IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 November 2024, accepted 15 November 2024, date of publication 2 December 2024, date of current version 18 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3509814

RESEARCH ARTICLE

Supervisory Configuration of Deep Learning Networks for Plant Stress Detection and Synthetic Dataset Generation

J. RENÁN VELÁZQUEZ-GONZÁLEZ[1], MADAIN PÉREZ-PATRICIO[1], J. A. DE JESÚS OSUNA-COUTIÑO[1], JORGE LUIS CAMAS-ANZUETO[1], ABIEL AGUILAR-GONZÁLEZ[2], N. A. MORALES-NAVARRO[1], AND CARLOS A. HERNÁNDEZ-GUTIÉRREZ[1]

[1]Department of Science, Tecnológico Nacional de México/Instituto Tecnológico de Tuxtla Gutiérrez, Tuxtla Gutiérrez, Chiapas 29050, Mexico
[2]Department of Computer Science, Instituto Nacional de Astrofísica, Óptica y Electrónica, San Andrés Cholula 72840, Mexico

Corresponding author: J. A. de Jesús Osuna-Coutiño (osuna_antonio_77@hotmail.com)

This work was supported by the Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT).

ABSTRACT In computer vision, plant stress detection involves the identification and classification of crop stresses. There are several approaches for the identification of green areas. The most recent approaches rely on machine-learning techniques or deep-learning networks to develop this task. Unfortunately, when attempting to use these networks to detect stressed plants, their performance drastically decreases. In most cases, these networks cannot detect plant stress. In addition, there are extensive repositories of plants on the internet. However, in most cases, these repositories do not include stressed plants. An alternative is to use networks to generate realistic synthetic images; nevertheless, these mathematical models frequently fail to produce accurate synthetic images (increasing supervision and collection times). Motivated by the latter, we propose a supervisory configuration of deep-learning networks to detect stressed plants and generate synthetic databases. This methodology consists of three phases. First, we collected a small set of Internet images of the stressed crops. Second, the process involves final layer training of the image generation model by introducing a new node into the network. Finally, we supervised the generative model using a classification neural network and a feedback loop. This supervision increased the quality of the generated synthetic images. Therefore, the experimental results were promising. The proposed configuration showed a 23.85% increase in average precision and a 10.8% increase in average recall compared with traditional classification architectures using the same synthetic dataset. These results demonstrated the feasibility of this configuration for the classification of stressed crops using synthetic datasets.

INDEX TERMS Plant stress detection, plant stress classification, deep learning, visual pattern.

I. INTRODUCTION

Plant stress detection involves the identification and classification of crop stress [1]. In agriculture, the identification and classification of stress in crops provides valuable information on the current state of the plant; therefore, accurate diagnosis of the type of stress in the crop is essential to provide appropriate treatment [2], [3]. In addition, plant stress identification can help farmers address plant health issues before they become severe [4]. This permits increased productivity and reduced costs [5]. Because of these characteristics, many tasks incorporate stressed crop detection as a fundamental tool for precision agriculture [6] and crop management [7].

There are several approaches for identifying plant stress. In most cases, these techniques use physiological and morphological monitoring of crops [8], [9]. This approach uses qualified experts to identify stress using visual information. In this case, experts infer a classification based on stress symptoms. Furthermore, this technique does not

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

require specialized equipment and can detect changes in plants in real-time. However, the disadvantages of using this approach include the subjectivity of the results, that is, the interpretation of visual characteristics may vary among observers [10]. In addition, this approach is susceptible to human error [11] and depends on skilled labor [12].

Some approaches use molecular and biochemical analyses [13], [14]. These techniques involve extracting a fraction or fluid from the crop to analyze the sample using biochemical techniques, enabling stress determination to achieve high performance. These methods can identify the presence of pathogens at the molecular level, and are employed when greater precision is required. However, they are not ideal in terms of cost-effectiveness and speed [15]. In addition, these methods are invasive, implying that chemical or mechanical intervention in crops is necessary during sample extraction for analysis [16]. These actions can affect plant development and growth through direct intervention by the experimenter [17].

Image processing is another approach to detect stressed crops [18], [19]. This approach uses an image to identify and detect visual characteristics that indicate stress symptoms. Unlike other approaches, computer vision allows the identification of crop stress without an expert agent or does not require direct contact. In addition, computer vision techniques have the main characteristic of being non-invasive. However, in classical works, the images need certain ideal factors in the illumination, sharpness, resolution, etc. These factors may result in low robustness [20]; that is, in most cases, these computer vision techniques operate in a controlled environment to be effective.

The deep learning approach does not require human intervention to extract visual features [21], [22]. This attribute is particularly relevant because it eliminates the laborious feature identification process. However, these methods generally encounter significant challenges because they require the collection of a large image dataset [23]. This step is crucial because the quality and diversity of the training data directly impact the model's generalization capabilities and accuracy [24]. The diversity is a noticeable limitation in the available datasets, particularly for stressed crops.

This study was motivated by recent results from computer vision studies using deep learning and the generation of synthetic datasets. Our study aimed to detect stressed plants using the proposed supervisory configuration. This approach integrates the generalization capability of deep learning with the generation of synthetic information, that is, the proposed method combines the strengths of both fields to address the classification challenge. Our methodology provides an approximation to automate the training process and the generation of synthetic datasets for Convolutional Neural Networks (CNN) for stressed plant detection. Also, this work facilitates the creation of datasets. Unlike previous studies, the proposed method allows us to determine the classification performance when applying our configuration. We used synthetic images to train the network because the state-of-the-art does not have sufficient datasets of stressed crop images, that is, the datasets available in the state-of-the-art are insufficient for training the proposed configuration.

