Machine Learning Based Abiotic Stress Assessment in Plants

A Thesis

Submitted in partial fulfilment of the requirements for the award of the degree of

DOCTOR OF PHILOSOPHY

by

Aswini Kumar Patra

Roll No: 186106001

Department of Biosciences and Bioengineering
Indian Institute of Technology Guwahati
Guwahati 781039, Assam, India
December 2025

DECLARATION

I do hereby declare that the matter embodied in this thesis entitled "**Machine Learning based Abiotic Stress Assessment in Plants**" is the result of work carried out in the Department of Biosciences and Bioengineering, Indian Institute of Technology Guwahati, India, under the supervision of **Prof. Lingaraj Sahoo**.

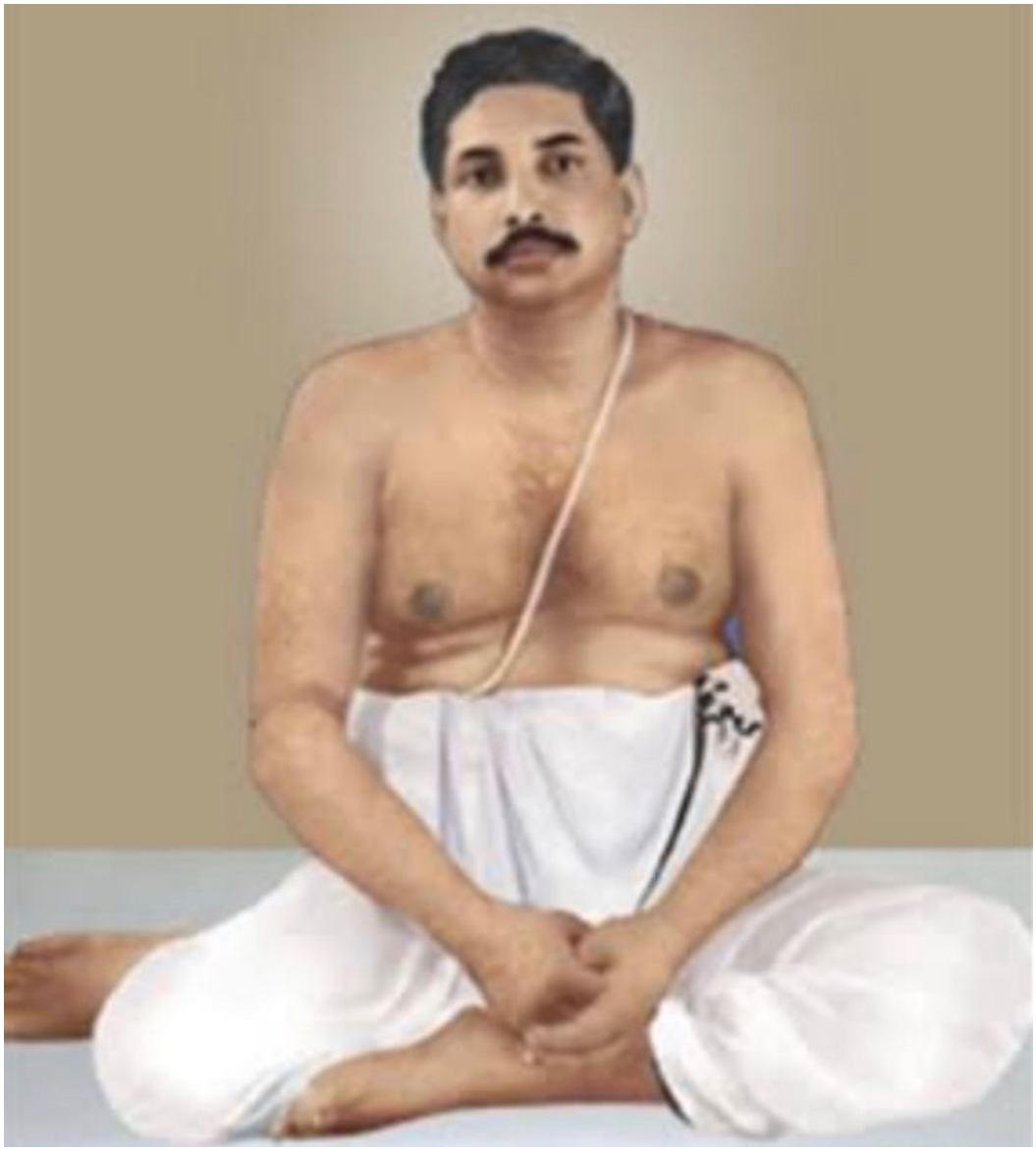
In keeping with the general practice of reporting of scientific observations, due acknowledgement has been made wherever the work described is based on the findings of other investigators.

Aswini Kumar Patra
Roll no. 186106001
Department of Biosciences and Bioengineering
Indian Institute of Technology Guwahati
Assam 781039, India

CERTIFICATE

It is certified that the work described in this thesis entitled "**Machine Learning based Abiotic Stress Assessment in Plants**" by Mr. Aswini Kumar Patra for the award of degree of Doctor of Philosophy is an authentic record of the results obtained from the research work carried out under my supervision in the Department of Biosciences and Bioengineering, Indian Institute of Technology Guwahati, India. The work embodied in this thesis has not been submitted elsewhere for a degree.

Prof. Lingaraj Sahoo
Thesis Supervisor
Department of Biosciences and Bioengineering
Indian Institute of Technology Guwahati
Assam 781039, India



This thesis is humbly dedicated to

my Guru, Param Dayal Param Premamaya Sree Sree Thakur,

His living embodiment, Param Pujyapad Sree Sree Acharyadev,

and Param Pujyapad Sree Sree Abin Da.

~Aswini Kumar Patra

Acknowledgments

I express my deepest gratitude to my supervisor and guide, Prof. Lingaraj Sahoo, for his invaluable guidance, encouragement, and support throughout the course of my doctoral research. His insights and constant motivation have been instrumental in shaping the direction and quality of this work. I also extend my sincere thanks to the Head of the Department of Biosciences and Bioengineering for providing me with the necessary facilities, support, and academic environment that enabled me to pursue my research successfully. I am equally indebted to the members of my Doctoral Committee - Prof. Selvaraju Narayanasamy (Chairman), Prof. Soumen Kumar Maiti, and Prof. Sreedeep S. - for their valuable suggestions, constructive criticism, and continuous encouragement.

I owe special thanks to my family members for their unconditional love, sacrifices, and encouragement. My heartfelt respect goes to my beloved Baba and Bou, whose blessings and values have always guided me. I am deeply indebted to my wife, Mamali, and my daughter, Vedanshi, for their patience, understanding, and constant inspiration during the most demanding phases of this journey. My brother Satya, my sisters Banita and Anita, along with my brothers-in-law Babula and Chintamani, and sister-in-law Gayatri have been a source of strength and affection, for which I remain ever grateful. A special place of gratitude goes to my late father-in-law and mother-in-law for their constant motivation, and to my brother-in-law Kanu for his unwavering support. I also extend my warm love and affection to my nephews Jagan, Somu, and Jagakalia, and my niece Swati, whose innocent smiles and encouragement have been a source of joy and inspiration.

I would like to express my special gratitude to Saroj Sahoo Sir and Nalini Madam for their valuable support throughout the journey. I also acknowledge the blessings and encouragement of Bhubani Angya, whose guidance during my formative years played a pivotal role in shaping my aspirations. My heartfelt thanks go to Namita Nani for her affection and blessings, and to all my school teachers, whose teachings laid the foundation of my academic life.

My sincere thanks also go to my labmates and friends — Sanjeev, Kiran,

Anurabh, Asif, Subhajit, Deepak, Maheswari, Mahesh, Ashrumochan, and Chitta— for their camaraderie, cooperation, and for making the lab atmosphere both productive and enjoyable. I also gratefully acknowledge my friends Bipra and Arup, whose encouragement during the initial years of this journey was of immense importance. I further extend my gratitude to my well-wishers and childhood friends from my village, who have always stood by me, and to my friends and colleagues at NERIST, whose support and motivation enriched this journey.

I also wish to express my heartfelt respect and gratitude to my Satsang guru bhai and behen — Bijita, Uday, Jeevan, Bhargavi, and Sreejev — for their spiritual guidance, motivation, and constant moral support, which have been a source of inner strength during this path.

Finally, I dedicate this work to all those who stood by me, believed in me, and contributed directly or indirectly to the successful completion of this thesis.

Aswini Kumar Patra

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under Curve
BPNN	Backpropagation Neural Network
CFD	Computational Fluid Dynamics
CGLCM	Correlation-based Gray-Level Co-occurrence Matrix
ChlF	Chlorophyll Fluorescence
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CT	Computed Tomography
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DT	Decision Tree
ECa	Soil Electrical Conductivity
ELM	Extreme Learning Machine
E-nose	Electronic Nose
GB / GBDT	Gradient Boosting / Gradient Boosting Decision Tree
GLCM	Gray-Level Co-occurrence Matrix
GPR	Gaussian Process Regression
Grad-CAM	Gradient-weighted Class Activation Mapping
GRNN	General Regression Neural Network
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HSV	Hue, Saturation, Value
HTP	High-Throughput Phenotyping
IDC	Iron Deficiency Chlorosis
IRIV	Iteratively Retaining Informative Variables
KNN / k-NN	k-Nearest Neighbor
KRR	Kernel Ridge Regression
LAI	Leaf Area Index

LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
LIBS	Laser-Induced Breakdown Spectroscopy
LiDAR	Light Detection and Ranging
LSTM	Long Short-Term Memory
MARS	Multivariate Adaptive Regression Splines
MEMD	Multivariate Empirical Mode Decomposition
ML	Machine Learning
MLP	Multi-Layer Perceptron
MLR	Multiple Linear Regression
MOS	Metal Oxide Semiconductor
MRI	Magnetic Resonance Imaging
NB	Naive Bayes
NDVI	Normalized Difference Vegetation Index
NNI	Nitrogen Nutrition Index
NUE	Nitrogen Use Efficiency
PCA	Principal Component Analysis
PET	Positron Emission Tomography
PLS / PLSR	Partial Least Squares / Partial Least Squares Regression
PNN	Probabilistic Neural Network
RBFNN	Radial Basis Function Neural Network
ReLU	Rectified Linear Unit
RF	Random Forest
RL	Reinforcement Learning
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RUSBoost	Random Under-Sampling Boosting
SIFT	Scale-Invariant Feature Transform
SIMCA	Soft Independent Modeling of Class Analogy
SISA	Sharded, Isolated, Sliced, and Aggregated
SPA	Successive Projections Algorithm
SPAD	Soil and Plant Analyzer Development (chlorophyll meter)
SVM	Support Vector Machine
SW-SVR	Sliding Window Support Vector Regression
UAV	Unmanned Aerial Vehicle
VIs	Vegetation Indices
ViT	Vision Transformer
XAI	Explainable Artificial Intelligence
XGB / XGBoost	Extreme Gradient Boosting

Thesis Abstract

Climate changes and nutrient deprivation severely impact crop production by increasing heat and water deficit stress, altering soil nutrient availability, and reducing the nutritional content of crops, leading to lower yields. Traditional stress monitoring approaches rely heavily on manual scouting or complex laboratory analyses, which limit scalability in real-world agricultural settings. This thesis establishes a suite of explainable, lightweight deep learning frameworks tailored for field-based stress identification and severity assessment to address these limitations. The research proposes an explainable deep learning pipeline for UAV-acquired RGB imagery in drought stress detection. A pre-trained CNN backbone, with custom layers, is used for dimensionality reduction and improved generalization. Gradient-based visualizations inspired by Grad-CAM are integrated to highlight the model's internal focus, enhancing interpretability and trust. This framework outperformed conventional CNN-based methods in natural agricultural conditions. Besides, an explainable Vision Transformer (ViT) framework is introduced, leveraging both an end-to-end ViT architecture and a hybrid ViT-SVM pipeline. The attention mechanism in ViTs enables precise identification of spatial regions within field images affected by drought stress, further strengthening model transparency and classification accuracy. To refine efficiency for deployment in resource-limited agricultural environments, a novel lightweight hybrid CNN is designed, inspired by ResNet, DenseNet, and MobileNetV2. This model achieves a 15-fold reduction in trainable parameters while retaining competitive accuracy. Performance is further enhanced through a gradient-guided machine unlearning mechanism, which systematically removes non-contributive training samples based on gradient magnitudes, reducing misclassification errors and improving robustness. Beyond drought detection, the thesis addresses nitrogen stress severity under combined stressors such as drought and weed competition. Two pipelines were developed using multimodal imaging datasets (RGB, multispectral, and infrared): a MobileNetV2-based spatial classifier and a MobileNetV2-LSTM hybrid for spatio-temporal modeling. The CNN-LSTM pipeline captured complex interactions between nutrient deficiency and other stresses, demonstrating superior accuracy in severity classification. Overall, this thesis contributes lightweight, interpretable, and scalable AI frameworks for stress monitoring, offering actionable insights for precision agriculture and supporting resilient, sustainable crop production.

Keywords: {Plant-stress Detection, Abiotic Stress, Precision Agriculture, Machine Learning, Deep Learning, Explainable AI}

Chapter 1

Introduction

Agriculture is the backbone of global food security, yet it is increasingly threatened by climate change, land degradation, and a rapidly expanding population. One of the most pressing consequences of these challenges is the prevalence of plant stresses, which hinder crop growth, reduce yield, and threaten food sustainability. Plants experience stress when environmental conditions deviate from the optimal range, leading to morphological, physiological, and biochemical disruptions [2]. Such stresses may be biotic, caused by pathogens, pests, or weeds, or abiotic, resulting from environmental or chemical factors; in practice, plants often encounter these stresses simultaneously [3]. Abiotic stresses—including drought, salinity, heat, cold, flooding, nutrient deficiency, and heavy metal contamination—pose major threats to agricultural productivity worldwide [4, 5, 6]. These stresses not only diminish yield but also impair grain quality, nutritional value, and resistance to secondary infections. Importantly, in real-world conditions, plants rarely experience stresses in isolation; rather, they frequently face multiple concurrent challenges, such as drought combined with heat or nitrogen deficiency coupled with weed competition [3, 5, 7]. This complexity emphasizes the urgency of developing robust, non-invasive, and scalable methods for stress detection, classification, and monitoring in crops, thereby enabling strategies for more sustainable and resilient agriculture.

Phenotyping—the precise measurement of plant traits such as morphology, physiology, and biochemical responses—is a cornerstone of stress research. Traditional methods rely heavily on manual, destructive sampling techniques such as chlorophyll assays or root excavations, which are labor-intensive, time-consuming, and unsuitable for large-scale studies [8]. Moreover, these methods lack temporal resolution, making it difficult to monitor stress progression over time. To overcome these limitations, non-invasive imaging-based phenotyping technologies have emerged as powerful alternatives [9, 10]. By capturing plant traits at cellular, organ, canopy, and field levels, these technologies provide rapid, scalable, and accurate insights into stress responses [11, 12]. High-throughput phenotyping (HTP), enabled by advanced sensors, drones, and automated platforms, allows for the evaluation of thousands of genotypes in a short time [13, 14]. This shift from

destructive sampling to image-driven digital phenotyping marks a paradigm change in plant science, where the focus is no longer on mere data collection and analysis alone, but on extracting meaningful biological insights from vast datasets [15, 16, 17].

The advent of advanced sensors and UAV-mounted platforms has greatly enhanced the precision of stress detection in plants. Imaging techniques now provide non-invasive and non-destructive approaches for assessing plant stress, utilizing modalities such as red-green-blue (RGB) imagery [18], thermal imaging [19], fluorescence imaging [20], multispectral, and hyperspectral imaging [1]. The integration of multiple sensing modalities enriches the detection of physiological responses, while spatio-temporal imaging—tracking changes over time—enables the early detection of stress conditions before visible symptoms appear [7, 21].

The rapid accumulation of high-dimensional imaging datasets has created challenges and opportunities in data management, analysis, and interpretation. While traditional machine learning techniques remain valuable for small datasets, deep learning models - particularly convolutional neural networks (CNNs) [9], CNN–LSTM hybrids[22], and Vision Transformers (ViTs)[23] - have emerged as state-of-the-art due to their ability to learn hierarchical and discriminative features. These models have substantially improved the accuracy of stress detection across modalities, yet challenges persist. Many deep learning architectures are computationally intensive, requiring large numbers of trainable parameters, which constrains deployment in field settings. Moreover, their "black-box" nature limits interpretability and explainability, reducing trust and usability for breeders, agronomists, and farmers [24, 25, 26]. These gaps highlight the importance of developing lightweight and explainable multimodal frameworks, capable of generalizing across diverse crops and environments.

Among abiotic stresses, drought and nutrient deficiencies (especially nitrogen) are particularly devastating. Drought severely reduces crop productivity by impairing photosynthesis, transpiration, and water-use efficiency [27], and is one of the most widely studied stressors. Nitrogen stress, conversely, reduces chlorophyll content, hampers growth, and degrades grain quality [28]. By addressing these two major stresses, AI-enabled imaging systems can directly contribute to climate-resilient and resource-efficient crop production. This thesis therefore reviews imaging-based ML and DL approaches for abiotic stress phenotyping, identifies research gaps, and focuses on advancing solutions tailored to drought identification and nitrogen stress severity classification.

1.1 Review of Literature

This section reviews the advances in plant stress phenotyping with a focus on imaging technologies, machine learning (ML), and deep learning (DL) methods applied to abiotic stress classification. The literature is organized thematically, beginning with imaging modalities used in phenotyping, followed by computational approaches in ML and DL. Subsequently, stress-specific studies are discussed with emphasis on drought and nitrogen stress, while other abiotic stresses such as salinity, heat, nutrition and heavy metals are also highlighted. The chapter concludes with multimodal and spatio-temporal frameworks, before synthesizing critical insights to identify gaps addressed by this thesis.

1.1.1 Imaging Modalities in Plant Stress Phenotyping

Various imaging techniques such as hyperspectral, fluorescence, RGB, X-ray and thermal imaging target specific wavelengths of light to detect important plant biochemicals including chlorophyll, proteins, polyphenols, nitrate, sugar, and water content. When light hits a plant leaf, it interacts in several ways: some light is absorbed by the leaf to drive photosynthesis and other physiological processes; some light is reflected either as specular reflection (mirror-like) or scattered (diffused) reflection; and some light is transmitted through the leaf to the other side. Imaging detectors capture this reflected or transmitted light, allowing analysis of plant properties based on the wavelengths and intensity of the light [1]. This approach leverages the unique spectral signatures of plant biochemical compounds and physiological traits, enabling detailed non-destructive monitoring of plant health and stress status through absorption, reflection, and transmission measurements as summarized in Fig. 1.1.

The biochemical characteristics of plants have long been associated with specific wavelengths of light that crops reflect and absorb within the visible and near-Infrared (NIR) spectra [29]. As a result, a range of sensors such as RGB, multispectral, hyperspectral, magnetic resonance imaging(MRI), X-ray Computed Tomography (CT), positron emission tomography(PET), light detection and ranging(LiDAR) are employed to detect traits associated with relevant abiotic stress. For instance, visible RGB imaging provides essential morphological and color information related to biomass and senescence but has limited spectral range and is influenced by lighting conditions[30]. Thermal imaging offers a potential avenue for effectively screening crop varieties that exhibit resistance to drought, tolerance to saline conditions, or cold resistance by assessing leaf temperatures [19, 31]. Multispectral and hyperspectral imaging offers spectral data concerning a range of parameters associated with physiological and biochemical characteristics. These parameters include but are not limited to the leaf area index (LAI), crop water content, leaf/canopy chlorophyll content, and nitrogen content [32, 33]. Fluorescence imaging eval-

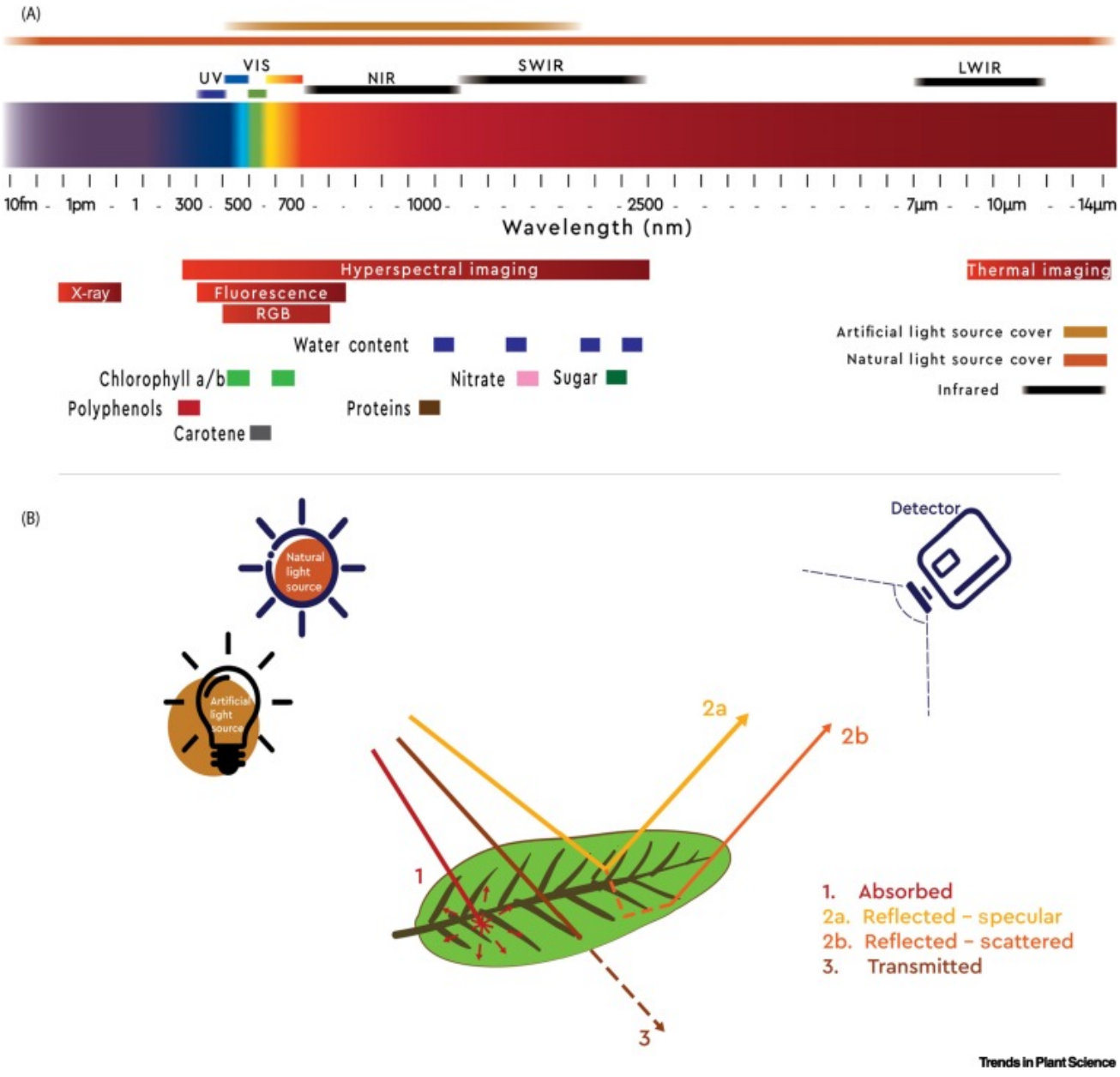


Figure 1.1: Spectral regions and imaging techniques used to assess plant biochemical and physiological traits[1]

uates photosynthetic efficiency by measuring chlorophyll fluorescence parameters and is useful for detecting photosynthetic stress but may require careful calibration and is influenced by environmental factors[30]. 3D imaging provides spatial and internal structural information, important for detailed morphological and anatomical assessment, yet tends to be resource-intensive and lower throughput[10]. Amongst the imaging based technologies, hyperspectral imaging is gaining much space as it provides the ability to measure traits across a wide range of wavebands simultaneously, enabling more detailed characterisation of plant performance and environmental interactions [1]. Incorporating data from multiple sensing modalities enriches the information available, enabling more effective assessment of abiotic stress.[34].

1.1.2 Machine Learning and Deep Learning in Stress Phenotyping

With the increasing availability of multi-modal imaging data—such as RGB, thermal, hyperspectral, and fluorescence images—machine learning (ML) and deep learning (DL) techniques have emerged as powerful tools for extracting meaningful patterns and predicting stress conditions. ML approaches offer robust analytical capabilities for stress classification, prediction, and pattern recognition, while DL models, particularly convolutional neural networks (CNNs), have demonstrated superior performance in processing complex image data and learning hierarchical features directly from raw inputs. Advanced DL techniques like Vision Transformers (ViT) and hybrid CNN-ViT models are excelling in plant stress detection by capturing complex, high-dimensional data and long-range dependencies[23]. ViT models, by treating images as patches, offer superior performance in tasks like disease detection[35]. Additionally, CNN-LSTM models are effective for tracking the temporal progression of stress traits, capturing the time-dependent changes in plant health due to environmental or disease factors[36]. These combined models enhance both spatial and temporal plant stress analysis.

The evolution of ML methods in plant phenotyping has expanded from traditional supervised and unsupervised learning to advanced paradigms such as active learning [37], transfer learning [38], few-shot learning [39] and semi-supervised learning [40]. These approaches address practical challenges such as limited labeled data and high variability in field conditions. For instance, transfer learning enables the adaptation of pre-trained models on large plant datasets to specific stress-related tasks, significantly reducing the need for extensive labeled training data [38]. Furthermore, there is increasing emphasis on developing lightweight and interpretable DL models suitable for real-time deployment on edge devices like drones and mobile platforms. Explainability methods are being integrated to provide transparent decision-support, helping agronomists trust and act upon AI model predictions in precision agriculture.

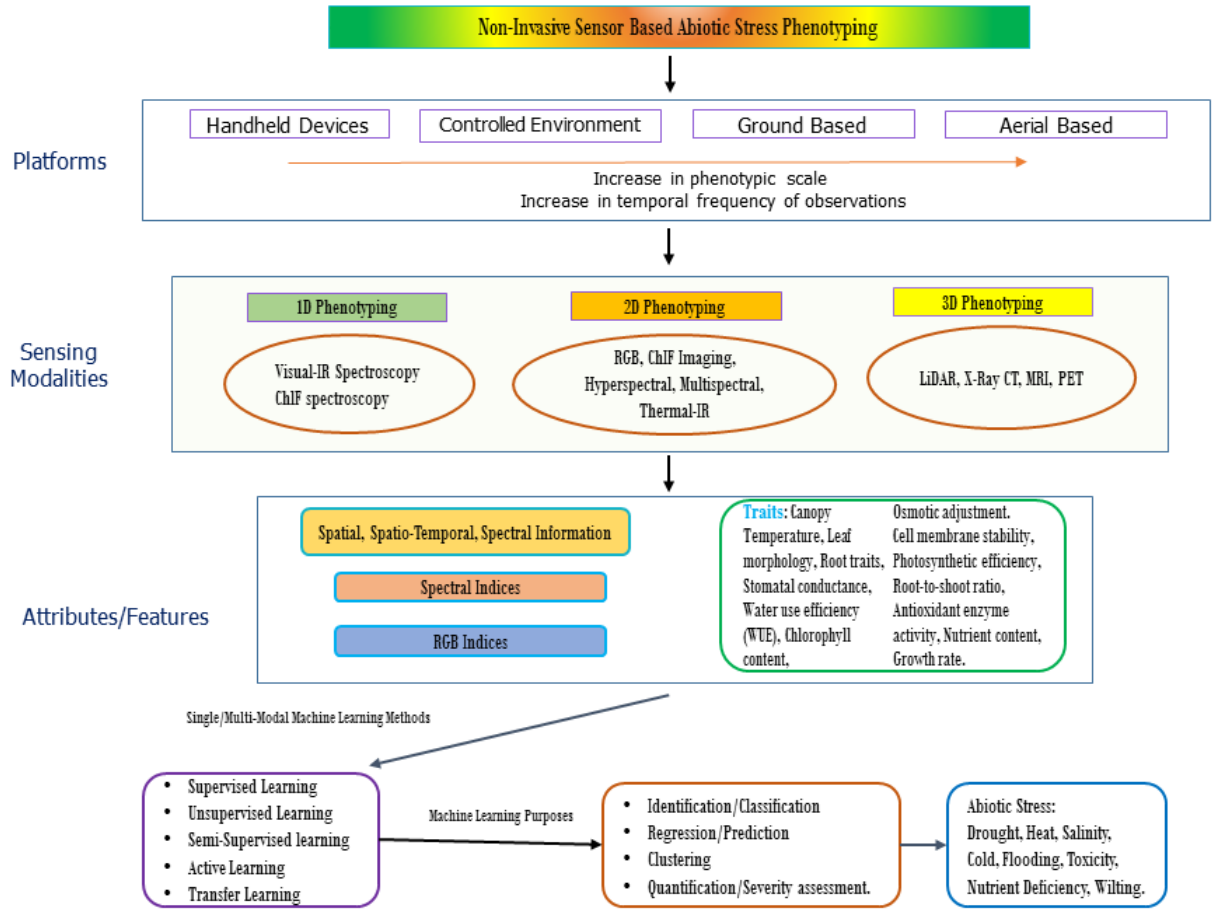


Figure 1.2: Non-invasive Abiotic Stress Phenotyping with Machine Learning

The Fig.1.2 presents a comprehensive workflow of non-invasive sensor-based abiotic stress phenotyping, linking platforms, sensing modalities, traits, and machine learning applications as observed in literature. Platforms range from handheld devices and controlled environments to ground-based and aerial systems (e.g., UAVs), enabling larger-scale and higher-frequency phenotypic observations. Sensing modalities span 1D (spectroscopy, ChlF), 2D (RGB, hyperspectral, multispectral, thermal-IR, ChlF imaging), and 3D (LiDAR, X-ray CT, MRI, PET) phenotyping approaches. These modalities capture spatial, spatio-temporal, and spectral attributes, including spectral and RGB indices, to quantify traits such as canopy temperature, root and leaf morphology, stomatal conductance, water-use efficiency, chlorophyll content, osmotic adjustment, photosynthetic efficiency, antioxidant activity, and nutrient content. Extracted features are processed through single- or multi-modal machine learning methods (supervised, unsupervised, semi-supervised, active, and transfer learning) for tasks such as identification/classification, regression/prediction, clustering, and severity assessment. Collectively, this pipeline enables high-throughput detection and analysis of key abiotic stresses including drought, heat, salinity, cold, flooding, toxicity, nutrient deficiency, and wilting.

1.1.3 Stress-Specific Studies

1.1.3.1 Drought Stress

Water stress may arise as a result of two conditions, either due to excess of water or water deficit. The more common water stress encountered is the water-deficit stress known as the drought stress [6]. Drought is a major abiotic stress factor that affects the growth and development of plants [41].

Early work by Zhuang et al. [42] relied on segmentation, color, and texture feature extraction combined with a gradient boosting decision tree (GBDT) to detect water stress in maize. A follow-up study demonstrated that a deep convolutional neural network (DCNN) outperformed GBDT on the same dataset [43]. Random Forest (RF) has also proven effective in relating spectral bands and vegetation indices to physiological traits such as foliar temperature and stomatal conductance. Brewer et al. [44] used UAV-derived spectral variables for maize, optimizing RF with 500 trees and selected variable subsets to predict temperature and stomatal conductance. Similar ML regression approaches have been used in hyperspectral studies: Asaari et al. [45] linked normalized hyperspectral profiles to four physiological traits (water potential, effective quantum yield, stomatal conductance, transpiration) using PLSR, Kernel Ridge Regression (KRR), and Gaussian Process Regression (GPR). Partial Least Squares Regression (PLSR) has also been employed to predict leaf stomatal conductance, transpiration, and photosynthesis by integrating hyperspectral, thermal, and canopy height data across soybean and maize, identifying crop-specific sensitive variables [46].

To reduce complexity and labeled data requirements, many studies focus on selecting hyperspectral bands most sensitive to drought. Sankararao et al. [47] introduced an ensemble method for waveband selection followed by stress classification in groundnut canopies. Dao et al. [48] further demonstrated the value of using first-order derivative spectra over full spectra, enabling deep learning models to better capture subtle drought-related changes compared to traditional spectral indices. Schmitter et al. [49] proposed an unsupervised domain adaptation framework that allowed ordinal SVM models trained on one dataset to generalize across species (barley, maize), environments, and sensors, improving early stress detection.

DL has consistently outperformed conventional ML in phenotyping. For instance, Ramos-Giraldo et al. [38] used DenseNet-121 to classify soybean drought severity into five wilting levels. Azimi et al. [21] developed a CNN-LSTM hybrid model for chickpea stress stages, while another study combined VGG16/InceptionV3-based CNN feature extractors with LSTM for chickpea shoot images [50]. Chandel et al. [51] compared AlexNet, GoogLeNet, and Inception V3 for maize, okra, and soybean, reporting

GoogLeNet as most accurate.

Sensor/Modality	**ML Models**	**Traits**	**Plant Species**	**ML Purpose**	**Reference**
Hyperspectral	RF, SVM, XGB	Spectral (canopy)	Groundnut	Waveband identification and stress classification	[47]
Spectroscopy	DT, RF, XGB	Spectral Reflectance	Arabidopsis thaliana	Classification	[26]
Direct Measurement/Manual Phenotyping	KSC, k-NN, RF	Root Traits	Faba Bean	Classification, Clustering	[52]
Chlorophyll Fluorescence	DT, RF, SVM, LR, LDA, NB, GB, k-NN	Chlorophyll fluorescence (PSII), texture/GLCM, morphological features, correlation-based features, and colour percentage of various nine bands	Wheat	Classification	[53]
Hyperspectral	K-means Clustering, SVM	Vegetation Indices (VI)	Barley	Classification, Clustering	[54]
Hyperspectral	PLSR, GPR, KRR	Hyperspectral reflectance data	Maize	Regression	[45]

Sensor/Modality	ML Models	Traits	Plant Species	ML Purpose	Reference
RGB, IR-Thermal, Multi-spectral	SVM	NDVI, gNDVI, temperature, color hue, color saturation, canopy size and plant height	Soybean	Classification	[55]
RGB	CNN, SVM	Color and Texture Features	Maize	Classification, Quantification	[56]
RGB	CNN, SVM, RF	Plant Area, Plant Height, Plant Width, Center of Mass Horizontal Distance (CM width), Center of Mass Height (CM height), Plant Mass	Soybean	Classification	[57]
RGB	CNN-LSTM	Spatiotemporal	Chickpea	Classification	[50]

Table 1.1: Drought/Water Stress and Machine Learning Models

Wilting has emerged as a robust phenotypic marker. Pathan et al. [58] introduced severity scoring (0–4), later operationalized through transfer learning on DenseNet-121 [38]. Yang et al. [57] extracted color, shape, and stem-based metrics from segmented RGB images of plants, which were classified using RF, SVM, and CNNs. Zhoua et al. [55] combined UAV RGB, thermal, and multispectral features (NDVI, gNDVI, canopy size, hue, temperature) in an SVM model that achieved up to 0.9 accuracy in classifying fast- vs slow-wilting soybean genotypes. Xiang et al. [59] extended this by segmenting 3D point clouds into plants and leaves for trait extraction, applying classifiers such as LASSO+Logistic regression, RF, AdaBoost, and SVM.

Gupta et al. [53] used chlorophyll fluorescence imaging of wheat, extracting features via correlation-based gray-level co-occurrence matrix (CGLCM) and color bands, with RF and Extra Trees classifiers showing superior accuracy. Chen et al. [60] demonstrated that SVM outperformed RF and PLSR in predicting drought tolerance coefficients from hyperspectral tea canopy data. Butte et al. [61] applied aerial imagery of potato canopies with multimodal DL models, and Patra et al. [62] later proposed an explainable DL framework built on CNNs and transfer learning to improve interpretability. Zhou et al. [34] combined UAV multispectral and thermal features with DL to classify soybean flooding injury scores, achieving 90% accuracy at 20 m altitude. Goyal et al. [63] curated an RGB maize dataset and showed their custom CNN surpassed five leading architectures (InceptionV3, ResNet50, DenseNet121, Xception, EfficientNetB1) for early drought detection.

Vegetation indices (VIs) derived from UAV or RGB imagery have also been linked to stress traits. Sarkar et al. [64] predicted leaf area index (LAI) and lateral growth (LG) in peanuts using multiple regression and neural networks, demonstrating that reduced biomass under drought correlated with yield declines. Zhou et al. [34] also showed how UAV-derived canopy temperature, NDVI, and canopy dimensions can support flooding and drought stress scoring. Quantification remains challenging, as most studies focus on classification. Automating severity assessment, traditionally reliant on expert scoring, is emerging as a critical next step.

While DL offers superior accuracy, its "black-box" nature limits adoption in practice. Few explainable AI (XAI) models exist in plant phenotyping. Ghosal et al. [25] identified soybean stress while highlighting visual features used in classification. In contrast, Nagasubramanian et al. [65] acknowledged interpretability but lacked methodological detail. These limitations emphasize the need for explainable frameworks that couple accuracy with transparency. Drought stress represents the most extensively studied form of abiotic stress with machine learning. While the majority of research focuses on identification and classification tasks, only a limited number have ventured into quantifying them. Quantification poses a significant challenge, primarily relying on expert assessments, and only a very few have highlighted the importance of automating quantification of these stress factors. Table 1.1 presents the list of studies on drought stress.

1.1.3.2 Nitrogen Stress

Nitrogen stress refers to the deficiency or imbalance of nitrogen in plants, which can significantly impact their growth, development, and overall productivity. Nitrogen is a vital nutrient for plants, and insufficient or excess nitrogen can lead to various physiological and biochemical stresses, affecting crop yield and quality. Recent research reflects a growing interest in nitrogen stress detection under both isolated and combined stress

conditions. Studies span across modeling, remote sensing, imaging, machine learning (ML), and deep learning (DL) approaches, using a wide variety of data modalities. Table 1.2 summarizes key works addressing nitrogen stress.

Clarke et al. [28] examines how spatial and temporal soil variability influences nitrogen use efficiency (NUE) in wheat using the Sirius crop simulation model and long-term field data. The study finds that soil electrical conductivity (ECa) can guide site-specific nitrogen management, with lower water-holding soils requiring less nitrogen but posing higher leaching risks. A reinforcement learning (RL) environment was developed by Kallenberg et al. [66], where agents learn crop management policies through crop growth models. In a nitrogen management case study for winter wheat, the RL agent successfully detected crop nitrogen requirements by analyzing growth states and guided optimal fertilizer application. Sarkar et al. [67] investigates how abiotic stressors—especially drought and temperature—affect nitrogen dynamics and crop productivity in dryland forage systems. Using field data and ML analysis, the study compares conventional tillage and no-till practices, along with the impact of green manures such as field peas. The results show that no-till systems with green manuring significantly improve NUE and reduce the negative effects of drought on plant growth.

