time field-level diagnostics. The pipeline's ability to generate balanced data across underrepresented disease classes is crucial in imbalanced datasets. Moreover, the use of synthetic progression opens opportunities for applying this model to other domains lacking temporal labels. The modular structure of the system supports easy plug-and-play integration with UAVs, mobile apps, and remote sensing platforms. In conclusion, this work bridges the gap between image classification and temporal disease modelling, making a strong case for progression-aware AI in digital agriculture.

Keywords: *Synthetic Temporal Sequences, CNN-LSTM, Plant Disease Progression, Severity Prediction, Frame-Wise Attention, Early Intervention, Dual-Task Learning, Grad-CAM, Corn Leaf Dataset, Interpretable AI.*

1. INTRODUCTION

Corn is one of the most important staple crops in the world, and any significant threat to its health can affect food security, economic stability and global supply chains. While traditional image-based detection systems can identify the presence of the

disease at a certain time, they fail to provide insight on how a disease develops over time, which is important to assess the urgency and plan intervention[13]. Temporal image sequences provide a rich dimension, which progress the symptoms, which progressively and text progress into the frame. Such progression-individual

analysis is especially important in diseases such as blight or grey leaf spots, where visual symptoms may appear subtle in the initial stages but grow rapidly. Initial detections in these stages allow timely fungicide application, quarantine, or targeted treatment, reduced crop loss. Static image classifier can remember or reduce initial-phase infections, especially in visually obscure cases. Temporal sequences follow these micro-to serious infections and allow the model to track gradual changes-an essential ability for high-dot crops such as cucumber[14]. In addition, capturing the temporary reference improves the strength of the model by reducing single-frame noise or environmental artifacts such as lighting and obstacles. In a real-world field setting, where the quality of the image can vary greatly, the ability to argue on the sequence of the frame increases the confidence of prediction. Ultimately, integrating temporary data in the disease detection structure not only provides accuracy, but also provides a forecast lens - the plant diagnostics are reactive to active.

1.1. Need for Tracking Disease Progression Over Time using Temporal Sequence

Over time, the progression of tracking disease is necessary to convert the diagnosis into action. Most existing plant disease detection systems provide only one snapshot of the problem, which lacks the ability to refer to it within a large timeline of change. This is the same to diagnose a patient based on a single X-ray without monitoring their symptoms over time. In agricultural settings, however, it is important to understand whether a disease is getting stable, deteriorating, or responding to treatment on time[15]. Time-aware models allow us to detect the presence of the disease but also guess the severity of the future. This enables the future capacity implementing agricultural strategies, reducing unnecessary chemical applications and adapting the crop deadline. For example, if the estimated severity curve reflects rapid progress, farmers can prioritize intervention in affected plots. In our structure, this temporary tracking frame-wise seriousness is obtained by following the predictions and imagining them as a decrease, which leads to interpretable and actionable action. Such trajectory treatment provides immediate insights into effectiveness and long-term crop health. In addition, the progression of progress patterns in tracking areas allows the facility of epidemiology

modelling at the farm or district level. This macro-level visibility is rapidly important in the era of climate instability and global food insecurity. Ultimately, the disease transforms tracking diagnosis into an active, data-powered loop-enables sustainable and accurate agriculture that develops with a crop life cycle.

1.2. Disease Progression Classification Using Temporal Image Sequences

Classifying the progression of the disease - not only the classification- is important for effective agricultural management. Temporal image sequences offer a unique opportunity to model the gradation of symptoms, enabled the model to learn the patterns that indicate whether a disease is starting, or stabilized[16]. In our structure, synthetic sequences are formed to mimic the progression of the real world, which begins with a pseudo-healthy frame and moves towards a fully developed transition. This allows the model to inspect the development of pathological characteristics over time, such as wound expansion, color fall and texture fragmentation. Using a temporary sequence, models can classify not only the type of disease (eg: Rust vs. Blight), but can also combine it with a progression phase-determined as a severity score in our double-head setup. This level of classification adds an important temporary axis to the disease understanding, which is important to assess yield loss or to plan re-treatment intervals. In practical deployment, the progression-comprehensive classification helps prefer immediate attention plots, rather than relying on binary healthy/diseased outputs only. In addition, mapping the stages of the disease over time can contribute to long-term crop health forecasting and agricultural research. In our architecture, severity regression heads the difference between effectively ranked and continuous diagnosis. By embedding this function directly into the learning process, we move beyond the traditional classifier and move towards time-sensitive, progression-comprehensive intelligence for crop disease management.

1.3. Sequential Modelling Types

The sequence modelling technique is important in understanding data that develops over time or has dependence in a sequence. Short-term, ordered data refers to scenarios where recent past directly affects the near future, such as language modelling or short-term stock trends. Reverse neural networks such as LSTM (long term short-term memory) and GRU (gated recurrent unit) excel in these functions due to their ability to

maintain references on low sequences[17]. On the other hand, long-term, uncontrolled data modelling emphasizes capturing the remote dependence where the sequence order matters less. This is the place where transformers thrive, especially in tasks such as document summary or DNA sequence modelling, where distant relationships are closely matured. Transformers take advantage of the self-eclipse mechanism to process the entire sequences simultaneously, which they are able to improve RNN in both speed and reference retention[18]. Meanwhile, some data, such as video or brain scans, demand modelling in space and time. It introduces high-dimensional spatial-temporal modelling. Tools such as 3D CNN and Timesformer can process this multidimensional input. A 3D CNN explains spatial features in multiple frames or slices, suitable for volumetric data such as MRI scans. Timesformer increase it by introducing cosmic attention mechanisms, which is ideal for dynamic sequences such as FMRI or video. Together, these models bridge the spatial and cosmic dependence in high-dimensional data currents, which play an important role in medical imaging and video understanding.

Some systems do not develop attested, but possibly, meaning that there is uncertainty in between transition states. To model it, Hidden Markov Models (HMM) and Variational Autoencoders (VAES) are used. HMMs are classical statistical tools that consider a sequence of observable events affected by the sequence of hidden states, often used in speech recognition and bio-informa science. Vaes, meanwhile, are deep common models that learn a probable distribution on latent variables, which are useful in scenarios such as discrepancy detection and potential forecast[19]. These models are particularly powerful when modelling uncertainty or variation in temporal sequences, providing more flexibility than deterministic networks. Another important type of progress is graphical progress, where data is best represented as nodes and edges - such as brain areas in neurological, social networks or traffic flows.

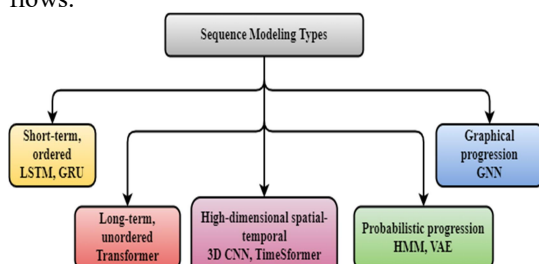


Figure 1: Classification of Sequence Modelling

In such cases, graphs neural networks (GNNs) are Go-Two models. Capturing structural and relationship dependence within GNNs graphs, can promote information in interconnected nodes. They enable powerful representation to learn where the data structure contributes significantly to its interpretation. In applications such as brain connectivity data from autism diagnosis, GNNs can model interactions, learning complex patterns that can miss traditional networks[20]. Together, these diverse modelling approaches provide a toolkit to deal with various sequences and structure-conducted tasks in machine learning.

1.4. Handling Temporal Dependencies Using RNN

The Recurrent Neural Network (RNN), and especially designed to process sequential data to their gated variants such as LSTM and GRU where the previous reference reports the current decision. Plant disease in modelling, translates to understand how the symptoms of the disease develop in gradual frames. Unlike CNN which are purely operated, RNN encodes temporary dependence while maintaining a hidden state which captures information from all previous inputs. This sequence-based diagnosis is important because early frames can show subtle signals, while the latter frames highlight the fully developed symptoms. By using LSTMs, we ensure that long-term temporary dependence is preserved without suffering from disappearance gradients-a common issue in Venilla RNN. The hidden state transition allows the frame-wise features to be maintained or forgotten by using gates learned gates. In our architecture, each frame is passed through a CNN to remove spatial features, which is later fed to LSTM. The final hidden condition encoded the progression of the entire disease, which serves as integrated representation for both classification and severity regression. This sequence enables the modelling network to estimate what the disease is, but how it has been treating over time. Additionally, the attention mechanism applied above the LSTM allows the model to disorient frame weight, focus on those most relevant for predictions. Overall, RNN disease provides the spine required for temporary arguments in progress systems.