Our method allows the automated creation of datasets for stressed plant detection. This detection can help growers identify problems in their crops early, allowing them to take quick action and reduce losses. Integrating these classification models into drones and real-time monitoring systems can provide automatic alerts, saving time and costs and increasing profitability. Using synthetic data also extends the scope of deep learning models, allowing them to adapt difficult or dangerous features to recreate with original data, saving time and resources. On the other hand, obtaining databases of stressed crops requires frequently damaging plants to capture images. This situation could be avoided with synthetic datasets.

In section II, we present related studies that determine the research location. The proposed method is described in section III. Section IV describes the experiments designed to evaluate the feasibility of the proposed method and the obtained results. Finally, conclusions and future work are presented in the last section.

II. RELATED WORK

This section provides a state-of-the-art overview of plant stress detection. We mainly focused on methodologies that have gained relevance in recent times, particularly those involving deep learning and synthetic image generation. The analysis covered deep learning applications and synthetic data to identify crop stress. In addition, we discuss the main advantages and challenges associated with these technologies in modern agriculture, as well as future trends in research and development in this field.

A. RGB IMAGE

Detection of plant stress using RGB imaging methods is an emerging area in precision agriculture and plant phenotyping [1]. These techniques focus on analyzing visual characteristics in plants to identify early signs of stress caused by various factors such as nutrient deficiencies, water scarcity, or disease. Convolutional Neural Networks (CNN) for classification are an alternative to detecting stressed crops by analyzing images [25], [26], [27]. These algorithms automatically extract features instead of using manual selection. In addition, these algorithms can be generalized from a training dataset and evaluated to new plant images, enabling better adaptation to different conditions and species. This is achievable because CNNs establish a relationship between the RGB image data and the classification task. However, these networks require training with a large dataset containing the expected features to generalize classification [28]. Unfortunately, in this field, datasets for crops under stress confront a scarcity of comprehensive and varied data or, in the worst-case scenario, are non-existent [29]. Therefore, networks for green area detection have low performance for stressed crop recognition, that is,

in most cases, these networks cannot detect stressed plants efficiently [30].

B. GENERATIVE

Variational Autoencoders (VAEs) comprise deep-learning neural networks for unsupervised learning [31]. These networks operate in two parts: encoder and decoder. The encoder transforms the input data into a distribution with a latent space generally smaller, whereas the decoder reconstructs the input data from this latent space. This network does not learn to map the input to a fixed point in the latent space, but to a probability distribution. This behavior allows them to generate new data similar to the training data, making them useful for tasks such as image generation and image quality improvement. However, VAE can suffer from post-collapse, where the encoder ignores the input data and generates a trivial latent space, leading to poor representation and reconstruction [32]. Furthermore, this architecture is sensitive to the choice of previous distribution, likelihood function, and regularization term, which can affect the balance between model fidelity and diversity [33]. In addition, the autoencoder process can lead to information loss, resulting in blurry, low-resolution images that lack sharpness and detail [34]. These particularities pose a significant constraint because training a classification network using these data may lead to inadequate performance.

Generative Adversarial Networks (GANs) enable the generation of synthetic plant images [35], [36], [37]. GANs are useful because datasets for stressed crops are currently scarce or non-existent. These networks operate through two competing neural networks: a generator that creates images that imitate real images and a discriminator that attempts to distinguish between authentic and synthetic images. This training process continuously improves the quality of the generated images, making GANs useful for expanding datasets and simulating stress effects in crops without the need for physical experimentation. Moreover, these networks can generate data for training crop-detection models [38]. However, the quality of the generated data may not always accurately reflect reality, i.e., the generated data deviate significantly from the real characteristics, and training with these data may be deficient [39]. In addition, these data often do not cover a variety of environments and can thus lead to overfitted models in a particular environment [40], [41]. This significant limitation restricts their application in certain scenarios.

Transformer networks have emerged as a promising solution for crop dataset generation in agriculture, particularly for stressed crops [42], [43]. These models can efficiently generate data sequences, allowing spatial pattern analysis in crop conditions with stress factors such as drought, diseases, and pests. In addition, the transformer architecture can train models that enable us to consider the environmental variety and stress conditions that may be scarce or difficult to capture in the natural environment. These benefits allow us to explore the impacts of different types of stress on crops. However, these networks have certain disadvantages. These models require training with massive datasets and feature variety in the images [44]. Although these networks manage to represent elements and environments precisely, not all the images they generate have all the learned characteristics because of the randomness of the generation; that is, occasionally, the networks produce an image lacking features typical of the real object [45]. These images are an important limitation because training a classification network using this information can result in poor performance.

C. CNN + TRANSFORMERS

Recent advancements in the application of transformer networks have significantly improved the performance of classification networks. The first model integrated a transformer network into the computer vision field [46]. In this hybrid architecture, a CNN extracts feature maps to serve as the input for the encoder of the transformer, and the final result is obtained through the transformer's decoder. Hybrid models allow greater precision in the classification and detection of stress in crops, benefiting from both the local precision of CNNs and the global analysis of transformers.

Currently, the dependence of CNNs on hybrid models is fundamental for image classification. The combination of CNNs and Transformers for stressed crop classification has significant advantages, such as robust visual feature extraction and the ability to model long-distance relationships due to the mechanisms of Transformers, resulting in improved crop stress identification [47], [48], [49]. In addition, their ability to scale with large datasets and their transferability to different crop types are favorable points. However, these advantages come with challenges, such as high computational complexity and extensive data requirements for training, which may limit their applicability in resource-limited settings. In the case of stressed crop identification, obtaining sufficiently large and labeled data sets can be a challenge [50]. The effectiveness of these hybrid models will depend on the availability of resources and the ability to handle their complexity in real agricultural applications. In addition, the classification accuracy of transformer-based models depends mainly on the parameters of the dataset, such as the size, balance, and correct labeling of the data. Finally, although models using transformers have been widely researched and applied to tasks such as image classification, their specific application to the detection of stressed crops is an emerging area that is gaining attention [51].