Combining SPAD data from multiple leaf positions significantly improves the estimation of the Nitrogen Nutrition Index (NNI), as demonstrated in a study by Wang et al. [68], where ML models like Random Forest and XGBoost outperformed linear regression in predicting NNI. Hyperspectral remote sensing combined with stepwise multiple linear regression is used to detect nitrogen and water stress in maize in a study by Naik et al. [69]. Nitrogen stress was most effectively identified at 540, 780, and 860 nm, with leaf nitrogen content accounting for up to 66% of yield variation at the tasseling stage. A spatio-temporal spectral framework integrating RGB, infrared, hyperspectral data and plant traits (canopy cover, height, biomass, vegetation indices) was applied to sugar beet. ML models, especially SVM, showed high accuracy with multi-modal features outperforming single ones [7]. Electrophysiological signals also proved effective for detecting nitrogen deficiency stress in tomato plants under greenhouse conditions, where deep learning—particularly an encoder-based architecture—outperformed models such as XGBoost [70].

Ghazal et al. [71] evaluates ML models for nitrogen stress detection in maize using RGB images under field conditions. Among tested models, EfficientNetB0 achieved the highest accuracy, outperforming Vision Transformers and other CNNs. A study developed ML and DL models for image-based nitrogen diagnosis in muskmelon using canopy leaf images and environmental data. Among all models, the hybrid DCNN–LSTM achieved the highest accuracy by combining spatial features and temporal light–temperature

inputs [36]. Liao et al. [72] proposed a hybrid DL model integrating CNN with an attention mechanism and LSTM to diagnose nitrogen (N) and potassium (K) nutrient levels in rice at the early panicle initiation stage. Hui et al. [73] estimated sugarcane nitrogen levels using digital images and regression-based ML models, including Random Forest (RF), Backpropagation Neural Network (BPNN), and a stacking fusion approach. The fusion model with PCA-based color–texture features outperformed both RF and BPNN. Chaparro et al. [74] estimated foliar nitrogen content in pineapple by integrating multispectral UAV imagery, IoT-based environmental sensors, and SPAD chlorophyll values with ML. Of the nine models tested, XGBoost and multi-layer perceptron (MLP) achieved the highest accuracies, while multi-sensor data fusion consistently outperformed image-only approaches.

Trung-Tin Tran et al. [75] employ Inception-ResNet v2 and a CNN-based Autoencoder to classify and predict nutrient deficiency symptoms, specifically related to calcium, potassium, and nitrogen. Azimi et al. [76] developed a 23-layer CNN to classify nitrogen deficiency stress in sorghum using shoot images. It outperformed classical ML methods and performed comparably to deeper models like ResNet18 and NasNet Large, with far fewer parameters. Overall, the literature demonstrates that nitrogen stress detection leverages a wide range of single and multi-modal datasets—including imaging and spectral, physiological, biochemical, environmental, electrophysiological, and visual trait data. These datasets, when processed through advanced ML and DL frameworks, significantly improve nitrogen stress detection, diagnosis, and management strategies across diverse crop systems.

Sensor/Modality	**ML Models**	**Traits**	**Plant Species**	**ML Purpose**	**Reference**
Soil variability + Sirius crop model	Crop simulation model	Soil ECa, water-holding capacity	Wheat	Optimize NUE, fertilizer management	Clarke et al. (2024)
Crop growth models (RL environment)	Reinforcement Learning (RL) agent	Growth states, nitrogen demand	Wheat	Optimize fertilizer application policies	Kallenberg et al. (2023)

Sensor/Modality	ML Models	Traits	Plant Species	ML Purpose	Reference
Field + ML analysis	ML (unspecified)	Soil moisture, tillage, green manures	Forage crops	Improve NUE under drought/temperature stress	Sarkar et al. (2025)
SPAD meter	RF, XGBoost, Linear regression	SPAD data (multi-leaf) → NNI	Wheat	Predict Nitrogen Nutrition Index	Wang et al. (2024)
RGB + IR + Hyperspectral	SVM	Canopy cover, height, biomass, VIs	Sugar beet	Multi-stress detection	Khanna et al. (2019)
RGB field images	EfficientNetB0, ViT, CNNs	Visual canopy traits	Maize	Classify nitrogen stress	Ghazal et al. (2024)
RGB canopy images + environment data	Hybrid DCNN–LSTM	Leaf image + temp/light	Muskmelon	Nitrogen diagnosis	Chan et al. (2021)
RGB images	CNN + Attention + LSTM	Canopy features	Rice	N & K diagnosis	Liao et al. (2024)
Digital RGB images	RF, BPNN, Stacking Fusion	PCA-based color–texture	Sugarcane	Predict N levels	Hui (You) et al. (2023)
UAV multispectral + IoT + SPAD	XGBoost, MLP	Multisensor fusion, chlorophyll	Pineapple	Estimate foliar N content	Chaparro et al. (2024)
Hyperspectral sensing	Stepwise MLR	Spectral bands (540, 780, 860 nm)	Maize	Detect N + yield variation	Naik et al. (2020)
Electrophysiological signals	Encoder-based DL, XGBoost	Plant electrical activity	Tomato	Nitrogen deficiency detection	González i Juclà et al. (2023)

Sensor/Modality	ML Models	Traits	Plant Species	ML Purpose	Reference
RGB leaf images	Inception-ResNet v2, CNN Autoencoder	Visual deficiency symptoms	Multiple (Ca, K, N)	Nutrient deficiency classification	Tran et al. (2019)
Shoot images	23-layer CNN, ResNet18, NasNet	Shoot traits	Sorghum	Nitrogen deficiency classification	Azimi et al. (2021)

Table 1.2: Nitrogen Stress Detection and Machine Learning Models

1.1.4 Other Abiotic Stresses

1.1.4.1 Heat Stress

Heat stress is a major contributor to increased chalkiness in rice, which compromises grain quality and reduces market value. To address this, an automated approach leveraging convolutional neural networks (CNNs) and Gradient-weighted Class Activation Mapping (Grad-CAM) has been developed for detecting chalkiness in rice grain images. The CNN model is first trained to distinguish between chalky and non-chalky grains, after which Grad-CAM is employed to localize the grain regions responsible for chalky classification. The Grad-CAM output provides smooth heatmaps that offer a quantitative measure of chalkiness severity. Experiments on both polished and unpolished rice grains demonstrate the effectiveness of this method in accurately identifying chalky grains and delineating affected regions, as validated using standard classification and segmentation metrics [77].

Beyond grain quality, heat stress also disrupts the water status of plant cells through osmotic perturbations caused by reduced photosynthetic capacity, lower sugar content, and increased transpiration rates [78]. Peng et al. [79] attempted to phenotype photosynthetic capacities using a stacked regression framework that combined artificial neural networks (ANN), support vector machines (SVM), least absolute shrinkage and selection operator (LASSO), random forests (RF), Gaussian processes (GP), and partial least squares (PLS). Model performance during training and testing was evaluated using the coefficient of determination (R^2) and root mean square error (RMSE). The stacked regression approach outperformed individual models, achieving an improvement of approximately 0.1 in R^2, thereby demonstrating its superior predictive ability for photosynthetic capacity phenotyping.

Identifying novel marker phenotypes that consistently respond to heat stress

and enhance tolerance remains a priority. One study employing chlorophyll fluorescence, RGB, and infrared (IR) imaging applied logistic regression with LASSO regularization to identify the most informative traits for classification [80]. The model highlighted temperature and morphological traits—such as compactness, isotropy, leaf slenderness, and perimeter—as the top predictors, indicating that morphological changes are particularly valuable for genotypic differentiation under heat stress. Additionally, chlorophyll fluorescence emerged as a critical genotype-specific indicator.

Accurate quantitative assessment of seed quality is equally essential for improving agricultural yields. A study utilizing hyperspectral imaging compared SVM and CNN models to classify rice seeds grown under heat stress and control conditions [81]. The SVM was evaluated in two modes—pixel-based reflectance and seed-based reflectance—yet in both cases, CNN consistently outperformed SVM, underscoring its potential for high-precision seed quality assessment.

Collectively, these studies demonstrate the diverse impact of heat stress on grain quality, photosynthetic efficiency, morphological traits, and seed viability. Advances in imaging technologies combined with machine and deep learning approaches have enabled more precise detection, classification, and quantification of heat stress effects. However, further development of lightweight, explainable, and field-deployable frameworks remains essential to translate these advances into practical breeding tools for enhancing crop heat tolerance. Table 1.3 compiles representative studies on heat stress.

Sensor/Modality	ML Models	Traits	Plant Species	Tissue	ML Purpose and Reference
RGB	CNN	Autonomously extracted	Rice	Grain	Classification [77]
Hyperspectral	ANN, SVM, LASSO, RF, GP, PLS	Hyperspectral reflectance	Tobacco	Leaves	Regression [79]
RGB, Chlorophyll Fluorescence, IR	Logistic Regression	plant size, morphology, and chlorophyll fluorescence traits	Arabidopsis Thaliana	Leaves	Classification [80]
Hyperspectral	SVM, CNN	Pixel information, Autonomously extracted	Rice	Seed	Classification [81]

Table 1.3: Heat Stress and Machine Learning

1.1.4.2 Salt Stress

Salinity stress can alter the emission of volatile organic compounds (VOCs) from plant leaves. An electronic system, augmented with the necessary pattern-recognition algorithm, is developed to detect the salinity stress in the Khasi Mandarin Orange plants. By incorporating temperature modulation in the system, MOS-based gas sensors can selectively detect leaf-emitted VOCs. The study reveals the successful classification of plants exposed to different levels of saline water from VOC information recorded by the prototype with an accuracy of 98.3% [82].

An ensemble feature selection method is proposed for identifying informative spectral features from a hyperspectral dataset containing images of four wheat lines with control and salt (NaCl) treatments. Six feature selection methods, including correlation-based feature selection, ReliefF, sequential feature selection, SVM-RFE, LASSO logistic regression, and random forest, form the base of the ensemble. Furthermore, this feature selection pipeline facilitated the transformation of hyperspectral data into a multispectral dataset, showcasing its potential for developing customized multispectral cameras for plant phenotyping applications[83].

Soil salinity is one of the most widespread abiotic stresses limiting agricultural productivity worldwide. Excess salt disrupts ion balance, reduces water uptake, and triggers oxidative stress, leading to growth retardation and yield loss across diverse crops. Recent studies have applied imaging technologies, molecular markers, and alternative sensing systems, coupled with machine learning (ML), to provide rapid, non-destructive, and scalable solutions for salt stress detection and classification. The following section reviews representative works in this area, highlighting their methodologies, accuracy, and practical significance.

Mohammadi and Asefpour Vakilian demonstrated combining leaf textural traits (GLCM), physiological/biochemical indices, and miRNA concentrations (miR-156a, -166i, -399g, -477b) measured via an electrochemical biosensor. Stress classification with SVM optimized by GA and PSO showed that imaging alone explained severity ($R^2 \approx 0.61$), while miRNA models achieved near-perfect accuracy ($R^2 \approx 0.99$), highlighting molecular specificity [84]

Chlorophyll fluorescence (ChlF) imaging offers non-destructive physiological readouts. Deng et al. captured three ChlF image types in soybean under NaCl stress; ResNet50 with feature fusion achieved 98.61% accuracy, outperforming other CNNs[85]. Tian et al. used multicolor fluorescence in *Arabidopsis*, reaching ∼98.5% accuracy by Day 9, demonstrating early stress detection potential[86]. Spectral imaging approaches provide deeper biochemical insights. Kecoglu et al. applied Raman spectroscopy to wheat

leaves under 0–150 mM NaCl, extracting vibrational signatures of pigments, cell-wall polymers, and amino acids. Preprocessed spectra were modeled with multiple regressors, with Gaussian Process Regression (rational quadratic kernel) achieving $R^2 \approx 0.92$–0.93 for salt quantification. Despite strong predictive power, Raman requires costly instrumentation and showed reduced accuracy in very young leaves [87].

Bridging research-grade hyperspectral imaging to practical field sensors, Moghimi et al. used hyperspectral reflectance (400–900 nm, 215 bands) on wheat under salinity stress. Six feature selection methods (ReliefF, CFS, LASSO, SVM-RFE, RF, SFS) were ensembled to rank informative bands. Dimensionality was reduced from 215 to 15 features, improving F1-score by 8.5%. Crucially, the 589 nm sodium absorption band consistently ranked highest, and a reduced three-band set (528, 589, 805 nm) reproduced hyperspectral results with <4% loss. This study demonstrated how multispectral systems tailored to stress-specific bands can replace bulky hyperspectral sensors for field deployment[88].

Genotype-phenotype integration enhances predictive power. Akbari et al. combined SSR markers and phenotypic traits in barley, with neural networks achieving $R^2 \approx 0.999$ and >97% accuracy[89]. Classical phenotyping with ML (Okumuş et al.)in forage pea used morphological and germination traits, with XGBoost achieving $R^2 = 0.92$–0.98. Okumuş et al. (2024) examined four forage pea cultivars under temperature (10–20 °C) and salinity (0–20 dS m^{-1}) gradients. Morphological and germination traits (shoot/root length, fresh/dry weight, germination percentage) were measured and modeled using XGBoost, MARS, and Gaussian Process Classifiers. R^2 values ranged from 0.92 to 0.98, with XGBoost excelling for biomass traits. Though reliable, this approach is destructive and lacks the early-warning capacity of imaging or molecular methods[90].

Seed-level imaging extends this spectrum. Vello et al. built *SeedML*, a web platform using visible and fluorescent (400–500 nm) images of *Camelina sativa* seeds. From morpho-colorimetric descriptors (area, circularity, intensity quartiles), ML classifiers were trained in WEKA. Fluorescence color features proved most sensitive, giving ∼93% accuracy for 200 mM NaCl vs control. Performance was lower at mild stresses, but the approach is non-destructive and scalable for high-throughput screening[91].

Non-imaging modalities offer additional routes. Sharma et al. developed an electronic nose (E-nose) comprising nine MOS gas sensors with heater temperature modulation, optimized using CFD simulations. Khasi Mandarin plants were subjected to salinity (3–16 dS m^{-1}), and leaf VOC emissions were measured. ML classifiers (RF, SVM-linear, SVM-RBF) were evaluated; SVM-RBF achieved 98.3% accuracy with ∼25 s response and ∼40 s recovery times. Salinity stress was validated by chlorophyll fluorescence and chlorophyll content, but imaging was not central to detection. The study

highlights VOC sensing as rapid and portable, though field robustness remains untested [82].

Collectively, these studies illustrate diverse approaches—from molecular assays and genotype markers to ChlF, Raman, hyperspectral imaging, and VOC sensing. Imaging methods provide non-destructive physiological insights enhanced by deep learning, molecular/genotype assays provide specificity, and VOC sensing offers speed. Most remain controlled-environment studies, highlighting gaps in field validation, cross-species transferability, cost-effectiveness, and multimodal integration.

1.1.4.3 Nutrient Stress

In nutrient stress studies, deficiencies of iron [92, 93], potassium [25, 75], and nitrogen [94, 95] have been identified as the most prominent. Table 1.4 details machine learning and deep learning applications in nutrient stress assessment.

Stress identification, classification and quantification (stress severity) are carried out by S Ghosal et al. [25]in a single framework without detailed symptom annotation by experts. The proposed work uses deep CNN to carry out the classification. Besides, they followed a novel approach to generate feature maps at specific layer of the model. An explanation map (EM) is created by calculating a weighted average of the top K feature maps out of 128 maps, with the feature importance(FI) metrics serving as their respective weights. The average intensity of the EM is utilized as a representation of the severity level, expressed as a percentage (where 0% signifies a symptom-free leaf and a higher value indicates significant symptoms). This percentage can then be discretized to determine the stress severity class. To streamline the process of removing the background and retaining only the plant canopy (foreground), each image was transformed from its original Red, Green, Blue (RGB) format into the HSV (Hue, Saturation, Value) format. The field visual rating was employed as the categorical output variable, defining the classes. Subsequently, the classification models were utilized to produce IDC ratings based on various input variables[92]. This work severity ratings.

Austin A. et al.[93] employs a three-step approach for IDC phenotyping utilizing RGB and Infrared images. Initially, the plant canopy is masked from the soil through k-means clustering. Subsequently, this masked plant canopy undergoes additional k-means clustering to classify pixels into green, yellow, and brown categories. These pixel features are then linked back to ground-based visual scores through random forest and neural network models. This methodology aids in classifying plots as tolerant or susceptible to IDC. Scale-invariant feature transform (SIFT) and Histogram of oriented gradients (HOG) features are extracted from RGB shoot images of Sorghum and have been given as input to the ML methods such as SVM, DT, KNN. Simultaneously, a 23-layered Con-

volutional Neural Network (CNN) was developed to autonomously learned features from these images[95]. This approach demonstrated superior performance in classifying nitrogen stress levels when compared to conventional methods. RGB, Infrared (IR), and multispectral images undergo a series of pre-processing steps, including vegetation segmentation, 3D reconstruction, and reflectance normalization. These steps aim to convert the raw data into indicators of plant traits such as canopy cover, average height, and normalized narrow-band reflectances across time. Subsequently, machine learning models are trained to utilize these indicators for the prediction of severity levels for water, nitrogen, and weed stress, employing a 5-fold cross-validation approach[7].

Sensor/Modality	**ML Models**	**Traits**	**Plant Species**	**ML Purpose**	**Reference**
Hyperspectral, RGB, IR	DT, LDA, SVM, k-NN, Bagged Trees, Boosted Trees	Canopy Cover, Height, Volume, Hyperspectral reflectances	Sugar Beet	Classification	Nitrogen, Weed, Drought [7]
RGB	CT, RF, NB, LDA, SVM, KNN, GMM	Canopy Features	Soybean	Classification	IDC [92]
RGB, NIR	RF, NN, K-means Clustering	Green, Yellow and Brown Pixels (canopy)	Soybean	Classification	IDC [93]
RGB	CNN, SVM, DT, KNN	SIFT and HOG features, autonomously learned features	Sorghum	Classification	Nitrogen [95]
RGB	CNN	Automated Leaf Feature Extraction	Tomato	Classification	Calcium, Potassium, Nitrogen [75]
RGB	CNN	Automated Leaf Feature Extraction	Soybean	Classification, Quantification	Iron, Potassium [25]

Sensor/Modality	ML Models	Traits	Plant Species	ML Purpose	Reference
Chlorophyll fluorescence	PCA, hierarchical k-means classification, and super-organising maps	Leaf Element Content and ChIF parameters	Rapeseed	Dimension Reduction, Classification	N, P, K, Ca, Mg, Cu, Fe, Zn [96]

Table 1.4: Nutrient Stress and Machine Learning

Two distinct models, Inception-ResNet v2 and Autoencoder based on convolutional neural networks, are employed by Trung-Tin Tran et al.[75] to classify and predict nutrient deficiency symptoms namely, Calcium, Potassium and Nitrogen. To enhance predictive validation accuracy, ensemble averaging was applied on these two predictive models. This ensemble approach drew inspiration from the work of Cheng Ju et al.[97], which demonstrated the effectiveness of ensembles utilizing deep CNNs for image classification. An interesting study aims to establish a link between element content in different soils, plant leaves grown on these soils, and variations in selected chlorophyll a fluorescence parameters to detect early plant stress caused by nutrient status in natural conditions. To achieve this objective, a mathematical procedure combining principal component analysis (for data complexity reduction), hierarchical k-means (for classification), and machine-learning via super-organising maps are employed detect nutrient deficiency in early stages[96]. RGB images of plant canopies are used for the identification of nutrient stress types and the classification of plants grown under varying nutrient levels, which included low, normal, and high stress conditions. A total of nine treatments were conducted, involving two potassium (K) levels, four nitrogen (N) levels, two phosphate (P) levels, and normal fertilizer application. A combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) is employed for the classification task. The study conducted a comparative analysis of various pre-trained CNN architectures, including VGG16, AlexNet, ResNet18, Inception3, and ResNet101, to select the most suitable deep CNN for feature extraction. Ultimately, the Inceptionv3-LSTM model achieved the highest overall classification accuracy, reaching 95%, outperforming the other methods[98].

1.1.4.4 Heavy Metal Stress

Over the past decade, heavy metal (HM) toxicity has emerged as a significant concern in the agricultural industry, posing a threat to crop productivity[99]. HMs encompass a

group of metals, including but not limited to lead (Pb), manganese (Mn), copper (Cu), nickel (Ni), cobalt (Co), cadmium (Cd), mercury (Hg), aluminum (Al), and arsenic (As), which can adversely affect the growth and development of host plants[100]. Certain metal-based micronutrients and macronutrients, such as Cu, Zn, Co, Ni, Fe, Cr, Mn, I, and Se, are vital for key enzyme functions and the regulation of metabolic processes like redox homeostasis, metabolism, DNA synthesis, and photosynthesis[101]. However, elevated levels of HMs can lead to toxicity and even fatality. Furthermore, some HMs, such as As, Ag, Hg, Cd, and Pb, hold no biological significance for plants but are instead harmful, carrying severe health implications for humans, including skin and lung cancer, urinary tract disorders, cardiovascular diseases, neurotoxicity, and diabetes, as well as impacting animals upon exposure[102]. There are numerous machine learning tasks primarily using spectral information on leaves and roots for heavy metal stress assessment. These are summarized in Table 1.5.

Jianhong Zhang et al.[103] propsed a work comprising of three steps namely, decomposition of the spectrum, selection of sensitive bands and classification. First, multivariate empirical mode decomposition (MEMD) is applied to decompose the original spectrum. This effectively eliminates noise while enhancing and amplifying subtle spectral information. Subsequently, techniques such as the successive projections algorithm (SPA), competitive adaptive reweighted sampling (CARS), and iteratively retaining informative variables (IRIV) are utilized to filter characteristic bands, reduce redundant data, and pinpoint essential spectral details. Finally, to classify the types of Cu and Pb elements, machine learning algorithms including extreme learning machine (ELM), support vector machine (SVM), and general regression neural network (GRNN) are employed to construct models. Notably, the MEMD-IRIV-SVM model is found to be well-suited for distinguishing Cu and Pb species.

Sensor/Modality	ML Models	Plant Tissue	Plant Species	ML Purpose	Heavy Metal and Reference
Hyperspectral	SVM, ELM, GRNN	Leaves	Corn	Classification	Copper (Cu) and Lead (Pb)[103]
LIBS	PCA, SVM, RF, SIMCA, ELM, KNN , LS-SVM, PLSR, RBFNN	Stems	Rice	Clustering, Classification, Regression	Cadmium(Cd) [104]
Visible and Near-Infrared (Vis-NIR) spectroscopy	LDA, RBFNN	Leaves	Tea	Classification	Lead(Pb) [105]
LIBS, HSI	PLSR, LS-SVM, ELM and RBFNN	Root	Rice	Regression	Cadmium(Cd) [106]
HSI	CNN, PLS-DA, LS-SVM, RF, ELM	Leaves	Apple Rootstocks	Classification	Copper(Cu) [107]
HSI	ELS, PLS, LS-SVM	Leaves	Rice	Classification	Cadmium(Cd) [108]
HSI	PCA, PLS-DA, LS-SVM	Leaves	Tobacco	Classification	Mercury(Hg) [109]
HSI	PCA, Stacked-Autoencoder, SVM	Leaves, root	Rape Oilseed	Classification	Lead(Pb) [110]
LIBS-Raman Spectroscopy	LS-SVM, MLP, RBFNN, PNN	Root, leaves	wheat	Classification	Lead(Pb) [111]

Table 1.5: Heavy Metal Stress and Machine Learning

Presently, the primary emphasis in heavy metal detection in plants using Laser-Induced Breakdown Spectroscopy (LIBS) is centered on leaves. In contrast, when it comes to the detection of heavy metals in plant roots, the methods predominantly rely on Raman and Hyperspectral Imaging (HSI) techniques, with limited incorporation of LIBS. LIBS-based approaches for heavy metal detection in plant leaves offer a notable advantage in terms of speed, typically taking less than 10 minutes, highlighting the spectroscopy's

capacity for rapid analysis[106]. Furthermore, in data processing, chemometrics proves to be more advantageous compared to linear fitting, especially considering the matrix effects involved in the analysis[106][104].

1.1.4.5 Combined Stress

In natural environments, plants are often subject to multiple stresses. It should be noted that a single or numerous stress sources, as well as biotic or abiotic stress combinations, can cause plants to have very similar physiological responses, making their evaluation problematic (Blum, 2016). To date, few studies have focused on disentangling environmental stress sources (Poblete et al., 2021). Although studies have shown that spectral screening methods can disentangle abiotic and biotic stress sources, most studies at this stage are still focused on a single stress level. Detection of coexisting stresses remains challenging and under-explored (Zhang et al., 2019). Farmers and breeders have long known that often it is the simultaneous occurrence of several abiotic stresses, rather than a particular stress condition, that is most lethal to crops[5]. Heat and drought stress co-exist. The flowering and grain-filling stages of crops are vulnerable to heat and drought stress, causing substantial yield losses. To comprehend how crops respond to these stresses, understanding key physiological traits is crucial. Phenotyping these traits necessitates advanced sensors, high-quality imagery, and machine learning techniques[112].

1.1.5 Multi-Modal Approaches

As we come across variety of data such as spatial, spectral, temporal and metabolite, there is ample scope for the fusion of these data. This integration approach is also known as multi-modal information fusion. Along with machine learning, it can be employed to analyze data sets with multiple sources (i.e., rainfall, temperature, multi-spectral image, soil data) where as each data type is a modality that will be analyzed and combined to increase model performance [113]. Sagi Levanon et al. [114] combined the RGB and thermal images of Banana Plantlets and given as input to a CNN model to predict the water stress. It demonstrates the integrated approach outperform the single modality approach. A recent work suggests that the combination of RGB and thermal/multispectral and ML applications can significantly contribute to monitoring responses to flooding stress (Zhou et al., 2021). Yukimasa Kaneda et al. [115] proposed a novel multi-modal sliding window-based support vector regression (multi-modal SW-SVR) method for accurate prediction of complicated water stress, which is a plant status, from two data types, namely environmental and plant image data. A combination of both RGB and Hyperspectral imaging methods can optimize a comprehensive assessment of the root system[116]. RGB images can be combined with other data sources, such as multispectral or thermal imagery, to provide a more comprehensive assessment of plant health and stress[7]. Plant metabolites

can both influence and reflect the plant's phenotype. Hence, there is substantial merit in the integration of comprehensive physical and spectral data with additional chemical information over time[117].

1.1.6 Spatio Temporal Studies

The role of spatiotemporal plant stress phenotyping is to monitor and analyze how various environmental stresses affect plants over both space and time. This approach involves assessing and understanding the spatial distribution and temporal progression of stress-induced changes in plant morphology, physiology, and overall health. In order to account for temporal information, various probabilistic and computational models (e.g. Hidden Markov Models (HMMs) Conditional Random Fields (CRFs) and RNNs have been used for a number of applications involving sequence learning and processing [118]. RNNs (and LSTMs in particular) are able to grasp and learn long-range and complex dynamics and have emerged as an effective technique for stress phenotyping. A modified version of the Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) network is employed to extract spatiotemporal patterns from the chickpea plant dataset. These patterns are subsequently utilized for the classification of water stress conditions[50]. Abdalla et al. encoded spatiotemporal information of plants in a single time-series model to evaluate the nutrient status of oilseed rape more efficiently[98] . Khanna et al. demonstrated that by using spatio-temporal spectral data, it is feasible to develop precise classification models to ascertain the identification of drought, nitrogen levels, and weed stress presence and severity[7]. Monitoring plant trait changes over time allows for the early detection of stress factors like drought, salinity, or extreme temperatures. Consequently, spatio-temporal studies are emerging as a pivotal approach for abiotic stress assessment

1.2 Research Gaps

Several research gaps are identified in the literature related to machine learning and abiotic stress phenotyping:

- **Lack of lighter models and ensemble frameworks:** Popular convolutional neural network (CNN) architectures are often computationally intensive. There is a need to explore and develop lighter models that can still perform well in ensemble frameworks, particularly when dealing with smaller datasets to avoid overfitting.
- **Lack of interpretable machine learning methods:** Interpretability is a critical aspect of machine learning, especially in applications where decisions impact human lives or have significant consequences. Developing interpretable models is crucial for understanding and trusting the decisions made by these models. This remains

an ongoing challenge in the field.

- **Limited research on combined abiotic stresses:** Most existing studies focus on individual stress factors, whereas real-world scenarios often involve multiple, interacting stresses.

1.3 Objectives

Considering the research gaps and based on the state-of-the-art studies, the following objectives are formulated to conduct the PhD thesis work:

- **Objective 1:** Explainable lightweight deep learning pipeline for improved drought stress identification
- **Objective 2:** An explainable Vision Transformer with transfer learning for efficient drought stress identification
- **Objective 3:** Gradient-Guided Unlearning in a Novel Lightweight Hybrid CNN for Enhanced Drought Stress Identification
- **Objective 4:** Improved Classification of Nitrogen Stress Severity in Plants Under Combined Stress Conditions Using Spatio-Temporal Deep Learning Framework

1.4 Organization of Thesis

The doctoral thesis is organized in seven chapters as follows:

- **Chapter 2:** It devises and investigates lightweight CNN frameworks using UAV-acquired RGB imagery to identify drought stress in potato crops, integrating transfer learning for improved performance and gradient-based explainability for interpretability.
- **Chapter 3:** In this chapter, we propose the use of Vision Transformers combined with transfer learning and support vector machine classifiers for enhanced drought stress identification in potatoes, accompanied by attention maps for model transparency.
- **Chapter 4:** We developed a novel hybrid lightweight CNN architecture inspired by ResNet, DenseNet, and MobileNet, incorporating a gradient influence-based machine unlearning mechanism to reduce model size and enhance adaptability without sacrificing accuracy.
- **Chapter 5:** In this chapter, we propose a spatio-temporal deep learning model

combining MobileNetV2 and LSTM to classify nitrogen stress severity in sugar beet under combined drought and weed pressures, leveraging multi-modal imaging data for superior accuracy.

- **Chapter 6:** In this chapter, we summarize the key findings, discuss the overall contributions and outline the future research directions.

Chapter 2

Explainable Lightweight Deep Learning Pipeline for Improved Drought Stress Identification

2.1 Abstract

Early identification of drought stress in crops is vital for implementing effective mitigation measures and reducing yield loss. Non-invasive imaging techniques hold immense potential by capturing subtle physiological changes in plants under water deficit. Sensor-based imaging data serves as a rich source of information for machine learning and deep learning algorithms. While these approaches yield favorable results, real-time field applications require algorithms specifically designed for the complexities of natural agricultural conditions. Our work proposes a novel deep learning framework for classifying drought stress in potato crops captured by unmanned aerial vehicles (UAV) in natural settings. The novelty lies in the synergistic combination of a pre-trained network with carefully designed custom layers. This architecture leverages the pre-trained network's feature extraction capabilities while the custom layers enable targeted dimensionality reduction and enhanced regularization, ultimately leading to improved performance. A key innovation of our work is the integration of gradient-based visualization inspired by Gradient-Class Activation Mapping (Grad-CAM), an explainability technique. This visualization approach sheds light on the internal workings of the deep learning model, often regarded as a "black box". By revealing the model's focus areas within the images, it enhances interpretability and fosters trust in the model's decision-making process. Our proposed framework achieves superior performance, particularly with the DenseNet121 pre-trained network, reaching a precision of 97% to identify the stressed class with an overall accuracy of 91%. Comparative analysis of existing state-of-the-art object detection algorithms reveals the superiority of our approach in achieving higher precision and accuracy. Thus, our explainable deep learning framework offers a powerful approach to drought stress identification with high accuracy and actionable insights.

Keywords: {Stress Pheno-typing, Drought Stress, Machine Learning, Deep Learning, Transfer Learning, Convolutional Neural Network}

2.2 Introduction

Abiotic stress adversely affects crop development, yield, and product quality, with drought stress being among the most critical constraints [119, 120]. Drought not only reduces plant productivity but also aggravates other stresses such as salinity, heat, nutrient deficiency, and pathogen attack, thereby amplifying damage to crops and soil biota. Early detection of drought stress is therefore essential to implement timely mitigation strategies such as irrigation, maximizing yield potential. However, the diverse physiological and biochemical responses induced by drought—operating at cellular and whole-plant levels—make early diagnosis increasingly challenging [27].

Spectral properties of plants, particularly absorption and reflectance in the visible and near-infrared (NIR) regions, are closely associated with stress responses [29]. This has driven the adoption of imaging technologies as non-invasive, high-throughput tools for stress phenotyping [10]. Various imaging modalities, including RGB imagery [18], thermal imaging [19], fluorescence imaging [20], multispectral, and hyperspectral imaging [1], have demonstrated considerable success in stress detection.

The recent convergence of imaging with artificial intelligence (AI) has further advanced plant stress phenotyping. Traditional machine learning (ML) approaches, such as decision trees, random forests, support vector machines, and boosting algorithms, have provided important benchmarks in stress classification [9, 121, 17]. Nevertheless, their reliance on manual feature extraction limits generalizability and scalability in real-world field conditions. Deep learning (DL), particularly convolutional neural networks (CNNs), overcomes this limitation by automatically learning hierarchical features from raw image data, significantly improving classification performance [122, 123].

Recent studies demonstrate substantial progress in applying ML and DL to drought stress detection across diverse crops and imaging modalities. For instance, segmentation-based feature extraction followed by gradient boosting decision tree (GBDT) has been applied for maize water stress detection [42], while deep convolutional neural networks (DCNN) showed superior performance on the same dataset [43]. Hyperspectral imaging combined with SVM, RF, and XGBoost has been used for groundnut stress classification [47], whereas transfer learning with DenseNet-121 enabled severity-level classification in soybean [38]. CNN–LSTM combinations have been implemented for chickpea water stress detection [21], and CNN architectures such as AlexNet, GoogLeNet, and Inception V3 have been tested in maize, okra, and soybean, with GoogLeNet showing the highest accuracy [51]. Other notable approaches include chlorophyll fluorescence imaging of wheat using RF and extra trees [53], hyperspectral regression in tea canopies [60], derivative spectral analysis [48], and custom CNN frameworks for maize outperform-

ing state-of-the-art architectures [63]. Aerial imagery of potato canopies has also been integrated into DL pipelines for stress identification [61].

While these advances highlight the superior performance of DL over conventional ML in drought stress detection, the lack of interpretability remains a major limitation. Few studies, such as explainable CNN models for soybean stress [25] or attribution-based frameworks [65], have addressed this gap. Building on this, the present work introduces a novel lightweight deep learning pipeline for potato drought stress detection, integrating transfer learning to overcome data limitations and gradient-based visualization for interpretability. The key contributions of the work are: 1) A transfer learning-based model that effectively leverages knowledge from larger datasets to address the limitations of smaller potato crop stress datasets, overcoming challenges like overfitting, 2) A lightweight DL pipeline specifically designed to enhance stress identification in potato crops, 3) Integration of Gradient-based visualisation for model explainability, highlighting the image regions most relevant to stress detection.

2.3 Materials and Methods

2.3.1 Data Set Description

The potato crop aerial images utilized in this study have been sourced from a publicly accessible dataset that encompasses multiple modalities [124, 61]. Collected from a field at the Aberdeen Research and Extension Center, University of Idaho, these images serve as valuable resources for training machine learning models dedicated to crop health assessment in precision agriculture applications. Acquired using a Parrot Sequoia multispectral camera mounted on a 3DR Solo drone, the dataset features an RGB sensor with a resolution of $4,608 \times 3,456$ pixels and four monochrome sensors capturing narrow bands of light wavelengths: green (550nm), red (660nm), red-edge (735nm), and near-infrared (790nm), each with a resolution of $1,280 \times 960$ pixels. The drone flew over the potato field at a low altitude of 3 meters, with the primary objective of capturing drought stress in Russet Burbank potato plants attributed to premature plant senescence.

The dataset comprises of 360 RGB image patches in JPG format, each measuring 750×750 pixels. These patches were derived from high-resolution aerial images through cropping, rotating, and resizing operations. Data augmentation was applied to an initial set of 300 images as per procedure in Butte et al. [61], expanding the training dataset to 1,500 images. The remaining 60 images were reserved exclusively for testing. No data augmentation was performed on the test images to ensure an unbiased evaluation. Training classification models requires labeled data with annotated regions of interest. In this study, the targets were regions containing healthy and stressed potato

(a) (b)

Figure 2.1: Field images showing a) Sample RGB image and b) Healthy and Stressed plants.

plants. These two conditions were visually distinguishable based on color—healthy plants appeared green, whereas stressed plants exhibited a yellowish hue. Manual annotation was performed using the open-source graphical tool LabelImg [125], allowing bounding boxes to be drawn around both healthy and stressed regions. The resulting annotations, including class labels and bounding box coordinates, were saved and used to generate ground truth data for training the proposed models.

Additionally, the dataset includes corresponding image patches from spectral sensors with red, green, red-edge, and near-infrared bands, each sized 416×416 pixels. However, we are only utilizing the RGB images due to the limitations of the low-resolution monochromatic images.

The augmented dataset, consisting of 1,500 images, was used for training, while the test set included 60 distinct images. From both training and test images, annotated windows (i.e., rectangular patches) were extracted based on the bounding-box annotations. Each extracted window was labeled as either “healthy” or “stressed”, corresponding to the visual condition of the crop. As illustrated in Fig. 2.1, the original image is shown in Fig.2.1a, and the corresponding extracted windows are depicted in Fig. 2.1b. In this example, six windows represent healthy regions, while three represent stressed regions. The final count of training images for the ”stressed” and ”healthy” classes were 11,915 and 8,200, respectively. The evaluation of the model was performed on a specific test set comprising 60 images, from which 401 healthy images and 734 stressed images were extracted using the bounding boxes similar to the process used for the training image set.

2.3.2 Proposed Methodology

We present an integrated approach for drought stress classification, featuring a CNN-based pipeline with transfer learning and an interpretability technique to enhance model transparency. This methodology combines data augmentation, transfer learning, and a CNN architecture for robust feature extraction, followed by explainability methods that leverage gradients to provide insights into the model's decision-making process. The methodology is structured as follows:

2.3.2.1 Deep Learning Pipeline with Transfer Learning

The proposed framework uses CNN-based architecture with transfer learning to differentiate between drought-stressed and healthy plants. Transfer learning enables the model to start with a pre-trained network, reducing training time and improving accuracy, especially with smaller datasets. The model is divided into three key components: data augmentation, a pre-trained network, and additional layers for final classification. The pipeline is depicted in Fig. 2.2.

- **Data Augmentation:** It tackles the challenge of limited training data by artificially expanding the dataset with variations of existing samples. This approach injects variability and improves the model's generalization ability to unseen data. Transformations like re-scaling, shearing, rotating, shifting, and flipping are applied to create a more diverse training set. This robustness to variations helps the model perform better on real-world data and reduces the risk of over-fitting.

- **Pre-trained Network:** Transfer learning is employed to speed up the training process and improve accuracy by starting with a pre-trained CNN. The pre-trained model serves as the backbone of the architecture, effectively extracting low-level and mid-level features from the images. Networks like EfficientNetB0, MobileNet, DenseNet121, and NASNetMobile, trained on vast and diverse datasets such as ImageNet, are repurposed to recognize drought stress by fine-tuning them for this specific task.

- **Additional Layers:** Two types of layers are applied after the pre-trained architecture: the Global Average Pooling Layer and Dense Layers.

  - **Global Average Pooling:** It reduces the dimensionality of spatial data (like feature maps from convolutional layers) into a single feature vector. It achieves this by calculating the average of all elements within each feature map, resulting in one value per feature map.