1.5. Limitations of the Traditional Approaches

While RNNs and CNNs provide a powerful toolkit for joint sequence-based modelling, they have boundaries in their current form that affect performance and scalability. First,

the RNNS procedure sequence gradually, which can lead to disabilities during both training and estimates. This naturally limits their ability to prolonged sequences or real-time applications. Additionally, they often struggle with modelling complex temporary patterns in noisy or irregularly sample sequences—a common phenomenon in the agricultural settings of the world. Even LSTMs, despite their gating mechanisms, can demonstrate a decline in performance when sequences have sudden infection or missing frames. Another challenge lies in overfitting for synthetic patterns, especially when the progression of real disease does not fully follow linear visual trends, as is artificially generated. The model can learn to guess uniform progress, allowing its generalization to be reduced in field conditions. In addition, the camera angle, lighting, and background dislocation can cause spatial anomalies in the frame, which confuses the temporary model. The RNNs lacks a clear mechanism for handling multiscale temporal patterns, such as rapid-onset disease vs. slow progress. From an interpretation point of view, while attention helps, it is difficult to detect long-term frame effects as the length of the sequence increases. Finally, deployment challenges arise when integrating RNN-based architecture in a low-power-edged equipment used in agricultural drones or mobile apps. These boundaries suggest better modelling techniques, such as Temporal transformers, spatial-temporal graph networks, or the need for hybrid self-execution mechanisms, which can offer better scalability, strength and interpretability.

The majority of previous analyses only analyzed the classification of a static image without the dynamics of plant diseases. Other models such as MMF-Net and MaizeNet attained high accuracy but it did not estimate the severity or gain temporal information. Most of the methods relied on handcrafted characteristics or simple CNNs that played the roles of burning down deeper, nonlinear patterns of the disease. Other techniques were segmentation based systems that were not integrated with scoring on disease severity. Also, there was the problem of class imbalance in several models, and they did not indicate the methods to overcome the lack of representation in certain disease types. Applicability in real-time was also often absent, and very little applications were conceived to be deployed in low resources settings. Interpretations of the results were more difficult

because explainability tools such as attention maps or severity curves were infrequently employed. In addition, the majority of models had been trained only on particular datasets and did not produce generalization over a variety of crops or even types of diseases. Last but not least, among the previous works, there was no case study on how to model a synthetic progression to model the change in time using still images.

1.6 Novelty of the proposed methodology

The uniqueness of this work is presenting a new method to analyze plant disease by building synthetic sequences to demonstrate disease progression over time with only a single static image. Rather than having to resort to real-time videos, the technique involves compositing a healthy appearance of a leaf with the diseased version so as to create a smooth display. It utilizes a specialized dual-head CNN-LSTM that is applied at detecting the disease and approximating its severity. This increases the usefulness of the model to farmers by providing more specific information. Attention mechanisms are also used in the system to pay attention to key frames and makes changes in between frames smooth. As compared to its predecessors, this approach justifies its decisions with the help of heatmaps and severity curves. It makes unbalanced datasets even by designing comparative population of samples in each type of disease. The model had a high accuracy, and low testing error. It can also be conveniently applied to practical devices such as drones or mobile application technologies. However, this introduces new attributes that make the detection of plant disease smarter and more helpful.

1.7 Objective

Their primary objective is to come up with a deep learning model, capable of not only classifying plant diseases, but also predicting their severity with the progress of time. The model employs the use of synthesized sequences based on the individual still images of leaves as the simulated images to give a image of the developing pathogen. This will enhance the prediction efficiency, manage the problem of class imbalance by creating equal samples of each disease, and help

interpret the results more easily because of attention maps and severity curves. It also contrasts results of the proposed model with old techniques to point out the increase in classification and ability to estimate severity.

2. LITERATURE SURVEY

Rubina Rashid et al [1] There are 4,188 images of corn leaves which contain healthy leaves along with leaves infected by blight, common rust and gray leaf spot. The method proposes an integrated MMF-Net which unites CNNs with IoT data. The proposed model uses three connected sub-networks that include the RL-block which analyzes image features at a coarse level while the PL-block 1 extracts fine detailed global information with PL-block 2 integrating real-time environment data in its process. The extracted features are combined through an ensemble system at the decision stage by multiple classifiers. Training with heterogeneous data containing numeric values and images enables improved performance of disease classification through the model. A majority voting program post-feature extraction serves to enhance disease identification accuracy by sustaining reliability in diagnosis.

L G Divyanth et al [2] The dataset includes 1,050 images contains GLS, NLB and NLS infections on corn leaves which were photographed in field conditions with multiple background elements including soil, weeds and shadows. A two-stage deep learning segmentation process relies on SegNet, UNet and DeepLabV3+ as the proposed methodology. The first stage of the methodology employs a UNet model to separate corn leaves against complex background attributes. In stage two the process removes disease lesions from segmented leaves by utilizing DeepLabV3+. The trained models operate on annotated images while augmentation processes support their operation. The convolutional layers in combination with pooling layers as well as activation functions work together for extracting features from the processed data. The model uses atrous convolutions through DeepLabV3+ for precise detection of small-sized lesion features.

PAN Shuai-qun et al [3] A total of 985 maize leaf images affected by NCLB were obtained from Jilin Province fields and greenhouses and this collection was expanded through data augmentation to reach 30,655 images. The proposed methodology utilizes DCNN for disease diagnosis. The DCNN architectures including four TL among those GoogleNet has high-efficient performance. Before analysis Images need to pre-processed through methods including resizing, cropping, rotation and flipping operations. A proper distribution of the data exists through training, validation and testing phases to confirm model reliability. The training process employs SGD optimization while tuning learning rate and momentum parameters. The Softmax loss combines A-Softmax loss, CosFace loss and ArcFace loss act as classification performance enhancement tools. The detection of NCLB benefits most from using GoogleNet with Softmax loss function.

Nidhi Kundu et al [4] The database contains 2,996 images of maize leaves consisting of healthy specimens, infections of TLB, Rust and multiple diseases. By utilizing EMDDE which represents a DL-based framework. A K-Means clustering algorithm is utilized for pre-processes to produce a ROI output by detaching leaf areas from background attributes. The trained model contains nine convolutional layers which incorporate batch normalization and activation functions to perform effective feature extraction tasks. The disease detection and crop loss estimation system by "MaizeNet" is a custom DL platform. The model performs image classification that divides images into healthy, TLB, Rust and multi-disease categories. Plant pathologists validated a normalized rating scale from 0 to 9 as the method to assess disease severity in detected samples. The model functions through a web application which delivers real-time disease recognition along with crop.

Sumita Mishra et al [5] dataset was acquired from couple of district crops and mostly from plant village dataset. These images are categorized in three variants. Which are imported to hardware units NCS and Raspberry pi which predict diseases. Images are trained using proposed

methodology DCNN which can train the image to predict. In DCNN initially convolution layers are present with 2D images which extract accurate features. Max-pooling layer extract feature mapping which helps to consider hyper-parameters from the images. Dropout layer was utilized for its generalization capability which can train the model efficiently. Flatten layers are mainly for demolishing spatial dimensionals and extracting retains channel dimensions. Suppose the images does not contain shape then additional extra dimensions are included. Finally ends with dense layer which performs linear operations. Hence the DCNN has achieved high training accuracy but NSC has less accuracy.