We have previously studied plant stress detection using an optical approach [52] or 3D information [54]. First, we proposed a new methodology for estimating the chlorophyll content in plant leaves using reflectance and transmittance as base parameters. For this purpose, we propose a novel optical arrangement for extracting the base parameters. In addition, we estimated the chlorophyll content using a learning algorithm, where the inputs were the reflectance and

transmittance. This approach provides significant advances in the processing of the reflectance information and sheet transmittance. However, in this proposal, it is necessary to dismember the leaves to analyze the crop. In the second approach [54], we propose a methodology to infer crop stress using deep learning and a 3D reconstruction sensor. Although depth sensors simplify 3D extraction, they restrict their applicability to indoor environments and increase implementation costs. For example, these sensors are prone to failure in outdoor scenarios because of solar radiation. In addition, they are not integrated into personal devices (such as mobile phones, personal assistants, or personal computers). Finally, their power consumption (in watts), cost, and size were higher than RGB sensors.

D. RELATED WORKS DISCUSSION

Although numerous advances and techniques allow us to augment databases, none of these studies directly address the problem of training neural networks and the generation of data with stressed crops. So, our research focuses on a new supervisory configuration of deep learning networks that, by generating and evaluating synthetic images, makes it possible to train a classifier network that allows us to identify real-world environments.

Our supervisory configuration allows for stressed plant detection and the generation of synthetic datasets. This approach integrates the generalization capability of deep learning with the generation of synthetic information; that is, the proposed methodology combines the abstraction power of deep learning with the self-generation of synthetic images to address the classification challenge. Unlike previous studies, the proposed approach enabled us to determine the classification performance when applying our configuration. In addition, this methodology facilitates the generation of datasets and training; that is, it is an approximation to automate the training process and data generation for stressed plant classification. In summary, although previous work has been successful in detecting stressed crops, our research introduces a new supervisory configuration of deep learning networks, which allows us to train with synthetic images to identify real-world environments.

III. THE PROPOSED METHOD

This section presents the supervisory configuration of deep learning networks for stressed plant detection and the generation of synthetic datasets. We define a configuration as a set of networks, steps, and parameters that interact to achieve a common objective and achieve results that they cannot achieve individually. Our approach integrates the generalization capability of deep learning with the generation of synthetic information, that is, our proposal combines the strengths of both fields to address the classification challenge. This methodology consists of three phases. First, we collected a small set of Internet images of stressed crops (**Section** III-A). Second, the process involves final layer training of the image generation model by introducing a new node into the network (**Section** III-B). Finally, we supervise the generative model using a classification neural network and feedback loop (**Section** III-C). **Fig.** 1 shows the block diagram of the proposed configuration.

The main contribution of this configuration lies in the strategic combination of two neural networks: a generative network and a classifier network. This method allows the classification of stressed crops by training synthetic images. The generative network creates detailed images. However, it often generates inconsistencies that affect image quality. By employing a classifier network to monitor and verify the generated images, we can ensure that they are free from these errors. This process results in more accurate and reliable images, facilitating automation and improving quality without human intervention. This approach is appropriate in applications that require a high volume of quality images or where databases are scarce or difficult to obtain, such as stressed crops.

A. SAMPLE RECOLLECTION

We collected a small set of Internet images with and without stressed crops (fern and tree) using 15 to 25 images. This set allows us to create a new node in the generating network to develop synthetic images of a plant type (**Section** III-B). These images consider the morphological characteristics from different perspectives, and their spatial information varies. In the case of similar images, this information may cause the sample to be unrepresentative and lower the methodology efficiency; that is, synthetic images may not allow generalized classification. Similarly, a repetitive environment can bias the creation of synthetic images in the same environment. This repetitive information may lower the classification performance when attempting to recognize a plant in another environment. **Fig.** 2 shows some images used to create a new node.

B. IMAGE GENERATION

The previously collected dataset (**Section** III-A) was used to retrain the last layer of the image-generation model by creating a new node in the network. A node is a feature set that represents a particular object or living thing. A generation network has several nodes acquired during the training, which allows the generation of synthetic images by combining features from one or more nodes. In the proposed methodology, we added a crop node. This node enables the generation of synthetic images with morphological characteristics and spatial information. However, a generation network does not always generate the required conditions (crop characteristics and the spatial environment). Therefore, we propose a supervisory classification network (**Section** III-C).

Fig. 3 shows a representation of the image generation architecture. This section uses the Stable Diffusion architecture [56]. This generation network receives a textual input (prompt), first encodes the input text, and subsequently, a diffusion network gradually transforms random noise into an image using the encoded text features. Finally, the image