  - **Dense layers:** Two fully connected dense layers are stacked sequentially af-

ter global average pooling. These layers perform computations to learn complex relationships between the features extracted by the pre-trained network. Dropout and L2 regularization are applied between each dense layer to prevent over-fitting. Dropout randomly drops a certain percentage of neurons during training, forcing the model to learn from different subsets of features and reducing its reliance on any specific feature. L2 regularization penalizes large weights in the model, discouraging the model from becoming overly complex. Each neuron in the dense layer applies a weighted sum and activation function (ReLU) to determine the probability of an image belonging to a particular class.

- **Output Layer:** This layer uses a sigmoid activation function to generate the final probability scores between 0 and 1, indicating stressed or healthy.

In essence, the model incorporates data augmentation to enrich the training data, takes advantage of a pre-trained network's feature extraction capabilities, and uses dense layers with regularization and dropout to learn a classification boundary between stressed and healthy images.

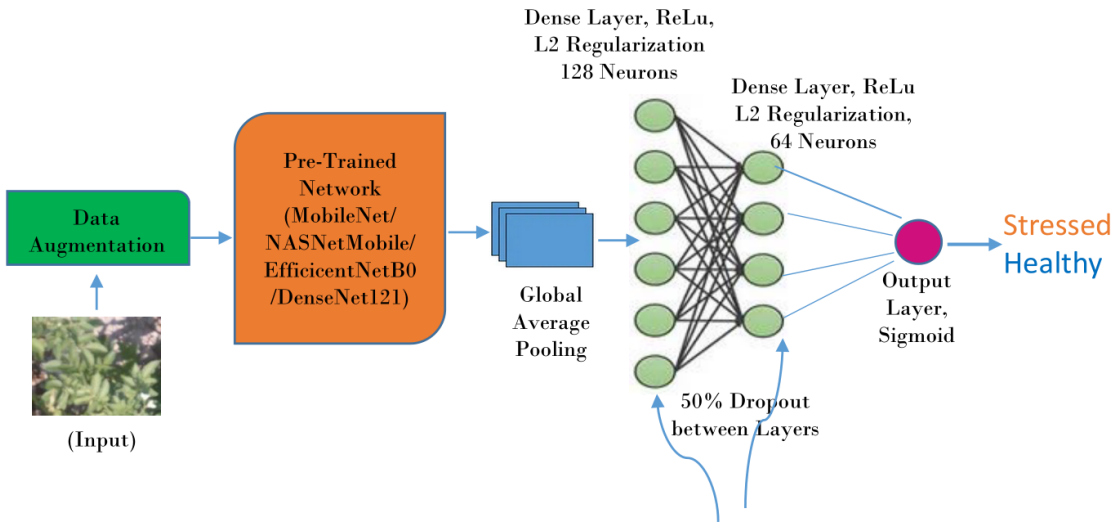


Figure 2.2: Deep Learning Framework for Drought Stress Identification

2.3.2.2 Explainability through Gradient-based Visualisation

We integrated a gradient-based explainable approach into our framework to ensure model transparency and to enhance interpretability. It is inspired by Grad-CAM [126], a technique that provides visual explanations by highlighting the regions of input images that contribute most to the model's predictions. While Grad-CAM focuses on class-specific, high-level features, gradient-based visualization emphasizes pixel-level sensitivities, offering a broader perspective on what influences the model's output. By visualizing the

areas most relevant to the model's decision, the devised explainable approach offers valuable insights into the decision-making process of the deep learning model. The practical application involves taking an input image that the deep learning pipeline can classify as healthy or stressed. According to the trained model, we can then use the identified stressed image to locate the specific areas of the field that are affected by stress. The following steps are involved in the proposed explainable approach, which takes its cue from Grad-CAM.

1. **Forward Pass**: The model output θ is computed by performing a forward pass through the deep learning model, represented as:

$$\theta = \phi(\xi)$$

where:

- θ represents the model output.
- $\phi(\cdot)$ represents the deep learning model.
- ξ represents the input image.

2. **Compute Gradients**: The gradients of the model output with respect to the input image are calculated, represented as:

$$\nabla_{\xi}\theta = \frac{\partial\theta}{\partial\xi}$$

where:

- $\nabla_{\xi}\theta$ represents the gradients of the model output with respect to the input image.
- $\frac{\partial\theta}{\partial\xi}$ represents the partial derivatives of the model output with respect to the input image.

3. **Gradient Visualization**: The absolute gradients are computed and visualized as a heatmap, represented as:

$$\text{Heatmap} = \text{abs}(\nabla_{\xi}\theta)$$

where:

- Heatmap represents the heatmap visualization of the gradients.
- abs$(\cdot)$ represents the absolute value function.

4. **Standardization**: The heatmap is optionally standardized by subtracting the mean and dividing by the standard deviation to improve visualization, represented as:

$$\text{Heatmap}_{\text{std}} = \frac{\text{Heatmap} - \mu}{\sigma}$$

where:

- $\text{Heatmap}_{\text{std}}$ represents the standardized heatmap.
- μ represents the mean of the heatmap.
- σ represents the standard deviation of the heatmap.

5. **Plotting**: Finally, the input image and the heatmap are plotted side by side for visualization.

Thus, the explainable approach based on Grad-CAM leverages the strength by analyzing gradients to pinpoint image regions crucial for the model's decisions, offering valuable insights into what triggers the model's stress responses.

2.3.3 Evaluation Metrics

The model's performance underwent assessment using various evaluation metrics, including accuracy, precision, and recall (sensitivity). These metrics are computed based on the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), which collectively form a 2x2 matrix known as the confusion matrix. The format is illustrated in Fig. 2.3, where the negative class represents the "healthy" class, and the positive class corresponds to the stressed class.

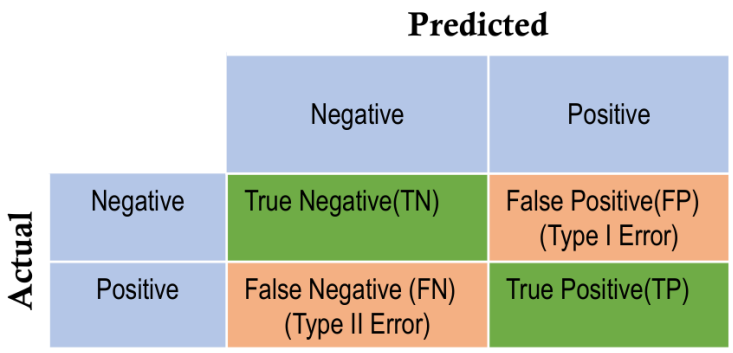


Figure 2.3: Confusion Matrix

In this matrix, TP and TN indicate the accurate predictions of water-stressed and healthy potato crops, respectively. FP, termed type 1 error, denotes predictions where the healthy class is inaccurately identified as water-stressed. FN, referred to as type 2 error, represents instances where water-stressed potato plants are incorrectly predicted as

healthy. The classification accuracy is a measure of the ratio between correct predictions for stressed and healthy images and the total number of images in the test set. Precision is the ratio of true positives to the sum of true positives and false positives, indicating the proportion of correctly identified positive instances out of all instances predicted as positive. Recall (sensitivity) is the ratio of true positives to the sum of true positives and false negatives, reflecting the model's ability to correctly identify all positive instances. The formulas for accuracy, precision and recall are given below.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Population}}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

2.3.4 Model Workflow

The proposed pipeline is a comprehensive framework that involves model training, evaluation, and explainability to provide a robust and transparent solution for identifying stressed plants in field images. It is demonstrated in Fig. 2.4. The training phase of the pipeline utilizes a dataset of 1500 augmented field images, each annotated with bounding boxes to delineate regions of healthy and stressed plants. These annotated windows were extracted from each augmented image, resulting in separate "healthy" and "stressed" classes with 8,200 and 11,915 images, respectively. The dataset is divided into 80% for training and 20% for validation to prevent over-fitting. In the testing phase, a distinct testing dataset comprising 60 field images are employed to evaluate the model's performance on unseen data. The evaluation is conducted on a test set of 60 images, with 401 healthy and 734 stressed images extracted using bounding boxes. The model's performance is assessed using standard evaluation metrics such as accuracy, precision, and recall. Then, to understand the model's decision-making process, the pipeline incorporates a devised explainable approach based on gradients. It involves using an already identified stressed image as input, leveraging the trained model and gradient-based visualization techniques to generate heatmaps highlighting the areas of the image the model identifies as affected by stress. These heatmaps provide valuable insights into the model's reasoning and can help identify the visual cues that indicate plant stress.

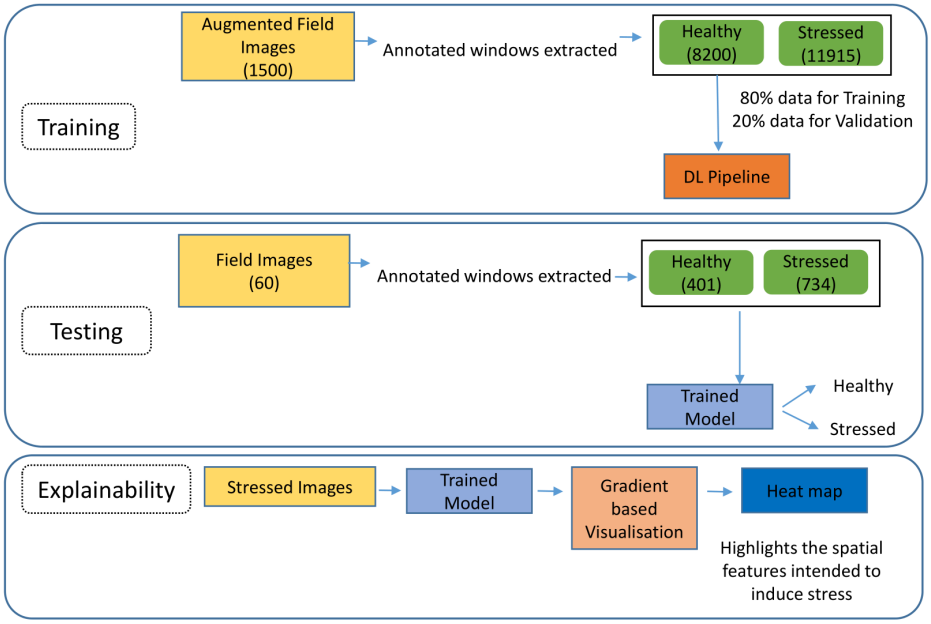


Figure 2.4: Workflow of the Model

2.4 Results and Discussion

The methodology employed in this study utilizes transfer learning, leveraging knowledge from models trained on the 'ImageNet' dataset and adapting it to address drought stress identification. By using pre-trained networks as a foundation rather than starting from scratch, the approach reduces storage requirements and computational demands. This approach results in a lightweight model, with trainable parameters ranging from 3.3 million to 7.36 million across various pre-trained networks, a notable departure from the considerably heavier models typically used in deep learning tasks. Specifically, the trainable parameters for EfficientNetB0, MobileNet, DenseNet121, and NASNetMobile are 4.18 million, 3.35 million, 7.09 million, and 4.37 million, respectively, as depicted in Fig. 2.5.

In our deep learning framework, *Python* version 3.8.8 serves as the programming language foundation, while *TensorFlow* and *Keras*, widely recognized and utilized libraries, are employed for model development and training. Additionally, various libraries such as *os*, *pandas*, *numpy*, and *sklearn* were employed to facilitate data manipulation and metric calculations.

In the proposed deep learning framework, the input data undergoes augmentation through various transformations. Four pre-trained networks are systematically evaluated, each serving as a backbone feature extractor. Additional layers are stacked on top of these networks to complement their ability to identify drought stress in images collected from natural settings. The following discussion provides an in-depth analysis

of the parameters, the pipeline's performance based on learning curves and confusion matrices, and the model's explainability by identifying stressed spatial features in field images. In addition, our approach is compared to previous works based on object detection algorithms, demonstrating that the proposed method outperforms them.

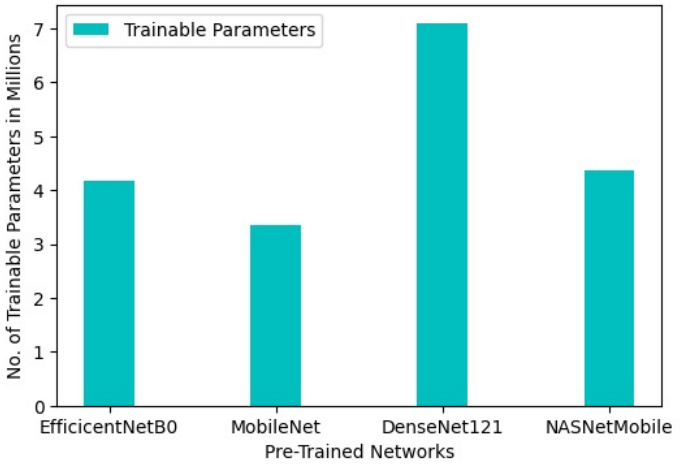


Figure 2.5: Number of Trainable Parameters of the Model with different Pre-trained CNN Architectures

2.4.1 Model Parameters

The input dataset is divided into two subsets for training and validating, utilizing a fixed random seed of 42. The $random\_state = 42$ parameter ensures re-producibility by setting a specific random seed, guaranteeing consistent results across different runs of the code. Separate generators are created for training, validation, and testing datasets using the $ImageDataGenerator$ function from $Keras$. Each generator is configured with specific settings tailored to the respective pre-trained architectures: EfficientNetB0, MobileNet, DenseNet121, and NASNetMobile within the deep learning framework as discussed in section 2.3.2.1. The target image sizes are set to 224x224 for EfficientNetB0, MobileNet, and DenseNet121, and 299x299 for NASNetMobile. The re-scaling factor, batch size, and class mode are standardized across all architectures, with values of 1/255, 128, and *binary*, respectively. The training generator is also equipped with data augmentation transformations to enhance the dataset's variability and improve model generalization. Key parameters governing these transformations, including the shear range, rotation range, width shift range, and height shift range, are configured as 0.2, 30, 0.2, and 0.2, respectively. Horizontal and vertical flipping are enabled with boolean values set to $True$ for both, while the fill mode is specified as *nearest*.

The proposed custom architecture builds on the pre-trained network by adding several layers. It begins with global average pooling, followed by two dense layers utilizing 128 and 64 neurons, respectively. Each dense layer utilizes ReLU activation for efficient

learning, dropout with a 50% rate to prevent over-fitting, and L2 regularization with a weight decay of 0.01 to further enhance robustness during feature extraction. The final layer of the network comprises a single neuron with sigmoid activation, outputting a value between 0 and 1, representing the probability of the input belonging to a specific class. The Adam optimizer is employed for training, starting with an initial learning rate of 0.001. An exponential decay schedule is applied to adjust the learning rate over epochs. This schedule gradually reduces the learning rate after every two epochs with a decay rate 0.9. The chosen loss function is *binary cross-entropy*, which measures the difference between the predicted probabilities and the actual class labels.

A callback function is utilized using *ModelCheckpoint* from *Keras* to save the best-performing version of the model during training. This callback monitors the validation loss and saves the model only when a new minimum validation loss is achieved. After training, the code identifies the epoch with the lowest validation loss and loads the corresponding model weights. These weights are then utilized to evaluate the model's performance on a separate test dataset. This strategy ensures that the model evaluated on unseen data represents the optimal performance attained during training.

2.4.2 Performance of the Model

We investigated four pre-trained networks individually as part of the proposed deep learning pipeline. While EfficientNetB0 and NASNetMobile achieved high training accuracies of 99.38% and 98.47%, respectively, their validation and test accuracies were notably lower, indicating potential weaknesses as evidenced by their loss and accuracy learning curves, which is discussed later in the section. In contrast, MobileNet demonstrated impressive performance with a training accuracy of 99.81%, a validation accuracy of 99.33%, a test accuracy of 88.72%, and a low validation loss of 0.033. Similarly, DenseNet121 showcased robust performance across training, validation, and test sets, achieving a training accuracy of 99.69%, a validation accuracy of 98.86%, and a test accuracy of 90.75%. Overall, DenseNet121 emerged as the best-performing model among those investigated, boasting the highest test accuracy, closely followed by MobileNet. The comparative performance is summarized in Table 2.1. Epochs in training are chosen based on observing the convergence pattern of the model, typically by monitoring performance metrics on a validation dataset. The training continues until the model's performance on the validation set plateaus or degrade, indicating convergence and preventing over-fitting. The deep learning pipeline was trained with EfficientNetB0, MobileNet, DenseNet121, and NASNetMobile for 30, 60, 60, and 30 epochs, respectively. The optimal performance for each model was achieved at epochs 30, 59, 55, and 28, correspondingly.

Analysis of learning curves: Analyzing the learning curves for training and

Table 2.1: Model Performance

Model	Train Acc	Val Acc	Val Loss	Test Acc	No. Epoch	Best Result Epoch
EfficientNetB0	99.38	74.30	0.5033	74.00	30	30
MobileNet	99.81	99.33	0.0330	88.72	60	59
DenseNet121	99.69	98.86	0.0508	90.75	60	55
NASNetMobile	99.47	59.81	0.6815	64.67	30	28

validation loss and training and validation accuracy offers valuable insights into how the model performs and behaves throughout the training process when employing different pre-trained networks. For EfficientNetB0, as illustrated in Fig. 2.6a, the training loss stabilizes at a low value, indicating that the model has learned most of the patterns in the training data and is not finding any substantial new information. On the other hand, the fluctuating validation loss indicates that the model's performance on unseen data (the validation set) is inconsistent, suggesting potential over-fitting or instability during training. Furthermore, the training accuracy remains consistently high, as shown in Fig. 2.7a. In contrast the validation accuracy fluctuates more, implying that the model performs well on the training data but struggles to generalize effectively to unseen data. Additionally, the noticeable gap between the training and validation accuracy further suggests over-fitting, where the model becomes too specialized to the training data and fails to generalize well to new data.

For NASNetMobile, as depicted in Fig. 2.6d, the learning curves for training and validation loss reveal evidence of over-fitting, given the considerable gap between the two curves. Regarding training and validation accuracy learning curves, a similar pattern is observed, as shown in Fig. 2.7d. This suggests that while these models perform well on the training data, their performance on unseen validation data is substantially lower.

For DenseNet121, the trends observed in the loss graphs indicate that the model is effectively learning from the data. Both training and validation loss curves (i.e., Fig. 2.6c) demonstrate a consistent decrease over time. While there is an initial gap between the training and validation loss curves, this gap gradually diminishes as the training progresses. This narrowing gap suggests the model is improving its generalization ability to unseen data. Additionally, the validation accuracy steadily increases throughout the training process and remains closely aligned with the training accuracy (i.e., Fig. 2.7c), indicating the model's positive performance on both training and validation sets. The performance of MobileNet exhibits a similar trend, where the loss graphs indicate effective learning by the model. Both training and validation loss curves (i.e., Fig. 2.6b) depict a consistent decrease over time. Nonetheless, a noticeable gap persists between the training and validation loss curves, suggesting a potential for over-fitting, although not severe, given the concurrent increase in validation accuracy (i.e., Fig. 2.7b). This

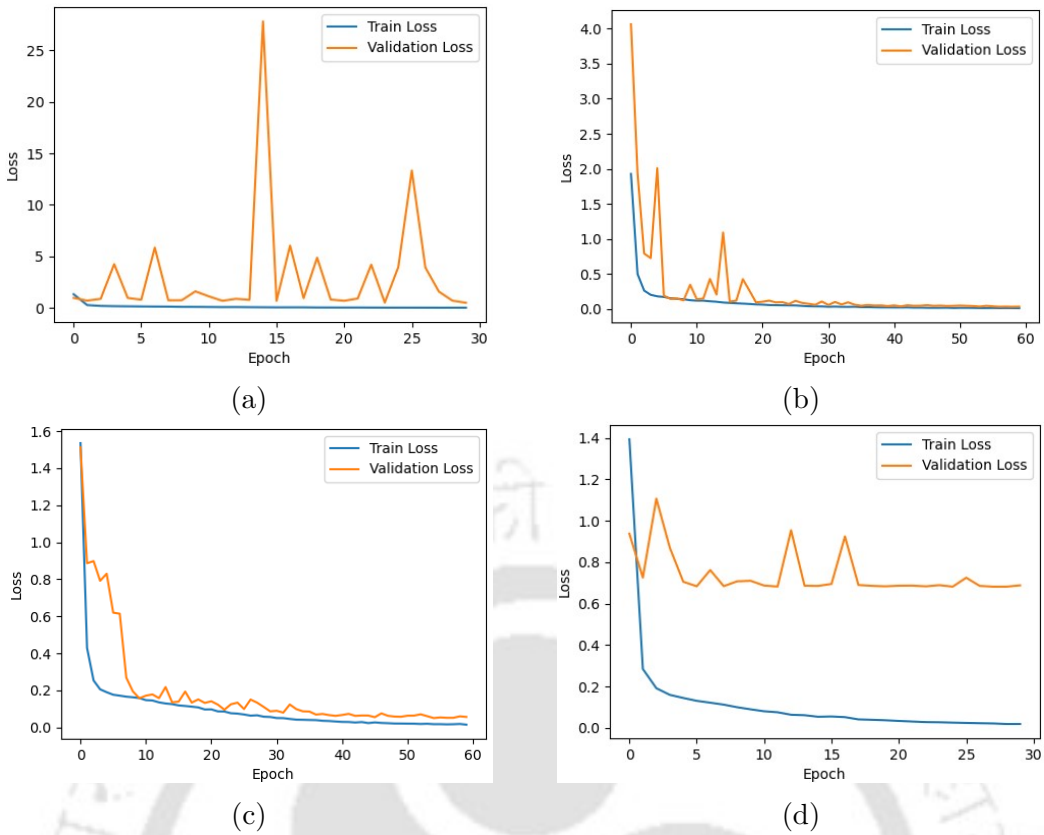


Figure 2.6: Training Loss vs Validation loss of the Model for the various Pre-trained Networks: a) EfficientNetB0, b) MobileNet, c) DenseNet121, and d) NASNetMobile.

indicates that the model is still able to generalize well to unseen data, despite the observed gap between the loss curves.

Analysis of Confusion Matrices: The analysis of confusion matrices is essential to provide deeper insight into the model's performance beyond accuracy. The deep learning pipeline utilizing various pre-trained CNN models were evaluated on an independent test set of 1,135 images that were not part of the training process. The model generated the confusion matrices shown in Fig. 2.8 using these already trained networks. The values within each confusion matrix were arranged according to the layout shown in Fig. 2.3. Following the similar pattern observed in the previous learning curves, the EfficientNetB0 and NASNetMobile showed the poorest performance on the test dataset. For EfficientNetB0, analysis of the confusion matrix (Fig. 2.8a) reveals a total of 295 misclassified instances out of 1135 predictions, comprised of 138 false positives (FP) and 157 false negatives (FN). This results in a misclassification rate of 26%. In contrast, the confusion matrix for NASNetMobile (Fig. 2.8d) indicates a strange behavior where the model correctly identifies all stressed images but fails to recognize any healthy ones. In the case of EfficientNetB0, the higher misclassification rate suggests suboptimal performance across both classes. Conversely, NASNetMobile's performance is characterized by a notable bias towards the "stressed" class, resulting in a complete

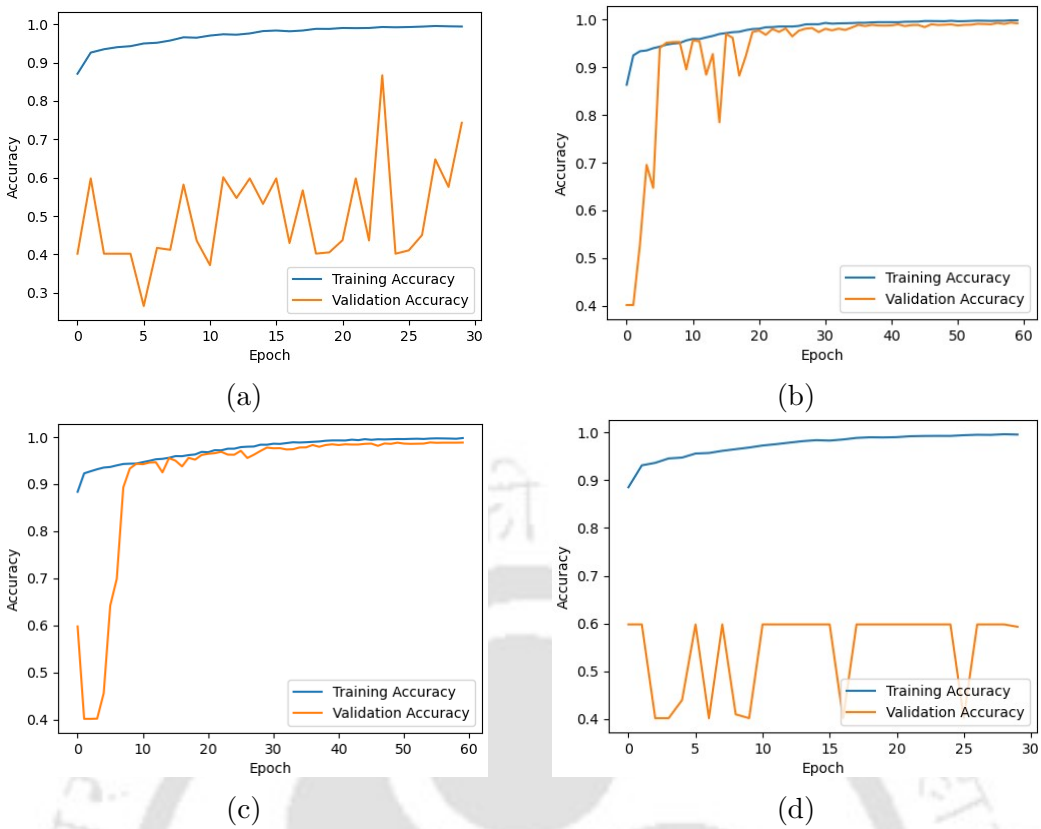


Figure 2.7: Training vs Validation accuracy of the Model for the various Pre-Trained Networks: a) EfficientNetB0, b) MobileNet, c) DenseNet121, and d) NASNetMobile.

oversight of the "healthy" class. Both scenarios are deemed undesirable, rendering the models ineffective for their intended purpose. On the other hand, both MobileNet and DenseNet121 achieve very low misclassification rates between healthy and stressed classes, as shown by the minimal Type I and Type II errors in their respective confusion matrices (Fig. 2.8b and Fig. 2.8c). This translates to high overall accuracies of 88.72% for MobileNet and 90.75% for DenseNet121.

DenseNet121 stands out as the top-performing backbone in the proposed deep learning pipeline, strengthened by data augmentation and additional layers. Analysis of both learning curves and confusion matrices shows that it generalizes better on unseen data and distinguishes healthy and stressed classes more effectively than EfficientNetB0, MobileNet, and NASNetMobile.

2.4.3 Explaining the Model

We employ a method to generate visual explanations for decisions made by the proposed deep learning pipeline, enhancing its transparency. We utilize the DenseNet121 pre-trained network in our pipeline due to its superior performance compared to other networks. The explanations are derived from analyzing gradients at two distinct stages

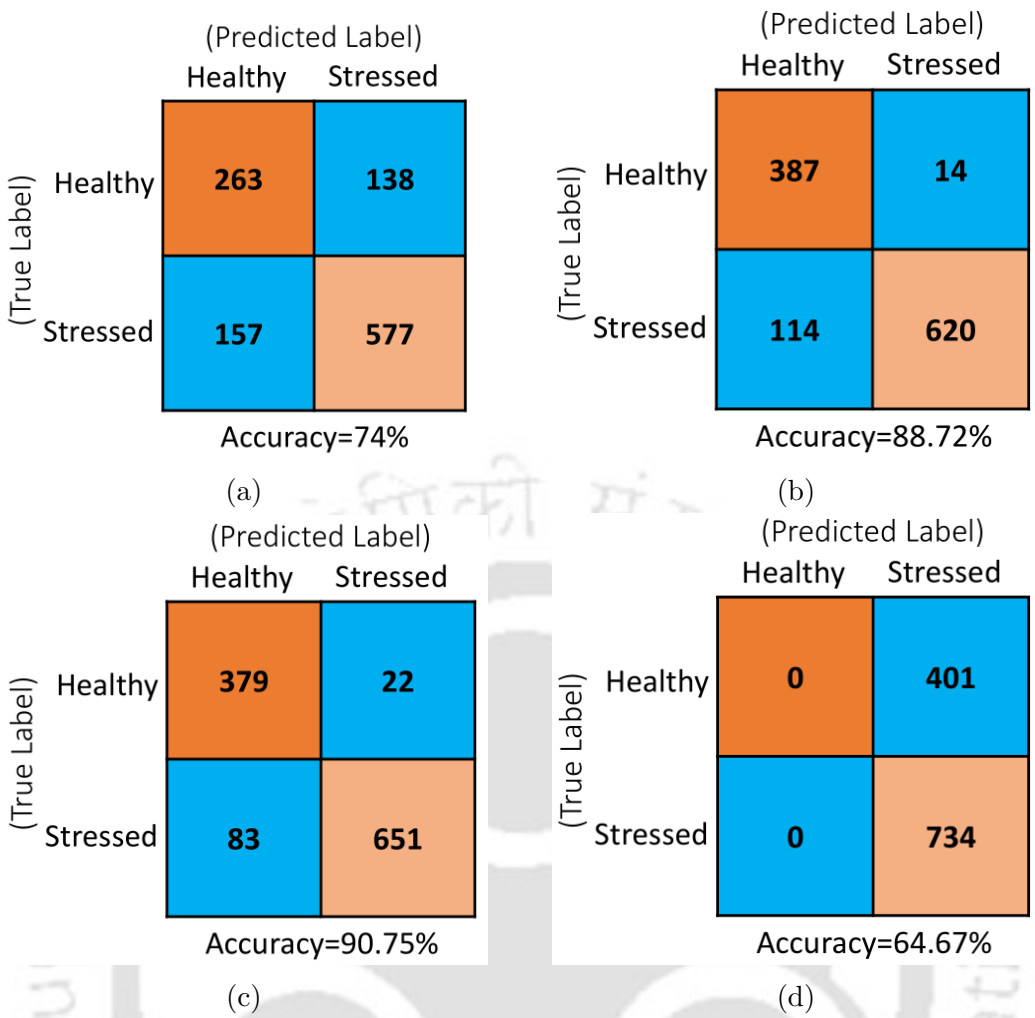


Figure 2.8: Confusion Matrix of the Model for the various Pre-Trained Networks with Test Data Set comprising of 1135 images: a) EfficientNetB0, b) MobileNet, c) DenseNet121, and d) NASNetMobile.

of the pipeline, resulting in two scenarios for investigation:

- Scenario 1: Gradients are considered at the last dense layer.
- Scenario 2: Gradients are considered at the last convolutional layer of DenseNet121.

These gradients are used to generate a coarse localization map for a specific target concept, such as drought stress. This map highlights key regions within the image that contribute greatly to predicting the concept. Analyzing an RGB image for drought stress involves examining various visual cues and patterns indicative of plant stress. In such images, areas of interest often exhibit discoloration, wilting, or reduced foliage density compared to healthy regions. The color spectrum may shift towards yellow or brown, signifying decreased chlorophyll content and photosynthetic activity. Additionally, leaf curling or necrotic spots may be visible, indicating water scarcity and cellular damage. The explainability process begins with pre-processing the drought-stressed image by resiz-

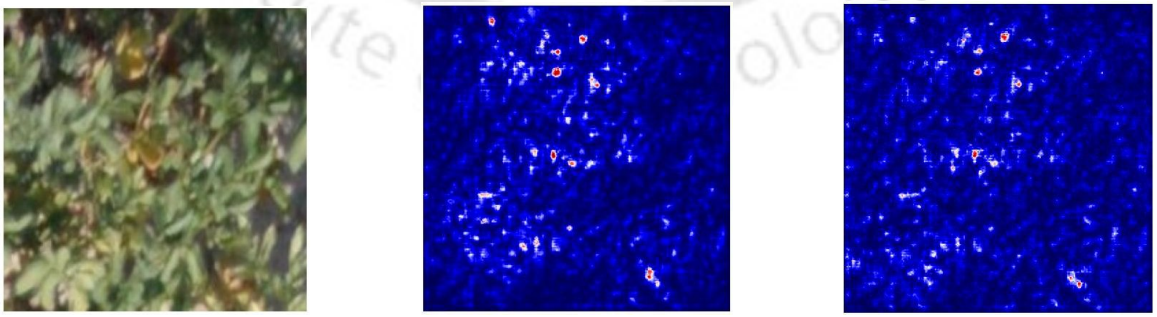
ing it to match the model's input dimensions and normalizing the pixel values to ensure consistent data representation. After pre-processing, the image is fed into the trained deep learning model. We used the model weights from the 55th epoch, as they provided the best performance in terms of classification accuracy, precision, and recall. Gradients are then calculated using *GradientTape*, a *TensorFlow* component that facilitates automatic differentiation. These gradients are subsequently used to generate a heatmap that effectively highlights the critical regions within the input image that contribute to the prediction of drought stress. The entire process is summarized in the Algorithm 1. The generated heatmap overlays the original image, highlighting areas where the model places greater importance in its decision-making process. The original image, along with the heatmaps for both Scenario 1 and Scenario 2, are shown in Fig. 2.9a, Fig. 2.9b, and 2.9c, respectively. The *seismic* colormap is used for heatmap visualization, where red shades highlight areas of high importance, blue indicates regions of low importance, and white represents the neutral point.

Algorithm 1: Procedure for visualizing image regions associated with stress using the trained model.

Input: Trained model M, input image I

Output: Visualization of input image with heatmap

1 Load the trained model M;
2 Read and preprocess image I;
3 Resize I to match model input size;
4 Normalize pixel values of I;
5 Compute gradients using *GradientTape* as described in Section 2.3.2.2;
6 Take absolute value of gradients and normalize;
7 Calculate mean and standard deviation of the gradient map;
8 Standardize to generate the heatmap;
9 Display input image I and heatmap side by side;



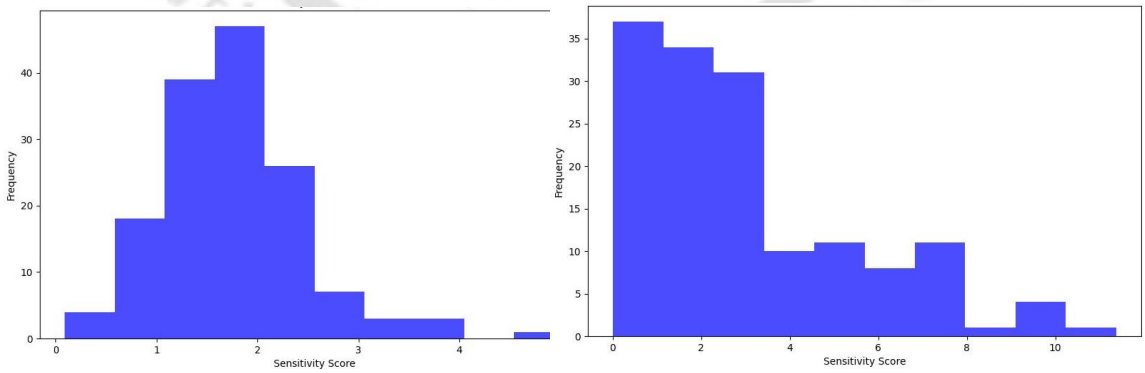
(a) Input Image (b) Heatmap 1 (c) Heatmap 2

Figure 2.9: Explaining the deep learning model using gradient-based visualisation.

The sensitivity analysis is conducted to evaluate the robustness of the model

against noise and small changes in the input in both scenarios. Specifically, it assesses how a trained model's predictions are influenced by perturbations in input images through the addition of Gaussian noise. This analysis involves introducing Gaussian noise to 148 images, which represent 20% of the drought-stressed images from the test set, using a variance of 0.01. Sensitivity is measured by calculating the absolute difference in prediction scores between the original and perturbed images, providing a numerical sensitivity score that reflects the model's resilience to slight variations in input data. To summarize the findings, key statistics are computed, including the average sensitivity, median sensitivity, and standard deviation of the sensitivity scores across all tested images. Furthermore, the distribution of these sensitivity scores is visualized using a histogram, with the score plotted against frequency and a bin size of 10.

In Scenario 1, the median sensitivity score is 1.65, close to the average sensitivity of 1.73, indicating a consistent response to noise across various inputs. These findings suggests that the model's predictions remain relatively stable and predictable, with minimal variation in how noise affects the different inputs. The standard deviation in Scenario 1 is 0.71, further emphasizing the model's consistency in handling noise, as the spread of sensitivity scores is narrow, and there are fewer outliers. In contrast, Scenario 2 exhibits a higher median sensitivity of 2.32. Still, it is notably lower than the average sensitivity of 3.01, indicating that while many inputs show moderate sensitivity to noise, a few outliers with much higher sensitivity skew the average upward. These observations suggests that the model's response to noise is less consistent in Scenario 2, as the presence of outliers introduces greater variability. The standard deviation 2.48 in Scenario 2 reflects this wider spread of sensitivity scores, highlighting the model's reduced robustness when gradients are taken from the convolutional layer. The greater variability indicates that some images are more affected by noise than others, making the model's behavior less predictable in this scenario.



(a) (b)

Figure 2.10: Distribution of Sensitivity Scores: a) Scenario 1 and b) Scenario 2.

The distribution of sensitivity scores for Scenario 1 (Fig. 2.10a) and Scenario

2 (Fig. 2.10b) further supports these findings. In Scenario 1, the distribution is concentrated around lower sensitivity scores, with most images showing sensitivity scores below 2. This pattern indicates that the model is generally less sensitive to noise, with fewer outliers, reflecting better stability and consistency across inputs. In contrast, Scenario 2 exhibits a wider range of sensitivity scores, from 0 to 10, indicating much higher variability. Some images show very high sensitivity scores, reaching up to 10, suggesting that some of the inputs are more affected by noise perturbations than others. Thus, Scenario 1 demonstrates better stability and interpretability, while Scenario 2 is more prone to noise and shows greater variability in its responses.

In summary, gradient-based visualization of drought-specific spatial features helps bridge the gap between a CNN model's 'black box' nature and human understanding. It empowers agricultural practitioners to interpret the model's reasoning and make informed decisions about plant health based on visual cues and analysis.

2.4.4 Performance Comparison with Object Detection Methodologies

We compare our proposed classifier, which incorporates gradient-based explainability, with the object detection algorithms implemented in a previous work [61]. This comparison is particularly insightful because the localization aspect of object detection models aligns with our proposed approach, which also focuses on pinpointing stress areas in crops. Both systems are designed to identify and distinguish between two classes (stressed and healthy) using the same dataset and bounding boxes.

The evaluation is based on precision and recall metrics to measure each model's effectiveness in accurately detecting instances of the target classes. Higher precision and recall indicate superior performance in classifying stressed and healthy conditions. Table 3.7 presents the performance metrics of our proposed model and compares them with those of models reported by [61]. Our proposed pipeline, based on DenseNet121, notably outperforms the other models. It achieves the highest precision for both stressed (0.967) and healthy (0.820) instances, along with the best recall for stressed instances (0.887). These results demonstrate its ability to accurately identify stressed conditions while maintaining high precision and minimizing false positives. In contrast, while Yolo v3 shows competitive recall for stressed plants (0.882), its low precision (0.407) indicates that it frequently misclassifies healthy plants as stressed. This result shows that our method provides reliable and accurate classification of stressed and healthy conditions compared to traditional object detection models.