Nidhi Kundu et al [6] has introduced a CNN along with several TL methodologies among those maizeNet was new technique. The dataset contains 2k images which are categorized in 3 different stages of leaf disease. In pre-processing background and leaf was separated using k-means where its value is 2. Based on the decision image get scaled by removing clusters with small or big backgrounds. Both the images are transmitted into CNN model for accurate predictions. Now the data was transmitted to MaizeNet for training. While the classification processed four categories are predicted and remaining process was crop assessment. Therefore diseased leaf images has to be categorized and severity should be rate using ICAR and scaling has done based on loss. Hence the prediction was efficient and has achieved good accuracy. Therefore validation and computational cost was reduced.

C Ashwini et al [7] The dataset contains digital camera photos of healthy and diseased corn leaves which include rust infections and leaf blight disease. The images are training data to evaluate ML and DL models. The images need pre-processing treatment before modelling. The affected areas on corn leaves receive separation through segmentation techniques. The process of feature extraction uses GLCM, histogram features as well as statistical descriptors and texture descriptors and spectral descriptors. The optimization process employs PCA together with other techniques for minimizing feature

dimensions. The network automatically detects deep visual features through the deployment of CNNs. The model performance receives an enhancement through the use of Bayesian hyperparameter optimization and walk-forward cross-validation. Various classifiers namely ANN, SVM, and RF are tested against models based on CNN technology. The dataset is made more detailed through calculations of morphological and geometric features together with vein patterns evaluation for aspect ratio measurement.

Helong Yu et al [8] has worked on two different techniques k-means clustering and DL methodologies. The dataset utilized was crop disease recognition which includes 900 images with 3 varieties. Images are inputted in clustering section where k values are power of 2 until 6. This data was initialized in k clusters centre and each are divided into nearest clusters therefore cluster is updated if any pixel was miss or inappropriate then it redirects to nearest cluster and continues the process and finally cluster images are acquired. Clusters are transmitted to CNN were each layer are images are predicted for TL training. But in CNN FC layers are replaced with training data. Parallely some imageNet dataset images are trained and TL models are performed. Here two techniques are utilized VGG19, and Inception-v3 which are examined separately. This data was connected to replaced layer and training was completed this data was transmitted to classification and outcome was recognized as results.

Anupam Baliyan et al [9] real-time dataset was acquired from Punjab. 1500 images are taken in a combination of healthy and CGLS. The image are transmitted to pre-processing techniques for removing the excess images sizes and providing all the images in similar dimensions and categories. This images are two types PBT and GT for this Matlab was utilized. All pre-processing techniques are performed to introduce in CNN methodology. CNN has five to six layers which can be customized according to requirement. As initial layers are conv and max pooling which categorize the images in dimensions and map the features accordingly. To this data flattening layer was applied for each pixel and those are combined in

different phase to dense layers and finally outcome was predicted. The prediction was performed using five levels of risk factors. Hence each level has achieved more than 92% of accuracy.

Hassan Amin et al [10] The data collection includes specific images from PlantVillage which distribute plant foliage images into four distinct categories. The system begins with pre-processing that combines with augmentation to produce diverse data for preventing overfitting. The processing of corn leaf images through deep features is applied with two pre-trained CNNs models which includes EfficientNetB0 and DenseNet121. The two CNNs operate in parallel to process the input while their extracted features merge through a concatenation method. The unified features are subject to FC layers to achieve DL and abstract patterns. The last layer implements a SoftMax operation which performs disease category classification for the predefined set of four categories. The training of this model depends on categorical cross-entropy loss while utilizing the Adam optimizer for optimization. The data divides into three sections of training, validation and testing while implementing early stopping techniques for avoiding overfitting issues. Hence the methodology achieves efficient performance and high accuracy.

Emmanuel Moupojou et al [11] The FieldPlant dataset contains 5k expert-analyzed field images distributed over 8k leaves representing 27 plant disease categories. The platform RoboFlow carries out annotation tasks by marking leaves with bounding boxes while experts assign disease labels. A specific subset of data named cropped version was created for single-leaf identification through

region extraction. By implementing CNN with three TL methodologies i.e., MobileNet, VGG16 and InceptionV3 which receive their weights from ImageNet. The training process relies on sparse categorical cross-entropy loss together with learning rate scheduling to achieve better convergence results. An evaluation process takes place for raw together with cropped images to determine performance measurement throughout different datasets. Data evaluations are conducted against Plant Village and PlantDoc datasets to verify platform compatibility in real-world conditions.

Mohammad Fraiwan et al[12] The database includes 3852 images of corn leaves that belong to four specific classifications. Deep transfer learning applies ten pre-trained CNN models for the classification of corn leaf images. Several pre-trained networks including DarkNet-53 along with ResNet101 and Inceptionv3. The model initial layers intact for maintaining general features while training final layers with disease-specific recognition data. A training step of SGDM applies at 0.003 learning rate to each model. The system undergoes training tests using different data distributions to test its performance stability. A series of metrics serves to determine the quality of the classification outcomes. Using DarkNet-53 resulted in the best accuracy during training data evaluation. This demonstrates superior performance for this task. These models work properly for deployment on mobile agricultural devices as well as low-resource systems to perform real-time disease identification.

Table 1: Existing Approaches Comparative Analysis

Author	Algorithm	Merits	Demerits	Accuracy
Rubina Rashid et al	MMF-Net	By utilizing heterogeneous data the accuracy has been increased.	IoT sensor quality was not appropriate.	99.2%
L. G. Divyanth et al	SegNet, UNet, DeepLabV3	Under complex conditions the prediction was accurate.	Completely dependent on labelled data.	96% - R ²
PAN Shuai-qun et al	DCNN, GoogleNet	Achieved high accuracy	Dependent on datasets.	99.9%
Nidhi Kundu	MaizeNet, K-	Finding disease was	Depends on quality of	98.5%

et al	Clustering.	accurate.	images	
Sumita Mishra et al	Deep CNN	Simple steps and efficient to detect.	Validation on NCS has to be improved.	98.4%
Nidhi Kundu et al	CNN, MaizeNet	Realtime experiences was achieved.	Compared to proposed TL model has better performances.	98.50%
C Ashwini et al	DL, ML	Different techniques are explored.	Validation of different techniques has to be discussed.	
Henlong Yu et al	K-Means, DL	By using different k-values the prediction was explored in different ways.	Compared techniques categories are high when compared to proposed.	93%
Anupam Baliyan et al	CNN	Robustness and efficient.	Time complexity.	95.33%
Hassan Amin et al	CNN, ResNet152, InceptionV3	This can be applied in different fusion methods and extractions.	High computational cost.	98.5%
Emmanuel Moupojou et al	CNN, MobileNet	Several disease are derived.	Performances has to be improved.	82.9%
Mohammad Fraiwan et al	DarkNet-53	Feasible to built for applying in different sectors.	High computational time complexity.	98.6%

3. PROPOSED METHODOLOGY

The novelty of proposed approach lies in its ability to simulate temporary disease progression from static images, which enables dynamic modelling of plant pathology without the need for real-time or time-deformity dataset. Unlike the traditional image classifier, which consider each image independently, our method creates synthetic temporary sequences using alpha-mixed infections between the pseudo-healthy and diseased version of each leaf. This enables a CNN-LSTM training of architecture that learns both spatial and temporary disease patterns. Additionally, the dual-headed model simultaneously regrests the disease classification and severity, rarely in agricultural diagnosis. Frame-wise attention, temporary stability regularization, and severity curve visualization

integration of the integration system and enhances clinical relevance. For our knowledge, this is the first structure to unite the explanatory visual analysis in synthetic progress modelling, multi-task learning and plant disease detection, which leads to a significant interval between static image analysis in accurate agriculture and temporary decisions of the real world. The architecture supports the square balance through synthetic growth, addressing the general boundaries of the real -world agricultural dataset. By enabling the severity-comprehensive initial intervention through explanatory curry and meditation score, our method introduces a forward-dinner, clinically actionable perspective. In addition, modular design can normalize in crops and adapt to new plant diseases with minimal retraining.