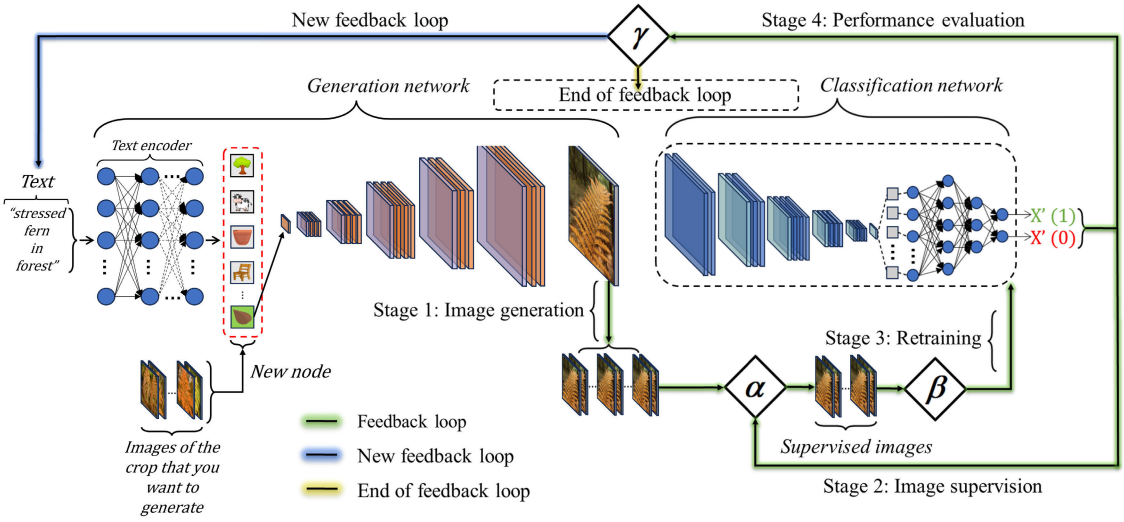


FIGURE 1. Block diagram of the proposed configuration.

FIGURE 2. Fern image samples for node generation.

decoder returns the output as an image reflecting the input textual description.

Our prompt requires the name of the newly added node ϕ, and we indicate the environment φ. The name of the new node ϕ must be a word that does not exist, for example, XtressedFern (stressed fern). This action allows the prompt not to be misinterpreted. An example of a prompt used is "a photo of XtressedFern in a garden". The generalized structure of our prompt is as follows: "a photo of $+\,\phi\,+$ in $+\,\varphi$". Using different environments in the prompt enhances the performance during training with synthetic images. **Fig.** 13-14 (b, d) show some synthetic images generated by our methodology. Furthermore, we present a series of images of stressed crops generated using different image-generation networks with the same prompt (see **Fig.** 11). When our supervision technique of synthetic crop images is not employed, the resulting images are often out of context, unstressed, or only partially stressed. Using these synthetic images to train networks to classify stressed crops may lead to deficient performance.

C. SUPERVISION

Generation networks offer an alternative for creating synthetic datasets. However, these models are often incorrect when generating crop images with morphological aspects (color and texture) or when they do not present the required environmental elements. Therefore, we propose a supervisory stage that allows us to monitor image generation. Initially, a generation network was used to develop images with specific characteristics. Subsequently, the classification network verifies that the image has the morphological characteristics of stressed and unstressed plants. Finally, we retrain the classification network by incorporating supervised synthetic images. This section utilizes the VGG-16 architecture [57] as the classification component because of its performance in experiments (see **Table** 1). **Fig.** 4 shows a representation of the classification network. In addition, we present a series of images with stressed crops discarded by our supervisory configuration (see **Fig.** 12). In most cases, these images showcase a different crop type, are out of context, unstressed, or only partially stressed. Discarding these images, we can create synthetic datasets that accurately represent the characteristics and morphology of stressed crops.

Initial Training: The classification network used a synthetic image collection from the generation network for training. Initially, we manually supervised a small dataset of 333 synthetic images to train the classification network. However, the number of images will increase automatically as the method progresses. Subsequently, the methodology initiates an iterative cycle. This cycle is called the feedback loop, which allows for retraining and the increase of training images.

Feedback Loop: Our methodology provides an approximation to automate the training process and the generation of synthetic datasets for Convolutional Neural Networks (CNN) for stressed plant detection. We obtained this automation

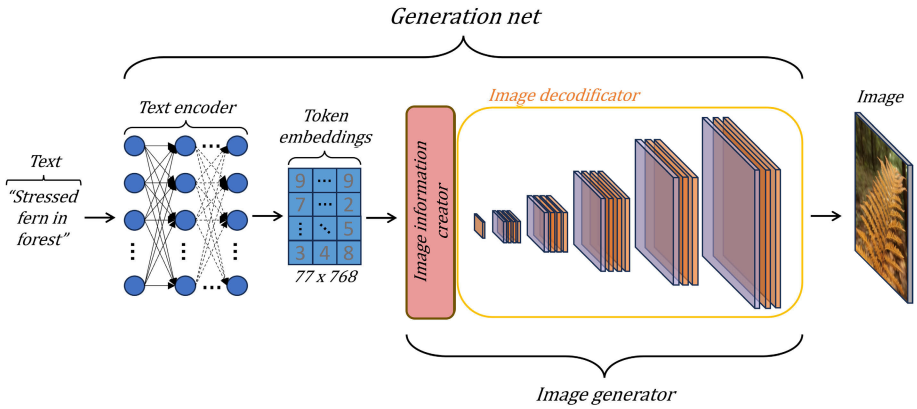


FIGURE 3. Image generation architecture.

by using feedback loops. The feedback loop consists of four phases. First, the methodology generates synthetic images using a generation network (**Section** III-B). Second, the classification network supervises the synthetic images produced by the generation network (**Section** III-C1). Subsequently, the network was retrained using supervised images (**Section** III-C2). Finally, we evaluate the performance of the classifier network using a real image dataset (**Section** III-C3).

1) IMAGE SUPERVISION (FEEDBACK LOOP)

The key concept behind this idea is the use of a classification network to monitor and evaluate the synthetic images generated by a generation network. The classification network supervises the new synthetic images produced by the generation network. These synthetic images approved supervision when their probability distribution exceeded a threshold. In our experiments, we used a membership threshold function $S(\alpha_i)$ of 90%; that is, the classification network classifies with more than 90% probability that the generated synthetic image belongs to the evaluated class; where i denotes the i-th generated synthetic image and w is an added iteration in the summation of β (**Eq.** 2). In every feedback loop, w is initialized at a value of 0. The membership threshold function $S(\alpha_i)$ is defined as follows:

$$S(\alpha_i) = \begin{cases} 1 & \text{if probability distribution} > 90\%, \\ 0 & \text{and } w+1 \qquad \text{otherwise}, \end{cases} \tag{1}$$

This monitoring ensures that only synthetic images that meet the established probability threshold contribute to the feedback and update process of the classification model. By using only images with a high probability of belonging to the evaluated class, we enhance the accuracy of the classifier and minimize the inclusion of incorrectly classified images.