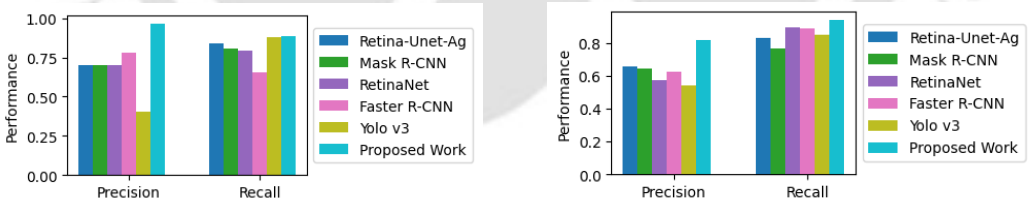
Moreover, the proposed classifier with explainability offers a better alternative to traditional object detection algorithms when interpretability and high precision in

identifying stressed plants are prioritized. The visual insights provided by the explainable approach enhance model transparency by highlighting critical regions used in the decision-making process. Such insights are especially useful for applications where understanding the model's reasoning is crucial, such as early stress detection in agriculture. On the other hand, traditional object detection algorithms may be more appropriate for tasks requiring precise object localization and real-time performance. Therefore, choosing between our classifier and object detection models depends on specific application requirements and priorities.

Our work advances non-invasive imaging techniques for crop monitoring by offering an interpretable, high-precision classifier that supports early stress detection. This advancement greatly aids decision-making in agriculture, ultimately contributing to better crop management practices. The performance comparison is further visualized in the histogram shown in Fig. 2.11.

Table 2.2: Performance of the models with the RGB images.

Model	Stressed		Healthy	
	Precision	Recall	Precision	Recall
Retina-Unet-Ag	0.702	0.841	0.659	0.832
Mask R-CNN	0.700	0.809	0.644	0.769
RetinaNet	0.698	0.795	0.578	0.899
Faster R-CNN	0.781	0.654	0.630	0.891
Yolo v3	0.407	0.882	0.541	0.855
Proposed Pipeline (with DenseNet121)	0.967	0.887	0.820	0.945



(a) Drought Stressed (b) Healthy

Figure 2.11: Comparison of Precision and Recall Metrics Across Various Models.

2.5 Summary

This study confirms the effectiveness of a deep learning pipeline, specifically utilizing DenseNet121 as the backbone, along with a data augmentation procedure and custom layers, to detect drought stress in potato crops with high accuracy. The results indicate that explainable machine learning methods yield actionable insights by identifying stress-specific regions within crop images. These findings support the hypothesis that early

detection of drought stress through non-invasive imaging enhances decision-making in agricultural practices.

The integration of gradient-based visualization significantly advances model transparency, enabling agricultural practitioners to more effectively trust and interpret the outputs of artificial intelligence-based systems. Such interpretability is essential for practical adoption in real-world agricultural contexts, where understanding the rationale behind model predictions is as important as predictive accuracy. This framework provides a promising tool for enhancing crop management, optimizing water use efficiency, and promoting sustainability by facilitating targeted interventions such as precision irrigation.

Although the framework has demonstrated strong performance, its application to additional crops and various environmental stressors requires further investigation. Future research should broaden its applicability, enhance real-time processing capabilities, and address scalability across diverse agricultural conditions. This study introduces an innovative, lightweight, and explainable approach to crop stress detection with the potential to transform current agricultural practices and promote more sustainable and efficient crop management strategies.

The work embodied in this chapter is published as:
Patra, A. K., & Sahoo, L. (2024). Explainable lightweight deep learning pipeline for improved drought stress identification. *Frontiers in Plant Science*, 15, 1476130.

Chapter 3

An Explainable Vision Transformer with Transfer Learning Based Efficient Drought Stress Identification

3.1 Abstract

Early detection of drought stress is critical for taking timely measures for reducing crop loss before the drought impact becomes irreversible. The subtle phenotypical and physiological changes in response to drought stress are captured by non-invasive imaging techniques and these imaging data serve as valuable resource for machine learning methods to identify drought stress. While convolutional neural networks (CNNs) are in wide use, Vision Transformers (ViTs) present a promising alternative in capturing long-range dependencies and intricate spatial relationships, thereby enhancing the detection of subtle indicators of drought stress. We propose an explainable deep learning pipeline that leverages the power of ViTs for drought stress detection in potato crops using aerial imagery. We applied two distinct approaches: a synergistic combination of ViT and support vector machine (SVM), where ViT extracts intricate spatial features from aerial images, and SVM classifies the crops as stressed or healthy and an end-to-end approach using a dedicated classification layer within ViT to directly detect drought stress. Our key findings explain the ViT model's decision-making process by visualizing attention maps. These maps highlight the specific spatial features within the aerial images that the ViT model focuses as the drought stress signature. Our findings demonstrate that the proposed methods not only achieve high accuracy in drought stress identification but also shedding light on the diverse subtle plant features associated with drought stress. This offers a robust and interpretable solution for drought stress monitoring for farmers to undertake informed decisions for improved crop management.

Keywords: {Stress Pheno-typing, Drought Stress, Machine Learning, Deep Learning, Vision Transformer, Support Vector Machine}

3.2 Introduction

In the previous chapter, we devised and investigated CNN based lightweight models for drought stress identification. CNNs rely on local receptive fields and often struggle to capture relationships across various parts of an image, whereas Vision Transformers (ViTs) offer notable advantages over CNNs in capturing long-range dependencies and global context due to their self-attention mechanism[127]. Additionally, ViTs exhibit greater flexibility with input image sizes and handle complex patterns and data variations more effectively [128].

The self-attention mechanism in ViTs enables more accurate classification by considering the entire image context, which enhances accuracy and robustness compared to traditional CNNs [129]. For example, Dosovitskiy et al. demonstrated that ViTs could outperform CNNs in image classification tasks, highlighting their potential for diversified applications [127]. Recent studies have effectively applied ViTs both in customized form [130, 35] and CNN- ViT hybrid form to plant disease identification [23]. Thakur et al. [131] developed a lightweight model that combines convolutional blocks from VGG 16 and Inception V7 with transformer components such as multi-head attention and multi-layer perceptron to effectively identify a wide range of plant diseases across multiple crops. The model leverages the local feature extraction capabilities of CNNs and the global feature modeling strength of Vision Transformers, enabling simultaneous extraction of both local and global features from images. ViT has outperformed Inception V3 in terms of accuracy when distinguishing among nine different tomato leaf disease classes [132]. Parez et al. fine-tuned the Vision Transformer with a four-fold reduction in training parameters, achieving higher accuracy compared to CNN models for disease identification across three datasets [133]. Yu et al. [134] proposed a framework that integrates soft split token embedding and depth-wise convolutional modules into the Vision Transformer architecture, resulting in improved accuracy. Replacing the MLP module in the ViT encoder block with an Inception module improved accuracy in multi-crop disease classification while reducing the number of trainable parameters [135]. Thai et al. optimized the ViT architecture for cassava leaf disease detection by pruning less important attention heads and using sparse matrix operations, achieving a 2% improvement in F1-score along with reduced model size and training costs [136]. Hemalatha et al. developed a plant disease localization and classification model which uses co-scale, co-attention, and cross-attention mechanisms with a Vision Transformer in a multi-task learning framework [137]. Li et al. integrated a convolutional block attention module into the standard ViT encoder, enabling the network to filter out irrelevant information and focus on essential features, leading to improved crop disease classification in rice, wheat, and coffee [138]. Vallabhajosyula et al. [139] proposed a novel framework that combines a transformer encoder

with ResNet9 for plant disease classification, outperforming several classical CNN-based models. From the extensive literature review, we observe that while most existing works focus primarily on improving accuracy, few address the reduction of trainable parameters, and none explore the explainability of transformers. Our proposed work differs in two key aspects: first, by deciphering the attention mechanism, we emphasize model explainability; second, we devised ViT and support vector machine (SVM) combined framework and investigated its performance.

In this study, we devised a Vision Transformer (ViT)-based framework and fine-tuned it specifically for drought stress identification, achieving improved accuracy in detecting stress conditions. To demonstrate the model's interpretability, we employed the inherent self-attention properties of Vision Transformers to produce attention maps. These maps offer meaningful insights into how the model arrives at its decisions, thereby increasing both the transparency and trustworthiness of its predictions. Additionally, we proposed a hybrid ViT+SVM framework that combines the rich feature representation capabilities of ViTs with the strong classification performance of Support Vector Machines (SVMs), resulting in a more robust drought stress identification model.

3.3 Materials and Methods

In this section, we begin by introducing the experimental dataset. Next, we present the drought stress classification model using two different approaches. In the first approach, we employ the Vision Transformer with transfer learning. In the second approach, we propose a framework that utilizes a Vision Transformer as a feature extractor, followed by the integration of an SVM as the classifier. Additionally, we investigate the interpretability of the model by generating and analyzing the attention maps. This comprehensive use of the ViT elucidates how the spatial features of drought stress can be precisely identified. Finally, we discuss the performance metrics for the proposed approaches.

3.3.1 Preparing the Data

We used the same data as used in Chapter 2.

3.3.2 Vision Transformer (ViT)

A Vision Transformer (ViT) is a type of neural network architecture that has revolutionized the field of computer vision [140]. Unlike traditional convolutional neural networks (CNNs), which process images pixel by pixel, ViT processes the input images in a sequential manner by dividing them into fixed-size patches, linearly embedding these patches, and then applying self-attention mechanisms for capturing global dependencies [127]. The

VIT architecture used for our work is inspired by Dosovitskiy et al.[127] and depicted in Fig 3.1a. It processes images by dividing them into fixed-size embedded patches, linearly transforming these patches, and treating them as sequences. The transformer architecture is then applied, incorporating multi-head self-attention mechanisms [141] that enable the model to capture long-range dependencies within the image. Layer normalization is applied before and after the multi-head attention, ensuring stable training, and a Multilayer Perceptron (MLP) head is added to the transformer's global representation for task-specific processing.

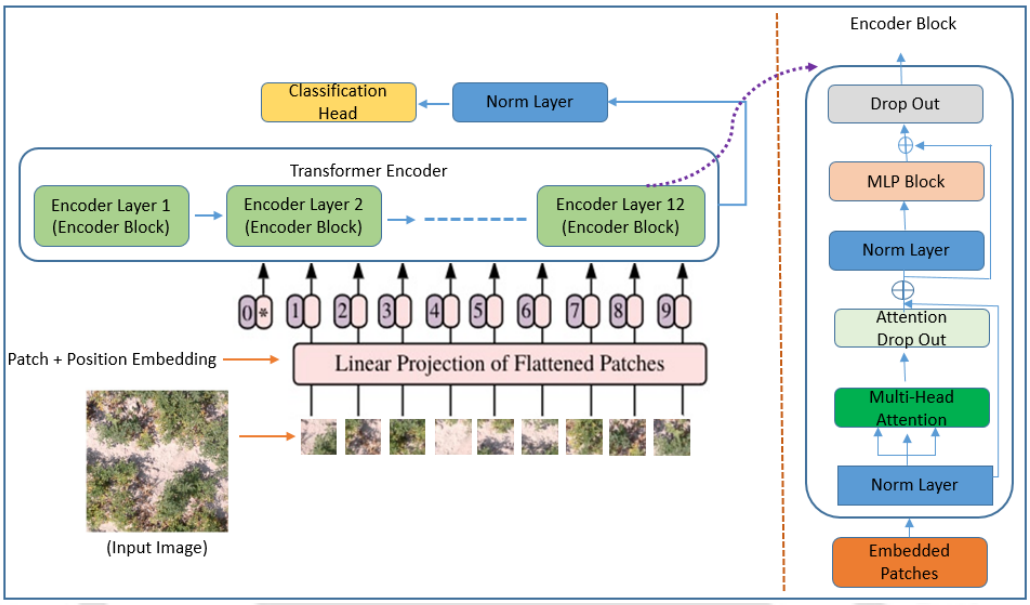
3.3.2.1 ViT Architecture

The proposed ViT architecture is based on the layers in [142] and Dosovitskiy et al.[127] and comprises five main components: patch embedding, positional encoding, transformer encoder , normalization layer and a classification head. This is illustrated in Fig 3.1a.

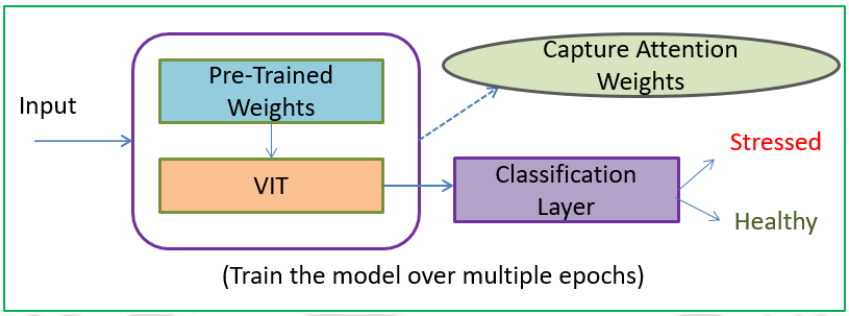
- **Patch Embedding**: Input images are divided into fixed-size patches, which are then linearly embedded to create a sequence of embeddings.
- **Positional Encoding**: To capture spatial information, positional encodings are added to the patch embeddings, allowing the model to understand the relative positions of different patches.
- **Transformer Encoder**: The embedded patches are fed into a Transformer Encoder, which is the core component of the ViT architecture. This encoder consists of twelve identical encoder layers stacked together. Each attention layer analyzes the relationships between pairs of patches, allowing the model to understand how different image regions interact and influence each other. Each attention layer consists of the following:
  - Multi-Head Self-Attention: This mechanism allows the model to weigh the importance of different parts of the image. It captures global dependencies between the patches.
  - MLP (Multi-Layer Perceptron) Block: This block introduces non-linearity to the network and further processes the information from the attention layer.
  - Normalization Layers: Layer normalization is applied after the multi-head attention and MLP blocks to stabilize training.
  - Dropout: The model uses dropout at two levels: within the attention mechanism and after the MLP block. Dropout is used to prevent over-fitting by randomly dropping out neurons during training.
- **Normalization Layer**: Layer normalization applied to the output of the encoder

serves several crucial purposes: Reduces internal co-variate shift, improves gradient flow, acts as regularization, handles varying input distributions and accelerates convergence.

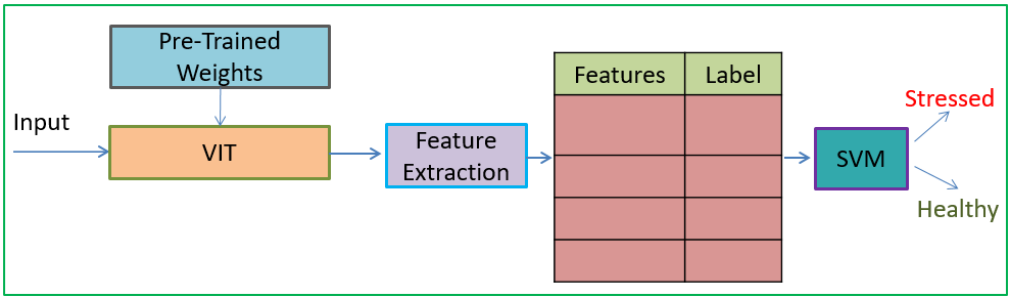
- **Classification Head**: This head typically consists of a simple MLP layer that maps the feature representation to the desired output, such as class probabilities.



(a) Vision Transformer Architecture



(b) Vision Transformer with Transfer Learning



(c) Integrating Vision Transformer and SVM

Figure 3.1: Vision Transformer based Approaches for Drought Stress Identification

3.3.2.2 Information Processing in ViT

At the core of the Vision Transformer (ViT) is the concept of a token, which plays a crucial role in the model's ability to process and understand images. A token is a fixed-size vector that represents a small patch of the input image. These tokens are at the core of the Vision Transformer model, forming the input sequence to the Transformer layers. This token-based approach enables the model to process and understand the image by focusing on the relationships between these patches through self-attention. The detailed explanation is given below.

- **Dividing the Image into Patches:**
  - The input image is divided into smaller, fixed-size patches. For example, an image of size 224x224 pixels might be divided into patches of size 16x16 pixels, resulting in $\left(\frac{224}{16}\right)^2 = 196$ patches.
- **Flattening and Embedding:**
  - Each image patch is then flattened into a one-dimensional vector. For instance, a 16x16 patch with 3 color channels (RGB) would be flattened into a vector of length $16 \times 16 \times 3 = 768$.
  - These flattened vectors (patch representations) are then linearly embedded into a higher-dimensional space. This is typically done using a learnable linear projection, transforming each vector into a fixed-size embedding, say of dimension 768.
- **Tokens:**
  - After linear embedding, each flattened and embedded patch becomes a "token." In this example, the image is transformed into a sequence of 196 tokens, each representing a 16x16 patch of the original image.
- **Adding Positional Encoding:**
  - Since the transformer model does not inherently understand the order or position of tokens, positional encodings are added to each token to incorporate information about its original position in the image. This helps the model understand spatial relationships between patches.
- **Processing by Transformer Layers:**
  - These tokens are then processed by the Transformer layers, which include self-attention mechanisms. The self-attention mechanism computes relationships between these tokens to understand how different parts of the image relate to

one another.

3.3.3 ViT with Transfer Learning

The proposed framework, as shown in Fig 3.1b effectively combines transfer learning, the power of the Vision Transformer and attention-based interpretability to address the challenging task of drought stress identification in potato crop images captured in natural settings. A core component of this approach is the utilization of pre-trained weights. This technique, known as *transfer learning*, involves leveraging knowledge gained from solving one problem (often a large-scale image classification task) and applying it to a different but related problem. Specifically, we used the model initialized with pre-trained weights from training on the ImageNet-1k dataset with 1000 classes. By employing pre-trained weights, the model can benefit from the rich feature representations learned from a massive dataset, accelerating training and potentially improving performance. Using this the ViT is trained and fine-tuned over multiple epochs using a combination of *binary cross-entropy* loss, *Adam/AdamW* optimizer and learning rate tuning. Experimenting with different learning rates is essential to find the optimal value for convergence and generalization. Eventually, the classification layer is responsible for making the final prediction. It takes the output of the ViT model and maps it to two classes: "healthy" and "stressed." This layer typically consists of a fully connected neural network with a sigmoid activation function for binary classification.

To enhance model interpretability, the architecture incorporates a mechanism to capture attention weights. Attention weights reveal which parts of the input image the model focuses on when making a decision. By visualizing these weights, researchers can gain insights into the model's decision-making process and identify key image features that contribute to the classification.

3.3.3.1 Attention Maps

Attention maps provide insights into how the model focuses on different parts of the image during the self-attention mechanism. This mechanism allows the model to selectively attend to specific areas while processing visual information. The image is first divided into patches. Self-attention then helps the model prioritize relevant patches and their relationships for effective feature extraction.

Attention maps act as a visual representation of the weights assigned by the self-attention mechanism to each patch. These maps can be visualized for each self-attention layer within the ViT model, as each layer progressively learns more intricate relationships between these image features. Higher weights indicate a greater focus on a specific patch and its connection to others. By analyzing these maps, we can essentially

see the model's "thought process" during image understanding. We can identify which regions it prioritizes for information extraction. The following section highlights the computation behind the attention mechanism [142].

At its core, self-attention computes a weighted sum of the values (features) based on the similarities (attention scores) between different positions in the sequence. This is achieved through three learnable matrices: Query (Q), Key (K), and Value (V).

Query Matrix (Q): The query matrix is responsible for capturing the information about the current token being processed. It learns to encode the features of the token in a format suitable for comparison with other tokens. Each token in the sequence is associated with a query vector, which represents its characteristics in the context of the entire sequence.

Key Matrix (K): The key matrix holds information about the relationship between the current token and other tokens in the sequence. It learns to encode the features that determine how relevant each token is to the current token.

Value Matrix (V):

The key matrix holds information about the relationship between the current token and other tokens in the sequence. It learns to encode the features that determine how relevant each token is to the current token.

Given a sequence of tokens $X = [x_1, x_2, ..., x_n]$, the attention scores A are computed as:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

where $Q = XW_Q$, $K = XW_K$, $V = XW_V$, and W_Q, W_K, W_V are weight matrices. d_k represents the dimensionality of the key vectors.

Once the attention scores are computed, they are used to compute a weighted sum of the values:

$$\text{Attention}(Q, K, V) = A \cdot V$$

where A is the attention matrix.

To capture different relationships between tokens, ViT (and Transformers in general) often employ multiple attention heads. Each head learns different sets of Q, K, V weight matrices and computes separate attention scores and weighted sums. The results

from all heads are concatenated and linearly transformed to maintain a consistent output dimension. Since self-attention does not inherently consider the order of tokens, positional encoding is typically added to the token embeddings to provide positional information.

Algorithm for Capturing Attention Weights: To enhance model interpretability, the architecture incorporates a mechanism to capture attention weights. The algorithm 2 outlines the steps during the training of ViT to capture and utilize attention weights for the drought identification task. The key components of this class include the initialization, forward pass, attention weight capture, output size determination, and retrieval of attention weights. By visualizing these weights, researchers can gain insights into the model's decision-making process and identify key image features that contribute to the classification.

Initialization The VisionTransformerBinary class is initialized with a pre-trained Vision Transformer model passed as `vit_model`. During initialization, the model assigns the provided `vit_model` to its own `vit` attribute. Additionally, it sets up a fully connected (linear) layer (`fc`) with a size appropriate to the output of the Vision Transformer. This layer is responsible for converting the output of the Vision Transformer into a format suitable for binary classification. An empty list `attn_weights` is also initialized to store the attention weights captured during the forward pass.

Forward Pass The `forward` function is central to the operation of the VisionTransformerBinary class. When an input image `x` is passed through the model, the function first clears any previously stored attention weights. This ensures that the attention weights list is fresh and only contains data relevant to the current input.

Next, the function registers hooks on the self-attention layers of each block in the Vision Transformer. These hooks are responsible for capturing the attention weights during the forward pass. For each block in the Vision Transformer's encoder layers, the self-attention layer is accessed, and a hook is registered to capture its attention weights using the `_capture_attn_weights` helper function. The hooks are stored in a list to facilitate their removal later.

With the hooks in place, the input image is passed through the Vision Transformer and then through the fully connected layer. This produces the model's output for the given input. After the forward pass is complete, the hooks are removed to free up memory, ensuring that they do not persist and interfere with future operations.

Attention Weights Capture The `_capture_attn_weights` helper function is designed to capture the attention weights during the forward pass. It is triggered by the hooks registered on the self-attention layers. This function receives the module, input, and output as arguments. From the input, it extracts the query, key, and value components,

which are essential for computing the attention weights. These components, along with the output, are appended to the `attn_weights` list. By capturing these components, the model can later compute and analyze the attention scores, which provide insights into the regions of the input image that the model focuses on during classification.

Output Size Determination The `_get_output_size` helper function determines the output size of the Vision Transformer model. This function performs a forward pass with a zero tensor of appropriate dimensions through the Vision Transformer. By doing so, it captures the shape of the output tensor produced by the Vision Transformer. The size of the last dimension of this output tensor is then returned, which is used to initialize the fully connected layer with the correct input size.

Retrieval of Attention Weights The `get_attention_weights` function provides a simple interface to retrieve the captured attention weights. It returns the `attn_weights` list, allowing external components or functions to access the attention weights for further analysis or visualization.

The modular design of the class, with separate functions for initialization, forward pass, attention weight capture, output size determination, and retrieval, provides a clear and maintainable structure. This design facilitates the integration of attention-based insights into the classification process, enhancing the interpretability and performance of the model.

Algorithm 2: Vision Transformer Class to capture Attention Weights

Input: Pre-trained Vision Transformer model

Output: Modified Vision Transformer model with attention weights capture

```
Class VisionTransformerBinary(vit_model):
    Data: vit_model: Vision Transformer model
    // Initialization
    Initialize vit with vit_model;
    Initialize fc with a linear layer of appropriate output size;
    Initialize attn_weights as an empty list;
    Function forward(x):
        // Clear previous attention weights
        Clear attn_weights;
        // Register hooks to capture attention weights
        Initialize hooks as an empty list;
        for each block in vit.encoder.layers do
            attn_layer ← block.self_attention;
            hook ← attn_layer.register_forward_hook(_capture_attn_weights);
            Append hook to hooks;
        // Pass input through the Vision Transformer
        x ← vit(x);
        x ← fc(x);
        // Remove hooks to free up memory
        for each hook in hooks do
            Remove hook;
        return x;
    Function _capture_attn_weights(module, input, output):
        // Capture attention weights
        Extract query, key, value from input;
        Append (query, key, value, output) to attn_weights;
    Function _get_output_size():
        // Determine the output size of the Vision Transformer
        Initialize output with a forward pass of zeros through vit;
        return size of the last dimension of output;
    Function get_attention_weights():
        return attn_weights;
```

3.3.4 Integrating Vision Transformer (ViT) and Support Vector Machine (SVM)

This framework focuses on combining a Vision Transformer (ViT) and a support vector machine (SVM) within a three-step approach for effective stress identification.

- Feature Extraction: For each image in the dataset, extract the final hidden state or pooled representation from the ViT model to obtain a feature vector. Let $\mathbf{X}$ represent the set of embedded features extracted from a dataset of images, and $\mathbf{y}$ represent the corresponding class labels.
- Training SVM: Train an SVM classifier using the extracted features $\mathbf{X}$ and their corresponding class labels $\mathbf{y}$.
- Classification: For a new, unseen image, extract features using the pre-trained ViT model and use the trained SVM classifier to predict its class.

3.3.4.1 Support Vector Machine (SVM)

Support Vector Machines [143] aim to find a hyperplane that best separates a given set of data points into different classes. Given a set of labeled data points $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$ where $\mathbf{x}_i$ is the feature vector of the i-th data point, and y_i is its corresponding class label ($y_i \in \{1, 0\}$ for binary classification), SVM seeks to find a hyperplane defined by the equation:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

where $\mathbf{w}$ is the weight vector and b is the bias.

The goal is to maximize the margin, which is the distance between the hyperplane and the nearest data point from each class. The margin is computed as the perpendicular distance from a data point $\mathbf{x}_i$ to the hyperplane:

$$\text{margin} = \frac{1}{\|\mathbf{w}\|} \cdot |\mathbf{w} \cdot \mathbf{x}_i + b|$$

Subject to the constraint that for all data points:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$$

This constraint ensures that data points are correctly classified and lie on the

correct side of the hyperplane.

The SVM optimization problem can be formulated as:

$$\text{Minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

Subject to the constraints:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \text{for all } i$$

This is the primal form of the optimization problem. However, the SVM problem is often reformulated in its dual form, which introduces Lagrange multipliers α_i to handle the constraints. The dual formulation is:

$$\text{Maximize} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

Subject to the constraints:

$$0 \leq \alpha_i \leq C \quad \text{for all } i$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

The solution to the dual problem provides the values of α_i, and the weight vector $\mathbf{w}$ and bias b can be obtained from these values.

In cases where the data is not linearly separable, SVM can be extended to handle non-linear decision boundaries using the kernel trick. The feature space is implicitly mapped to a higher-dimensional space, making it possible to find a linear separating hyperplane in that space.

Classification is performed based on the decision function derived from the trained model. Once the SVM model is trained with a set of support vectors, it identifies a hyperplane that best separates different classes in the feature space. The decision function for a new data point $\mathbf{x}$ is given by:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

Here, $\mathbf{w}$ is the weight vector, and b is the bias term. The sign of $f(\mathbf{x})$ determines the predicted class:

$$\begin{cases} \text{Class 1}, & \text{if } f(\mathbf{x}) > 0 \\ \text{Class 0}, & \text{if } f(\mathbf{x}) < 0 \end{cases}$$

The magnitude of $f(\mathbf{x})$ provides a measure of how far the data point is from the decision boundary. Larger magnitudes indicate greater confidence in the classification.

The key role of support vectors in this process is that they are the data points lying closest to the decision boundary. Support vectors effectively determine the position and orientation of the hyperplane. The optimization process in SVM aims to maximize the margin between the classes, and support vectors are the data points defining the edges of this margin. In practice, many data points do not significantly contribute to the definition of the decision boundary. Only the support vectors, with non-zero Lagrange multipliers α_i in the dual formulation, are crucial for determining the hyperplane. This property makes SVM memory-efficient and computationally faster, especially in high-dimensional spaces.‘

3.3.4.2 ViT+SVM Framework

The process begins with input images that are fed into the ViT. The ViT, pre-trained on a vast dataset, is adept at extracting meaningful features from the images. Once the features are extracted by the ViT, they are compiled into a feature matrix, which also includes the corresponding labels indicating whether the plants in the images are healthy or stressed. This matrix forms the input to the SVM, a robust classifier known for its effectiveness in handling high-dimensional data. The SVM is trained to discern between the two classes—healthy and stressed—based on the features provided. Finally, the trained SVM predicts the class of new images, categorizing them as either healthy or stressed. The entire approach is depicted in Fig 3.1c. The efficacy of this framework is evaluated using a designated test set and k-fold cross-validation.

3.3.5 Performance Evaluation Metrics

The model's performance underwent assessment using various evaluation metrics, including accuracy, precision, recall (sensitivity) and the Receiver Operating Characteristic (ROC) curve. These metrics are computed based on the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), which collectively form a 2x2 matrix known as the confusion matrix. In this matrix, TP and TN indicate the accurate predictions of water-stressed and healthy potato crops, respectively.

FP, termed as type 1 error, denotes predictions where the healthy class is inaccurately identified as water-stressed. FN, referred to as type 2 error, represents instances where water-stressed potato plants are incorrectly predicted as healthy. The classification accuracy is a measure of the ratio between correct predictions for both stressed and healthy images and the total number of images in the test set.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Population}}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Cross-validation is crucial in model development for two key reasons: it helps prevent over-fitting by assessing a model's performance across different subsets of the data, and it ensures the model's generalization ability, providing a reliable estimation of its effectiveness under various conditions. The model was trained and evaluated using k-fold cross-validation, a robust technique for assessing the generalization performance of the model. In each iteration of the k-fold cross-validation process, the dataset was partitioned into k folds, and the model was trained on $k-1$ folds while being validated on the remaining fold. This process was repeated k times, ensuring that every fold had the opportunity to serve as the validation set. For each fold, the Receiver Operating Characteristic (ROC) curve was plotted, illustrating the trade-off between true positive rate and false positive rate at various thresholds. After completing the k-fold cross-validation, the individual ROC curves were aggregated, and the mean ROC curve was calculated and plotted. AUC (area under the curve) measures the entire two-dimensional area underneath the entire ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.This comprehensive approach provides a more reliable estimation of the model's performance, capturing its consistency across different subsets of the data and enhancing the overall assessment of its predictive capabilities.

3.4 Results and Discussion

In this section, we present the experimental results of our proposed model for identifying drought stress in potato crop field images. First, we distinguish between healthy and stressed images. Then, we identify the spatial features responsible for the stress. Our experiments with the proposed Vision Transformer (ViT) framework were conducted in two ways:

- ViT with Transfer Learning (ViT-TL): Leveraging pre-trained weights.
- ViT+SVM with Optimal Weights. The optimal weights are the ones at which the model performs best while executing the ViT with pre-trained weights in the first case.

We used the *PyTorch* library and its sub-packages to implement deep learning functionalities, particularly employing *torch* and *torch.nn* for tensor operations and neural network construction. For handling image data, we utilized *Torchvision's* models for accessing pre-trained architectures and transforms for preprocessing, which included resizing images to (224, 224) pixels and converting them into tensors. Additionally, *TQDM* was used to generate progress bars for better training visibility. For data analysis and pre-processing, we employed *pandas*, *NumPy*, and *scikit-learn* for structured data manipulation, numerical computations, and machine learning utilities.

3.4.1 Performance of ViT with Transfer Learning

To adapt the Vision Transformer (ViT) architecture (as depicted in Fig. 3.1a) for our specific task of binary classification, we began by configuring the model using the *models.ViT-B/16* variant. Subsequently, we loaded custom pre-trained weights into the ViT model to realize Vision Transformer with transfer learning approach as shown in Fig 3.1b. This step was crucial as it transferred learned representations from a previously trained model to our current architecture, leveraging prior knowledge to enhance performance. A custom class was designed (as shown in Algorithm 2) to configure encoder layers as trainable or frozen, with methods to adjust various parameters. It also includes functionality to capture attention weights, crucial for analyzing the model's focus on specific image regions.

Table 3.1 presents the training configurations adopted across eleven experimental scenarios involving Vision Transformer (ViT)-based models. To balance accuracy with computational efficiency, we primarily fine-tuned a limited number of encoder layers within the pre-trained ViT models instead of retraining the entire architecture. Specifically, the number of trainable encoder blocks was varied across scenarios, starting with

Table 3.1: Training parameters of the model under different scenarios.

Scenario	Model	No. of Trainable Layers	Learning Rate & Optimizer	Callback Parameters		Batch Size	Attention Dropout	MLP Dropout
				Patience	Factor			
Scenario 1	ViT-B/16	Last encoder block	0.001(Adam)	5	0.2	128	0	0
Scenario 2	ViT-B/16	Last two encoder blocks	0.001(Adam)	5	0.2	128	0	0
Scenario 3	ViT-B/16	Last three encoder blocks	0.001(AdamW)	5	0.2	128	0	0
Scenario 4	ViT-L/16	Last three encoder blocks	0.001(AdamW)	5	0.2	128	0	0
Scenario 5	ViT-B/16	Last three encoder blocks	0.001(AdamW)	2	0.2	128	0	0
Scenario 6	ViT-B/16	Last three encoder blocks	0.001(AdamW)	2	0.2	64	0	0
Scenario 7	ViT-B/16	Last two encoder blocks	0.001(AdamW)	5	0.2	128	0.1	0.1
Scenario 8	ViT-B/16	Last two encoder blocks	0.001(AdamW)	5	0.2	128	0.1	0.2
Scenario 9	ViT-B/16	All encoder blocks	0.001(AdamW)	5	0.2	128	0	0
Scenario 10	ViT-B/16	All encoder blocks	0.001(AdamW)	5	0.2	64	0	0
Scenario 11	ViT-B/16	Last two encoder blocks	0.001(Adam)	5	0.2	128	0.1	0.2

only the final block in Scenario 1 and gradually increasing up to all encoder blocks in Scenarios 9 and 10. The ViT-B/16 model, known for its relatively lightweight design, served as the backbone for most experiments. An exception was Scenario 4, where we employed the larger ViT-L/16 model, which has approximately four times the number of parameters compared to ViT-B/16. Despite its increased capacity, ViT-L/16 did not yield a notable improvement in accuracy, leading us to retain ViT-B/16 in subsequent scenarios for better scalability and efficiency. Key training elements were systematically varied to evaluate their effects. Both Adam and AdamW optimizers were tested with a fixed learning rate of 0.001, with AdamW being preferred in most cases due to its improved regularization capabilities. Callback settings included early stopping and learning rate reduction on plateau, controlled by a patience of 5 and a factor of 0.2. However, Scenarios 5 and 6 employed a reduced patience of 2 to accelerate convergence. Batch sizes were set to either 64 or 128 to investigate their influence on model convergence and generalization. Finally, to combat overfitting, additional attention and MLP dropout layers were integrated in Scenarios 7, 8, and 11.

Table 3.2: Model performance across different scenarios.

Scenario	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Test Accuracy	Epoch No.
Scenario 1	0.9899	0.9816	0.0272	0.0466	0.9039	16
Scenario 2	0.9929	0.9831	0.0190	0.0563	0.9083	20
Scenario 3	0.9939	0.9906	0.0178	0.0294	0.9057	16
Scenario 4	0.9431	0.9443	0.1468	0.1433	0.8960	14
Scenario 5	0.9935	0.9901	0.0191	0.0322	0.8819	17
Scenario 6	0.9940	0.9876	0.0180	0.0370	0.8995	16
Scenario 7	0.9833	0.9796	0.0421	0.0444	0.9127	18
Scenario 8	0.9765	0.9747	0.0613	0.0876	0.9162	20
Scenario 9	0.9377	0.9513	0.1589	0.1344	0.9119	16
Scenario 10	0.9320	0.9274	0.1721	0.1747	0.9075	19
Scenario 11	0.9707	0.9672	0.0780	0.0896	0.8942	16

Table 3.2 summarizes the training and validation accuracy and loss for each scenario along with the final test accuracy and the epoch number at which early stopping was triggered. Across most scenarios, the training and validation accuracies exceed 97%, demonstrating strong convergence. Scenario 5, despite high training accuracy (99.35%), showed comparatively lower test accuracy (88.19%), indicating potential overfitting. In contrast, Scenario 8 exhibited the highest generalization with a test accuracy of 91.62%.

Table 3.3: Confusion matrix components and test accuracy across different scenarios.

Scenario	TP	TN	FP	FN	Test Accuracy
Scenario 1	647	379	22	87	0.9039
Scenario 2	661	370	31	73	0.9083
Scenario 3	650	378	23	84	0.9057
Scenario 4	638	379	22	96	0.8960
Scenario 5	617	384	17	117	0.8819
Scenario 6	639	382	19	95	0.8995
Scenario 7	663	373	28	71	0.9127
Scenario 8	661	379	22	73	0.9162
Scenario 9	645	390	11	89	0.9119
Scenario 10	638	392	9	96	0.9075
Scenario 11	637	378	23	97	0.8942

The confusion matrix components for each scenario are provided in Table 3.3, including the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These values corroborate the overall test accuracy and provide insights into class-wise prediction reliability. Scenario 8 again stands out with a balanced count of TP and TN and a lower FN, contributing to its highest accuracy.

The loss and accuracy curves for all 11 scenarios are depicted in Fig. 3.2 and Fig. 3.3, respectively. These plots offer visual confirmation of training convergence and generalization. From the loss and accuracy curves, it is evident that most scenarios converge smoothly, with minimal overfitting. Scenario 5 is a notable exception, exhibiting a pronounced gap between training and validation performance. This is further reflected in its confusion matrix, where the number of false negatives (117) significantly exceeds other scenarios, explaining its poor generalization (test accuracy: 88.19%). Scenario 8, despite modest fluctuations during training, achieves the highest test accuracy (91.62%) and maintains a balanced distribution of true and false predictions. Scenarios 9 and 10 notably report the lowest false positive counts (11 and 9, respectively), making them suitable for applications where false alarms are critical. In contrast, Scenario 7 shows the highest recall (TP: 663, FN: 71), which is necessary in settings where missing stressed cases could be detrimental.