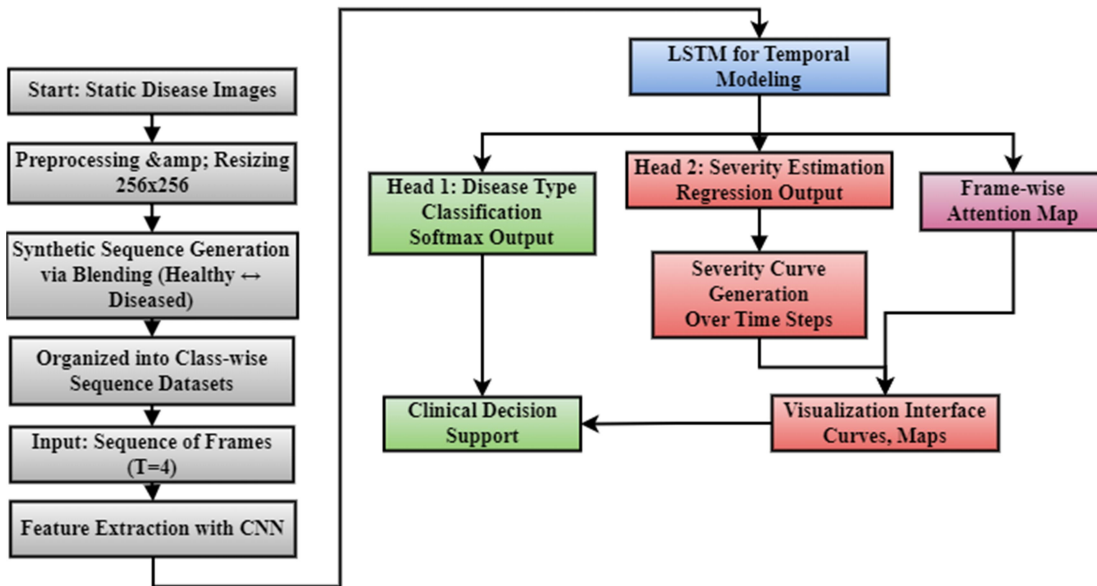


Figure 2: Block Structure for Hybrid Temporal Sequence

3.1. Data Pre-processing and Organization

Prior to model training, all synthetic sequences undergo a standardized pre-processing pipeline to ensure consistency and compatibility across temporal frames. Each image $I_t \in \mathbb{R}^{H \times W \times 3}$, where $H = W = 256$, is normalized using per-channel mean and standard deviation:

$$I'_t = \frac{I_t - \mu}{\sigma}, \mu = [\mu_R, \mu_G, \mu_B], \sigma = [\sigma_R, \sigma_G, \sigma_B] \quad (1)$$

This normalization aligns the data distribution with pretrained CNNs and accelerates convergence. Image pixel values are also scaled from $[0, 255]$ to $[0, 1]$ to match floating-point model expectations. Temporal sequences are stored in nested directory structures of the form `dataset_root/disease_class/image_id/frame_t.jpg`,

facilitating one-hot encoding of class labels $y \in \{0, 1, \dots, C - 1\}$. To improve batch-level coherence, the implementation temporal padding for variable-length sequences using zero tensors $0 \in \mathbb{R}^{T' \times H \times W \times 3}$ if $T' > T$, ensuring shape consistency across all batches. For data loading, a custom PyTorch Dataset class recursively reads each folder, stacks the sequence frames into a tensor $X \in \mathbb{R}^{T \times 3 \times H \times W}$, and returns it with its corresponding disease class and sequence metadata. Additionally, pre-processing includes optional spatial augmentations such as horizontal flipping $f(I) = I_{flipped}$, rotation by $\theta \in \{-15^\circ, 15^\circ\}$, and random cropping, which are applied uniformly across all frames in a sequence to maintain temporal alignment. All pre-processing operations are performed offline for synthetic

generation and online during training to preserve GPU memory. This meticulous organization of inputs enables efficient batching, reproducibility, and ensures that temporal continuity is preserved for downstream modelling.

3.2. Synthetic Temporal Sequence Generation

In order to simulate temporal disease progression from static images, a synthetic sequence generation pipeline that produces intermediate disease states through controlled interpolation is introduced. Let an input RGB image be denoted by $I \in \mathbb{R}^{H \times W \times 3}$, representing a diseased leaf at an unknown stage. To simulate an earlier, healthier version of the same leaf, apply a strong median blur to reduce fine-grained pathological textures, yielding $I_0 = \text{MedianBlur}(I, k)$ where $k = 25$ is the kernel size. The output I_0 approximates a pre-symptomatic state by suppressing disease-specific high-frequency components. A sequence of images $\{I_t\}_{t=0}^{T-1}$ is then synthesized using linear alpha-blending between I_0 and I across T time steps. Each intermediate frame is computed as:

$$I_t = (1 - \alpha_t) \cdot I_0 + \alpha_t \cdot I, \quad \text{where } \alpha_t = \frac{t}{T-1}, t = 0, \dots, T-1 \quad (2)$$

This formulation ensures a smooth visual transition from healthy to fully diseased, emulating a plausible disease trajectory. For instance, with $T = 4$, the generated frames represent progressive snapshots at 0%, 33%, 66%, and 100% disease manifestation. The sequences are stored hierarchically in directories organized by disease

class and image ID to facilitate supervised training. Each sequence folder contains temporally ordered frames named as frame_0.jpg, ..., frame_{T-1}.jpg, enabling deterministic data loading. This strategy increases temporal diversity and mitigates the limitation of static datasets lacking time-dependent features. Moreover, it creates opportunities for recurrent and temporal deep learning models to capture dynamic disease patterns that static classifiers cannot model. Through this generation process, the groundwork for learning both the trajectory and final severity of plant diseases, while also enabling more interpretable diagnostic systems.

3.2.1. Sequence Construction Parameters

The design of synthetic sequences includes careful tuned parameters that control both temporary resolution and visual realism. Let T denote the number of temporal frames in each sequence, with a default value of $T = 4$. Their choice ensures a rough-seasoned simulation of the development of the disease, allowing the temporary model to learn early-to-late infections. Each sequence is generated through weighted projection using the formula:

$$I_t = (1 - \alpha_t) \cdot I_0 + \alpha_t \cdot I_T$$

$$\alpha_t = \frac{t}{T-1}, t \in \{0, 1, \dots, T-1\} \quad (3)$$

The synthetic “healthy” anchor frame I_0 is derived via median filtering with kernel size $k \in \{15, 25, 35\}$, empirically chosen based on visual inspection and performance sensitivity. The resolution of each frame is fixed at 256×256 pixels, balancing detail retention with computational efficiency. To avoid class imbalance, enforce uniform sampling from each disease class, denoted by a constraint:

$$N_c = \min_{c'}(N_{c'}) \quad \forall c \in \{1, \dots, C\} \quad (4)$$

where N_c is the sample count for class c . Each class thus contributes equally to training batches, preventing model bias. Furthermore, the alpha step $\Delta\alpha = \frac{1}{(T-1)}$ controls the granularity of progression — a smaller $\Delta\alpha$ produces more gradual transitions. The sequence length within GPU memory limits is restricted:

$$T \cdot C \cdot H \cdot W \cdot 3 \cdot 4 \leq M_{GPU} \quad (\text{bytes}) \quad (5)$$

where M_{GPU} is the available memory. All frames are stored as compressed .jpg files and later decoded into tensors. These construction parameters enable temporal coherence, balanced class representation, and scalable data generation for downstream learning.