2) RETRAINING (FEEDBACK LOOP)

Another factor to consider is the number of images that surpasses the membership threshold function $S(\alpha_i)$. If this set of images that surpasses the membership threshold function $S(\alpha_i)$ is equal to (β), a new retraining phase is triggered. These images constitute the new training set of the classification network. In addition, we considered the image limit to evaluate κ, where w is an additional iteration included in **Eq.** 1 and **Eq.** 2. We used an image limit of $\beta = 333$ images; that is, we paused the generation of images when 333 images surpassed the membership threshold function $S(\alpha_i)$. We found that a small number β increased the number of feedback loops. However, for a large number β, the last feedback loop will probably use an unnecessary image number during training. The number of evaluated images β is defined in **Eq.** 2. The image generation ends and retraining begins when $\beta = \kappa$.

$$\beta = \sum_{i=1}^{\kappa+w} S(\alpha_i) \tag{2}$$

3) PERFORMANCE EVALUATION (FEEDBACK LOOP)

After completing the retraining, we evaluated the performance of the classifier network (ϖ) with a real image dataset (Internet images). We use this process to determine whether the synthetic images have the required characteristics or if a new training cycle is required to expand the training set. We use the F1-Score as the evaluation metric in the performance threshold (γ). At the performance threshold γ, we defined a desired performance (ξ) of 90%. However, we can indicate the desired performance ξ (greater or lesser), for example, 80%, 95%, 98%, and so on. Unlike previous research, the proposed approach enables us to determine the classification performance. For the evaluation, we downloaded the Internet dataset from Google Images. Although we trained the classification network using synthetic images, we evaluated its performance using a real image dataset (ψ). The feedback loop stops when the performance of the classifier network ϖ surpasses the desired performance ξ. Otherwise, a new feedback loop is initiated. The performance threshold γ is

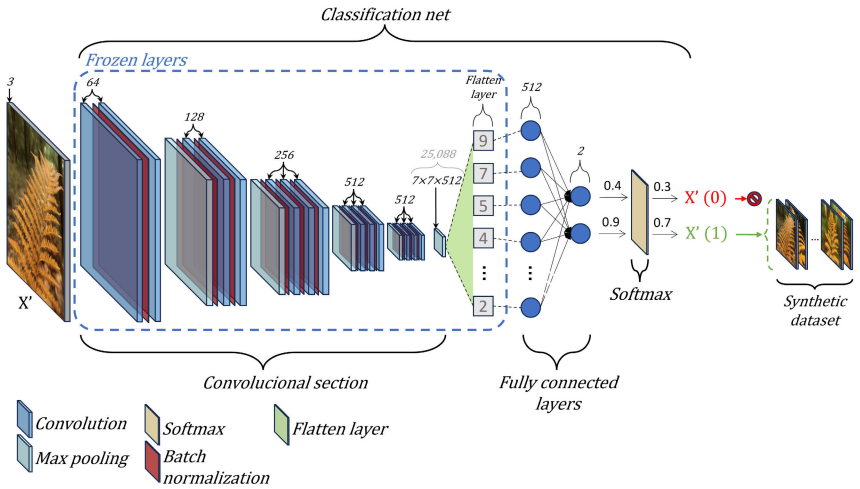


FIGURE 4. Image classification architecture.

defined as follows:

$$\gamma = \begin{cases} \text{End of feedback loop} & \text{if} \quad \varpi > \xi, \\ \text{New feedback loop} & \text{otherwise}, \end{cases} \tag{3}$$

In the evaluation dataset ψ, we downloaded all the images that can be downloaded from Google Images of the crop to be analyzed (fern, stressed fern, tree, and stressed tree). We purged the icons, images with a resolution lower than 250, repeated images, and images that did not belong to the crop. In addition, we augmented the evaluation images with data augmentation by mirroring, translating, and making brightness adjustments. The datasets utilized for training and evaluation are discussed in the **Subsection** IV-B.

D. CLASSIFICATION NETWORK

The input of the CNN is an RGB image X' with a size of 512 × 512 pixels. We use the VGG-16 architecture [57] as the classification component due to its performance in the experiments (see **Table** 1). The network consists of two stages. The first stage consists of 13 convolutional layers, 13 batch normalization layers, and five max-pooling layers, which extract feature maps with different resolutions from the input image. This stage is the backbone of the network because the extracted features are shared in the second stage. The second stage combines all the local features found in the previous convolutional layers. For this, we use a flatten layer and dense layers. Finally, we use a second dense layer to obtain one of the two possible labels (crop without stress $X'(0)$ and stressed crop $X'(1)$). **Fig.** 4 shows the architecture of the CNN for classification and supervision.