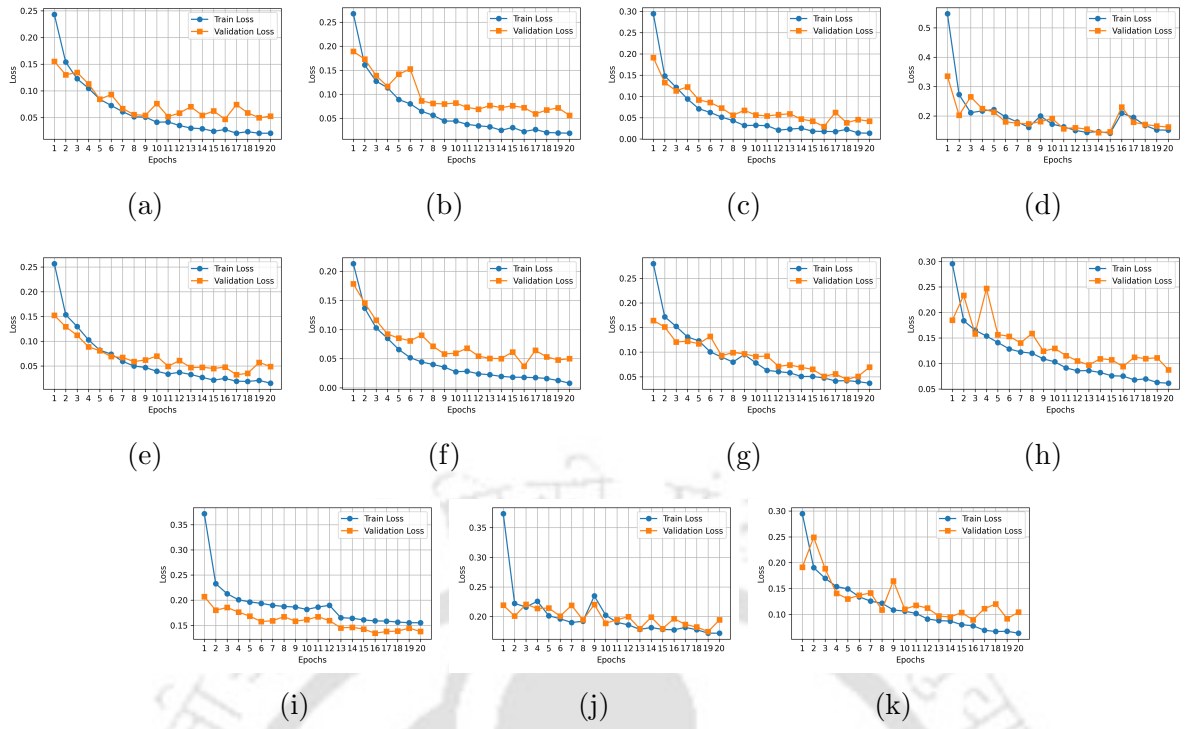


Figure 3.2: Loss curves for 11 scenarios: Fig. a–k corresponding to scenario 1 to 11.

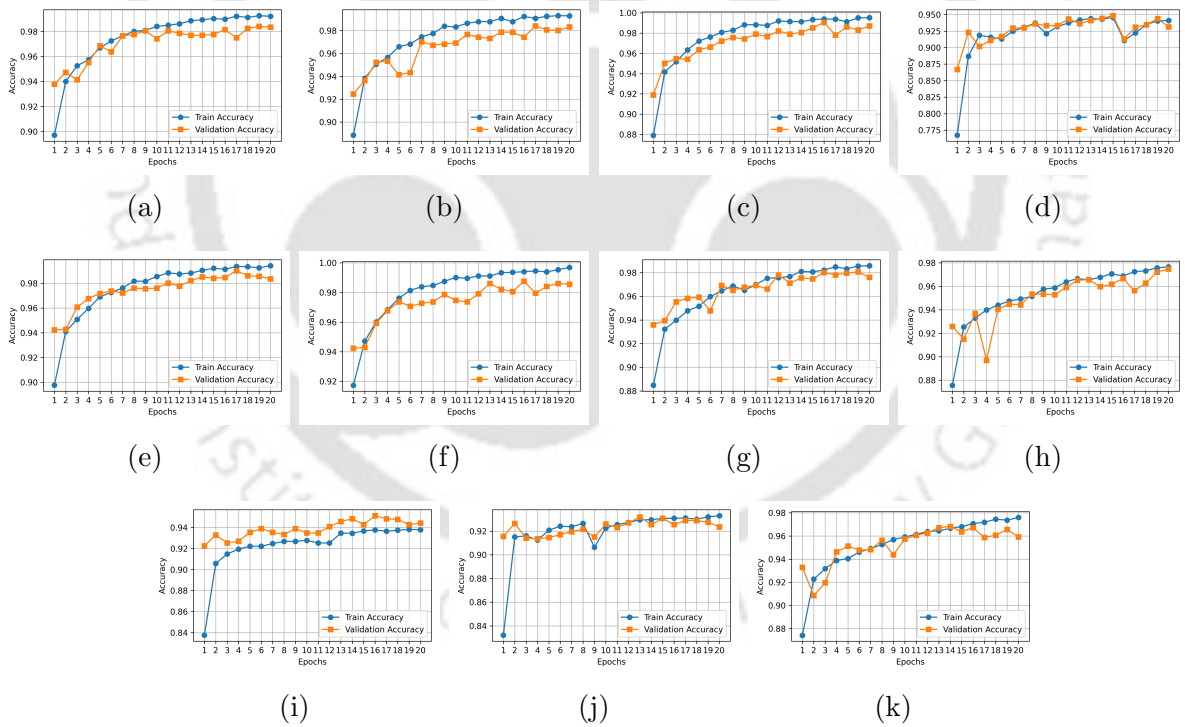


Figure 3.3: Accuracy curves for 11 scenarios: Fig. a–k corresponding to scenario 1 to 11.

This systematic experimentation helped us analyze the trade-off between model complexity and performance. The results showed that fine-tuning only the last 2–3 encoder blocks of ViT-B/16, combined with suitable regularization, yielded performance comparable to that of training the entire network or using ViT-L/16, but with

significantly fewer trainable parameters and faster training times.

3.4.1.1 Analyzing Attention Maps

Visualizing attention weights provides insights into how the ViT focus on different parts of an input during processing. By examining these weights, researchers and practitioners can understand the model's decision-making process, diagnose potential biases, and improve interpretability.

The following pseudocode outlines a systematic approach to calculate and visualize attention weights from a Vision Transformer model. This process involves capturing attention weights during the forward pass, computing attention scores, and generating visual representations of these scores.

Initialization: The process begins by initializing an empty list to store the attention weights that will be captured during the forward pass of the Vision Transformer model. This list will later be used to compute and visualize the attention maps.

Forward Pass and Capture Attention Weights: The next step involves iterating through each layer of the Vision Transformer model. For each layer, a hook is registered to capture the attention weights. The input image is then passed through the Vision Transformer to compute the output features. This stage ensures that attention weights are collected during the forward pass for later analysis.

Calculate Attention Score: After capturing the attention weights, the algorithm processes each weight to compute the attention scores. This involves extracting the query, key, and value tensors from the hook outputs. The attention score is computed as per the principles discussed in section 3.3.3.1. The attention scores are then normalized to ensure they are in a range suitable for visualization.

Visualize Attention Maps: In the visualization phase, each normalized attention map is resized to match the dimensions of the input image. A *colormap* (e.g., 'hot') is applied to the attention map to highlight areas of high attention. The attention map is then overlaid on the original image to create a visual representation of where the model is focusing.

Display: Finally, both the original image and the overlaid attention maps are displayed, allowing for an interpretation of how the Vision Transformer model is making its decisions based on different regions of the input image.

Algorithm 3: Calculate and Visualize Attention Weights

Input: Input image, Vision Transformer model

Output: Attention maps visualization

Function `main`**:**

    Data: Input image, Vision Transformer model

    `// Forward Pass and Capture Attention Weights`

    Initialize attention weights list

    for *each layer in Vision Transformer* **do**

        Register hook to capture attention weights

        Pass input image through Vision Transformer

        Compute output features

    `// Calculate Attention Score`

    for *each captured attention weight* **do**

        Extract query, key, value from the hook output

        Compute attention score as $\texttt{Attention} = \texttt{Query} \times \texttt{Key}^{\texttt{T}}$

        $\texttt{Attention(Q,K,V)} = \texttt{A} \times \texttt{V}$

        Normalize attention score

    `// Visualize Attention Maps`

    for *each attention score* **do**

        Resize attention map to match input image dimensions

        Apply colormap (e.g., 'hot') to visualize attention weights

        Overlay attention map on original image

    Display the original image and attention maps

The stressed image along with the corresponding attention maps from the 12 encoder blocks of the Vision Transformer (ViT) is shown in Fig 3.4. Several key observations can be made from these attention maps, including spatial relevance, hierarchical processing, interpreting model decisions, visualization of learned features, and using them as a diagnostic tool for model improvement. Each attention map is resized and overlaid on the original image, with colors indicating the attention intensity:

- Red/Hot Colors: Indicate areas of high attention.
- Yellow/Warmer Colors: Show areas of moderate attention.
- Dark/Cold Colors: Represent areas of low attention.

Spatial Relevance: The spatial relevance can give insights into which parts of the image the model finds important for differentiating between classes. In Layer 1, the model's attention is broadly distributed with some central intensity, indicating that

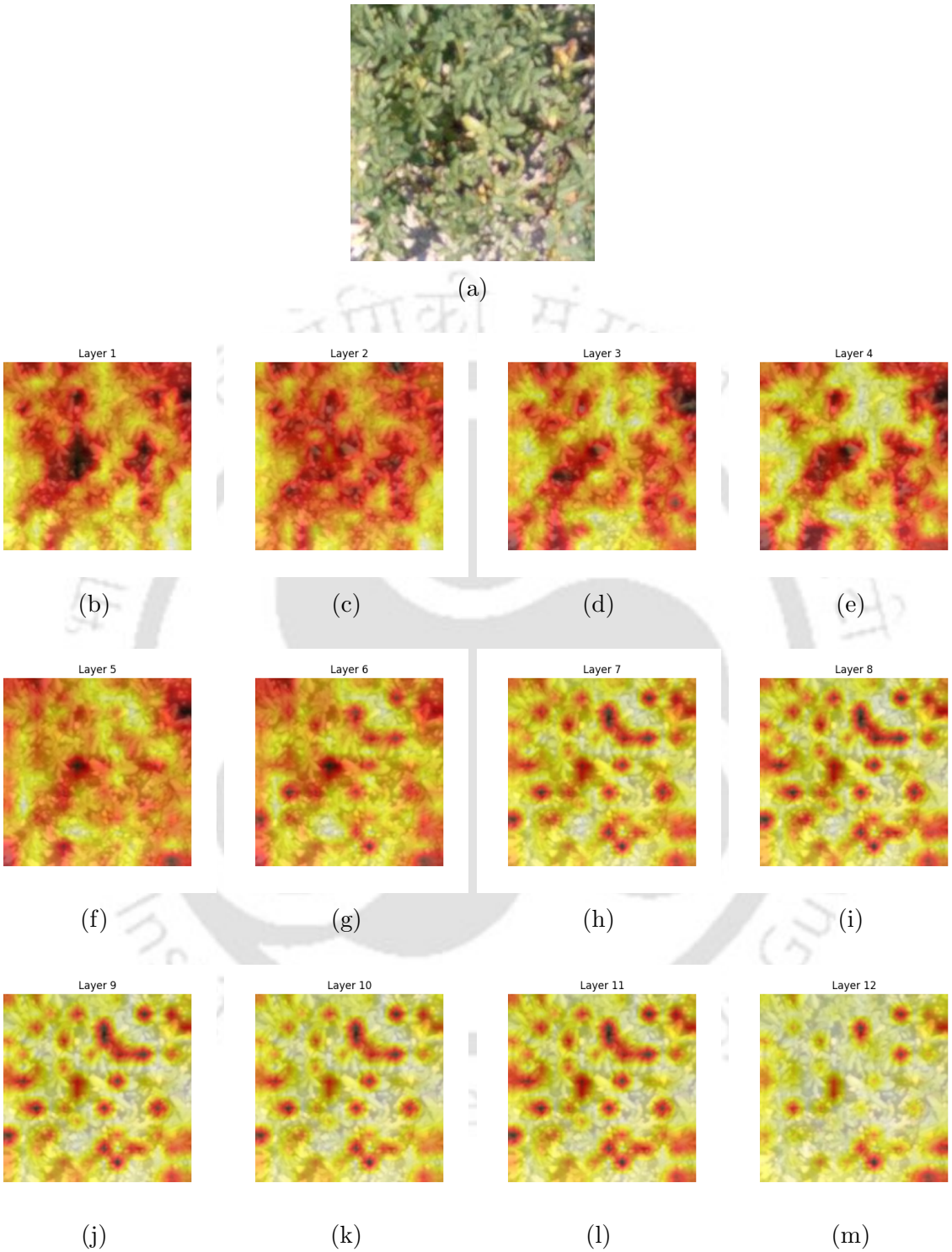


Figure 3.4: A Sample Image (Stressed) and Corresponding Attention Maps from 12 Encoder Blocks.

the model initially captures coarse, global structures of the image. Moving to Layers 2 ,3 and 4, attention becomes increasingly localized, suggesting the model is beginning to identify distinct regions and features relevant to the classification task. Between Layers 5 and 6, the attention patterns grow sharper and more discriminative. These layers appear to focus on intermediate-level features—regions that are potentially indicative of stress patterns but not yet fine-grained. Layers 7 through 11 show a high concentration of attention in specific, small regions with strong contrast, indicating that the model is now attending to fine details, such as localized stress indicators in the vegetation. Notably, Layer 12 diverges from the preceding layers. The attention becomes more diffused and less sharply defined compared to Layers 8–11. This indicates that the final encoder block form a more holistic representation by aggregating information from previous layers, striking a baalnce of local details with global context for the final classification decision.

Hierarchical Processing: As we move through the layers of the ViT, attention maps can show how the model progressively refines its understanding of the image. In the lower layers, attention is distributed across large regions, capturing global context. As we ascend through the layers, the attention narrows down to more specific features, highlighting finer details and important objects within the image. This hierarchical processing is crucial for the model to effectively balance global and local information.

Interpreting Model Decisions: By visualizing attention maps, we can interpret why the model made certain predictions. For instance, if the attention maps highlight specific objects or patterns in an image, it suggests that those elements influenced the classification decision. This interpretability can help validate the model's decisions and identify potential biases or weaknesses.

Visualization of Learned Features: The attention maps provide a visual representation of the features learned by the ViT. Unlike abstract feature vectors, these maps directly relate model activations to spatial locations in the input image. This visualization helps in understanding how the model processes visual information and forms its internal representations.

Diagnostic Tool for Model Improvement: Analyzing attention maps can serve as a diagnostic tool for improving model performance. By examining where the model attends and comparing it with ground truth or human perception, we can identify areas where the model might be lacking or where it might over-emphasize certain features. This feedback loop can guide model refinement and training strategies.

In summary, attention maps in image classification tasks with Vision Transformers provide a transparent view into the inner workings of the model, highlighting which parts of the input image contribute most to its decision-making process.

3.4.2 Performance of ViT+SVM

We used a pre-trained Vision Transformer (ViT) model to extract features from both the training and testing datasets. These extracted features were then utilized by an SVM to identify stress, aiming to distinguish between stressed and healthy images. The implementation was done using the *PyTorch* framework, leveraging both its core library and the *torchvision* module. Key libraries such as *torch*, *torchvision*, *csv*, *pandas*, and *scikit-learn* were imported to facilitate feature extraction, data transformations, file handling, and classification tasks.

We employed the Vision Transformer (*ViT*) models for feature extraction and subsequent evaluation through a Support Vector Machine (SVM) classifier. Primarily, we utilized the *ViT-B/16* architecture, sourced from *torchvision.models*, and loaded pre-trained weights using *torch.load* from a specified path. The model was initialized and set to evaluation mode to facilitate inference-based feature extraction. To standardize the inputs, all images were resized to 224×224 pixels and transformed into tensors. Data loaders were constructed for both the training and testing datasets, using a batch size of 32. Feature extraction was encapsulated in a dedicated function invoked separately for the training and testing data loaders. The resulting features, along with their corresponding labels, were saved to CSV files for downstream processing. This process was executed within a no-gradient context *torch.no_grad()* to optimize computational efficiency. The hyperparameters and implementation details used for ViT-based feature extraction and classification are summarized in Table 3.4.

Parameters	
ViT Parameters	**SVM Parameters**
Image size for resizing: 224x224	Learning rate: 0.001
Batch size for data loaders: 32	Kernel: radial basis function
–	All other parameters are default in scikit-learn

Table 3.4: Parameters in ViT+SVM Framework

To analyze the influence of model capacity, we extracted features under two experimental scenarios. In the first, we used the *ViT-B/16* model (final weights from scenario 8). In the second scenario, we utilized *ViT-L/16*, taking into account the weights in scenario 4. In both settings, the extracted features were fed into a Support Vector Machine (SVM) for classification. Performance evaluation was conducted on specified test set as well as using 5-fold cross-validation to generate the Receiver Operating Characteristic (ROC) curves, compute the mean Area Under the Curve (AUC), and estimate the mean classification accuracy.

The confusion matrix components, as shown in Table 3.5 , show that ViT-

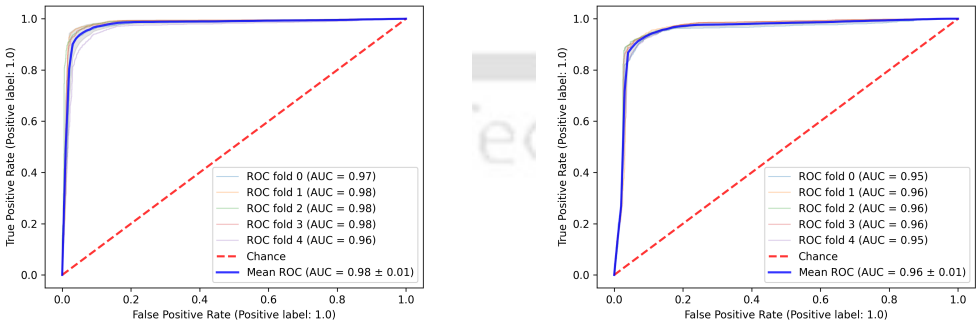
Table 3.5: Confusion matrix components and test accuracy across for ViT+SVM.

Scenario	TP	TN	FP	FN	Test Accuracy
ViT-B/16 +SVM	650	353	48	84	0.8837
ViT-L/16 +SVM	640	364	37	94	0.8845

Table 3.6: Mean Accuracy and AUC for k-fold cross validation

Approaches	Accuracy for Specified Test Set	Mean Accuracy with 5-Fold	Mean AUC with 5-Fold
ViT-B/16 +SVM	0.8837	0.9435	0.98
ViT-L/16 +SVM	0.8845	0.9208	0.96

B/16 produces a lower number of false negatives (FN), a critical metric in drought stress identification tasks, where missing stressed cases can lead to substantial consequences. The overall performance evaluation of the proposed models reveals the superiority of the ViT-B/16 + SVM architecture over its larger counterpart, ViT-L/16 + SVM. Table 3.6 detailed that ViT-B/16 consistently achieved higher mean accuracy, mean AUC, indicating better generalization capability and robustness. Additionally, as shown in Fig. 3.5a, the ROC curve for ViT-B/16 exhibits a higher mean AUC (0.98 $\pm$ 0.01) compared to ViT-L/16 (0.96 $\pm$ 0.01 in Fig. 3.5b), with consistently strong performance across all folds. The tighter clustering of the ROC curves around the upper-left corner for ViT-B/16 also indicates more reliable classification across varying thresholds. These results demonstrate that despite having a smaller architecture, ViT-B/16 offers more accurate and explainable performance, making it a preferable feature extractor for efficient and dependable drought stress detection.



(a) ViT-B/16 +SVM (b) ViT-L/16 +SVM

Figure 3.5: ROC curves depicting the model's performance

3.4.3 Comparison of the Models

Performance on Specified test set:

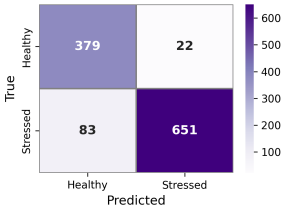
Our two proposed approaches are compared with a previously published model [62] on the same dataset. Table 3.7 presents a comparative evaluation of the three models for drought stress detection, reporting precision, recall, F1-score, and overall accuracy on the designated test set. The CNN-based framework demonstrates competitive performance, particularly in terms of stressed plant precision (0.9673). However, it lags behind in F1-score and overall accuracy when compared to the Vision Transformer with Transfer Learning. The ViT+SVM model shows relatively weaker performance, especially in classifying healthy plants, suggesting that the SVM integration may not fully exploit the representational capacity of ViTs for this task. The Vision Transformer with Transfer Learning outperforms both the CNN-based and ViT+SVM models across all metrics, making it the most effective model for distinguishing drought-stressed from healthy plants.

Fig. 4.5 presents the confusion matrices for three models—CNN Framework, ViT+SVM, and ViT with Transfer Learning—which correspond to the performance metrics previously computed and detailed in Table 3.7 for the specified test set. In drought stress detection, minimizing false negatives (FN) is crucial, as higher FN could lead to significant oversight in applications like drought stress detection, where failing to identify stressed plants might delay critical interventions. Among the three models compared, the Vision Transformer with Transfer Learning (ViT-TL) demonstrates the lowest number of false negatives (73), highlighting its superior sensitivity to stress conditions. In contrast, the CNN-based framework and the ViT+SVM model yield higher false negatives—83 and 84 respectively—indicating a comparatively weaker ability to detect all truly stressed plants. Apparently, false positives (FP) are generally less severe than false negatives, they still represent inefficiencies in resource utilization, such as unnecessary irrigation or pesticide application to healthy plants. Both ViT-TL and the CNN framework exhibit equally low false positives (22), whereas the ViT+SVM model produces a significantly higher FP count of 48, suggesting an over-prediction of stress in healthy plants. Overall, ViT-TL achieves a well-balanced performance with the lowest FN and equally low FP, indicating its robustness in both detecting stressed plants accurately and avoiding unwarranted false alarms.

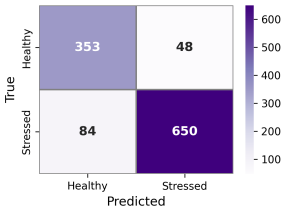
K-fold cross validation: Table 3.8 presents a comparative performance analysis between two model configurations: the Vision Transformer (ViT) with transfer learning and ViT-B/16 combined with a Support Vector Machine (SVM) classifier. In the first approach, the ViT model is fine-tuned end-to-end on the target dataset, whereas in the second, ViT is used as a feature extractor, and an SVM classifier is trained on the

Table 3.7: Performance comparison of the models for the specified test set.

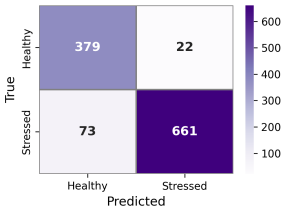
Model	Stressed			Healthy			Accuracy
	Precision	Recall	F1-score	Precision	Recall	F1-score	
CNN Based Framework [62]	0.9673	0.8869	0.9252	0.8203	0.9451	0.8788	0.9075
ViT+SVM	0.9312	0.8856	0.9078	0.8078	0.8803	0.8421	0.8837
ViT-TL	**0.9678**	**0.9005**	**0.9328**	**0.8385**	**0.9451**	**0.8883**	**0.9162**



(a) CNN Framework [62]



(b) ViT+SVM



(c) ViT-TL

Figure 3.6: Confusion matrices comparison for CNN, ViT+SVM, and ViT with Transfer Learning models.

extracted features. The evaluation was performed using k-fold cross-validation, and the results are reported in terms of the mean F1-scores for both the Healthy and Stressed classes, along with the overall mean accuracy. The ViT with transfer learning demonstrated superior performance with F1-scores of 0.97 and 0.98 for the Healthy and Stressed classes, respectively, and a mean accuracy of 97.43%. In comparison, the ViT-B/16 + SVM configuration achieved F1-scores of 0.93 (Healthy) and 0.95 (Stressed), and a mean accuracy of 94.35%. These results highlight the effectiveness of end-to-end fine-tuning, which enables the model to learn task-specific representations more effectively than the frozen feature extractor approach.

Table 3.8: Performance comparison between ViT with Transfer Learning and ViT+SVM with Optimal Weights for k-fold cross validation.

Model	Mean F1-score		Mean Accuracy
	Healthy	Stressed	
ViT with Transfer Learning	0.97	0.98	0.9743
ViT-B/16 + SVM	0.93	0.95	0.9435

3.5 Summary

Drought stress represents a severe threat to crop yield and quality, disrupting normal plant growth and survival rates. Detecting early signs of drought stress is crucial for effective crop management and intervention. Traditional methods, primarily reliant on

Convolutional Neural Networks (CNNs), have made significant strides in capturing spatial hierarchies in image data. However, Vision Transformers (ViTs) offer a compelling alternative by leveraging self-attention mechanisms to capture long-range dependencies and complex spatial relationships, thus enhancing the detection of subtle drought stress indicators.

Our study successfully addressed the challenge of stress identification using smaller datasets by harnessing the feature extraction capabilities of Vision Transformers. Unlike conventional CNN architectures, Vision Transformers utilize self-attention mechanisms to effectively capture relationships across different parts of the image, enabling them to model long-range dependencies, which is particularly advantageous for complex datasets where traditional CNNs may struggle to capture global context. Moreover, Vision Transformers can handle images of varying resolutions without necessitating architectural modifications, enhancing their flexibility in handling diverse datasets. Our framework also emphasized explainability by generating attention maps, which provide insights into the model's focus areas within the images, thereby offering transparency in its decision-making process.

The comparative analysis of a CNN-based framework, ViT-TL, and ViT+SVM showed that Vision Transformers (ViT) with transfer learning significantly improve classification performance. Its performance is consistent across various data splits, thus offering a reliable and accurate solution for drought stress identification, with attention maps providing valuable insights into the model's decision-making process. These findings highlight the potential of advanced deep learning techniques to enhance agricultural practices and decision-making, paving the way for more effective crop management strategies.

The work embodied in this chapter is published as:
Patra, A. K., Varshney, A., & Sahoo, L. (2025). An explainable Vision Transformer with transfer learning based efficient drought stress identification. *Plant Molecular Biology*, 115(4), 98.

Chapter 4

Gradient-Guided Unlearning in a Novel Lightweight Hybrid CNN for Enhanced Drought Stress Identification

4.1 Abstract

Precise detection of drought stress is essential to safeguard crop productivity, requiring advanced technologies for timely intervention and improved agricultural management. Traditional methods are limited, leading to the development of innovative techniques such as hyperspectral remote sensing, machine learning with deep learning architectures, and geophysical imaging to non-invasively monitor crop health and water status. These modern approaches offer enhanced precision and scalability, enabling targeted interventions to mitigate losses and enhance crop resilience against drought. In recent years, Convolutional Neural Network (CNN) and Vision Transformer architectures have been widely explored for drought stress identification; however, these models generally require a large number of trainable parameters, limiting their use in resource-limited and real-time agricultural settings. To address this challenge, we propose a novel lightweight hybrid CNN framework inspired by ResNet, DenseNet, and MobileNet architectures. The framework achieves a remarkable 15-fold reduction in trainable parameters compared to conventional CNNs and ViTs, while maintaining competitive accuracy. In addition, we introduce a machine unlearning mechanism based on a gradient-norm-based influence function, enabling targeted removal of the influence of specific training data and thereby improving model adaptability and robustness. The method was evaluated on an aerial image dataset of potato fields with expert-annotated healthy and drought-stressed regions. Experimental results show that our framework achieves high accuracy and low computational costs, highlighting its potential as a practical, scalable, and adaptive solution for drought stress monitoring in precision agriculture, particularly under resource-constrained conditions.

Keywords: {Deep Learning, Machine Unlearning, Lightweight Convolutional Neural Network, Drought Stress, Precision Agriculture}

4.2 Introduction

Drought stress is one of the most severe abiotic factors threatening global crop productivity and food security. The early and precise identification of drought stress is essential for timely intervention, efficient resource management, and sustaining agricultural yields [12, 144, 27]. Conventional assessment methods, including manual field inspections and physiological measurements, are often labor-intensive, subjective, and not scalable for large-scale agricultural operations. Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have enabled automated and high-throughput analysis of plant health from aerial imagery, opening new possibilities for precision agriculture [145].

Initial research on drought and water stress relied heavily on traditional machine learning and handcrafted features. For example, Zhuang et al. [42] used color and texture features with a gradient boosting decision tree (GBDT) for water stress detection in maize, though later studies showed CNNs significantly outperformed GBDT in both accuracy and robustness [43]. Further progress has been made through advanced imaging modalities and machine learning pipelines, such as hyperspectral imaging in groundnut [47], transfer learning with DenseNet-121 for soybean drought severity [38], and hybrid CNN-LSTM approaches for chickpea [21]. Comparative studies have also reported GoogLeNet as highly accurate for multiple crops [51] and tree-based classifiers (e.g., Random Forest, Extra Trees) as effective for chlorophyll fluorescence-based stress detection [53].

In potato crops, Butte et al. [61] applied deep learning to multi-modal aerial imagery, while Patra et al. [146] proposed an explainable CNN framework that improved classification accuracy. Other works confirmed the reliability of SVMs and Random Forests with hyperspectral data [60, 48], while Goyal et al. [63] designed a custom CNN surpassing state-of-the-art models for maize drought detection.

Recently, Vision Transformers (ViTs) have gained traction in plant stress identification due to their self-attention mechanism, which captures long-range dependencies and global context [127, 128, 129]. ViTs and hybrid variants have achieved superior performance in plant disease and stress classification [130, 35, 23, 131]. For example, ViTs have outperformed Inception V3 in tomato disease classification [132], while Parez et al. [133] demonstrated accuracy gains with a fine-tuned ViT using fewer parameters. Recent innovations include reduced transformer encoders for drought stress identification [147], lightweight modules [135], attention-head optimization [136], and transformer–CNN hybrids [139]. Despite these successes, most works emphasize accuracy, with limited focus on parameter efficiency—a key barrier for real-time deployment

in resource-constrained agricultural environments.

Both advanced CNNs and ViT-based models remain computationally demanding and parameter-heavy, which limits their suitability for edge deployment and large-scale agricultural applications. To address this challenge, some recent attempts have explored lightweight models for drought stress detection, such as the approaches proposed by Patra et al. [146, 147], Li et al. [138] and Gole et al. [135]. As CNN architectures such as ResNet [148], DenseNet [149], and MobileNet [150] employ relatively fewer parameters while incorporating architectural innovations that enhance both accuracy and efficiency, their combined strengths present a promising direction. Therefore, developing a lightweight hybrid model that integrates these architectures can be highly effective and is the focus of investigation in this work.

Equally important, deployed model must adapt to evolving environmental conditions, mislabeled data, and outliers. In this context, *machine unlearning*—the process of selectively removing the influence of specific training data—has emerged as a tool for enhancing adaptability, and error correction [151, 152]. While the bulk of unlearning research focuses on privacy, there are emerging connections to plant stress detection by supporting adaptive models. Bourtoule et al.[153] proposed the foundational "Sharded, Isolated, Sliced, and Aggregated" (SISA) approach for efficient unlearning in deep models, while Cao and Yang [154] earlier formalized the notion of statistical unlearning by bounding the effect of deleted samples. Later, Ginart et al.[155] explored "amnesiac" machine learning that selectively forgets training data. Practical advances in deep neural networks include gradient-based influence estimation for fast unlearning [156], certified removal guarantees [157], and unlearning mechanisms in computer vision tasks [158]. These studies provide a strong foundation for adapting unlearning to agricultural stress identification, where mislabeled samples and environmental noise are common.

In this work, we propose a framework that addresses both efficiency and adaptability. Specifically, we:

- Design a novel lightweight hybrid CNN architecture, inspired by ResNet, DenseNet, and MobileNet, that achieves a 15-fold reduction in trainable parameters compared to CNN and ViT models, while maintaining high accuracy for drought stress identification.
- Develop a gradient norm-based machine unlearning mechanism that enables selective removal of the influence of specific training samples, thereby enhancing adaptability.
- Validate the proposed framework on a challenging real-world aerial imagery dataset of potato crops [61], demonstrating superior efficiency, performance, and scalability

compared to state-of-the-art methods.

4.3 Material and Methods

4.3.1 Data Set Description

We used the same dataset as used in Chapter 2 and Chapter 3.

4.4 Methodology

The proposed model is a novel convolutional neural network (CNN) architecture that synergistically combines the principles of **MobileNetV2**, **Resnet** and **DenseNet**. The design goal is to achieve an optimal balance between model *efficiency*, *representational power*, and *gradient flow*, critical for binary image classification tasks where both precision and computational feasibility are essential.

The proposed framework begins with an initial convolutional layer for low-level feature extraction, followed by four residual blocks that enable deeper feature learning through skip connections. A dense block is then employed to encourage feature reuse and efficient representation learning, after which a transition layer reduces dimensionality and controls model complexity. This is followed by a bottleneck block that captures the most salient features in a compact form. The high-level representations are aggregated using global average pooling, passed through a fully connected dense layer for further processing, and finally mapped to the output layer for prediction. The overall architecture of the proposed framework is depicted in Fig. 4.1, illustrating the sequence of layers from the initial convolution through to the output layer. The detailed steps and parameters of each constituent block in the proposed model are summarized in Algorithm 4. A brief description of each block follows.

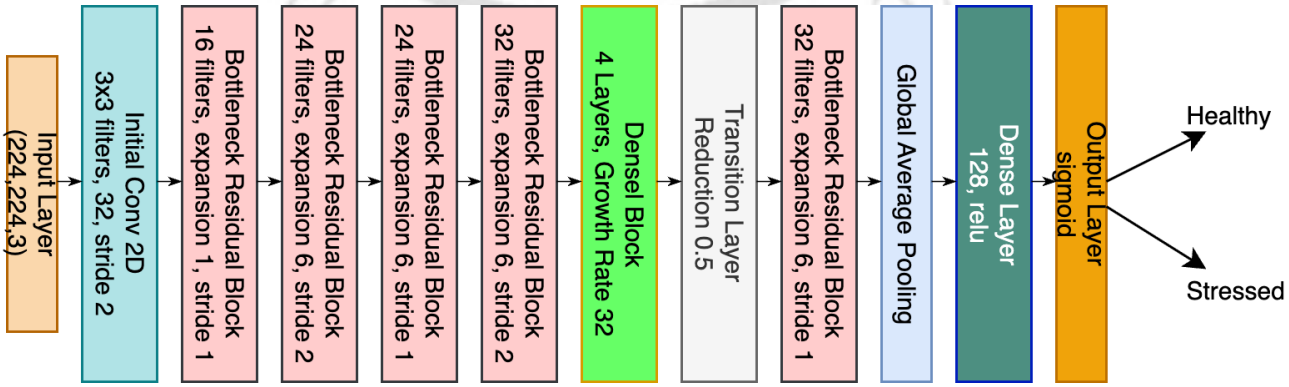


Figure 4.1: Schematic diagram of the lightweight network architecture

4.4.0.1 Input and Initial Convolution

The model processes RGB input images with a resolution of 224×224. An initial convolutional layer applies 32 filters of size 3×3 with stride 2 and "same" padding. This layer is followed by batch normalization and ReLU6 activation to standardize feature distributions and introduce non-linearity while preserving numerical stability for low-precision computation.

4.4.0.2 Bottleneck Residual Blocks

Inspired by MobileNetV2, the Bottleneck Residual Block comprises three phases:

- **Expansion Phase**: A 1×1 pointwise convolution increases channel dimensionality, enabling a richer representation of features.
- **Depthwise Convolution**: A 3×3 depthwise convolution performs lightweight spatial filtering independently across channels, drastically reducing computational cost.
- **Projection Phase**: A second 1×1 convolution projects the features back to a lower dimension.

These blocks integrate *skip connections* when input and output dimensions match, preserving information and enhancing gradient flow during backpropagation. Their significance lies in reducing computation while retaining accuracy—ideal for scalable deep learning models.

4.4.0.3 Dense Block

A Dense Block consisting of four convolutional units is embedded mid-network. Each unit applies BatchNorm $\rightarrow$ ReLU $\rightarrow$ 3×3 Conv2D and concatenates its output with all preceding feature maps. This design enforces: feature reuse across layers, improved gradient propagation, diminished vanishing gradient issues, especially in deeper networks.

Dense connectivity introduces a collective memory mechanism, where each layer benefits from the cumulative knowledge of all previous layers, enhancing both convergence speed and generalization.

4.4.0.4 Transition Layer

To manage the increasing number of channels from the Dense Block, a Transition Layer is applied. It performs:

- Channel compression using a 1×1 convolution,

- Downsampling via 2×2 average pooling.

This module plays a *regularization role*, reducing model complexity and the risk of overfitting, while maintaining vital feature information. The transition also helps in controlling memory usage and computation.

4.4.0.5 Final Processing and Classification

Following the transition, a final Bottleneck Residual Block is applied. The feature map is then reduced to a 1D vector using Global Average Pooling (GAP), which reduces overfitting and parameter count compared to fully connected layers. A Dense Layer with 128 units and ReLU activation processes the features, followed by a sigmoid-activated output unit to generate binary classification results.

4.4.0.6 Optimization Strategy

The model is trained using the *Adam* optimizer with a scheduled exponential learning rate decay, which gradually lowers the learning rate to promote stable convergence. The initial learning rate is set to 0.001, with a decay rate of 0.9 applied every $2 \times$ steps_per_epoch. Here, the steps per epoch are calculated as

$$\text{steps\_per\_epoch} = \left\lfloor \frac{\text{num\_train\_samples}}{\text{batch\_size}} \right\rfloor.$$

The **binary crossentropy** loss function is employed to handle binary classification, particularly in cases of class imbalance. The optimizer's learning rate schedule is designed to balance fast convergence at the start with fine-grained updates toward the end.

4.4.1 Machine Unlearning Mechanism

Machine unlearning refers to the process by which a model forgets or discards data used for training. This is particularly useful when certain data becomes irrelevant or when privacy issues arise (such as removing data associated with a specific individual).

In the process of machine unlearning, the model is first trained on the entire dataset. For each sample, an influence score is computed based on the gradients of the model's predictions with respect to the loss function. This score measures how much a particular sample contributed to the model's learning. The influence score for sample i can be expressed as:

$$\text{Influence Score for sample i} = \|\nabla \mathcal{L}(f(\mathbf{x}_i), y_i)\|$$

Algorithm 4: Customized lightweight CNN Model

Input: Input shape: $224 \times 224 \times 3$, Training samples: *num_train_samples*

Output: Trained CNN model with bottleneck residual blocks and DenseNet layers

1 **Sub-functions:**

- `BottleneckBlock(`x, *filters, expansion_factor, stride*`)`:
  - Conv2D($filters \times expansion\_factor$, $(1, 1)$), BatchNorm, ReLU(6.0);
  - DepthwiseConv2D$(3, 3)$, stride = *stride*, BatchNorm, ReLU(6.0);
  - Conv2D($filters$, $(1, 1)$), BatchNorm;
  - If $stride == 1$ & $x.shape[-1] == filters$, skip connection;
- `DenseBlock(`x, *num_layers, growth_rate*`)`: **for** $i = 1$ *to num_layers* **do**
  - BatchNorm, ReLU, Conv2D(*growth_rate*, $(3, 3)$);
  - Concatenate input x with Conv2D output;
- `TransitionLayer(`x, *reduction*`)`:
  - BatchNorm, ReLU, Conv2D($reduction \times x.shape[-1]$, $(1, 1)$);
  - AveragePooling $(2, 2)$, strides = 2;

Main Model:

- Input: $224 \times 224 \times 3$;
- Conv2D(36, $(3, 3)$), stride = 2, BatchNorm, ReLU(6.0);
- `BottleneckBlock(`x, *16, 1, 1*`)`, `BottleneckBlock(`x, *24, 6, 2*`)`;
- `BottleneckBlock(`x, *24, 6, 1*`)`, `BottleneckBlock(`x, *32, 6, 2*`)`;
- `DenseBlock(`x, *4, 32*`)`;
- `TransitionLayer(`x, *0.5*`)`;
- `BottleneckBlock(`x, *32, 6, 1*`)`;
- Global Average Pooling;
- Dense(128, ReLU), Dense(1, Sigmoid);
- Initial learning rate: 0.001;
- Steps per epoch: *num_train_samples*/*batch_size*;
- Exponential decay learning rate schedule;
- Compile model (Adam, Binary Crossentropy, Accuracy);

Return: Compiled CNN model;

where $\nabla\mathcal{L}$ is the gradient of the loss function $\mathcal{L}$ with respect to the model's prediction $f(\mathbf{x}_i)$ for sample i, and y_i is the true label.