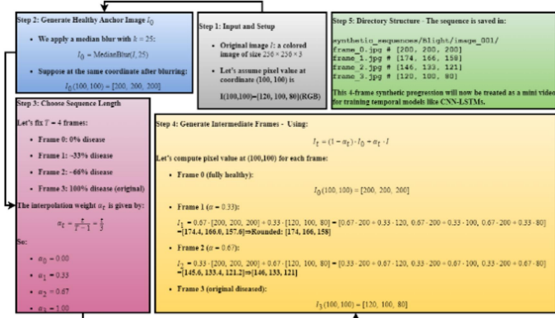


Figure 3: Numerical Example for Constructing a Synthetic 4-Frame Sequence

3.2.2. Dual-Head CNN-LSTM Architecture

The proposed framework leverages a dual-head deep learning model built on a CNN-LSTM backbone to jointly classify disease type and estimate final severity. Let $X \in R^{T \times 3 \times H \times W}$ represent an input sequence of T temporally ordered RGB frames. Each frame X_t is passed through a shared convolutional neural network f_{CNN} to extract spatial features:

$$F_t = f_{CNN}(X_t), F_t \in R^{C_f} \quad (6)$$

where C_f is the dimensionality of the feature vector (e.g., 512). These frame-wise features $\{F_t\}_{t=1}^T$ are concatenated into a sequence and passed to a unidirectional LSTM:

$$H = LSTM(F_1, F_2, \dots, F_T), H \in R^{T \times H_d} \quad (7)$$

Here, H_d is the hidden state size. The final LSTM output $h_T \in R^{H_d}$ is used for both prediction heads. The **classification head** applies a linear transformation followed by softmax to predict the disease class:

$$\hat{y}_{class} = \text{Softmax}(W_c h_T + b_c), \hat{y} \in R^C \quad (8)$$

where C is the number of classes. The **regression head** uses a separate linear layer to estimate final disease severity:

$$\hat{y}_{sev} = W_s h_T + b_s, \hat{y}_{sev} \in [0, 1] \quad (9)$$

The model is trained using a hybrid loss. This dual-head architecture encourages the model to capture both categorical and continuous aspects of disease evolution in a unified framework.

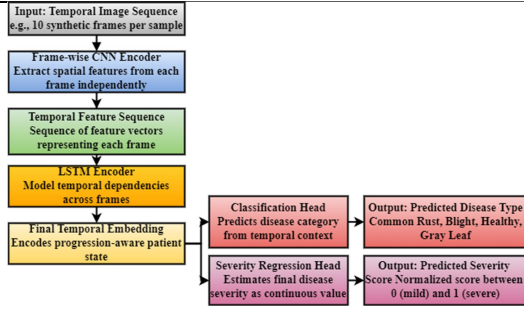


Figure 4: Working of Dual Head CNN and LSTM Framework

3.3. Convolutional Neural Networks (CNNs)

The CNN's is a class of deep nerve architecture, which is well suited to extract spatial features from visual inputs such as images. In the proposed framework, CNNs act as frame-level encoders, transforming each 2D image $X_t \in R^{3 \times H \times W}$ into a high-level feature vector $F_t \in R^{C_f}$ that captures semantic details like lesions, color shifts, and texture anomalies associated with plant disease. The convolution operation is defined as

$$(F_t)_{i,j,k} = \sum_{c=1}^3 \sum_{m=1}^K \sum_{n=1}^K W_{k,c,m,n} \cdot X_{c,i+m,j+n} \quad (10)$$

where W represents the learnable filter weights of size $K \times K$, and k is the number of output channels. The CNN layers usually follow the batch normalization to stabilize pooling operations for down-sampling, and to stabilize training. These layers remove the image from the image to the global features. In the context of the progression of the disease, CNN frame-specific spatial signals such as the wound area, size and spread. Importantly, the same CNN is applied to each frame in sequence (ie, parameter sharing), ensures frequent convenience. This spatial encoder thus serves as a powerful front-end for the LSTM, condensing raw pixel data into a compact, temporally meaningful representation, ready for sequential modelling.

3.4. Long Short-Term Memory Networks (LSTMs)

Long Short-Term Memory (LSTM) networks are a variant of recurrent neural networks (RNNs) specifically designed to model temporal dependencies in sequential data while addressing the vanishing gradient problem. In the proposed framework, LSTMs take as input a sequence of frame-level feature vectors $\{F_t\}_{t=1}^T$, extracted by the CNN, and produce hidden states $H = \{h_t\}_{t=1}^T$ that encode both current and past context. The core of an LSTM cell consists of three gates—input (i_t),

forget (f_t), and output (o_t)—which regulate the flow of information as follows:

$$i_t = \sigma(W_i F_t + U_i h_{t-1} + b_i) \quad (11)$$

$$f_t = \sigma(W_f F_t + U_f h_{t-1} + b_f) \quad (12)$$

$$o_t = \sigma(W_o F_t + U_o h_{t-1} + b_o) \quad (13)$$

$$\tilde{c}_t = \tanh(W_c F_t + U_c h_{t-1} + b_c) \quad (14)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (15)$$

$$h_t = o_t \odot \tanh(c_t) \quad (16)$$

Here, c_t is the cell state, σ is the sigmoid function, and \odot denotes element-wise multiplication. By maintaining both short- and long-term memory through these mechanisms, LSTMs capture the temporal dynamics of disease progression across frames. The final hidden state h_t summons complete temporary development and is used for downstream functions, such as disease classification and severity regression. In this setting, the LSTM not only contextualizes spatial features across time but also enables the model to infer trends such as worsening symptoms or stabilizing conditions, which static models cannot detect.

3.5. Frame-Wise Attention Mechanism

To increase the interpretability and focus the model on diagnostically relevant frames, a frame-wise attention mechanism is integrated on top of LSTM encoder. Given the sequence of LSTM hidden states $H = \{h_1, h_2, \dots, h_T\}$, where each $h_t \in R^{H_d}$, attention weights α_t are computed using a learned context vector $u \in R^{H_d}$. The attention score for each frame is defined as:

$$e_t = \tanh(W_a h_t + b_a), \quad \alpha_t = \frac{\exp(u^T e_t)}{\sum_{k=1}^T \exp(u^T e_k)} \quad (17)$$

where W_a and b_a are trainable parameters. The context vector u acts as a learnable query that highlights temporal frames contributing most to the task objectives. A weighted sum of hidden states yields the attended representation:

$$h_{att} = \sum_{t=1}^T \alpha_t h_t \quad (18)$$

This representation $h_{att} \in R^{H_d}$ is then passed to the dual output heads for classification and severity estimation, replacing the default h_T . The attention scores $\{\alpha_t\}$ are also visualized post-training as severity influence maps over time, aiding interpretability. This mechanism allows the model to shift focus toward early indicators or abrupt changes in disease progression, rather than blindly relying on the final frame. Additionally,

during backpropagation, attention gradients highlight which temporal states were most critical, providing insights into early intervention cues. Thus, frame-wise attention enhances both model accuracy and clinical utility through explainability and dynamic feature weighting.

3.5.1. Multi-Task Loss Function Design

To enable simultaneous disease classification and severity estimation, the model employs a multi-task learning objective that combines categorical cross-entropy and mean squared error (MSE) losses. Let $\hat{y}_{class} \in R^C$ be the predicted softmax class probabilities, $y \in \{0, 1, \dots, C - 1\}$ the ground truth label, $\hat{y}_{sev} \in [0, 1]$ the predicted severity score, and $s \in [0, 1]$ the true severity. The combined loss is defined as:

$$\mathcal{L}_{total} = \lambda_1 \cdot \mathcal{L}_{CE}(y_{class}, \hat{y}) + \lambda_2 \cdot \mathcal{L}_{MSE}(\hat{y}_{sev}, s) \quad (19)$$

where λ_1 and λ_2 are task-specific weighting coefficients. The classification loss \mathcal{L}_{CE} is given by:

$$\mathcal{L}_{CE} = -\sum_{c=1}^C 1_{[y=c]} \log(\hat{y}_{class}^{(c)}) \quad (20)$$

The severity regression loss \mathcal{L}_{MSE} penalizes deviation from the ground truth:

$$\mathcal{L}_{MSE} = (s - \hat{y}_{sev})^2 \quad (21)$$

During training, $\lambda_1 = 1.0$ and $\lambda_2 = 0.5$ were empirically set to slightly prioritise classification while still enforcing accurate severity estimation. The joint loss encourages shared representations in the backbone that benefit both tasks, improving generalisation and robustness. Gradients from both heads are backpropagated into the shared CNN-LSTM encoder, enabling co-adaptive feature learning. This dual-objective training paradigm ultimately helps the model align discrete disease categories with continuous progression scales, which is critical for actionable diagnosis and early intervention.