On the other hand, the architecture is structured as follows. The convolutional layers consist of several feature maps. Each feature map is connected to the preceding layer via a kernel, that is, a size-fixed weight matrix. In each iteration, this kernel performs a convolution operation on a group of neighboring neurons within the local area of the preceding layer. The kernel then slides with a fixed stride until this operation is performed on all neurons. After adding a bias to the convolution item, the output of the convolutional layer is activated by a nonlinear activation function, such as a Rectified Linear Unit (ReLU), sigmoid, tanh, and so on. The ReLU function was chosen as the activation function in the convolutional layer because of its ability to avoid the vanishing gradient and its fast convergence speed.

$$y_i = \text{ReLU}(y_{i-1} * W_{i,i-1} + b_i) \tag{4}$$

$$\text{ReLU}(x) = \begin{cases} x, & x > 0, \\ 0, & x \leq 0 \end{cases} \tag{5}$$

where y_{i-1} and y_i denote the outputs of two successive convolutional layers, $W_{i,i-1}$ denotes the connection weight matrix between them, $*$ represents the convolution operation, and b_i refers to the bias. Reducing the resolution of the convolutional layer can preserve a steady scale. Hence, the max pooling layer is introduced to perform the downsampling operation after the convolutional layer, as shown in **Fig.** 4. In the pooling layer, the downsampling operation aims to derive a unique statistic from a local region of the convolutional layer using a pooling strategy.

E. PROPOSED TRANSFER LEARNING TECHNIQUE

The process initializes the weights using classic transfer learning with the ImageNet dataset [58], freezing all convolutional network weights (**Fig.** 4). This process allowed us to use the generalized convolutional features of ImageNet [58] training for relevant feature extraction (classical transfer learning technique). We then added two fully connected layers with a dimension of 512 and two neurons (**Fig.** 4). These two small layers enable us to obtain a positive classification trend without a large training dataset. The proposed transfer learning technique begins using the weights obtained in the first feedback loop as initial weights in

the second feedback loop (ϖ'). At the end of the second feedback loop, we compare the performance of the classifier network using the weights obtained in the first feedback loop (ϖ') with the performance of the classifier network with the weights of the new training (ϖ). The technique updates the weights if the new performance (ϖ) surpasses the previous performance (ϖ'); otherwise, the system retains the previous weights. This process allowed us to use the relevant feature extraction of ImageNet [58] and to transfer the best weights for crop classification. We used the F1-Score as the evaluation metric (**Section** III-C3), and the system compares the performance using a real image dataset (ψ). In the proposed transfer learning technique, the weight-updating processes ϑ are defined as:

$$\vartheta = \begin{cases} \text{To update the weights} & \text{if } \varpi > \varpi', \\ \text{To retain the previous weights} & \text{otherwise,} \end{cases} \quad (6)$$

IV. DISCUSSION

We present evaluation metrics, approaches, and classification results. A quantitative evaluation was performed using recall, precision, accuracy, and F1-score measures. We compared nine different approaches for plant stress detection (Inception [59], DenseNet [60], ResNet50 [61], VGG-16 [57], Sup.-N Inception, Sup.-N DenseNet, Sup.-N ResNet50, Sup.-N VGG-16, and Supervisory-Net+TL). Finally, the results of the approaches are discussed using evaluation metrics.

A. HYPERPARAMETERS

We use a selected combination of hyperparameters to maximize the performance of different convolutional architectures (VGG16, ResNet, DenseNet, and Inception). The fully connected layers were configured with 512 and 2 neurons, using ReLu activation functions for the inner layers and Softmax for the output layer, ensuring proper modeling of classification tasks. The learning rate was set to 0.001 to ensure stable convergence, and a batch size of 32 was used for training. The binary cross entropy loss function was employed, along with the Adam optimizer, known for its efficiency in optimizing neural networks. The model was trained for 10 epochs, achieving a good balance between training time and final accuracy. Additionally, for the Stable Diffusion network used, the following hyperparameters were applied: 80 sampling steps, a guidance scale of 7.5, a seed of 42, a resolution of 512×512, a denoising strength of 0.75 for inpainting tasks, and a latent dimensionality of 256.

B. DATASETS

For this work, we use both real and synthetic datasets. Due to a shortage of specific crop datasets under stress conditions, Google was utilized to obtain images of crops experiencing various stress levels in different real-world environments. Also, data augmentation techniques were subsequently applied to diversify the dataset and improve the ability of the model to generalize. We collected in Google Images and data augmentation 9,480 fern images (5,034 healthy ferns and 4,446 stressed). Similarly, We collected 9,175 tree images (4,239 healthy trees and 4,936 stressed). We refer to this set as the "Google dataset". Both datasets include images of crops under stressed conditions, with features and environments that describe the conditions of interest of the study.

1) DATASET FOR TRAINING

The training process was conducted exclusively with synthetic images. We used 17,982 fern synthetic images and 54,612 tree synthetic images for the training. In the evaluation of training, we use 80% of the "Google dataset". We employed 7,584 images from the fern dataset and 7,338 images from the tree dataset for evaluation. These datasets were employed for the experiments presented in the **Tables**1-2.

2) DATASET FOR EVALUATION

The remaining 20% of the Google dataset was assigned for the performance evaluation phase (**Section** III-C3). For the fern dataset, this represents a total of 1,896 images. Similarly, the tree dataset used 1,834 images for this phase. By reserving 20% of the data for evaluation, the model is tested on unseen data during training, enabling a more accurate assessment of its performance and generalization ability to new instances. Also, a 5-fold cross-validation was applied, dividing the dataset into five parts and alternating the validation subset in each iteration. This cross-validation approach allows for a more robust evaluation and minimizes overfitting in the results.

C. EVALUATION

We analyze the statistical significance of the proposed methodology. For that, we use ANOVA for multiple comparison tests. In the **Table** 1, a † symbol expresses a significant difference between a classification approach and our work in the recall, precision, or F-score measures. A 5-fold cross-validation was applied, dividing the dataset into five parts and alternating the validation subset in each iteration. The **Tables** 1-2 show the cross-validation results. This cross-validation approach allows for a robust evaluation and minimizes overfitting in the results, enhancing the model's generalization.