Based on these influence scores, certain samples (such as the least influential ones) are removed from the training dataset. The rationale is that removing these samples will not significantly degrade the model's performance, but will help reduce the model's reliance on less useful or sensitive data. Formally, we can identify and discard the samples with the smallest influence scores:

$$S_{\text{remove}} = \{\mathbf{x}_i : \mathcal{I}(\mathbf{x}_i) \text{ is among the lowest in } S\}$$

After removing the data points with the lowest influence scores, the model is retrained on the reduced dataset. This mimics a "forgetting" mechanism, as the model no longer has access to those removed data points, but retains the important features learned from the remaining data. Let $S_{\text{new}} = S \setminus S_{\text{remove}}$ be the new training set. The model is retrained using the reduced dataset:

$$\hat{f}_{\text{new}} = \arg\min_{f} \sum_{\mathbf{x}_i \in S_{\text{new}}} \mathcal{L}(f(\mathbf{x}_i), y_i)$$

Thus, the model $\hat{f}_{\text{new}}$ is updated without the influence of the removed data points, effectively forgetting them while retaining the learned features from the remaining data. Figure 4.2 illustrates the overall machine unlearning framework, depicting key steps from influence score calculation to data removal and model retraining.

The influence score for each image in the dataset is calulated based on the gradients of the model's predictions with respect to the input image, as explained below. The influence score indicates how much an individual image influences the model's decision. The higher the influence score, the more sensitive the model's prediction is to changes in that image.

4.4.1.1 Influence Score Calculation

To quantify how much a sample contributes to learning, we define an influence score as follows.

Given a trained model M with parameters θ, the prediction for an input image $\mathbf{x}$ is

$$\hat{y} = f(\mathbf{x}; \theta).$$

To quantify the effect of an image on the model's decision, we compute an *influence score*

using gradients. The procedure consists of the following steps:

1. **Gradient Computation:** For each image, compute the gradient of the prediction $\hat{y}$ with respect to the input $\mathbf{x}$:
$$\nabla_{\mathbf{x}} \hat{y} = \frac{\partial \hat{y}}{\partial \mathbf{x}}.$$
This captures pixel-level sensitivity.

2. **Flattening:** Each gradient tensor is flattened into a one-dimensional vector $\mathbf{g}$ for ease of handling.

3. **Concatenation:** Gradients from all layers, $\mathbf{g}_1, \ldots, \mathbf{g}_n$, are concatenated into a single vector:
$$\mathbf{v} = \text{Concat}(\mathbf{g}_1, \ldots, \mathbf{g}_n).$$

4. **Norm Calculation:** The L2 norm of $\mathbf{v}$ gives the influence score:
$$s = \|\mathbf{v}\|_2 = \sqrt{\sum_i v_i^2}.$$

5. **Final Result:** Repeating the above steps for the dataset D yields an array of scores:
$$\mathbf{S} = [s_1, s_2, \ldots, s_N].$$

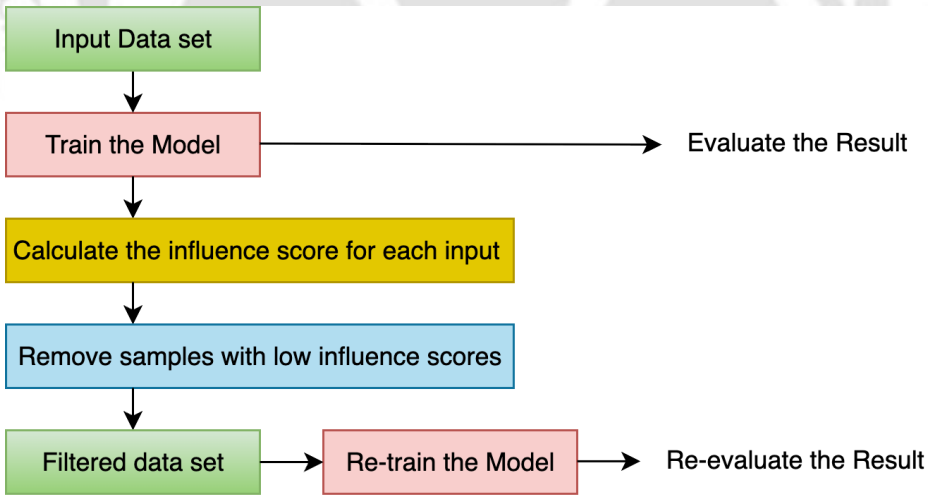


Figure 4.2: Machine Unlearning Framework

Algorithm 5 outlines the step-by-step procedure for computing influence scores. For each image in the dataset, the model gradients are computed, flattened, and concatenated across layers. The L2 norm of this concatenated gradient vector yields a single scalar influence score, which quantifies the sensitivity of the model's prediction to that

Algorithm 5: Calculate Influence Scores

Input: Trained model M, DataFrame D with image filenames and labels
Output: Array of influence scores S

Initialize empty list $S \leftarrow [\,]$;
foreach *row* r *in* D **do**
    $p \leftarrow$ filename from r;
    $y \leftarrow$ label from r converted to float32;
    $X \leftarrow$ `PreprocessImageForCustomCNN`(p);
    $G \leftarrow$ `ComputeGradients`$(M, X, \text{ExpandDims}(y))$;
    Initialize empty list F;
    foreach *gradient tensor* g *in* G **do**
        Append `Reshape`$(g, (-1))$ to F;
    $V \leftarrow$ `Concat`(F);
    $s \leftarrow \| V \|_2$;
    Append s to S;
return S;

image. By iterating over the dataset, the algorithm produces an array of influence scores that highlights which images exert the strongest effect on the model's decision-making. These influence scores enable targeted data removal to facilitate efficient and effective machine unlearning.

4.5 Results and Discussion

The model was implemented in Python version 3.10.14 using machine learning libraries such as *Keras*, *TensorFlow*, *Scikit-learn*, *Pandas*, *NumPy*, and *Matplotlib*. Training was performed with a batch size of 128 for 50 epochs, optimized with the *Adam* optimizer (initial learning rate = 0.001) and *categorical cross-entropy* as the loss function.

4.5.1 Performance of the Proposed Model

The customized CNN model was evaluated under three distinct scenarios: (i) training without augmentation, (ii) training with augmentation, and (iii) retraining with augmentation after machine unlearning (removal of 5% training data).

Eight augmentation techniques were applied to the training dataset, including rescaling, rotation, width and height shifting, shear transformation, zooming, and horizontal flipping. These parameters (Table 4.1) expanded data variability and helped mitigate overfitting. Their role in stabilizing training dynamics is further highlighted in the learning curve analysis (Fig. 4.4).

To quantify the contribution of individual training samples, influence scores

Table 4.1: Data Augmentation Parameters Used for Model Training

Parameter	Value
Rescale	$\frac{1}{255}$
Rotation Range	30
Width Shift Range	0.2
Height Shift Range	0.2
Shear Range	0.2
Horizontal Flip	True
Vertical Flip	True
Fill Mode	Nearest

were calculated as described in Section 3.1.1. The histogram of scores (Fig. 4.3) revealed that while most samples exerted moderate influence, a subset exhibited either very high or very low values. In the proposed unlearning strategy, the least influential 5% of samples were excluded, since their removal was expected to reduce noise and redundancy without significantly degrading model performance. This formed the basis of the machine unlearning step, aimed at reducing reliance on less useful data. The resulting reduced dataset was then used to retrain the model, effectively forgetting the discarded samples while retaining the critical patterns learned from the remaining data.

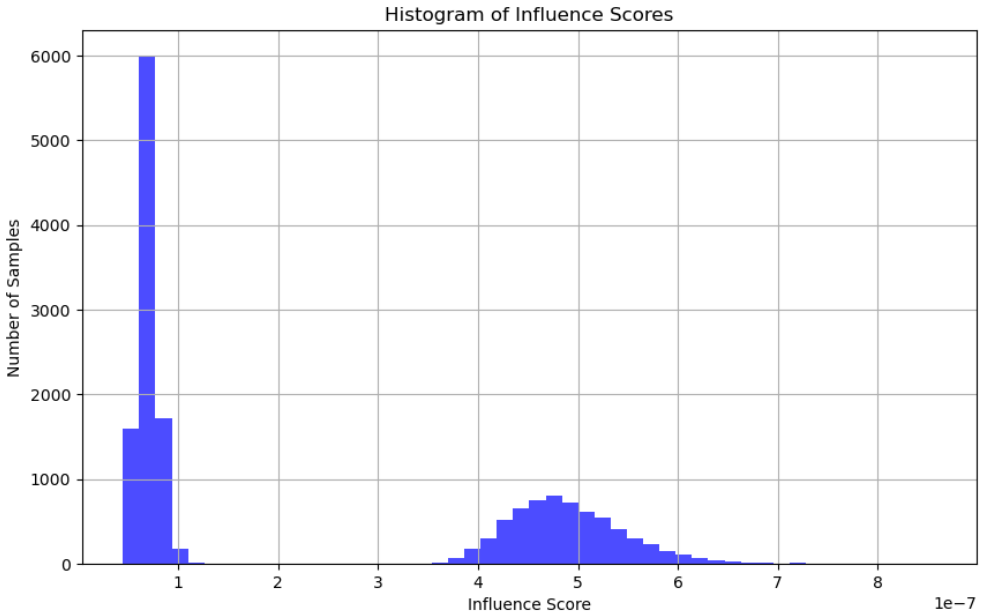
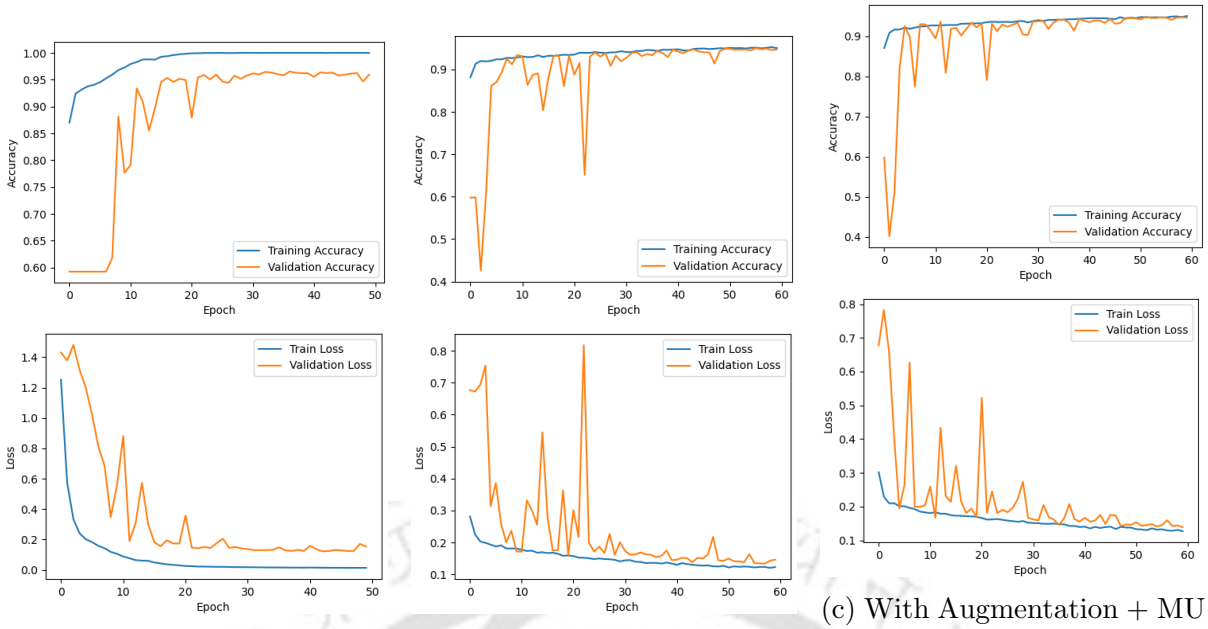


Figure 4.3: Distribution of influence scores for training samples (50 bins).

4.5.1.1 Learning curves

The training and validation behavior of the CNN under different experimental settings is shown in Fig 4.4. In the no augmentation scenario (Fig. 4.4a), training accuracy



(a) No Augmentation (b) With Augmentation (c) With Augmentation + MU (5%)

Figure 4.4: Learning curves (accuracy and loss) for the CNN under three scenarios: (a) without augmentation, (b) with augmentation, and (c) with augmentation + machine unlearning (5% data removal). Each subfigure shows Accuracy (top) and Loss (bottom).

increased rapidly and plateaued near 0.99, whereas validation accuracy fluctuated widely during the first 20 epochs before stabilizing around 0.95. The divergence between training and validation loss, along with the instability of validation loss, indicates overfitting and limited generalization despite high training accuracy.

With the introduction of data augmentation (Fig. 4.4b), both training and validation curves exhibited improved stability. Validation accuracy consistently tracked training accuracy above 0.90, and validation loss displayed fewer spikes compared to the no-augmentation setting. These results confirm that augmentation enhanced generalization and convergence stability.

The augmentation with machine unlearning (5%) configuration (Fig. 4.4c) produced the most stable training dynamics. Training and validation accuracies converged smoothly above 0.93, with both loss curves showing steady declines and minimal divergence. Importantly, validation fluctuations were further reduced compared to the augmentation-only scenario, suggesting that machine unlearning contributed to robust generalization, reduced overfitting, and improved reliability across epochs.

4.5.1.2 Confusion Matrices and Classification Report

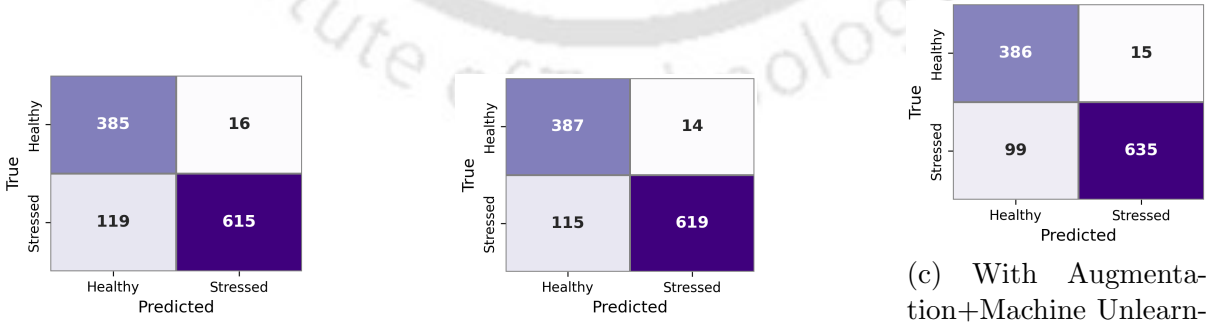
The quantitative evaluation results are summarized in Table 4.2, with confusion matrices shown in Fig. 4.5.

Scenario-wise classification outcomes In the no augmentation setting, the framework achieved an overall accuracy of 88.1%. The stressed class exhibited high precision (0.97) but comparatively lower recall (0.84), leading to mis-classification of stressed samples as healthy. The confusion matrix (Fig. 4.5a) confirms this, with 119 stressed samples incorrectly labeled as healthy, highlighting the model's tendency to under-detect stress conditions despite the healthy class achieving a strong recall of 0.96.

The augmentation-only scenario improved both stability and generalization. Accuracy rose slightly to 88.6%, with stressed precision maintained at 0.98 and recall remaining at 0.84. The confusion matrix (Fig. 4.5b illustrates a modest reduction in misclassifications, with 115 stressed samples misclassified as healthy and only 14 false positives among healthy samples. This indicates that augmentation reduced the imbalance and improved class separability.

The augmentation with machine unlearning (Aug + MU, 5%) scenario yielded the best overall results. Accuracy peaked at 90.0%, with stressed recall improving to 0.87 and F1-score to 0.92, while the healthy class achieved precision of 0.80 and recall of 0.96. As depicted in the confusion matrix (Fig. 4.5c), false negatives for stressed samples decreased to 99, the lowest across all scenarios, with only 15 false positives for the healthy class. This demonstrates that machine unlearning reduced error rates, and minimized false negatives compared to the other two settings.

Comparative insights Taken together, the results from Table 4.2 and Fig. 4.5 highlight that stressed class predictions benefited the most from augmentation and unlearning, as recall consistently improved (0.84 → 0.87). Meanwhile, the healthy class maintained strong recall across all scenarios but exhibited notable gains in precision under the Aug + MU configuration. These improvements emphasize that the integration of data augmentation with machine unlearning achieves the best trade-off between precision and recall, yielding a more generalizable and balanced framework for classification.



(a) No Augmentation (b) With Augmentation (c) With Augmentation+Machine Unlearning

Figure 4.5: Confusion matrices for the three scenario.

Table 4.2: Comparative performance summary of the proposed framework in three scenarios. Bold values indicate the best within each column.

Scenario	Stressed			Healthy			Accuracy	Key Notes
	Precision	Recall	F1-score	Precision	Recall	F1-score		
No Augmentation	0.97	0.84	0.90	0.76	**0.96***	0.85	0.881	Overfits, unstable validation
With Augmentation	**0.98***	0.84	0.91	0.77	**0.97***	0.86	0.886	Better generalization, stable curves
With Aug + MU (5%)	**0.98***	**0.87**	**0.92**	**0.80**	0.96	**0.87**	**0.900**	Best balance, robust generalization, fewer false negatives

Table 4.3: Comparative performance evaluation of various existing works

Model	Stressed		Healthy		F1-Score (Stressed)	F1-Score (Healthy)	Test Accuracy	Trainable Parameters
	Precision	Recall	Precision	Recall				
MobileNet based Pipeline [146]	0.978	0.845	0.773	0.965	0.906	0.862	0.887	3.5 Million
DenseNet121 based Pipeline [146]	0.967	0.887	0.820	0.945	0.925	0.880	0.907	7.09 Million
ViT-TL [147]	0.968	0.901	0.839	0.945	0.934	0.890	0.916	14M
Proposed Framework (Aug + MU) *	0.980	0.870	0.800	0.960	0.922	0.874	0.900	0.231M

4.5.2 Comparison of Performance

The comparative performance of the proposed CNN framework against existing state-of-the-art models is summarized in Table 4.3. Prior pipelines such as MobileNet [146] and DenseNet121 [146] achieved respectable accuracies of 88.7% and 90.7%, respectively, while the ViT-TL model [147] reached the highest reported accuracy of 91.6%. These transformer-based and deep CNN architectures, however, come with significantly larger parameter counts (ranging from 3.5M to 12M), which demand higher computational resources and longer training times.

In contrast, the proposed framework attains a competitive accuracy of 90.0% with stressed and healthy F1-scores of 0.922 and 0.874, respectively. Notably, this is achieved with only 0.231M trainable parameters, representing a 15 to 60 fold reduction in model complexity compared to existing pipelines. The lightweight design translates to reduced memory footprint, faster inference, and improved deployability on edge devices or low-resource settings, where computational constraints often limit the use of heavier architectures.

Another critical insight lies in class-specific performance. While MobileNet and DenseNet pipelines tend to favor healthy class recall, and ViT-TL maximizes stressed class recall, our proposed model achieves a more balanced trade-off between precision and recall across both classes. This is particularly important for stress detection tasks, where false negatives (misclassifying stressed plants as healthy) carry greater risk for agricultural monitoring and decision-making. The machine unlearning strategy further

reduced such errors, resulting in the lowest stressed-class false negative rate (13.5%) among the evaluated models.

Taken together, these findings highlight that the proposed framework delivers state-of-the-art performance with minimal computational overhead, making it well-suited for real-world applications where resource efficiency and balanced predictive power are equally critical. By bridging the gap between accuracy and efficiency, the framework offers a practical alternative to heavy-weight architectures, without compromising generalization.

4.6 Summary

This study proposed a lightweight hybrid CNN integrated with gradient-guided machine unlearning for drought stress identification in potato crops. The framework achieved 90.0% accuracy with only 0.231M parameters, offering state-of-the-art performance with at least 15-fold fewer parameters than existing models. Unlike heavier CNN or transformer-based pipelines, the proposed model delivered a balanced trade-off between precision and recall, while reducing false negatives in stressed plants—a critical requirement for agricultural monitoring. Its compact design makes it highly suitable for real-time deployment in UAVs, edge devices, and other resource-constrained settings. Future work will emphasize expanding to multimodal imagery, improving interpretability, and validating performance across diverse crops and field conditions.

Chapter 5

Improved Classification of Nitrogen Stress Severity in Plants Under Combined Stress Conditions Using Spatio-Temporal Deep Learning Framework

5.1 Abstract

In the real-world scenario multiple stresses often appear together in the field such as drought and nutrient. As drought can inhibit nutrient uptake and increase water requirements for nutrient conversion, while imbalances in plant nutrients can also weaken cell membranes, increasing susceptibility to drought. This combined stress leads to synergistic negative effects on plant growth and yield, causing greater reductions in seed quality and biomass than either stress alone. Early detection of these stresses is therefore crucial for protecting plant health and implementing effective management strategies. This study proposes a novel deep learning framework to accurately classify nitrogen stress severity in a combined stress environment. Our model uses RGB, multispectral, and two infrared wavelengths to capture a wide range of physiological plant responses from canopy images. These images, provided as time-series data, document plant health across three levels of nitrogen availability (low, medium, and high) under varying water stress and weed pressures. The core of our approach is a spatio-temporal deep learning pipeline that merges a Convolutional Neural Network (CNN) for extracting spatial features from images with a Long Short-Term Memory (LSTM) network to capture temporal dependencies. We also devised and evaluated a spatial-only CNN pipeline for comparison. Our CNN-LSTM pipeline achieved an impressive accuracy of 98%, surpassing significantly the 80.45% accuracy of spatial-only model and the accuracy of around 76% of previously reported machine learning methods. The results show strong potential as a reliable tool for identifying nitrogen stress early and accurately, helping farmers make better management decisions and maintain healthier crops.

Keywords: {Deep Learning, Nitrogen Stress, Combined Stress, Spatio Temporal Framework, Precision Agriculture}

5.2 Introduction

Among all essential macro-nutrients, nitrogen (N) deficiency represents a major constraint on plant growth, development, and productivity [159]. As a fundamental component of amino acids, proteins, nucleic acids, and chlorophyll [160], nitrogen plays a central role in multiple physiological and metabolic processes. Its deficiency disrupts these pathways, resulting in reduced leaf area, chlorosis, lower leaf count, and stunted plant height [161]. Beyond nutrient limitations, abiotic stressors such as drought and biotic pressures like weed competition frequently co-occur, compounding the negative effects on plant health. For example, water stress restricts nutrient mobility and uptake, thereby intensifying the impacts of nitrogen deficiency [162]. In natural environments, plants rarely face single stress factors in isolation. Rather, stress events often occur simultaneously or sequentially, interacting in synergistic or antagonistic ways [163, 164]. These multi-stress combinations induce overlapping phenotypic symptoms, complicating efforts to diagnose the underlying causes [165]. Despite this, the majority of plant stress phenotyping studies have focused on single-stress scenarios, with relatively limited progress in disentangling or classifying coexisting stresses [166, 167, 168]. This gap demands the need for advanced tools capable of modeling the intricate, multi-dimensional dynamics of plant responses under combined stress conditions.

Recent advances in imaging and sensor technologies have transformed stress detection through high-throughput phenotyping platforms [10]. These approaches, coupled with machine learning (ML) and deep learning (DL), enable non-invasive, scalable monitoring of plant health and yield-related traits [16, 121]. However, the majority of existing frameworks remain restricted to single stress scenarios, overlooking the complex interactions that occur when nutrient deficiency coincides with other environmental pressures. This gap limits the applicability of current models in real-world conditions, where stresses such as nitrogen deficiency, drought, and weed pressure often co-occur. To address this challenge, we propose a spatio-temporal deep learning framework that leverages pre-trained CNNs in combination with LSTMs to capture both spatial features and temporal growth dynamics, enabling accurate classification of nitrogen stress severity under combined drought and weed pressure. The key contributions of our work are as follows:

- We developed a hybrid MobileNetV2–LSTM model that leverages transfer learning and temporal encoding to classify nitrogen stress severity. Our model achieves a high classification accuracy of 98%, significantly outperforming traditional spatial-only and machine learning-based approaches.
- To validate our architecture, we implemented a spatial-only CNN pipeline, enhanced

via data augmentation and transfer learning, achieving 80.45% accuracy. This serves as a baseline to demonstrate the advantage of temporal modeling.

- Our results demonstrate that integrating spatial information over time is significantly more effective for predicting nitrogen stress severity than relying on spatial information alone. The proposed spatio-temporal framework outperforms our spatial-only pipeline as well as other machine learning methods employed in previous studies.

5.3 Materials and Methods

In this section, we first introduce the experimental dataset used in the study. We then describe the proposed spatio-temporal framework, followed by the spatial-only network architecture. Finally, we outline the performance metrics employed for model evaluation.

5.3.1 Data Description

Table 5.1: Combined Stress Treatment

Nitrogen Input	Water Input	Weed Pressure	Box Numbers
Low	Sufficient	None	22,23,24
Medium	Sufficient	None	4,5,6
High	Sufficient	Medium	7,8,9
High	Sufficient	High	10,11,12
Medium	Low	None	13,14,15
High	Sufficient	High	16
High	Sufficient	High	17
High	Sufficient	High	18
Low	Sufficient	Medium	25,26,27
Medium	Low	High	19,20,21
Low	Low	None	28,29,30

The dataset utilized in this study is derived from the work by Khanna et al.[7], who established a comprehensive plant phenotyping framework to investigate the physiological effects of combined abiotic (drought) and biotic (weed competition) stresses alongside nitrogen deficiency in sugar beet (Beta vulgaris L.). Their experimental design closely mimicked field-realistic stress scenarios, enabling systematic evaluation of plant responses under factorial combinations of low, medium, and high nitrogen supply, with varying water availability and weed presence. This design aimed to disentangle the complex interactions between multiple, simultaneously occurring stressors, which often induce overlapping phenotypic responses such as reductions in leaf area, biomass, and visible symptoms like chlorosis.To classify nitrogen stress levels in sugar beet plants, we utilized

canopy images from multiple modalities, namely RGB, infrared, and multi-spectral. The images were collected using an Intel® RealSense™ ZR300 camera—providing RGB and dual infrared (stereo IR) channels and a Ximea MQ022HG-IM-SM5X5 camera capturing multi-spectral images.

Each stress factor was applied at different severity levels. Nitrogen availability was assessed using three levels—low, medium, and high—representing deficient, sufficient, and surplus nitrogen supply, equivalent to 20, 40, and 80 kg/ha, respectively. Weed pressure was categorized as no weeds, medium pressure (chickweeds), or high pressure (chickweeds and grasses). Water supply was manually regulated at two levels—sufficient and low. As a result, plants experienced varying combinations of these three stressors at any given time. The experiment was intentionally structured to emulate real-world conditions by applying nitrogen deficiency, drought, and weed competition both individually and in combination. Nitrogen deficiency levels with varying water and weed pressure captured by RGB, stereo infrared, and hyperspectral sensors on a specific day are illustrated in Fig. 5.1. The detailed stress treatments are summarized in Table 5.1, which lists the combinations of nitrogen input, water input, and weed pressure, along with the corresponding cultivation box numbers. The treatment matrix (i.e., Table 5.1) assigns 27 cultivation boxes to these combinations. The dataset includes images from 16 measurement dates throughout the growth period, featuring 27 boxes (9 per nitrogen level). Images from 14 dates were retained for analysis, excluding the first two dates due to early-stage germination where stress symptoms were minimal. For each nitrogen level category, nine boxes were imaged across four modalities, yielding 504 images per category (14 dates × 9 boxes × 4 modalities). Altogether, the dataset comprised 1,512 processed images spanning all three nitrogen levels. To ensure data quality, all images were preprocessed by cropping to remove irrelevant background regions prior to analysis.

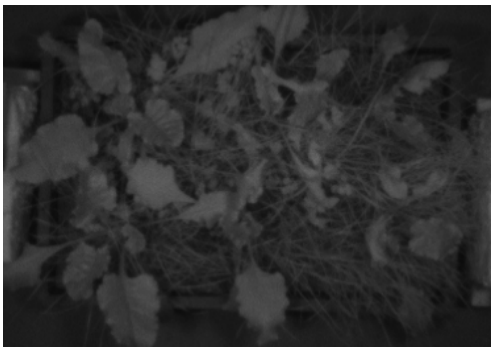
5.3.2 Proposed Framework

5.3.2.1 Spatio-Temporal Framework

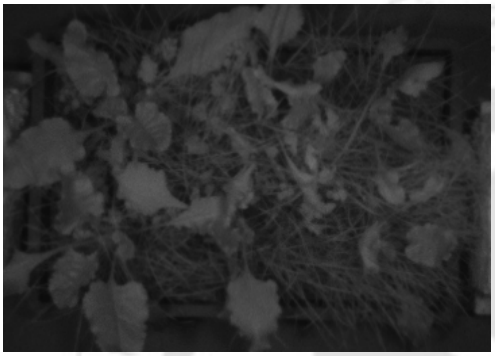
In this study, we present a deep learning framework that integrates spatial and temporal features to classify images into three nitrogen severity levels: low, medium, and high. The model integrates a CNN for feature extraction and a LSTM network for temporal sequence modeling. The overall architecture of the proposed framework is depicted in Fig. 5.2. The architecture of the proposed CNN-LSTM hybrid consists of the following components:

1. **Feature Extractor:** A pre-trained `MobileNetV2` (with `imagenet` weights) served as the base CNN, where the classification head was removed. The network's output was passed through a Global Average Pooling layer to obtain a fixed-size feature

(a) Low nitrogen (sufficient water, medium weed), RGB



(b) Medium nitrogen (low water, high weed), Infrared 1



(c) Medium nitrogen (low water, high weed), Infrared 2

(d) High nitrogen (sufficient water, high Weed), Multi-Spectral

Figure 5.1: Nitrogen deficiency levels(with varying levels of water and weed) on a specific day captured by RGB, Infrared and Multi-spectral sensor.

vector per image. This CNN was wrapped within a `TimeDistributed` layer to process each frame of the sequence independently while sharing weights.

2. **Temporal Encoder:** The sequence of image features was then fed into an LSTM layer with 128 hidden units to learn temporal patterns across the sequence.

   LSTM networks are an extension of recurrent neural networks (RNNs) designed to address the vanishing gradient problem and effectively capture long-term dependencies in sequential data [169]. In LSTM models, a memory cell with gating mechanisms enables the network to retain and utilize information over extended sequences, allowing for the reading, writing, and deletion of information from its memory. These gating mechanisms, comprised of forget, input, and output gates, play crucial roles in managing the flow of information within the LSTM unit [170]. An LSTM unit consists of three main components:

   (a) **Forget Gate (f_t)**: Evaluates the relevance of existing information stored in the memory cell. It decides which information to retain and which to discard based on the input at the current time step (x_t) and the previous hidden state

(h_{t-1}). Mathematically, the output of the forget gate (f_t) is computed using a sigmoid activation function:

$$f_t = \sigma\left(W_{f_h} h_{t-1} + W_{f_x} x_t + b_f\right)$$

where W_{f_h} and W_{f_x} are weight matrices, and b_f is the bias.

(b) **Input Gate (i_t) and Candidate Cell State ($\tilde{c}_t$)**: Determines how much new information should be added to the memory cell. It consists of a sigmoid layer that controls the update and a "tanh" layer that generates a vector of new candidate values. The input gate output (i_t) and the candidate cell state ($\tilde{c}_t$) are computed as follows:

$$i_t = \sigma(W_{ih} h_{t-1} + W_{ix} x_t + b_i)$$

$$\tilde{c}_t = \tanh(W_{ch} h_{t-1} + W_{cx} x_t + b_c)$$

The candidate cell state represents the new information to be added to the memory cell.

(c) **Memory Update and Output Gate**: Updates the memory cell content based on the forget gate output (f_t), input gate output (i_t), and candidate cell state ($\tilde{c}_t$). The updated cell state (c_t) is calculated as:

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

where $\odot$ denotes element-wise multiplication.

Finally, the output gate controls which parts of the cell state contribute to the output. The output gate's output (o_t) and the final hidden state (h_t) are computed as:

$$o_t = \sigma\left(W_{oh} h_{t-1} + W_{o_x} x_t + b_o\right)$$

$$h_t = o_t \odot \tanh\left(c_t\right)$$

The output gate output (o_t) determines the relevance of the current cell state,

and the final hidden state (h_t) represents the LSTM's output at the current time step.

In summary, LSTM models utilize gated memory cells to effectively capture and retain long-term dependencies in sequential data, addressing the limitations of traditional RNNs. The forget, input, and output gates enable the LSTM to selectively process and utilize information, making it a powerful tool for tasks involving sequential data analysis and prediction.

3. **Fully Connected Layers:** The output of the LSTM layer (128 units) was followed by Batch Normalization and Dropout (0.25), then passed through a Dense layer with 64 units and ReLU activation with L2 regularization, followed by another Batch Normalization and Dropout (0.25), and finally an output layer with softmax activation for multiclass classification.

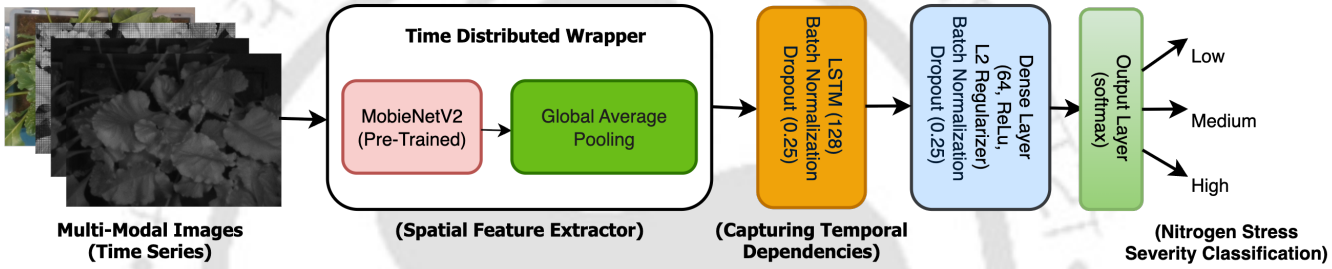


Figure 5.2: Spatio-Temporal Deep Learning Framework for Nitrogen Stress Severity Classification

Algorithm 6 outlines the steps involved in training and evaluating the proposed framework. The dataset comprises images grouped into three class-based folders, with each image filename containing a date stamp in the format `YYYYMMDD`. These dates were extracted and parsed to create a chronological order within each class. To capture the temporal dynamics, we generated overlapping image sequences of fixed length (5 images per sequence), preserving their temporal order. Each sequence was labeled according to its class, resulting in a structured dataset for temporal learning. All images were resized to `224 × 224` pixels and normalized to a [0, 1] range. Each sequence was stacked into a 4D tensor with dimensions `(sequence_length, height, width, channels)`, i.e., `(5, 224, 224, 3)`, forming the input to the model. After sorting the images by date, sequences were created using a sliding window approach within each class. Each sequence of five consecutive images was treated as one sample, and the corresponding class label was assigned. Label encoding was performed using `LabelEncoder`, and categorical labels were one-hot encoded to be used with softmax-based classification.

To ensure reliable evaluation and generalization, we adopted a 5-fold Stratified Cross-Validation scheme. Stratification maintained class distribution across folds,

Algorithm 6: Training and Evaluating the CNN-LSTM model with K-Fold Cross Validation

Data: Dataset from directory `DATA_DIR`, sequence length `SEQUENCE_LEN`, number of folds `N_SPLITS`, epochs `EPOCHS`, batch size `BATCH_SIZE`, random state `RANDOM_STATE`

Result: Learning curves, validation reports, and trained model for each fold

for *each class_name in DATA_DIR* **do**
    Read images from each class directory;
    for *each filename in class directory* **do**
        Extract date from filename and store image paths, class, and date in records list;
Create DataFrame `df` from records with columns `filename`, `class`, and `date`;
Convert date column to datetime format;
for *each (class_name, group) in df.groupby("class")* **do**
    Sort the group by date and generate sequences of length `SEQUENCE_LEN` with corresponding labels;
Encode labels using `LabelEncoder`;
One-hot encode the labels;
Define function `load_seq_batch(seq_file_list)` to load and preprocess image sequences;
for *fold = 1 to N_SPLITS* **do**
    Split the data into training and validation sets using StratifiedKFold;
    Load the training and validation image sequences using `load_seq_batch`;
    Define CNN base model using MobileNetV2 with pre-trained weights;
    Freeze CNN layers;
    Define feature extractor with GlobalAveragePooling2D;
    Define LSTM model with TimeDistributed wrapper, LSTM, BatchNormalization, Dropout, Dense, and final softmax layer;
    Compile with Adam optimizer and categorical crossentropy loss;
    Set up model checkpoint based on validation loss;
    Train model with training and validation data;
    Store training history for each fold;
    Plot and save learning curves (loss and accuracy);
    Evaluate model on validation set and store accuracy;
    Generate confusion matrix and classification report;

allowing balanced training and validation splits. This also enabled assessment of the model's stability across multiple runs.

Freezing the layers of the CNN model refers to preventing the weights of the pre-trained convolutional layers from being updated during the training process. This approach ensures that only the newly added layers, such as the LSTM and Dense layers, are trained. Freezing the CNN layers is particularly beneficial when working with small datasets, as it enables the model to retain the learned feature representations from the pre-trained model. This allows the model to focus on learning the temporal patterns from the sequential data using the LSTM layers without altering the feature extraction process that has already been established by the CNN.

The model is trained using experiments on different subsets of parameters, namely learning rate, sequence length, batch size, and number of epochs. A fixed random state is used to ensure reproducibility of the results. The model was compiled using the *Adam* optimizer, with categorical cross-entropy as the loss function and accuracy as the evaluation metric. To prevent overfitting, the CNN base was frozen during training, and `ModelCheckpoint` was used to save the best-performing model based on validation loss. For each fold, the model was evaluated on the validation dataset using accuracy score, classification report (including precision, recall, and F1-score), and confusion matrix.

To monitor the model's learning behavior, we plotted training and validation loss curves, training and validation accuracy curves, and confusion matrices annotated with prediction counts and color maps. These visualizations supported qualitative assessment and helped identify potential overfitting or underfitting trends.

5.3.2.2 Spatial Framework

A spatial-only baseline architecture is proposed as a reference to compare the results achieved through temporal modeling in the CNN–LSTM framework. For the spatial-only setup, we employed pre-trained MobileNetV2 with weights initialized from the *ImageNet* dataset. The original top layer, configured for 1,000 ImageNet classes, was removed so the backbone could function as a feature extractor. Custom classification layers were appended to adapt the model for our three-class classification task. By leveraging pre-trained weights, we utilized the rich feature representations learned from large-scale data while fine-tuning the model to our target domain.