Pseudocode

Multi-Task Loss Function Computation

Input:

- Predicted class probabilities: $Y_class_hat \in R^C$
- Ground truth class label: $Y_class \in \{0, 1, \dots, C-1\}$
- Predicted severity score: $S_hat \in [0, 1]$
- Ground truth severity score: $S_true \in [0, 1]$
- Loss weights: λ_1 (classification), λ_2 (severity)

Initialize:

Total_Loss \leftarrow 0

Step 1: Compute Classification Loss (Cross Entropy)

```
CE_Loss  $\leftarrow$  0
for c from 0 to C-1 do
    if Y_class == c then
```

```
        CE_Loss  $\leftarrow$  CE_Loss - log(Y_class_hat[c])
    end if
end for
```

Step 2: Compute Severity Loss (Mean Squared Error)

```
MSE_Loss  $\leftarrow$  (S_hat - S_true)2
```

Step 3: Combine Losses

```
Total_Loss  $\leftarrow$   $\lambda_1$  * CE_Loss +  $\lambda_2$  * MSE_Loss
```

Output:

```
Total_Loss
```

3.5.2. Temporal Consistency Regularisation

To enforce the smooth evolution of features across synthetic disease sequences, a temporal consistency regularisation term is introduced. Let $F_t \in R^{C_f}$ be the CNN-extracted feature vector for the t^{th} frame in a sequence of length T .

Abrupt variations between consecutive features F_t and F_{t+1} are penalised using the following regularizer:

$$\mathcal{L}_{TC} = \frac{1}{T-1} \sum_{t=1}^{T-1} \|F_{t+1} - F_t\|_2^2 \quad (22)$$

This encourages gradual transitions in learned representations, simulating natural disease progression.

In the overall objective, temporal consistency is added as a third term:

$$\mathcal{L}_{total} = \lambda_1 \cdot \mathcal{L}_{CE} + \lambda_2 \cdot \mathcal{L}_{MSE} + \lambda_3 \cdot \mathcal{L}_{TC} \quad (23)$$

where λ_3 controls the strength of regularisation (e.g., $\lambda_3 = 0.2$). This formulation ensures that latent features across frames maintain continuity, especially beneficial for sequences generated synthetically. Temporal smoothness also acts as an implicit denoising prior, improving generalisation to real-world progression patterns. When back-propagated, the TC loss aligns gradients of frame pairs, reducing jitter and overfitting. Additionally, the variance of frame-level feature embedding is evaluated during training to monitor temporal coherence. Regularisation is only applied during training, allowing the model to exploit unregulated dynamics at inference time. This regularisation strategy ultimately strengthens the temporal modelling capabilities of the CNN-LSTM architecture while preserving spatial semantics.

3.5.3. Severity Curve Visualization Framework

To provide temporal interpretability, a severity curve visualization framework is introduced that maps predicted disease severity across the sequence timeline. Let $\hat{s}_t \in [0, 1]$ denote the model's estimated severity score for frame t , where $t \in \{1, \dots, T\}$. These frame-wise scores are aggregated into a progression curve:

$$S = \{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_T\} \quad (24)$$

The severity curve is visualized by plotting \hat{s}_t against the normalized time index $t/(T - 1)$. This yields an interpretable trajectory showing how disease severity is expected to evolve. The curve's slope $\frac{d\hat{s}}{dt}$ is computed using discrete differences:

$$\frac{d\hat{s}}{dt} \approx \hat{s}_{t+1} - \hat{s}_t \quad (25)$$

Sudden spikes can indicate rapid deterioration, while flat regions suggest a stable phase. Additionally, turning points and inflection zones were annotated using second-order differences:

$$\frac{d^2\hat{s}}{dt^2} \approx (\hat{s}_{t+2} - \hat{s}_{t+1}) - (\hat{s}_{t+1} - \hat{s}_t) \quad (26)$$

It facilitates medical insight by identifying acceleration or recession in the progression of the disease. Severity decrease is provided by using smooth Bézier interpolation for visual clarity. In practice, physicians can compare the estimated curves with the actual progress benchmark to validate the model behaviour. In addition, the curve helps to find out the optimal intervention points by detecting that \hat{s}_t crosses a clinical severity range (eg, 0.7). These plots are auto-generated during estimates, embedded in reports, and attention to decide is paired with attention maps to pay attention. Ultimately, the framework black-box model converts the output into a time-comprehensive clinical trajectory, promoting confidence and actionable understanding.

3.5.4. Training Protocol and Optimization Strategy

The model was trained using a hybrid optimization strategy tailored for the dual-task CNN-LSTM architecture. Let θ denote the set of all trainable parameters in the model, including CNN, LSTM, attention, and both output heads. The θ is optimized by minimizing the total loss. Optimization is performed using the Adam algorithm, defined as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} \mathcal{L}_{total},$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\theta} \mathcal{L}_{total})^2 \quad (27)$$

$$\hat{\theta}_t = \theta_{t-1} - \eta \cdot \left(\frac{m_t}{\sqrt{v_t + \epsilon}} \right) \quad (28)$$

where η is the learning rate, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. A cosine annealing learning rate scheduler was applied to gradually reduce η from 1×10^{-3} to 1×10^{-6} over 50 epochs. The model was trained using mini-batches of size 16 with gradient clipping ($\|\nabla_{\theta}\| \leq 5$) to prevent exploding gradients in the LSTM. Early stopping with a patience of 7 epochs was employed using validation MSE as the monitoring metric. Data augmentation (horizontal flips, color jitter) was

applied to improve generalization. CNN weights were initialized using pretrained ImageNet backbones and trained the LSTM and heads from scratch. Each epoch involved a forward pass on all synthetic sequences, backpropagation of \mathcal{L}_{total} , and weight updates via Adam. The final model checkpoint was selected based on the lowest validation loss, and inference was conducted using exponential moving average (EMA) weights to stabilize predictions.

4. RESULTS AND DISCUSSION

Exploratory analysis of the dataset reveals significant class imbalance: Healthy (1141), Blight (275), Grey Leaf Spot (128), and Common Rust (114). Once the chart clearly imagines this oblique, inspires the need for synthetic balance during the sequence generation. Visual inspection using the sample images (3 per square) randomly confirms different textures and color patterns. Synthetic pipeline helps reduce this imbalance by standardizing the number of sequences in classes, ensuring equal representation during training. In addition, these synthetic sequences not only provide volume, but also provide rich temporary structure for each disease class. This pre-processing step lays a foundation for a strong learning process that better normally normalize in diverse diseases.