D. METRICS

Quantitative evaluation was performed using eight measures recall (RE), precision (PR), specificity (SP), jaccard index (JI), accuracy (AC), F1-score (F1), Area Under the Curve (AUC) and Polygon Area Metric (PAM) based on the number of true positives, true negatives, false positives, and false negatives. The true positives (Tp) count the number of images whose classification was predicted correctly concerning the ground truth. To count the number of true negatives (Tn), we proceed as follows: Suppose that we are interested in the tree label, then all those images corresponding to other crops rather than a tree, according to the ground truth, should have

TABLE 1. Classification results for trees and ferns using different convolutional networks and metrics (Recall, Precision, and F-score). In this experiment, we use the same number of training synthetic images and the evaluation dataset. Also, we validate the differences between the networks with statistical significance.

	Trees (54,612 images)			Ferns (17,982 images)			Average		
	PR	RE/SE	F1	PR	RE/SE	F1	PR	RE/SE	F1
Inception [59]	0.7403	0.6840	0.7110	0.6798	0.7823	0.7274	0.7100†	0.7331†	0.7192†
DenseNet [60]	0.6148	0.7527	0.6765	0.6990	0.7190	0.7085	0.6569†	0.7359†	0.6925†
ResNet50 [61]	0.6709	0.7430	0.7050	0.6809	0.7253	0.7023	0.6759†	0.7342†	0.7037†
VGG-16 [57]	0.6889	0.7007	0.6944	0.7121	0.7651	0.7375	0.7005†	0.7329†	0.7159†
Supervisory-Net Inception	0.7853	0.6601	0.7170	0.7592	0.7766	0.7678	0.7723†	0.7183†	0.7424†
Supervisory-Net DenseNe	0.7949	0.7556	0.7713	0.7418	0.7729	0.7570	0.7683†	0.7643†	0.7641†
Supervisory-Net ResNet50	0.6175	**0.9220**	0.7377	0.7751	0.8159	0.7947	0.6951†	**0.8689**	0.7632†
Supervisory-Net VGG-16	0.7702	0.7730	0.7710	0.8195	**0.8721**	0.8450	0.7949†	0.8226†	0.8080†
Supervisory-Net+TL	**0.9458**	0.8655	**0.9037**	**0.9399**	0.8718	**0.9046**	**0.9429**	0.8686	**0.9041**

A † symbol indicates a significant difference between the proposed approach (Supervisory-Net+TL) and classification approaches.
Bold emphasis indicates the best value obtained in the experimental run.

TABLE 2. Classification results for trees and ferns using different convolutional networks and metrics. In this experiment, we use the same number of training synthetic images and the evaluation dataset.

	Trees (54,612 images)					Ferns (17,982 images)					
	SP	JI	AC	AUC	PAM	SP	JI	AC	AUC	PAM	Avg. PAM
Inception [59]	0.7882	0.5510	0.7400	0.7391	0.4956	0.7086	0.5717	0.7411	0.7421	0.5103	0.5030
DenseNet [60]	0.7343	0.5113	0.7408	0.7276	0.4883	0.7851	0.5490	0.7578	0.7501	0.5142	0.5013
ResNet50 [61]	0.7445	0.5442	0.7443	0.7376	0.5011	0.8154	0.5412	0.7836	0.7631	0.5330	0.5171
VGG-16 [57]	0.7342	0.5319	0.7185	0.7165	0.4697	0.7388	0.5841	0.7506	0.7499	0.5225	0.4961
Supervisory-Net Inception	0.8066	0.5581	0.7293	0.7361	0.4907	0.7856	0.6231	0.7814	0.7804	0.5686	0.5297
Supervisory-Net DenseNe	0.8304	0.6309	0.7896	0.7955	0.5869	0.7756	0.6091	0.7744	0.7726	0.5562	0.5716
Supervisory-Net ResNet50	0.6144	0.5850	0.7373	0.7691	0.5288	0.7905	0.6597	0.8024	0.8025	0.6065	0.5677
Supervisory-Net VGG-16	0.8019	0.6274	0.7871	0.7868	0.5775	0.8362	0.7316	0.8528	0.8521	0.6936	0.6356
Supervisory-Net+TL	**0.9538**	**0.8246**	**0.9106**	**0.91331**	**0.8024**	**0.9446**	**0.8258**	**0.9084**	**0.9098**	**0.7996**	**0.8010**

Bold emphasis indicates the best value obtained in the experimental run.

received any other predicted label except tree; if that is the case, each of these images are counted as true negatives. False positives (Fp) correspond to all images with incorrect labels. Finally, false negatives (Fn) correspond to those images that should have received a specific label, but the prediction did not assign it correspondingly, for instance, those images corresponding to a tree should have received a tree label. However, if any image did not receive such a label, then those are counted as false negatives (Fn).

We used recall (RE) to measure the proportion of images whose respective labels were predicted correctly regarding the image number in the ground truth labeled with such a label. In simple terms, it is the ground truth that was correctly predicted. Precision (PR) is the proportion of labels that were classified correctly, that is, considering our classification, the proportion classified correctly.

We employed specificity (SP) to measure the proportion of true negatives correctly identified in all negative cases. That is, the probability that the test is classified as negative when it is negative. The jaccard index (JI) is utilized to evaluate the similarity and diversity of the predicted labels compared to the ground truth labels. The jaccard index is calculated as the number of values belonging to both sets (intersection) divided by the unique number across both sets (union).