To retain essential feature extraction capabilities, the first 18 layers of MobileNetV2 were frozen, while the subsequent layers were fine-tuned. On top of the backbone, a GlobalAveragePooling2D layer reduced spatial dimensions, followed by two dense layers (128 and 64 neurons) with ReLU activation and L2 regularization. Dropout layers with a rate of 0.5 were added after each dense layer to improve generalization. The

final classification layer used softmax activation to predict probabilities across the three categories. The architecture is illustrated in Fig. 5.3.

To improve model performance and address limited training data, extensive data augmentation is performed using random rotations, shear transformations, horizontal and vertical flips, and spatial translations. The model is trained using the *Adam* optimizer with an exponentially decaying learning rate and evaluated under a 5-fold stratified cross-validation protocol.

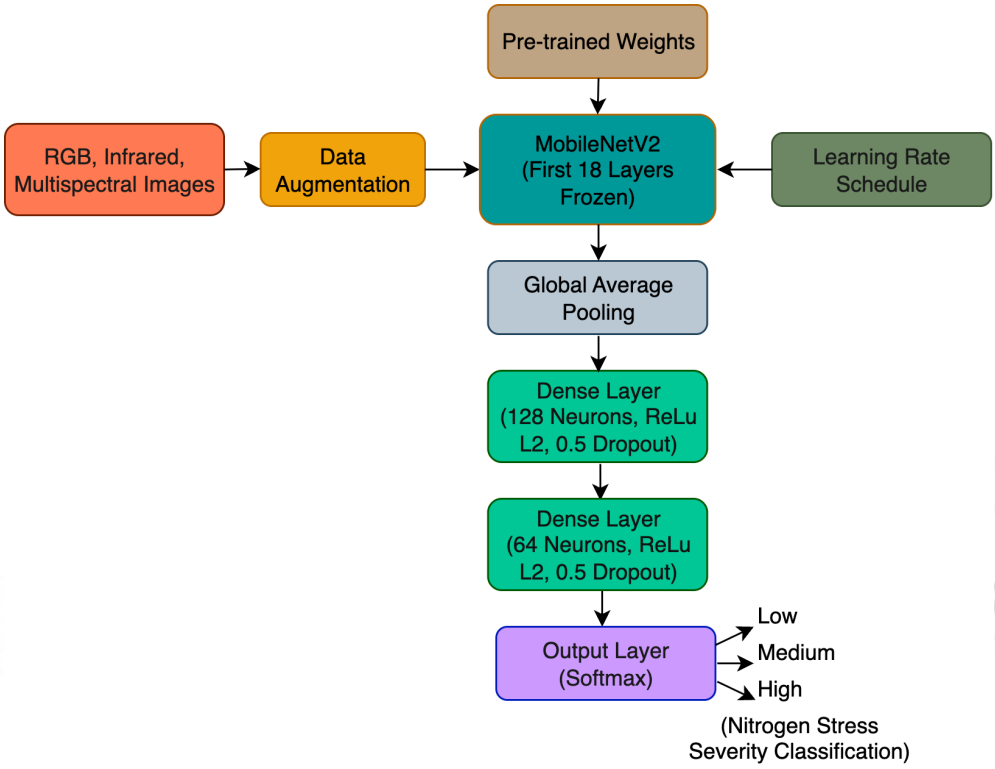


Figure 5.3: Spatial Deep Learning Framework for Nitrogen Stress Severity Classification

5.4 Results and Discussion

The spatio-temporal and spatial-only frameworks were implemented in Python (version 3.10.14) using machine learning libraries, including *Keras*, *TensorFlow*, *Scikit-learn*, *Pandas*, *NumPy*, and *Matplotlib*.

5.4.1 Performance Evaluation of Spatial Temporal Framework

The proposed MobileNetV2–LSTM framework, illustrated in the Fig. 5.2 and detailed in the Algorithm 6, was tested with different subsets of parameters for 20 epochs. The best performance was obtained using the parameter settings summarized in Table 5.2. A 5-fold cross-validation was performed, with the data split controlled using a fixed random state

of 42 to ensure reproducibility. This pure k-fold cross-validation approach maximizes the use of available data by combining the validation and test roles within each fold, making it particularly suitable for relatively small datasets where retaining an entirely separate test set would significantly reduce the amount of training data.

Table 5.2: Best Parameter Settings for the MobileNetV2–LSTM Framework

Parameter	**Value**
Batch Size	16
Sequence Length	5
Learning Rate	0.001
Epochs	20

The model achieved consistently high performance across all folds of cross-validation. As shown in Table 5.3, training, validation, and test accuracies exceeded 98% in every fold, with a mean accuracy of $98.47 \pm 0.0045\%$. This demonstrates the stability and generalization power of the spatio-temporal pipeline.

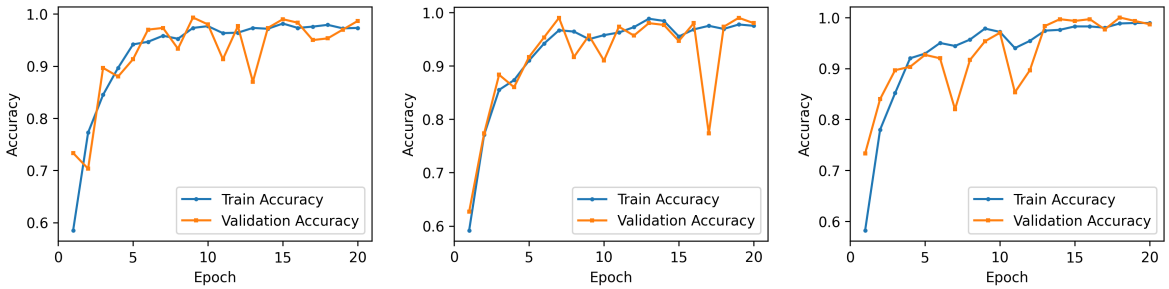
Table 5.3: Fold-wise best performance metrics of MobileNetV2-LSTM spatio-temporal framework during 5-fold cross-validation.

Fold	**Train Accuracy**	**Train Loss**	**Val Accuracy**	**Val Loss**	**Test Accuracy**	**Epoch**
1	0.9733	0.1424	0.9867	0.0920	0.9867	20
2	0.9775	0.1547	0.9900	0.1123	0.9800	19
3	0.9892	0.0989	0.9867	0.0925	0.9867	20
4	0.9758	0.1581	0.9800	0.1373	0.9800	20
5	0.9800	0.1275	0.9900	0.1000	0.9900	20
Mean	0.9792	0.1363	0.9867	0.1068	0.9847	–
Std	0.0061	0.0241	0.0041	0.0189	0.0045	–

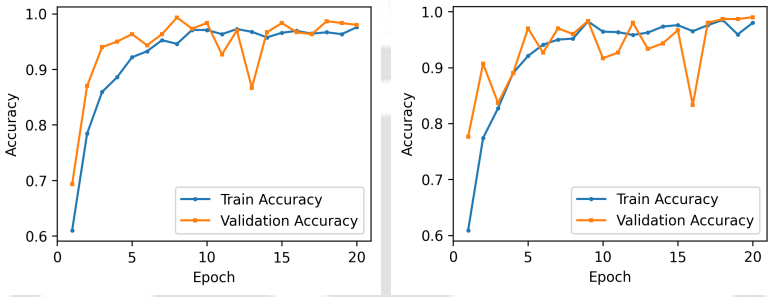
Fig. 5.4 presents the accuracy curves across folds, confirming rapid convergence and minimal variance between training and validation accuracy. Similarly, the loss curves (Fig. 5.5) show stable optimization without overfitting, further supported by confusion matrices in Fig. 5.6, which illustrate near-perfect classification across nitrogen stress levels.

Table 5.4 highlights class-specific performance. Precision, recall, and F1-scores consistently exceeded 0.97 for all nitrogen categories (low, medium, high), confirming that the model effectively captured subtle spectral and morphological features. The macro-averaged F1-score of 0.99 ensures the overall reliability of the spatio-temporal model.

While Khanna et al.[7] utilized the same dataset, leveraging vegetation indices, hyperspectral signatures, and 3D point cloud features over a two-month crop cycle, their modeling approach lacked dynamic learning components such as LSTMs that are capable of capturing temporal dependencies inherent in stress progression. In contrast,



(a) Fold 1 (b) Fold 2 (c) Fold 3

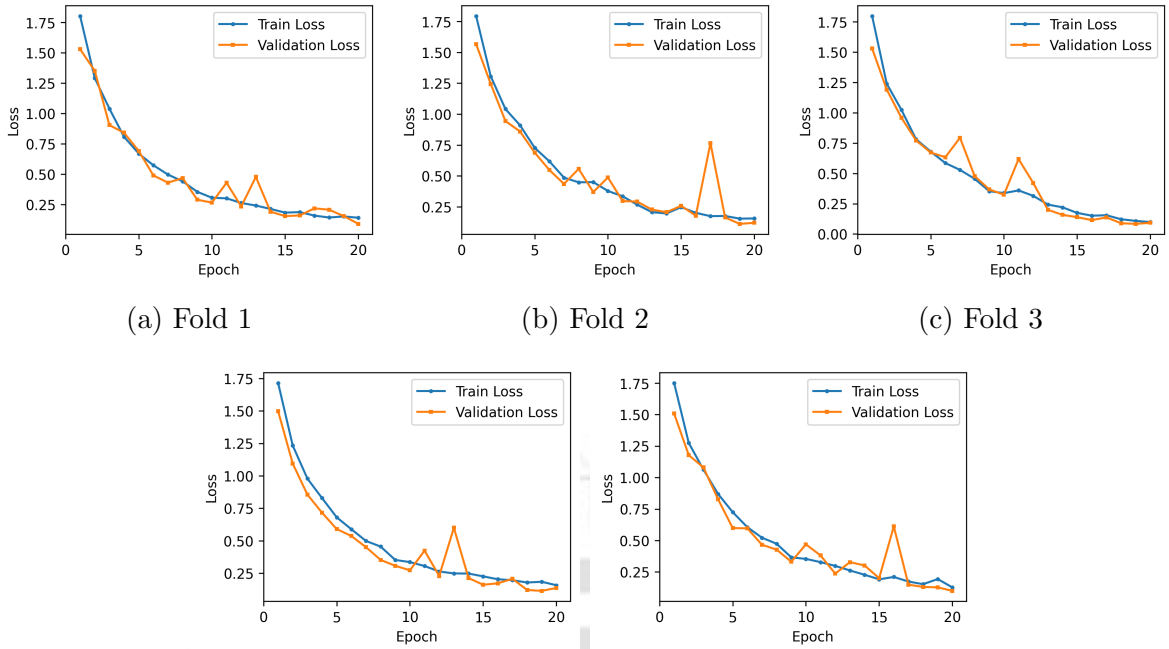


(d) Fold 4 (e) Fold 5

Figure 5.4: Accuracy curves of MobileNetV2-LSTM for 5-fold cross-validation (a–e correspond to Fold 1–5).

Table 5.4: Precision, Recall, and F1-score across 5 folds in MobileNetV2-LSTM

Class	**Metric**	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**
High	Precision	0.98	1.00	0.99	0.97	0.99
	Recall	0.99	0.95	1.00	1.00	1.00
	F1-score	0.99	0.97	1.00	0.99	1.00
Low	Precision	0.99	0.99	0.99	0.98	0.99
	Recall	1.00	1.00	0.97	0.98	0.98
	F1-score	1.00	1.00	0.98	0.98	0.98
Medium	Precision	0.99	0.95	0.98	0.99	0.99
	Recall	0.97	0.99	0.99	0.96	0.99
	F1-score	0.98	0.97	0.99	0.97	0.99



(a) Fold 1 (b) Fold 2 (c) Fold 3

(d) Fold 4 (e) Fold 5

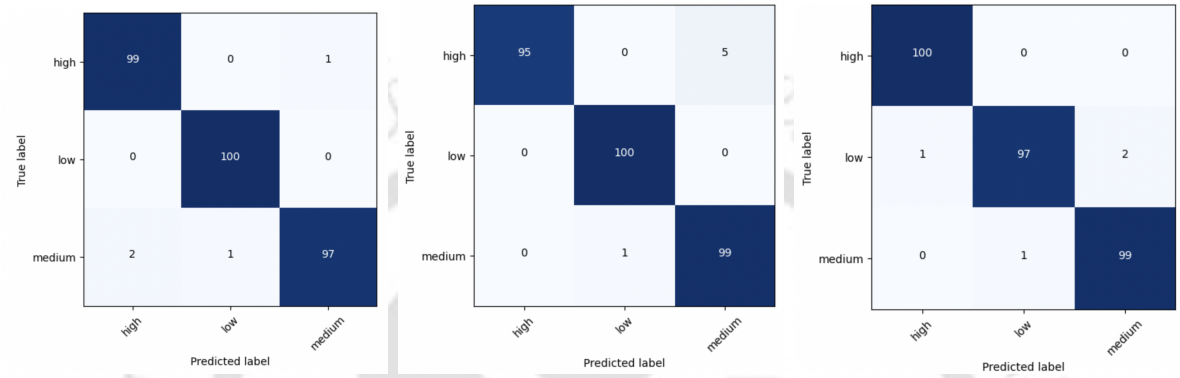
Figure 5.5: Loss curves of MobileNetV2-LSTM for 5-fold cross-validation (a–e correspond to Fold 1–5).

our results demonstrate that the temporal evolution of nitrogen stress is more accurately modeled using the sequential learning capacity of LSTMs, particularly when integrated with lightweight CNN backbones like MobileNetV2. Furthermore, the spatial feature representations extracted via MobileNetV2 enabled superior early stress detection by capturing fine-grained morphological variations, which were not effectively addressed by the handcrafted features employed in Khanna et al.'s pipeline. This advantage is practically relevant during early phenological stages, where visible symptoms may be subtle, and precise morphological cues become essential for timely and accurate stress diagnosis.

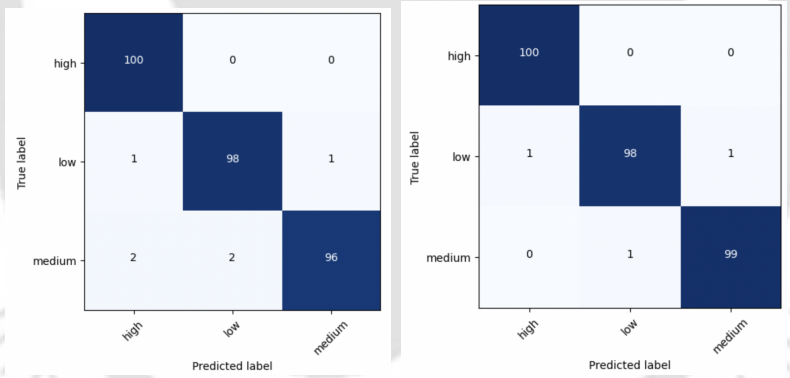
5.4.2 Performance Evaluation of Spatial Framework

To establish a baseline for comparison with the CNN–LSTM temporal framework, a spatial-only CNN model based on MobileNetV2 was designed. The model was trained for 250 epochs with a batch size of 64, ensuring stable gradient updates while balancing computational efficiency. To address limited training data and simulate real-world variability, extensive data augmentation was applied through the `ImageDataGenerator` class with the following parameters: rescale = 1/255, shear range = 0.2, rotation range = 30°, width and height shift range = 0.2, horizontal and vertical flips = `True`, and fill mode = `nearest`, thereby enhancing generalization and reducing overfitting.

For optimization, the Adam optimizer was used with an exponentially decaying learning rate schedule. The initial learning rate was set to 0.001. The decay



(a) Fold 1 (b) Fold 2 (c) Fold 3



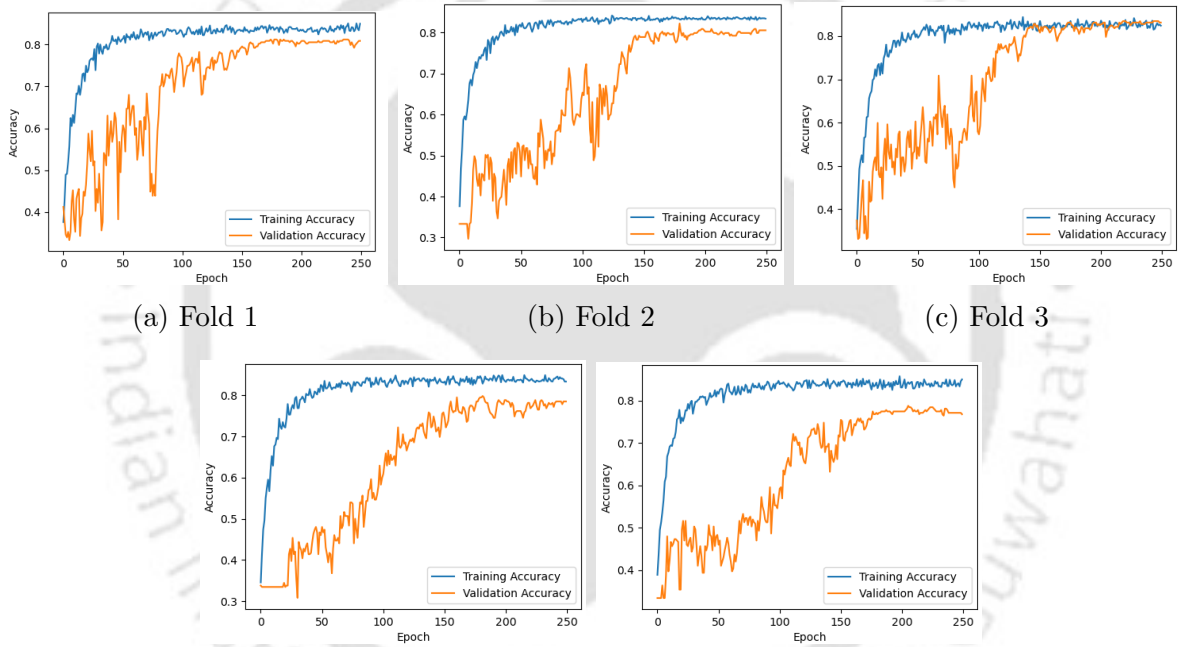
(d) Fold 4 (e) Fold 5

Figure 5.6: Confusion Matrices of MobileNetV2-LSTM for 5-fold cross-validation (a–e correspond to Fold 1–5).

schedule followed an `ExponentialDecay` policy with decay_steps = steps_per_epoch × 10, decay_rate = 0.9, and a staircase update, ensuring the learning rate decreased gradually as training progressed. This strategy stabilized convergence and avoided premature overfitting.

The network employed L2 regularization (0.01) on the dense layers and dropout (rate = 0.5) after each fully connected layer to further prevent overfitting. The training incorporated a `ModelCheckpoint` callback, saving the best-performing model weights per fold based on the lowest validation loss.

The model is trained and evaluated under a 5-fold stratified cross-validation strategy. Fig. 5.3, 5.7, and 5.8 illustrate the architecture, accuracy, and loss curves respectively. Table 5.5 summarizes fold-wise training, validation, and test results, while Table 5.6 reports the precision, recall, and F1-scores across classes.

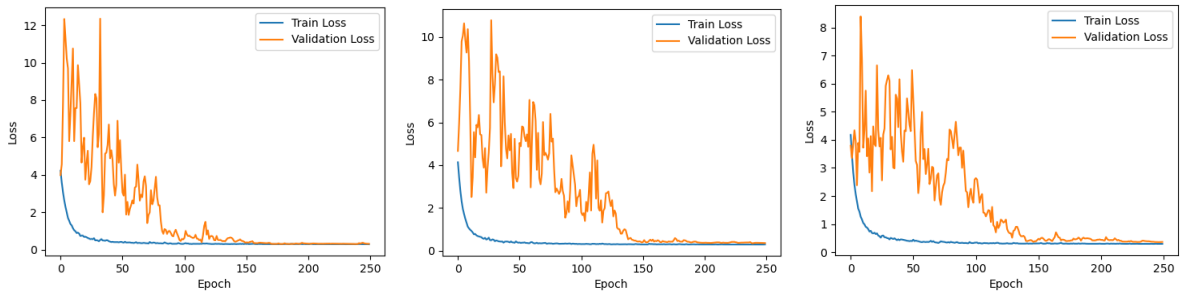


(a) Fold 1 (b) Fold 2 (c) Fold 3

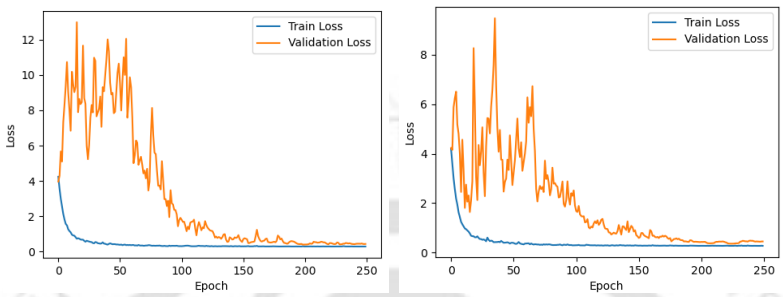
(d) Fold 4 (e) Fold 5

Figure 5.7: Accuracy curves of Spatial Framework for 5-fold cross-validation (a–e correspond to Fold 1–5).

As illustrated in Fig. 5.7, across all folds, the training accuracy rapidly converged to approximately 0.83, while validation accuracy improved gradually and stabilized in the range of 0.78–0.83. Fold 3 demonstrated the strongest alignment between training and validation curves, whereas Folds 4 and 5 displayed a slightly larger gap, suggesting mild overfitting. The corresponding loss curves (Fig. 5.8) revealed sharp decreases in training loss, with validation loss exhibiting high variance in the early epochs but stabilizing after 150 epochs. These observations confirm that data augmentation, dropout, and L2 regularization were effective in mitigating overfitting, while the exponentially



(a) Fold 1 (b) Fold 2 (c) Fold 3



(d) Fold 4 (e) Fold 5

Figure 5.8: Loss curves of Spatial Framework for 5-fold cross-validation (a–e correspond to Fold 1–5).

decaying learning rate ensured stable convergence.

Table 5.5: Training, validation, and test performance across 5 folds in Spatial Framework

Fold	Train Loss	Train Acc.	Val. Loss	Val. Acc.	Test Acc.	Epochs
1	0.2855	0.8412	0.2949	0.8119	0.8119	238
2	0.2943	0.8337	0.3470	0.8053	0.8053	240
3	0.3022	0.8165	0.3636	0.8344	0.8344	245
4	0.2830	0.8322	0.3968	0.7881	0.7881	200
5	0.2747	0.8331	0.3605	0.7848	0.7848	224
Mean	0.2879	0.8313	0.3226	0.8049	0.8049	
Std	0.0098	0.0081	0.0370	0.0186	0.0186	

Table 5.5 shows that the model achieved an average training accuracy of 83.13% and a validation accuracy of 80.49% across folds, with low standard deviation (1.86%). The test accuracy mirrored the validation accuracy (80.49%), highlighting the model's ability to generalize well across unseen data. The highest validation accuracy was observed in Fold 3 (83.44%), while the lowest occurred in Fold 5 (78.48%). Training and validation losses were consistent across folds, with only minor fluctuations.

As shown in Table 5.6, class-wise evaluation revealed balanced predictive capacity, with F1-scores ranging from 0.76 to 0.85 across all classes and folds. For the *High* class, precision was strong in most folds ($\geq$ 0.97) but recall was comparatively lower (0.69–0.75), except in Fold 1 where recall reached 1.00. Conversely, the *Low* class showed

Table 5.6: Precision, Recall, and F1-score across 5 folds in Spatial Framework

Class	Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
High	Precision	0.64	0.99	0.97	0.99	0.99
	Recall	1.00	0.72	0.75	0.73	0.69
	F1-score	0.78	0.83	0.85	0.84	0.81
Low	Precision	1.00	0.64	0.96	0.97	0.61
	Recall	0.72	0.96	0.76	0.65	0.99
	F1-score	0.84	0.77	0.85	0.78	0.76
Medium	Precision	1.00	0.96	0.69	0.62	0.99
	Recall	0.71	0.73	0.99	0.98	0.67
	F1-score	0.83	0.83	0.81	0.76	0.80

complementary trends, with high recall in Folds 2 and 5 (≥ 0.95) but reduced precision (0.61–0.64). The *Medium* class was the most variable, with precision fluctuating between 0.62 and 1.00, though recall generally remained high (0.71–0.99). These results suggest that inter-class boundaries are occasionally ambiguous, leading to trade-offs in precision and recall.

Overall, the spatial framework achieved an average test accuracy of 80.49%, with stable performance across folds and balanced per-class F1-scores. While this indicates strong baseline capability, the variability in class-specific precision and recall reveals the limitations of a purely spatial approach. The nearly 18% performance gap compared to the CNN–LSTM framework (98.47%) clearly demonstrates the critical role of temporal dynamics in nitrogen stress identification. Spatial features primarily capture structural and color-based traits, whereas temporal sequences encode progression patterns essential for distinguishing overlapping symptoms caused by drought, weeds, and nitrogen deficiency.

5.4.3 Comparison with Machine Learning Methods

Notably, this performance surpasses that of traditional models, as reported by Khanna et al.[7]. The comparative analysis of performances is presented in Table 5.7.

Method	Mean Nitrogen Train Accuracy (%)	Nitrogen Test Accuracy (%)	Reference
Decision Trees	63.66	47.62	[7]
LDA	68.47	78.57	
SVM	75.68	80.95	
KNN	62.16	55.95	
Bagged Trees	67.57	63.10	
Subspace Discriminant	70.57	75.00	
Subspace KNN	60.66	66.67	
RUSBoosted Trees	69.37	63.10	
Proposed Spatial Framework	83.13	80.49	[Proposed]
Proposed Spatio-Temporal Framework	97.92	98.47	

Table 5.7: Training and test set classification accuracy for Nitrogen stress using different machine learning methods.

To contextualize our findings, we compared the performance of proposed framework against that of conventional machine learning models and the spatial-only CNN. Traditional classifiers such as Decision Trees, KNN, and Bagged Trees achieved test accuracies below 70%, while SVM achieved a higher accuracy of 80.95%, comparable to the spatial-only CNN. In contrast, the spatio-temporal CNN-LSTM framework markedly outperformed all baselines, achieving 98.47% test accuracy. This performance gain demonstrates that sequential modeling provides a substantial advantage in resolving confounding stress symptoms and effectively predicting nitrogen severity classes.

Khanna et al. [7] rely on explicitly derived plant trait indicators—canopy cover, height, spectral indices, reflectance statistics, and 3D point-cloud features as inputs to classical ML classifiers (SVM, RF, k-NN, etc.). In contrast, our DL frameworks learn discriminative spatial features directly from raw multi-modal images, eliminating the need for domain-specific handcrafted traits and enabling more scalable deployment across crops and environments. Thus, beyond the expected accuracy gap, 5.7 highlights that traditional ML models are fundamentally constrained by handcrafted feature dependence, weak temporal modelling, and reduced sensitivity to overlapping stress phenotypes. Our deep learning–based spatio-temporal frameworks address these limitations by learning hierarchical representations directly from raw data and explicitly modelling stress dynamics over time, leading to substantially improved precision and accuracy under combined stress conditions.

5.5 Summary

This study demonstrates the effectiveness of a spatio-temporal deep learning framework for classifying nitrogen stress severity in sugar beet under combined drought and weed pressure. By integrating MobileNetV2 for spatial feature extraction with LSTM for temporal sequence modeling, the proposed CNN-LSTM approach achieved 98.47% accuracy, substantially outperforming both the spatial-only CNN (80%) and conventional machine learning models (¡76%). The inclusion of temporal dynamics is particularly valuable because it enables early detection of nitrogen deficiency before visible symptoms become severe. Detecting stress at earlier growth stages allows for timely corrective interventions, preventing yield loss and reducing excessive fertilizer use. This advantage underscores the importance of modeling stress progression rather than relying solely on static imaging.

From an application standpoint, the proposed framework offers a lightweight and transferable solution for precision agriculture. It's high accuracy, coupled with the capacity for early detection, makes it suitable for guiding variable-rate fertilizer management, optimizing resource efficiency, and minimizing environmental impacts.

While the results obtained in this study are promising, it is important to note that they were based on a controlled experimental dataset. The generalizability of the findings to unstructured, open-field environments remains a subject of further investigation. Future work should focus on scaling the dataset with field-level images, developing advanced data augmentation strategies, and exploring multi-sensor fusion to further enhance generalizability. These improvements would accelerate the deployment of spatio-temporal deep learning in real-world crop monitoring systems, ensuring sustainable and proactive agricultural practices.

Chapter 6

Conclusion and Future Perspectives

6.1 Conclusion

This thesis investigates a critical challenge in modern agriculture: achieving early, accurate, interpretable, and scalable detection of abiotic stress in crops. Conventional assessment methods, including manual scouting, destructive sampling, and laboratory-based analyses, are unsuitable for large-scale or high-frequency monitoring. To overcome these limitations, the study systematically develops and evaluates lightweight, explainable, and field-deployable deep learning frameworks for plant stress phenotyping using non-invasive imaging data. The research is organized around four core methodological contributions, each addressing a specific gap identified in the literature.

The study introduces an explainable deep learning approach utilizing transfer learning with convolutional neural networks (CNNs) to identify drought stress from UAV-acquired RGB images. The model achieved 90.75% accuracy and employed gradient-based visualizations to highlight stressed regions within plant canopies, thereby enhancing interpretability and transparency and addressing the "black-box" issue associated with standard CNNs. This method demonstrated efficiency, interpretability, and readiness for real-time field deployment.

The research further developed a Vision Transformer (ViT) model and a hybrid ViT-SVM system for drought stress detection. The ViT leverages self-attention mechanisms to identify salient regions in images and capture spatial patterns. It achieved 91.62% accuracy on the designated test set, with performance increasing to 94% (ViT-SVM) and 97% (ViT) with expanded datasets and cross-validation. These models outperformed traditional CNNs, and the hybrid approach enhanced accuracy through margin-based learning. The framework also offered attention-based explanations, facilitating interpretation of model predictions in practical scenarios.

To improve deployment feasibility in resource-constrained agricultural settings, a novel hybrid CNN architecture was proposed. Inspired by ResNet, DenseNet,

and MobileNetV2, this architecture achieved 90% accuracy while reducing the number of trainable parameters by nearly fifteen-fold. Additionally, a gradient-guided machine unlearning mechanism was introduced to systematically eliminate non-contributive or noisy training samples, leading to improved accuracy, reduced overfitting, and enhanced generalization. The resulting lightweight model is adaptive and suitable for real-time deployment on edge platforms.

The research was further extended to assess the severity of nitrogen stress under combined conditions, including drought and weed competition. Two complementary pipelines were developed: a spatial classifier based on MobileNetV2 and a spatio-temporal MobileNetV2-LSTM framework. The spatio-temporal model achieved up to 98% accuracy and effectively captured the progression and interaction of multiple stress factors over time, offering insights into the dynamic nature of plant stress. This framework underscores the importance of temporal context in stress phenotyping, enabling more accurate assessments of stress severity and supporting real-time agricultural monitoring.

The proposed frameworks make substantial contributions to precision agriculture by enabling early detection of plant stress, often preceding the appearance of visible symptoms, thereby supporting proactive decision-making. The explainability of these models facilitates targeted resource management, such as efficient irrigation, fertilizer application, and high-throughput phenotyping, which enhances resource utilization, reduces input costs, and promotes sustainable practices. Additionally, the scalability and deployability of these lightweight models render them suitable for field deployment on UAVs, edge devices, or mobile platforms, ensuring practical application in real-world agricultural environments.

In summary, this thesis demonstrates the potential of lightweight, explainable, and spatio-temporal deep learning models to advance plant stress phenotyping in precision agriculture. By integrating explainability and multimodal information fusion, the research represents a substantial improvement over existing methods and establishes a strong foundation for AI-driven agricultural solutions. The proposed models are adaptable to various crops, stress types, and imaging modalities, ensuring broad applicability across diverse agro-ecological contexts. These frameworks bridge the gap between advanced AI research and practical agricultural applications, providing significant value for farmers, agronomists, and the wider agricultural community.

6.2 Future Perspectives

Recent advances in plant phenotyping increasingly utilize machine learning (ML) and deep learning (DL) to analyze large-scale, multi-modal datasets, enabling more accurate

identification, classification, and quantification of abiotic stresses. Imaging-based ML methods, multi-sensor fusion, and spatio-temporal modeling have emerged as effective tools to address the limitations of traditional phenotyping, such as reliance on manual scoring and the need for extensive labeled datasets. Techniques including transfer learning, semi-supervised learning, and active learning enhance efficiency in scenarios with limited annotations, while self-supervised learning supports robust feature extraction from unlabeled data.

Despite these advancements, several limitations persist in current phenotyping frameworks. Reliance on labeled datasets remains a significant challenge, as models often require time-consuming and subjective dataset generation. The limited diversity of stress types evaluated, with a primary focus on drought and nitrogen stress, restricts applicability to other abiotic stresses such as heat, salinity, and flooding. Furthermore, environmental variability, including extreme lighting and fluctuating weather conditions, can impact model performance in real-world field settings. The study's single-season and single-site constraints further limit the generalizability of findings across diverse agro-climatic zones.

To address these limitations, future research should investigate multi-stress and stress-interaction modeling, extending beyond the current emphasis on drought and nitrogen stress. Integrating multimodal sensing, including hyperspectral, thermal, and physiological sensors, will enhance model robustness and expand the range of plant responses captured. Additionally, self-supervised and semi-supervised learning frameworks can reduce dependence on large labeled datasets, thereby improving model performance in data-scarce environments. Further innovations, such as real-time deployment on edge and IoT platforms, longitudinal studies across multiple seasons and locations, and the integration of genomic and phenomic data, will strengthen the accuracy, scalability, and applicability of AI-driven plant stress monitoring.

Spatio-temporal learning methods, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and temporal convolutional networks, enable modeling of stress progression over time, thereby improving stress management strategies. Incorporating continuous ground-truth measurements and linking stress identification with crop yield prediction models will facilitate more accurate severity estimation and provide deeper insights into the agronomic impact of stress.

Lightweight architectures optimized for edge deployment, in combination with Internet of Things (IoT) sensor networks, can facilitate real-time, high-throughput phenotyping in resource-constrained environments using UAVs or mobile devices. Techniques such as few-shot learning and federated learning can reduce reliance on large labeled datasets and enable privacy-preserving model training across distributed farms. The ap-

plication of machine unlearning techniques, which selectively remove non-contributive or misleading data, will further enhance generalization and robustness in heterogeneous field conditions.

As ML and DL models advance toward practical agricultural deployment, interpretability becomes essential. Explainable AI (XAI) techniques, including saliency maps, Layer-wise Relevance Propagation (LRP), Grad-CAM, and SHAP, will provide transparency into model decisions, fostering trust among domain experts and supporting biological validation. The development of standardized benchmarks and metrics for explainability will further ensure reliability and encourage adoption in real-world scenarios.

In conclusion, the integration of advanced learning paradigms with high-throughput imaging, multimodal sensing, and spatio-temporal modeling is poised to transform abiotic stress phenotyping. These innovations will enhance precision, efficiency, and interpretability, while accelerating the development of climate-resilient crop varieties and establishing a foundation for robust, scalable, and deployable AI systems in sustainable agriculture.

List of Publications and Pre-Prints

Publications

- Patra, A. K., & Sahoo, L. (2024). Explainable lightweight deep learning pipeline for improved drought stress identification. *Frontiers in Plant Science*, 15, 1476130.
- Patra, A. K., Varshney, A., & Sahoo, L. (2025). An explainable Vision Transformer with transfer learning based efficient drought stress identification. *Plant Molecular Biology*, 115(4), 98.

Pre-Prints

- Patra AK, Sahoo L (2025) "MRD-LiNet: A Novel Lightweight Hybrid CNN with Gradient-Guided Unlearning for Improved Drought Stress Identification." arXiv preprint: `https://arxiv.org/abs/2509.06367`
- Patra AK, Sahoo L (2025) Improved Classification of Nitrogen Stress Severity in Plants Under Combined Stress Conditions Using Spatio-Temporal Deep Learning Framework. arXiv preprint: `https://arxiv.org/abs/2509.06625`

List of Conferences

- A.K.Patra, Priyam Kurmi, L Sahoo, Machine learning for sustainable mangement of drought in tea and other crops of NER, Japan-NER Sustainable Technologies Cooperation Symposium 2025 (JNSTCS 2025), IIT Guwahati.

Bibliography

[1] Rijad Sarić, Viet D. Nguyen, Timothy Burge, Oliver Berkowitz, Martin Trtílek, James Whelan, Mathew G. Lewsey, and Edhem Čustović. Applications of hyperspectral imaging in plant phenotyping. *Trends in Plant Science*, 27(3):301–315, March 2022.

[2] Kareem A. Mosa, Ahmed Ismail, and Mohamed Helmy. Introduction to Plant Stresses. In Kareem A. Mosa, Ahmed Ismail, and Mohamed Helmy, editors, *Plant Stress Tolerance: An Integrated Omics Approach*, SpringerBriefs in Systems Biology, pages 1–19. Springer International Publishing, Cham, 2017.

[3] Nobuhiro Suzuki, Rosa M. Rivero, Vladimir Shulaev, Eduardo Blumwald, and Ron Mittler. Abiotic and biotic stress combinations. *New Phytologist*, 203(1):32–43, 2014.

[4] Andy Pereira. Plant Abiotic Stress Challenges from the Changing Environment. *Frontiers in Plant Science*, 7, 2016.

[5] Ron Mittler. Abiotic stress, the field environment and stress combination. *Trends in Plant Science*, 11(1):15–19, January 2006.

[6] Shilpi Mahajan and Narendra Tuteja. Cold, salinity and drought stresses: An overview. *Archives of Biochemistry and Biophysics*, 444(2):139–158, December 2005.

[7] Raghav Khanna, Lukas Schmid, Achim Walter, Juan Nieto, Roland Siegwart, and Frank Liebisch. A spatio temporal spectral framework for plant stress phenotyping. *Plant Methods*, 15(1):13, February 2019.

[8] Qiaosheng Guo and Zaibiao Zhu. Phenotyping of Plants. In *Encyclopedia of Analytical Chemistry*, pages 1–15. John Wiley & Sons, Ltd, 2014.

[9] Zhenbo Li, Ruohao Guo, Meng Li, Yaru Chen, and Guangyao Li. A review of computer vision technologies for plant phenotyping. *Computers and Electronics in Agriculture*, 176:105672, September 2020.

[10] Nadia Al-Tamimi, Patrick Langan, Villo Bernad, Jason Walsh, Eleni Mangina, and Sonia Negrao. Capturing crop adaptation to abiotic stress using image-based technologies. *OPEN BIOLOGY*, 12(6), JUN 22 2022.

[11] Stijn Dhondt, Nathalie Wuyts, and Dirk Inze. Cell to whole-plant phenotyping: the best is yet to come. *Trends in Plant Science*, 18(8):428–439, August 2013.

[12] Jose Luis Araus and Jill E. Cairns. Field high-throughput phenotyping: the new crop breeding frontier. *Trends in Plant Science*, 19(1):52–61, January 2014.

[13] Daniel T. Smith, Andries B. Potgieter, and Scott C. Chapman. Scaling up high-throughput phenotyping for abiotic stress selection in the field. *THEORETICAL AND APPLIED GENETICS*, 134(6, SI):1845–1866, JUN 2021.