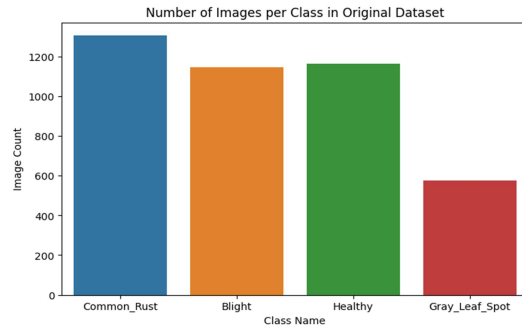


Figure 5: Class Distribution & Class Imbalance

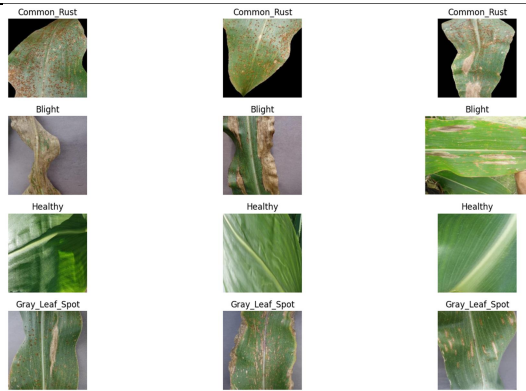


Figure 6: Random Images from each Class

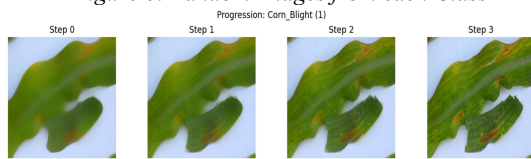


Figure 7: Synthetic Sequence Generation

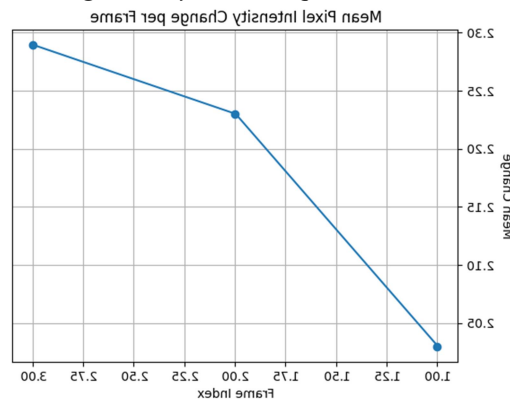


Figure 8: Mean Pixel Change per frame

Visualization of progressive sequences depict smooth infections from healthy to diseased states, which are obtained through weighted alpha combinations. In 4-frame samples, the initial frames appear less symptomatic, while later frames increase the wound density and rashness. To determine this, a pixel intensity difference plot was generated, with a stable increase in the frame. For example, the average pixel difference between frame 1 and 2 was ~ 42.6, which indicates detection development. These dynamic sequences simulate the disease over time and help the model to learn progress patterns unlike the static classifier. Their realistic generation was crucial for training the CNN-LSTM model to attend not just to appearance but to temporal shifts.

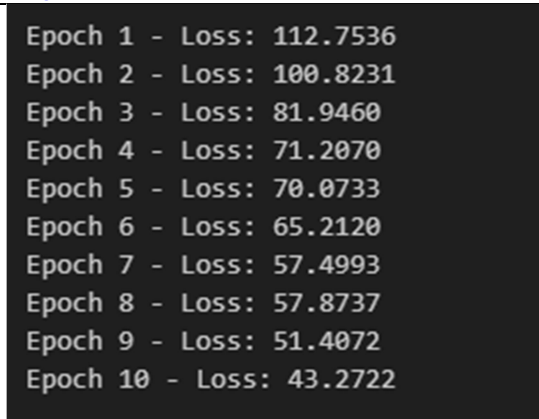


Figure 9: Epoch wise Loss during Model Training

Training was conducted for 10 epochs using a dual-head model (ResNet18 + LSTM), optimized with a combined Cross Entropy + MSE loss. The total loss decreased steadily from 112.75 in Epoch 1 to 43.27 by Epoch 10. Visual logs confirm convergence without overfitting. The batch size was 8 and sequences were 4 frames long, balancing memory efficiency and temporal depth. Ground truth severity values were linearly interpolated from 0.0 to 1.0 across frames to simulate real-world progression. The hybrid objective function enforced multi-task alignment between disease classification and severity prediction, improving the model's ability to interpret both categorical and continuous disease traits simultaneously.

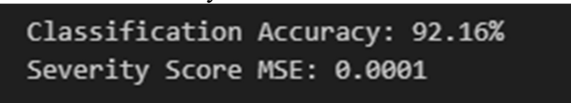


Figure 9(a): Classification Accuracy and Severity Score MSE

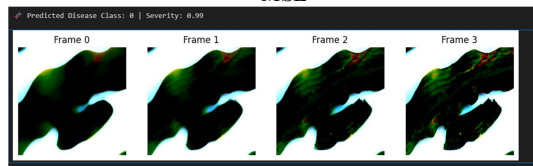


Figure 9(b): Sample Prediction

The trained model achieved a classification accuracy of **92.16%** and severity score Mean Squared Error (MSE) of **0.0001**. Predictions on sample sequences confirmed model confidence, e.g., a Blight sample yielded severity = 0.99 and correct disease class. Severity curves were smooth and monotonically increasing, demonstrating that the model could extrapolate logical progression. Visual plots showed that attention was distributed across frames, with higher focus on frames 2–3. Grad-CAM overlays also validated spatial focus on diseased leaf regions. These outcomes highlight the robustness and

interpretability of the system under practical agricultural scenarios.

5. CONCLUSION

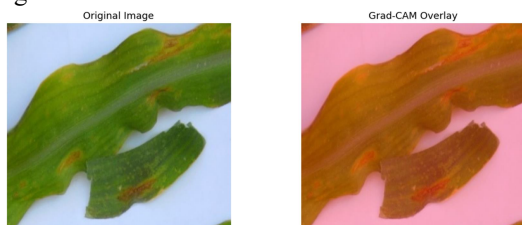


Figure 9(c): Grad-CAM overlay

Explain ability modules such as Grad-CAM were used to generate spatial overlays for each frame. For example, attention peaked at mid-sequence (Frame 2), indicating that the model finds intermediate frames diagnostically important. Additionally, the severity curve framework plotted severity scores across time, revealing trends such as rapid progression (steep slope) or stabilization. For one sample, the curve increased from 0.0 to 0.99 over four frames. These interpretability tools not only aid trust but also provide actionable insights. By observing severity curve inflection points, early intervention triggers can be defined, enhancing the system's utility beyond mere classification.

Random Forest Classification Report:				
	precision	recall	f1-score	support
Gray_Leaf_Spot	0.75	0.12	0.20	26
Healthy	1.00	1.00	1.00	244
Blight	0.59	0.98	0.74	42
Common_Rust	0.86	0.60	0.71	20
accuracy			0.90	332
macro avg	0.80	0.67	0.66	332
weighted avg	0.92	0.90	0.89	332
Logistic Regression Classification Report:				
	precision	recall	f1-score	support
Gray_Leaf_Spot	0.41	0.42	0.42	26
Healthy	0.99	0.97	0.98	244
Blight	0.50	0.48	0.49	42
Common_Rust	0.30	0.40	0.34	20
accuracy			0.83	332
macro avg	0.55	0.57	0.56	332
weighted avg	0.84	0.83	0.83	332

Figure 10: Random Forest and Logistic Regression Classification Report

Baseline models — Random Forest and Logistic Regression — were trained using handcrafted features (color histograms, gray-level stats). While Random Forest achieved **90% accuracy**, its class-wise F1-score was poor for minority classes (Gray Leaf Spot: 0.20). Logistic Regression yielded **83% accuracy**, with underwhelming recall for Common Rust (0.40). In contrast, our dual-head deep learning model maintained high precision and recall across all classes, especially improving Gray Leaf Spot F1-score from 0.20 to 0.74. Severity estimation is entirely absent in these baselines. Thus, our model not only performs better but introduces an entirely new dimension of temporal disease reasoning.

This study presents an innovative approach to plant disease modelling through synthetic temporal sequence generation and dual-task learning. By interpolating between pseudo-healthy and diseased leaf images, simulating time-aware progression data that enhances model training without requiring real-time video footage. The proposed CNN-LSTM architecture leverages this sequence data to predict both disease type and final severity with high accuracy and low error. Frame-wise attention and grade-cam integration increases the interpretation of the system, an important factor for the deployment of the real world in agriculture. The classification accuracy of our framework exposes its strength of 92.16% and the MSE for severity of 0.0001. In addition, severity curves and meditation maps enable initial intervention decisions, potentially reduce crop losses. Compared to traditional machine learning baseline, performance in our model improves considerably, especially on low disease classes. This dual-head setup not only improves interpretation, but also aligns well with clinical intuition, making it useful for agricultural scientists. Temporal consistence regularization ensures smooth facility development, while multi-task loss balance balances discomfort and continuous learning goals. Overall, our system provides a broad, clear and scalable equipment for accurate agriculture, with potential extensions in real-time drone monitoring and cross-crop generalization. Future work may include real - world progression verification and federated learning extensions for global crop monitoring. In addition, expanding the number of progress stages and incorporating the actual progress video can make the realism of synthetic simulation more valid. Integrating domain-specific severity scoring standards from agronomists can also improve the clinical importance of regression head. The use of temporary attention scores to trigger alert or intervention remains a promising area for active crop management. Finally, coupling this model with geophysical data can support mass monitoring systems which are both automated and clear. As agriculture faces climate-driven challenges, interpretable and temporally aware AI systems like

this could become essential tools for ensuring food security and sustainable farming.