[14] Andrew W. W. Herr, Alper Adak, Matthew E. E. Carroll, Dinakaran Elango, Soumyashree Kar, Changying Li, Sarah E. E. Jones, Arron H. H. Carter, Seth C. C. Murray, Andrew Paterson, Sindhuja Sankaran, Arti Singh, and Asheesh K. K. Singh. Unoccupied aerial systems imagery for phenotyping in cotton, maize, soybean, and wheat breeding. *CROP SCIENCE*, 2023 JUN 22 2023.

[15] Sunny Arya, Karansher Singh Sandhu, Jagmohan Singh, and Sudhir kumar. Deep learning: as the new frontier in high-throughput plant phenotyping. *Euphytica*, 218(4):47, March 2022.

[16] Wanneng Yang, Hui Feng, Xuehai Zhang, Jian Zhang, John H. Doonan, William David Batchelor, Lizhong Xiong, and Jianbing Yan. Crop Phenomics and High-Throughput Phenotyping: Past Decades, Current Challenges, and Future Perspectives. *Molecular Plant*, 13(2):187–214, February 2020.

[17] Arti Singh, Baskar Ganapathysubramanian, Asheesh Kumar Singh, and Soumik Sarkar. Machine learning for high-throughput stress phenotyping in plants. *TRENDS IN PLANT SCIENCE*, 21(2):110–124, FEB 2016.

[18] Alanna V. Zubler and Jeong-Yeol Yoon. Proximal methods for plant stress detection using optical sensors and machine learning. *Biosensors*, 10(12):193, 2020.

[19] MÃ³nica Pineda, Matilde BarÃ³n, and MarÃa-Luisa PÃ©rez-Bueno. Thermal Imaging for Plant Stress Detection and Phenotyping. *Remote Sensing*, 13(1):68, January 2021.

[20] Reeve Legendre, Nicholas T. Basinger, and Marc W. van Iersel. Low-cost chlorophyll fluorescence imaging for stress detection. *Sensors (Basel, Switzerland)*, 21(6):2055, 2021.

[21] Shiva Azimi, Rohan Wadhawan, and Tapan K. Gandhi. Intelligent monitoring of stress induced by water deficiency in plants using deep learning. *IEEE Transactions on Instrumentation and Measurement*, 70:1–13, 2021.

[22] Fubing Liao, Xiangqian Feng, Ziqiu Li, Danying Wang, Chunmei Xu, Guang Chu, Hengyu Ma, Qing Yao, and Song Chen. A hybrid cnn-lstm model for diagnosing rice nutrient levels at the rice panicle initiation stage. *Journal of Integrative Agriculture*, 23(2):711–723, 2024.

[23] Yasamin Borhani, Javad Khoramdel, and Esmaeil Najafi. A deep learning based approach for automated plant disease classification using vision transformer. *Scientific Reports*, 12(1):11554, 2022.

[24] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, January 2022.

[25] Sambuddha Ghosal, David Blystone, Asheesh K. Singh, Baskar Ganapathysubramanian, Arti Singh, and Soumik Sarkar. An explainable deep machine vision framework for plant stress phenotyping. *Proceedings of the National Academy of Sciences*, 115(18):4613–4618, May 2018.

[26] Ana Barradas, Pedro M. P. Correia, Sara Silva, Pedro Mariano, Margarida Calejo Pires, Ana Rita Matos, Anabela Bernardes da Silva, and Jorge Marques da Silva. Comparing Machine Learning Methods for Classifying Plant Drought Stress from Leaf Reflectance Spectra in Arabidopsis thaliana. *Applied Sciences*, 11(14):6392, January 2021.

[27] M. Farooq, A. Wahid, N. Kobayashi, D. Fujita, and S. M. A. Basra. Plant Drought Stress: Effects, Mechanisms and Management. In Eric Lichtfouse, Mireille Navarrete, Philippe Debaeke, Souchere Veronique, and Caroline Alberola, editors, *Sustainable Agriculture*, pages 153–188. Springer Netherlands, Dordrecht, 2009.

[28] David E. Clarke, Elizabeth A. Stockdale, Jacqueline A. Hannam, Benjamin P. Marchant, and Stephen H. Hallett. Spatial-temporal variability in nitrogen use efficiency: Insights from a long-term experiment and crop simulation modeling to support site specific nitrogen management. *European Journal of Agronomy*, 158:127224, 2024.

[29] Compton J. Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8(2):127–150, May 1979.

[30] Dapeng Ye, Libin Wu, Xiaobin Li, Tolulope Opeyemi Atoba, Wenhao Wu, and

Haiyong Weng. A synthetic review of various dimensions of non-destructive plant stress phenotyping. *PLANTS-BASEL*, 12(8), APR 2023.

[31] Ting Wen, Jian-Hong Li, Qi Wang, Yang-Yang Gao, Ge-Fei Hao, and Bao-An Song. Thermal imaging: The digital eye facilitates high-throughput phenotyping traits of plant growth and stress responses. *Science of The Total Environment*, 899:165626, November 2023.

[32] Guijun Yang, Jiangang Liu, Chunjiang Zhao, Zhenhong Li, Yanbo Huang, Haiyang Yu, Bo Xu, Xiaodong Yang, Dongmei Zhu, Xiaoyan Zhang, Ruyang Zhang, Haikuan Feng, Xiaoqing Zhao, Zhenhai Li, Heli Li, and Hao Yang. Unmanned Aerial Vehicle Remote Sensing for Field-Based Crop Phenotyping: Current Status and Perspectives. *Frontiers in Plant Science*, 8, 2017.

[33] Puneet Mishra, Mohd Shahrimie Mohd Asaari, Ana Herrero-Langreo, Santosh Lohumi, BelÃ©n Diezma, and Paul Scheunders. Close range hyperspectral imaging of plants: A review. *Biosystems Engineering*, 164:49–67, December 2017.

[34] Jing Zhou, Huawei Mou, Jianfeng Zhou, Md Liakat Ali, Heng Ye, Pengyin Chen, and Henry T. Nguyen. Qualification of Soybean Responses to Flooding Stress Using UAV-Based Imagery and Deep Learning. *Plant Phenomics*, 2021, June 2021.

[35] Aadarsh Kumar Singh, Akhil Rao, Pratik Chattopadhyay, Rahul Maurya, and Lokesh Singh. Effective plant disease diagnosis using vision transformer trained with leafy-generative adversarial network-generated images. *Expert Systems with Applications*, 254:124387, 2024.

[36] Catherine Chan, Peter R. Nelson, Daniel J. Hayes, Yong-Jiang Zhang, and Bruce Hall. Predicting Water Stress in Wild Blueberry Fields Using Airborne Visible and Near Infrared Imaging Spectroscopy. *Remote Sensing*, 13(8):1425, January 2021.

[37] Koushik Nagasubramanian, Talukder Jubery, Fateme Fotouhi Ardakani, Seyed Vahid Mirnezami, Asheesh K Singh, Arti Singh, Soumik Sarkar, and Baskar Ganapathysubramanian. How useful is active learning for image-based plant phenotyping? *The Plant Phenome Journal*, 4(1):e20020, 2021.

[38] Paula Ramos-Giraldo, Chris Reberg-Horton, Anna M. Locke, Steven Mirsky, and Edgar Lobaton. Drought Stress Detection Using Low-Cost Computer Vision Systems and Machine Learning Techniques. *IT Professional*, 22(3):27–29, May 2020.

[39] Jiachen Yang, Xiaolan Guo, Yang Li, Francesco Marinello, Sezai Ercisli, and Zhuo Zhang. A survey of few-shot learning in smart agriculture: developments, applications, and challenges. *Plant Methods*, 18(1):28, 2022.

[40] Koushik Nagasubramanian, Asheesh Singh, Arti Singh, Soumik Sarkar, and Baskar Ganapathysubramanian. Plant phenotyping with limited annotation: Doing more with less. *The Plant Phenome Journal*, 5(1):e20051, 2022.

[41] Amal Harb, Arjun Krishnan, Madana M.R. Ambavaram, and Andy Pereira. Molecular and Physiological Analysis of Drought Stress in Arabidopsis Reveals Early Responses Leading to Acclimation in Plant Growth. *Plant Physiology*, 154(3):1254–1271, November 2010.

[42] Shuo Zhuang, Ping Wang, Boran Jiang, Maosong Li, and Zhihong Gong. Early detection of water stress in maize based on digital images. *Computers and Electronics in Agriculture*, 140:461–468, 2017.

[43] Jiangyong An, Wanyi Li, Maosong Li, Sanrong Cui, and Huanran Yue. Identification and Classification of Maize Drought Stress Using Deep Convolutional Neural Network. *Symmetry*, 11(2):256, February 2019.

[44] Kiara Brewer, Alistair Clulow, Mbulisi Sibanda, Shaeden Gokool, John Odindi, Onisimo Mutanga, Vivek Naiken, Vimbayi G. P. Chimonyo, and Tafadzwanashe Mabhaudhi. Estimation of maize foliar temperature and stomatal conductance as indicators of water stress based on optical and thermal imagery acquired using an unmanned aerial vehicle (uav) platform. *Drones*, 6(7), 2022. All Open Access, Gold Open Access, Green Open Access.

[45] Mohd Shahrimie Mohd Asaari, Stien Mertens, Lennart Verbraeken, Stijn Dhondt, Dirk InzÃ©, Koirala Bikram, and Paul Scheunders. Non-destructive analysis of plant physiological traits using hyperspectral imaging: A case study on drought stress. *Computers and Electronics in Agriculture*, 195:106806, April 2022.

[46] veronica Sobejano-Paz, Teis Norgaard Mikkelsen, Andreas Baum, Xingguo Mo, Suxia Liu, Christian Josef Koppl, Mark S. Johnson, Lorant Gulyas, and Monica Garcia. Hyperspectral and Thermal Sensing of Stomatal Conductance, Transpiration, and Photosynthesis for Soybean and Maize under Drought. *Remote Sensing*, 12(19):3182, January 2020.

[47] Adduru U. G. Sankararao, P. Rajalakshmi, and Sunita Choudhary. Machine Learning-Based Ensemble Band Selection for Early Water Stress Identification in Groundnut Canopy Using UAV-Based Hyperspectral Imaging. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5, 2023.

[48] Phuong D. Dao, Yuhong He, and Cameron Proctor. Plant drought impact detection using ultra-high spatial resolution hyperspectral images and machine learning. *International Journal of Applied Earth Observation and Geoinformation*, 102:102364,

October 2021.

[49] P. Schmitter, J. SteinrÃ¼cken, C. RÃ¶mer, A. Ballvora, J. LÃ©on, U. Rascher, and L. PlÃ¼mer. Unsupervised domain adaptation for early detection of drought stress in hyperspectral images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 131:65–76, September 2017.

[50] Shiva Azimi, Rohan Wadhawan, and Tapan K. Gandhi. Intelligent monitoring of stress induced by water deficiency in plants using deep learning. *IEEE Transactions on Instrumentation and Measurement*, 70:1–13, 2021.

[51] Narendra Singh Chandel, Subir Kumar Chakraborty, Yogesh Anand Rajwade, Kumkum Dubey, Mukesh K. Tiwari, and Dilip Jat. Identifying crop water stress using deep learning models. *Neural Computing and Applications*, 33(10):5353–5367, May 2021.

[52] Jiangsan Zhao, Peter Sykacek, Gernot Bodner, and Boris Rewald. Root traits of European Vicia faba cultivarsâ€"Using machine learning to explore adaptations to agroclimatic conditions. *Plant, Cell & Environment*, 41(9):1984–1996, 2018.

[53] Ankita Gupta, Lakhwinder Kaur, and Gurmeet Kaur. Drought stress detection technique for wheat crop using machine learning. *PeerJ Computer Science*, 9:e1268, May 2023.

[54] Jan Behmann, JÃ¶rg SteinrÃ¼cken, and Lutz PlÃ¼mer. Detection of early plant stress responses in hyperspectral images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:98–111, July 2014.

[55] Jing Zhou, Jianfeng Zhou, Heng Ye, Md Liakat Ali, Henry T. Nguyen, and Pengyin Chen. Classification of soybean leaf wilting due to drought stress using UAV-based imagery. *Computers and Electronics in Agriculture*, 175:105576, August 2020.

[56] Shuo Zhuang, Ping Wang, Boran Jiang, and Maosong Li. Learned features of leaf phenotype to monitor maize water status in the fields. *Computers and Electronics in Agriculture*, 172:105347, May 2020.

[57] Changye Yang, Sriram Baireddy, ValÃ©rian MÃ©line, Enyu Cai, Denise Caldwell, Anjali S. Iyer-Pascuzzi, and Edward J. Delp. Image-based plant wilting estimation. *Plant Methods*, 19(1):52, May 2023.

[58] S. M. Pathan, J.-D. Lee, D. A. Sleper, F. B. Fritschi, R. E. Sharp, T. E. Carter Jr., R. L. Nelson, C. A. King, W. T. Schapaugh, M. R. Ellersieck, H. T. Nguyen, and J. G. Shannon. Two Soybean Plant Introductions Display Slow Leaf Wilting and Reduced Yield Loss under Drought. *Journal of Agronomy and Crop Science*,

200(3):231–236, 2014.

[59] Lirong Xiang, Trevor M. Nolan, Yin Bao, Mitch Elmore, Taylor Tuel, Jingyao Gai, Dylan Shah, Ping Wang, Nicole M. Huser, Ashley M. Hurd, Sean A. McLaughlin, Stephen H. Howell, Justin W. Walley, Yanhai Yin, and Lie Tang. Robotic Assay for Drought (RoAD): an automated phenotyping system for brassinosteroid and drought responses. *The Plant Journal*, 107(6):1837–1853, 2021.

[60] Sizhou Chen, Jiazhi Shen, Kai Fan, Wenjun Qian, Honglian Gu, Yuchen Li, Jie Zhang, Xiao Han, Yu Wang, and Zhaotang Ding. Hyperspectral machine-learning model for screening tea germplasm resources with drought tolerance. *Frontiers in Plant Science*, 13, 2022.

[61] Sujata Butte, Aleksandar Vakanski, Kasia Duellman, Haotian Wang, and Amin Mirkouei. Potato crop stress identification in aerial images using deep learning-based object detection. *Agronomy Journal*, 113(5):3991–4002, 2021.

[62] Aswini Kumar Patra and Lingaraj Sahoo. Explainable light-weight deep learning pipeline for improved drought stress identification. `http://arxiv.org/abs/2404.10073`, 2024. version: 2.

[63] Pooja Goyal, Rakesh Sharda, Mukesh Saini, and Mukesh Siag. A deep learning approach for early detection of drought stress in maize using proximal scale digital images. *Neural Computing and Applications*, 36(4):1899–1913, 2024.

[64] Sayantan Sarkar, Alexandre-Brice Cazenave, Joseph Oakes, David McCall, Wade Thomason, Lynn Abbott, and Maria Balota. Aerial high-throughput phenotyping of peanut leaf area index and lateral growth. *Scientific Reports*, 11(1):21661, November 2021.

[65] Koushik Nagasubramanian, Asheesh K. Singh, Arti Singh, Soumik Sarkar, and Baskar Ganapathysubramanian. Usefulness of interpretability methods to explain deep learning based plant stress phenotyping, July 2020. arXiv:2007.05729 [cs].

[66] Michiel G. J. Kallenberg, Hiske Overweg, Ron van Bree, and Ioannis N. Athanasiadis. Nitrogen management with reinforcement learning and crop growth models. *Environmental Data Science*, 2:e34, 2023.

[67] Reshmi Sarkar, Brian K. Northup, Charles R. Long, and Vijay P. Singh. Machine learning the abiotic stressor impacts on nitrogen availability and photo energy use in dryland forage systems under different tillage and green manuring practices. 2(1):5, 2025.

[68] Yuan Wang, Peihua Shi, Yinfei Qian, Gui Chen, Jiang Xie, Xianjiao Guan, Weim-

ing Shi, and Haitao Xiang. Enhancing nitrogen nutrition index estimation in rice using multi-leaf SPAD values and machine learning approaches. *Frontiers in Plant Science*, 15, 2024.

[69] B. Balaji Naik, H. R. Naveen, G. Sreenivas, K. Karun Choudary, D. Devkumar, and J. Adinarayana. Identification of water and nitrogen stress indicative spectral bands using hyperspectral remote sensing in maize during post-monsoon season. *Journal of the Indian Society of Remote Sensing*, 48(12):1787–1795, 2020.

[70] Daniel González I Juclà, Elena Najdenovska, Fabien Dutoit, and Laura Elena Raileanu. Detecting stress caused by nitrogen deficit using deep learning techniques applied on plant electrophysiological data. *Scientific Reports*, 13(1):9633, 2023.

[71] Sumaira Ghazal, Namratha Kommineni, and Arslan Munir. Comparative analysis of machine learning techniques using RGB imaging for nitrogen stress detection in maize. *AI*, 5(3):1286–1300, 2024.

[72] Fubing Liao, Xiangqian Feng, Ziqiu Li, Danying Wang, Chunmei Xu, Guang Chu, Hengyu Ma, Qing Yao, and Song Chen. A hybrid CNN-LSTM model for diagnosing rice nutrient levels at the rice panicle initiation stage. *Journal of Integrative Agriculture*, 23(2):711–723, 2024.

[73] Hui You, Muchen Zhou, Junxiang Zhang, Wei Peng, and Cuimin Sun. Sugarcane nitrogen nutrition estimation with digital images and machine learning methods. *Scientific Reports*, 13(1):14939, 2023.

[74] Jorge Enrique Chaparro, José Edinson Aedo, and Felipe Lumbreras Ruiz. Machine learning for the estimation of foliar nitrogen content in pineapple crops using multispectral images and internet of things (IoT) platforms. *Journal of Agriculture and Food Research*, 18:101208.

[75] Trung-Tin Tran, Jae-Won Choi, Thien-Tu Huynh Le, and Jong-Wook Kim. A Comparative Study of Deep CNN in Forecasting and Classifying the Macronutrient Deficiencies on Development of Tomato Plant. *Applied Sciences*, 9(8):1601, January 2019.

[76] Shiva Azimi, Taranjit Kaur, and Tapan K. Gandhi. A deep learning approach to measure stress level in plants due to nitrogen deficiency. *Measurement*, 173:108650.

[77] Chaoxin Wang, Doina Caragea, Nisarga Kodadinne Narayana, Nathan T. Hein, Raju Bheemanahalli, Impa M. Somayanda, and S. V. Krishna Jagadish. Deep learning based high-throughput phenotyping of chalkiness in rice exposed to high

night temperature. *Plant Methods*, 18(1), 2022. All Open Access, Gold Open Access, Green Open Access.

[78] Naveen Puppala, Spurthi N. Nayak, Alvaro Sanz-Saez, Charles Chen, Mura Jyostna Devi, Nivedita Nivedita, Yin Bao, Guohao He, Sy M. Traore, David A. Wright, Manish K. Pandey, and Vinay Sharma. Sustaining yield and nutritional quality of peanuts in harsh environments: Physiological and molecular basis of drought and heat stress tolerance. *Frontiers in Genetics*, 14, 2023. All Open Access, Gold Open Access, Green Open Access.

[79] Peng Fu, Katherine Meacham-Hensold, Kaiyu Guan, and Carl J. Bernacchi. Hyperspectral Leaf Reflectance as Proxy for Photosynthetic Capacities: An Ensemble Approach Based on Multiple Machine Learning Algorithms. *Frontiers in Plant Science*, 10, 2019.

[80] Ge Gao, Mark A. Tester, and Magdalena M. Julkowska. The use of high-throughput phenotyping for assessment of heat stress-induced changes in arabidopsis. *PLANT PHENOMICS*, 2020, 2020.

[81] Tian Gao, Anil Kumar Nalini Chandran, Puneet Paul, Harkamal Walia, and Hongfeng Yu. Hyperseed: An end-to-end method to process hyperspectral images of seeds. *Sensors*, 21(24), 2021. All Open Access, Gold Open Access, Green Open Access.

[82] Chayanika Sharma, Anandita Dey, Hiramoni Khatun, Jyotshna Das, and Utpal Sarma. Design and development of a gas sensor array to detect salinity stress in khasi mandarin orange plants. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 72, 2023.

[83] Ali Moghimi, Ce Yang, and Peter M. Marchetto. Ensemble feature selection for plant phenotyping: A journey from hyperspectral to multispectral imaging. *IEEE Access*, 6:56870 – 56884, 2018. All Open Access, Gold Open Access.

[84] Parvin Mohammadi and Keyvan Asefpour Vakilian. Machine learning provides specific detection of salt and drought stresses in cucumber based on mirna characteristics. *Plant Methods*, 19(1):123, 2023.

[85] Yixin Deng, Nan Xin, Longgang Zhao, Hongtao Shi, Limiao Deng, Zhongzhi Han, and Guangxia Wu. Precision detection of salt stress in soybean seedlings based on deep learning and chlorophyll fluorescence imaging. *Plants*, 13(15):2089, 2024.

[86] Ya Tian, Limin Xie, Mingyang Wu, Biyun Yang, Captoline Ishimwe, Dapeng Ye, and Haiyong Weng. Multicolor fluorescence imaging for the early detection of salt

stress in arabidopsis. *Agronomy*, 11(12):2577, 2021.

[87] Ibrahim Kecoglu, Merve Sirkeci, Mehmet Burcin Unlu, Ayse Sen, Ugur Parlatan, and Feyza Guzelcimen. Quantification of salt stress in wheat leaves by raman spectroscopy and machine learning. *Scientific Reports*, 12(1):7197, 2022.

[88] Ali Moghimi, Ce Yang, and Peter M. Marchetto. Ensemble Feature Selection for Plant Phenotyping: A Journey From Hyperspectral to Multispectral Imaging. *IEEE Access*, 6:56870–56884, 2018.

[89] Mahjoubeh Akbari, Hossein Sabouri, Sayed Javad Sajadi, Saeed Yarahmadi, and Leila Ahangar. Classification and prediction of drought and salinity stress tolerance in barley using genphenml. *Scientific Reports*, 14(1):17420, 2024.

[90] Onur Okumuş, Ahmet Say, Barış Eren, Fatih Demirel, Satı Uzun, Mehmet Yaman, and Adnan Aydın. Using machine learning algorithms to investigate the impact of temperature treatment and salt stress on four forage peas (pisum sativum var. arvense l.). *Horticulturae*, 10(6):656, 2024.

[91] Emilio Vello, Megan Letourneau, John Aguirre, and Thomas E Bureau. Integrated web portal for non-destructive salt sensitivity detection of camelina sativa seeds using fluorescent and visible light images coupled with machine learning algorithms. *Frontiers in Plant Science*, 14:1303429, 2024.

[92] Hsiang Sing Naik, Jiaoping Zhang, Alec Lofquist, Teshale Assefa, Soumik Sarkar, David Ackerman, Arti Singh, Asheesh K. Singh, and Baskar Ganapathysubramanian. A real-time phenotyping framework using machine learning for plant stress severity rating in soybean. *Plant Methods*, 13(1):23, 2017.

[93] Austin A. Dobbels and Aaron J. Lorenz. Soybean iron deficiency chlorosis high-throughput phenotyping using an unmanned aircraft system. *Plant Methods*, 15(1):97, August 2019.

[94] Kira M. Veley, Jeffrey C. Berry, Sarah J. Fentress, Daniel P. Schachtman, Ivan Baxter, and Rebecca Bart. High-throughput profiling and analysis of plant responses over time to abiotic stress. *Plant Direct*, 1(4):e00023, 2017.

[95] Shiva Azimi, Taranjit Kaur, and Tapan K. Gandhi. A deep learning approach to measure stress level in plants due to nitrogen deficiency. *MEASUREMENT*, 173, MAR 2021.

[96] Hazem M. Kalaji, Wojciech Baba, Krzysztof Gediga, Vasilij Goltsev, Izabela A. Samborska, Magdalena D. Cetner, Stella Dimitrova, Urszula Piszcz, Krzysztof Bielecki, Kamila Karmowska, Kolyo Dankov, and Agnieszka Kompala-Baba. Chloro-

phyll fluorescence as a tool for nutrient status identification in rapeseed plants. *PHOTOSYNTHESIS RESEARCH*, 136(3):329–343, JUN 2018.

[97] Cheng Ju, AurÃ©lien Bibaut, and Mark van der Laan. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 45(15):2800–2818, November 2018.

[98] Alwaseela Abdalla, Haiyan Cen, Liang Wan, Khalid Mehmood, and Yong He. Nutrient status diagnosis of infield oilseed rape via deep learning-enabled dynamic model. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 17(6):4379–4389, JUN 2021.

[99] Manamele Dannies Mashabela, Priscilla Masamba, and Abidemi Paul Kappo. Applications of metabolomics for the elucidation of abiotic stress tolerance in plants: A special focus on osmotic stress and heavy metal toxicity. *Plants*, 12(2), 2023.

[100] Dagmar Hosiner, Susanne Gerber, Hella Lichtenberg-FratÃ©, Walter Glaser, Christoph SchÃ¼ller, and Edda Klipp. Impact of Acute Metal Stress in Saccharomyces cerevisiae. *PLoS ONE*, 9(1):e83330, January 2014.

[101] Riyazuddin Riyazuddin, Nisha Nisha, Bushra Ejaz, M. Iqbal R. Khan, Manu Kumar, Pramod W. Ramteke, and Ravi Gupta. A Comprehensive Review on the Heavy Metal Toxicity and Sequestration in Plants. *Biomolecules*, 12(1):43, December 2021.

[102] Uchenna Okereafor, Mamookho Makhatha, Lukhanyo Mekuto, Nkemdinma Uche-Okereafor, Tendani Sebola, and Vuyo Mavumengwana. Toxic Metal Implications on Agricultural Soils, Plants, Animals, Aquatic life and Human Health. *International Journal of Environmental Research and Public Health*, 17(7):2204, March 2020.

[103] Jianhong Zhang, Min Wang, Keming Yang, Yanru Li, Yaxing Li, Bing Wu, and Qianqian Han. The new hyperspectral analysis method for distinguishing the types of heavy metal copper and lead pollution elements. *INTERNATIONAL JOURNAL OF ENVIRONMENTAL RESEARCH AND PUBLIC HEALTH*, 19(13), JUL 2022.

[104] Wei Wang, Wenwen Kong, Tingting Shen, Zun Man, Wenjing Zhu, Yong He, Fei Liu, and Yufei Liu. Application of laser-induced breakdown spectroscopy in detection of cadmium content in rice stems. *FRONTIERS IN PLANT SCIENCE*, 11, DEC 18 2020.

[105] Alireza Sanaeifar, Wenkai Zhang, Haitian Chen, Dongyi Zhang, Xiaoli Li, and Yong He. Study on effects of airborne pb pollution on quality indicators and accumulation in tea plants using vis-nir spectroscopy coupled with radial basis function neural

network. *Ecotoxicology and Environmental Safety*, 229, 2022.

[106] Wei Wang, Zun Man, Xiaolong Li, Rongqin Chen, Zhengkai You, Tiantian Pan, Xiaorong Dai, Hang Xiao, and Fei Liu. Response mechanism and rapid detection of phenotypic information in rice root under heavy metal stress. *Journal of Hazardous Materials*, 449, 2023.

[107] Junmeng Li, Zihan Yang, Yanru Zhao, and Keqiang Yu. Hsi combined with cnn model detection of heavy metal cu stress levels in apple rootstocks. *Microchemical Journal*, 194, 2023.

[108] Tingting Shen, Chu Zhang, Fei Liu, Wei Wang, Yi Lu, Rongqin Chen, and Yong He. High-throughput screening of free proline content in rice leaf under cadmium stress using hyperspectral imaging with chemometrics. *Sensors (Switzerland)*, 20(11):1 – 15, 2020.

[109] Keqiang Yu, Shiyan Fang, and Yanru Zhao. Heavy metal hg stress detection in tobacco plant using hyperspectral sensing and data-driven machine learning methods. *Spectrochimica Acta - Part A: Molecular and Biomolecular Spectroscopy*, 245, 2021.

[110] Xin Zhou, Chunjiang Zhao, Jun Sun, Kunshan Yao, and Min Xu. Detection of lead content in oilseed rape leaves and roots based on deep transfer learning and hyperspectral imaging technology. *Spectrochimica Acta - Part A: Molecular and Biomolecular Spectroscopy*, 290, 2023.

[111] Zihan Yang, Junmeng Li, Lingming Zuo, Yanru Zhao, and Keqiang Yu. Collaborative estimation of heavy metal stress in wheat seedlings based on libs-raman spectroscopy coupled with machine learning. *JOURNAL OF ANALYTICAL ATOMIC SPECTROMETRY*, 2023 SEP 5 2023.

[112] Nathan T Hein, Ignacio A Ciampitti, and S. V. Krishna Jagadish. Bottlenecks and opportunities in field-based high-throughput phenotyping for heat and drought stress. *Journal of Experimental Botany*, 72(14):5102 – 5116, 2021. All Open Access, Green Open Access, Hybrid Gold Open Access.

[113] Monica F. Danilevicz, Philipp E. Bayer, Benjamin J. Nestor, Mohammed Bennamoun, and David Edwards. Resources for image-based high-throughput phenotyping in crops and data sharing challenges. *Plant Physiology*, 187(2):699–715, October 2021.

[114] Sagi Levanon, Oshry Markovich, Itamar Gozlan, Ortal Bakhshian, Alon Zvirin, Yaron Honen, and Ron Kimmel. Abiotic Stress Prediction from RGB-T Images

of Banana Plantlets. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, Lecture Notes in Computer Science, pages 279–295, Cham, 2020. Springer International Publishing.

[115] Yukimasa Kaneda, Shun Shibata, and Hiroshi Mineno. Multi-modal sliding window-based support vector regression for predicting plant water stress. *Knowledge-Based Systems*, 134:135–148, October 2017.

[116] Gernot Bodner, Mouhannad Alsalem, Alireza Nakhforoosh, Thomas Arnold, and Daniel Leitner. RGB and Spectral Root Imaging for Plant Phenotyping and Physiological Research: Experimental Setup and Imaging Protocols. *Journal of Visualized Experiments: JoVE*, 126:56251, August 2017.

[117] Robert D. Hall, John C. D'Auria, Antonio C. Silva Ferreira, Yves Gibon, Dariusz Kruszka, Puneet Mishra, and Rick van de Zedde. High-throughput plant phenotyping: a role for metabolomics? *TRENDS IN PLANT SCIENCE*, 27(6):549–563, JUN 2022.

[118] Sarah Taghavi Namin, Mohammad Esmaeilzadeh, Mohammad Najafi, Tim B. Brown, and Justin O. Borevitz. Deep phenotyping: deep learning for temporal phenotype/genotype classification. *Plant Methods*, 14(1):66, August 2018.

[119] Robson Luis Silva de Medeiros, Rinaldo Cesar de Paula, João Vitor Oliveira de Souza, and João Pedro Peixoto Fernandes. Abiotic stress on seed germination and plant growth of zeyheria tuberculosa. *Journal of Forestry Research*, 34(5):1511–1522, 2023.

[120] Ojasvini Ahluwalia, Poonam C. Singh, and Ranjana Bhatia. A review on drought stress in plants: Implications, mitigation and the role of plant growth promoting rhizobacteria. *Resources, Environment and Sustainability*, 5:100032, 2021.

[121] Taqdeer Gill, Simranveer K. Gill, Dinesh K. Saini, Yuvraj Chopra, Jason P. de Koff, and Karansher S. Sandhu. A Comprehensive Review of High Throughput Phenotyping and Machine Learning for Plant Stress Phenotyping. *Phenomics*, 2(3):156–183, June 2022.

[122] Asheesh Kumar Singh, Baskar Ganapathysubramanian, Soumik Sarkar, and Arti Singh. Deep Learning for Plant Stress Phenotyping: Trends and Future Perspectives. *Trends in Plant Science*, 23(10):883–898, October 2018.

[123] Yu Jiang and Changying Li. Convolutional Neural Networks for Image-Based High-Throughput Plant Phenotyping: A Review. *Plant Phenomics*, 2020, April 2020.

[124] A dataset of multispectral potato plants images, university of idaho. `https://www.`

idahofallshighered.org/vakanski/Multispectral_Images_Dataset.html, 2021.

[125] Tzutalin. Labelimg. https://github.com/tzutalin/labelImg, 2019.

[126] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[127] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. https://arxiv.org/abs/2010.11929v2, 2021. version: 2.

[128] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. TransUNet: Transformers make strong encoders for medical image segmentation. http://arxiv.org/abs/2102.04306, 2021.

[129] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. http://arxiv.org/abs/2103.14030, 2021.

[130] Auvick Chandra Bhowmik, Md. Taimur Ahad, Yousuf Rayhan Emon, Faruk Ahmed, Bo Song, and Yan Li. A customised vision transformer for accurate detection and classification of java plum leaf disease. *Smart Agricultural Technology*, 8:100500, 2024.

[131] Poornima Singh Thakur, Shubhangi Chaturvedi, Pritee Khanna, Tanuja Sheorey, and Aparajita Ojha. Vision transformer meets convolutional neural network for plant disease classification. *Ecological Informatics*, 77:102245, 2023.

[132] Utpal Barman, Parismita Sarma, Mirzanur Rahman, Vaskar Deka, Swati Lahkar, Vaishali Sharma, and Manob Jyoti Saikia. ViT-SmartAgri: Vision transformer and smartphone-based plant disease detection for smart agriculture. *Agronomy*, 14(2):327, 2024.

[133] Sana Parez, Naqqash Dilshad, Norah Saleh Alghamdi, Turki M. Alanazi, and Jong Weon Lee. Visual intelligence in precision agriculture: Exploring plant disease detection via efficient vision transformers. *Sensors*, 23(15):6949, 2023.

[134] Sheng Yu, Li Xie, and Qilei Huang. Inception convolutional vision transformers for plant disease identification. *Internet of Things*, 21:100650.

[135] Pushkar Gole, Punam Bedi, Sudeep Marwaha, Md Ashraful Haque, and Chandan Kumar Deb. TrIncNet: a lightweight vision transformer network for identification of plant diseases. *Frontiers in Plant Science*, 14, 2023.

[136] Huy-Tan Thai, Kim-Hung Le, and Ngan Luu-Thuy Nguyen. FormerLeaf: An efficient vision transformer for cassava leaf disease detection. *Computers and Electronics in Agriculture*, 204:107518, 2023.

[137] S. Hemalatha and Jai Jaganath Babu Jayachandran. A multitask learning-based vision transformer for plant disease localization and classification. *International Journal of Computational Intelligence Systems*, 17(1):188, 2024.

[138] Guoqiang Li, Yuchao Wang, Qing Zhao, Peiyan Yuan, and Baofang Chang. PMVT: a lightweight vision transformer for plant disease identification on mobile devices. *Frontiers in Plant Science*, 14, 2023.

[139] Sasikala Vallabhajosyula, Venkatramaphanikumar Sistla, and Venkata Krishna Kishore Kolli. A novel hierarchical framework for plant leaf disease detection using residual vision transformer. *Heliyon*, 10(9), 2024.

[140] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):87–110, 2023.

[141] Ankur P. Parikh, Oscar Tackstrom, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. `https://arxiv.org/abs/1606.01933`, 2016.

[142] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. `http://arxiv.org/abs/1706.03762`, 2023.

[143] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[144] Ali Razzaq, Parwinder Kaur, Naheed Akhter, Shabir Hussain Wani, and Fozia Saleem. Next-generation breeding strategies for climate-ready crops. *FRONTIERS IN PLANT SCIENCE*, 12, JUL 21 2021.

[145] A. Kamilaris and F. X. Prenafeta-BoldÃº. A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science*, 156(3):312–322, April 2018.

[146] Aswini Kumar Patra and Lingaraj Sahoo. Explainable light-weight deep learning pipeline for improved drought stress identification. *Frontiers in Plant Science*, 15:1476130, 2024.

[147] Aswini Kumar Patra, Ankit Varshney, and Lingaraj Sahoo. An explainable vision transformer with transfer learning based efficient drought stress identification. *Plant Molecular Biology*, 115(4):98, 2025.

[148] Brett Koonce. Resnet 50. In *Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization*, pages 63–72. Springer, 2021.

[149] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

[150] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[151] Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(3):2150–2168, 2024.

[152] Chunxiao Li, Haipeng Jiang, Jiankang Chen, Yu Zhao, Shuxuan Fu, Fangming Jing, and Yu Guo. An overview of machine unlearning. *High-Confidence Computing*, 5(2):100254, 2025.

[153] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.

[154] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[155] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524, 2021.

[156] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9304–9312, 2020.

[157] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

[158] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987, 2023.

[159] Bertrand Hirel, Thierry Tetu, Peter J. Lea, and Frederic Dubois. Improving nitrogen use efficiency in crops for sustainable agriculture. *Sustainability*, 3(9):1452–1485, 2011.

[160] Prabha Singh, Krishan Kumar, Abhishek Kumar Jha, Pranjal Yadava, Madan Pal, Sujay Rakshit, and Ishwar Singh. Global gene expression profiling under nitrogen stress identifies key genes involved in nitrogen stress adaptation in maize (zea mays l.). *Scientific Reports*, 12(1):4211, 2022-03-10.

[161] Samrat Das, Dalveer Singh, Hari S Meena, Shailendra K Jha, Jyoti Kumari, Viswanathan Chinnusamy, and Lekshmy Sathee. Long term nitrogen deficiency alters expression of mirnas and alters nitrogen metabolism and root architecture in indian dwarf wheat (triticum sphaerococcum perc.) genotypes. *Scientific reports*, 13(1):5002, 2023.

[162] Shah Saud, Shah Fahad, Chen Yajun, Muhammad Z. Ihsan, Hafiz M. Hammad, Wajid Nasim, Amanullah, Muhammad Arif, and Hesham Alharby. Effects of nitrogen supply on water stress and recovery mechanisms in kentucky bluegrass plants. *Frontiers in Plant Science*, 8, 2017.

[163] Abraham Blum. Stress, strain, signaling, and adaptation - not just a matter of definition. *Journal of Experimental Botany*, 67(3):562–565, 2016.

[164] Prachi Pandey, Vadivelmurugan Irulappan, Muthukumar V Bagavathiannan, and Muthappa Senthil-Kumar. Impact of combined abiotic and biotic stresses on plant growth and avenues for crop improvement by exploiting physio-morphological traits. *Frontiers in plant science*, 8:537, 2017.

[165] Heidi Webber, Ehsan Eyshi Rezaei, Masahiro Ryo, and Frank Ewert. Framework to guide modeling single and multiple abiotic stresses in arable crops. *Agriculture, Ecosystems & Environment*, 340:108179, 2022.

[166] Ramamurthy Mahalingam. Consideration of combined stress: a crucial paradigm for improving multiple stress tolerance in plants. In *Combined stresses in plants: Physiological, molecular, and biochemical aspects*, pages 1–25. Springer, 2014.

[167] Jiating Li, Peng Fu, and Carl J Bernacchi. Enhancing plant resilience under combined stress: the role of reflectance spectroscopy. *Journal of Experimental Botany*, page eraf368, 2025.

[168] Maja Zagorščak, Lamis Abdelhakim, Natalia Yaneth Rodriguez-Granados, Jitka Široká, Arindam Ghatak, Carissa Bleker, Andrej Blejec, Jan Zrimec, Ondřej Novák, Aleš Pěnčík, et al. Integration of multi-omics data and deep phenotyping provides insights into responses to single and combined abiotic stress in potato. *Plant physiology*, 197(4):kiaf126, 2025.

[169] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[170] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.