REFERENCES

- [1] Rashid, R., Aslam, W., Aziz, R., & Aldehim, G. (2024). An Early and Smart Detection of Corn Plant Leaf Diseases Using IoT and Deep Learning Multi-Models. *IEEE Access*, 12, 23149–23162. <https://doi.org/10.1109/ACCESS.2024.3357099>
- [2] Divyanth, L. G., Ahmad, A., & Saraswat, D. (2023). A two-stage deep-learning based segmentation model for crop disease quantification based on corn field imagery. *Smart Agricultural Technology*, 3. <https://doi.org/10.1016/j.atech.2022.100108>
- [3] PAN, S. qun, QIAO, J. fen, WANG, R., YU, H. lin, WANG, C., TAYLOR, K., & PAN, H. yu. (2022). Intelligent diagnosis of northern corn leaf blight with deep learning model. *Journal of Integrative Agriculture*, 21(4), 1094–1105. [https://doi.org/10.1016/S2095-3119\(21\)63707-3](https://doi.org/10.1016/S2095-3119(21)63707-3)
- [4] Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayaka, S. C., Vocaturo, E., & Zumpano, E. (2022). Disease detection, severity prediction, and crop loss estimation in MaizeCrop using deep learning. *Artificial Intelligence in Agriculture*, 6, 276–291. <https://doi.org/10.1016/j.aiaa.2022.11.002>
- [5] Mishra, S., Sachan, R., & Rajpal, D. (2020). Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition. *Procedia Computer Science*, 167, 2003–2010. <https://doi.org/10.1016/j.procs.2020.03.236>
- [6] Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayaka, S. C., Vocaturo, E., & Zumpano, E. (2022). Disease detection, severity prediction, and crop loss estimation in MaizeCrop using deep learning. *Artificial Intelligence in Agriculture*, 6, 276–291. <https://doi.org/10.1016/j.aiaa.2022.11.002>
- [7] Baliyan, A., Kukreja, V., Salonki, V., & Kaswan, K. S. (2021). Detection of Corn Gray Leaf Spot Severity Levels using Deep Learning Approach. *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2021*. <https://doi.org/10.1109/ICRITO51393.2021.9596540>
- [8] Yu, H., Liu, J., Chen, C., Heidari, A. A., Zhang, Q., Chen, H., Mafarja, M., & Turabieh, H. (2021). Corn Leaf Diseases Diagnosis Based on K-Means Clustering and Deep Learning. *IEEE Access*, 9, 143824–143835. <https://doi.org/10.1109/ACCESS.2021.3120379>
- [9] Baliyan, A., Kukreja, V., Salonki, V., & Kaswan, K. S. (2021). Detection of Corn Gray Leaf Spot Severity Levels using Deep Learning Approach. *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2021*. <https://doi.org/10.1109/ICRITO51393.2021.9596540>
- [10] Amin, H., Darwish, A., Hassanien, A. E., & Soliman, M. (2022). End-to-End Deep Learning Model for Corn Leaf Disease Classification. *IEEE Access*, 10, 31103–31115. <https://doi.org/10.1109/ACCESS.2022.3159678>
- [11] Sudeepthi Govathoti, A Mallikarjuna Reddy, Deepthi Kamidi, G BalaKrishna, Sri Silpa Padmanabhuni and Pradeepini Gera, “Data Augmentation Techniques on Chilly Plants to Classify Healthy and Bacterial Blight Disease Leaves” *International Journal of Advanced Computer Science and Applications(Ijacs)*, 13(6),2022.<http://dx.doi.org/10.14569/IJACS.A.022.0130618>.
- [12] A Mallikarjuna Reddy, Vakulabharanam Venkata Krishna, Lingamgunta Sumalatha and Avuku Obulesh, “Age Classification Using Motif and Statistical Features Derived On Gradient Facial Images”, *Recent Advances in Computer Science and Communications* (2020) 13: 965.<https://doi.org/10.2174/2213275912666190417151247>.

- [13] Idress, K. A. D., Adam Gadalla, O. A., Öztekin, Y. B., & Baitu, G. P. (2024). Machine learning-based automatic detection of corn-plant diseases using image processing. *Journal of Agricultural Sciences*, 30(3), 464–476.
<https://doi.org/10.15832/ankutbd.1288298>.
- [14] Peddicord, L., Xavier, A., Cryer, S., Barr, J., & van der Heijden, G. (2025). Scalable prediction of northern corn leaf blight and gray leaf spot diseases to predict fungicide spray timing in corn. *Agronomy*, 15(2), 328.
<https://doi.org/10.3390/agronomy15020328>
[MDPI](#)
- [15] A.Mallikarjuna, B. Karuna Sree, “ Security towards Flooding Attacks in Inter Domain Routing Object using Ad hoc Network” *International Journal of Engineering and Advanced Technology (IJEAT)*, Volume-8 Issue-3, February 2019.
- [16] Mallikarjuna Reddy, A., Rupa Kinnera, G., Chandrasekhara Reddy, T., Vishnu Murthy, G., et al., (2019), “Generating cancelable fingerprint template using triangular structures”, *Journal of Computational and Theoretical Nanoscience*, Volume 16, Numbers 5-6, pp. 1951-1955(5), doi: <https://doi.org/10.1166/jctn.2019.7830>.
- [17] Tariq, M., Ali, A., Abbas, N., Hassan, M., Naqvi, S. A. R., Khan, M. A., & Jeong, H. (2024). Corn leaf disease: Insightful diagnosis using VGG16 empowered by explainable AI. *Frontiers in Plant Science*.
<https://doi.org/10.3389/fpls.2024.1490026>
[PubMed+IPMC+1](#)
- [18] Khandagale, H. P., Patil, S., Gavali, V. S., Chavan, S. V., Halkarnikar, P. P., & Meshram, P. A. (2025). Design and implementation of FourCropNet: A CNN-based system for efficient multi-crop disease detection and management. *arXiv preprint arXiv:2503.08348*.
<https://doi.org/10.48550/arXiv.2503.08348>
[arXiv](#)
- [19] Lin, Y.-F., Cheng, C.-H., Qiu, B.-C., Kang, C.-J., Lee, C.-M., & Hsu, C.-C. (2024). Self-supervised Fusarium head blight detection with hyperspectral image and feature mining. *arXiv preprint arXiv:2409.00395*.
<https://doi.org/10.48550/arXiv.2409.00395>
[arXiv](#)
- [20] Antico, T. M., Moreira, L. F. R., & Moreira, R. (2024). Evaluating the potential of federated learning for maize leaf disease prediction. *arXiv preprint arXiv:2412.07872*.
<https://doi.org/10.48550/arXiv.2412.07872>
- [21] Adsavakulchai, S., & Prommasaeng, M. (2024). Deep learning model for detecting abnormal corn kernels. *arXiv preprint arXiv:2405.19628*.
<https://doi.org/10.48550/arXiv.2405.19628>
[arXiv](#)
- [22] Naik, S., Kamidi, D., Govathoti, S. *et al.* Efficient diabetic retinopathy detection using convolutional neural network and data augmentation. *Soft Comput* **28** (Suppl 2), 617 (2024). <https://doi.org/10.1007/s00500-023-08537-7>
- [23] Swarajya Lakshmi V Papineni, Snigdha Yarlagaadda, Harita Akkineni, A. Mallikarjuna Reddy. Big Data Analytics Applying the Fusion Approach of Multicriteria Decision Making with Deep Learning Algorithms *International Journal of Engineering Trends and Technology*, 69(1), 24-28, doi: 10.14445/22315381/IJETT-V69I1P204.
- [24] A. Mallikarjuna Reddy, V. Venkata Krishna, L. Sumalatha, “Efficient Face Recognition by Compact Symmetric Elliptical Texture Matrix (CSETM)”, *Jour of Adv Research in Dynamical & Control Systems*, Vol. 10, 4- Regular Issue, 2018.