Accuracy (AC) is the proportion of correct predictions (Tp and Tn) divided by the number of examined cases. The F1-score (F1) helps summarize the performance of the predictions returned by the system. For a system with good performance, both recall and precision should tend to be one, meaning that most of the system's predictions tend to be correct and that such predictions tend to cover most of the ground truth.

$$\text{recall(RE)/sensitivity(SE)} = \frac{\text{Tp}}{\text{Tp} + \text{Fn}} \tag{7}$$

$$\text{precision(PR)} = \frac{\text{Tp}}{\text{Tp} + \text{Fp}} \tag{8}$$

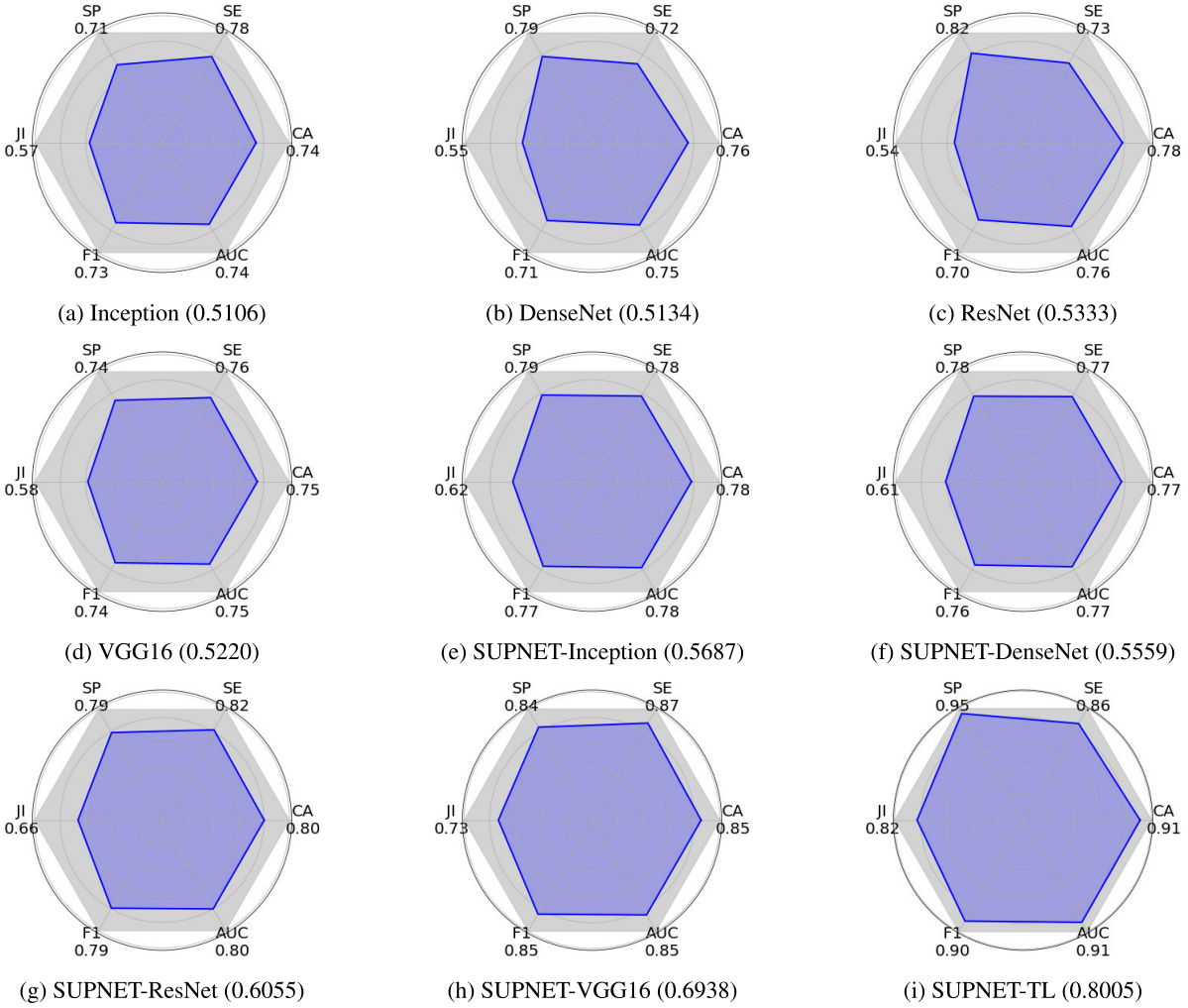
$$\text{specificity(SP)} = \frac{\text{Tn}}{\text{Tn} + \text{Fp}} \tag{9}$$

$$\text{jaccard index(JI)} = \frac{\text{Tp}}{\text{Tp} + \text{Fp} + \text{Fn}} \tag{10}$$

$$\text{accuracy(AC)} = \frac{\text{Tp} + \text{Tn}}{\text{Tp} + \text{Tn} + \text{Fp} + \text{Fn}} \tag{11}$$

$$\begin{aligned} \text{F1} &= 2\frac{\text{recall} * \text{precision}}{\text{recall} + \text{precision}} \\ &= \frac{\text{Tp}}{\text{Tp} + \frac{1}{2}(\text{Fp} + \text{Fn})} \end{aligned} \tag{12}$$

The Area Under the Curve (AUC) provides a measure of the model to discriminate between the classes. This metric represents the area under the receiver operating characteristic (ROC) curve, which plots the true positive rate against the false positive rate at various threshold settings. Where f(x) is a receiver operating characteristic curve in which the true-positive rate (SE) is plotted in the function of the false-positive rate (1-SP) for different cut-off points. The Polygon Area Metric (PAM) [62] is calculated by determining the area of the polygon formed by the points



(a) Inception (0.5106) (b) DenseNet (0.5134) (c) ResNet (0.5333)

(d) VGG16 (0.5220) (e) SUPNET-Inception (0.5687) (f) SUPNET-DenseNet (0.5559)

(g) SUPNET-ResNet (0.6055) (h) SUPNET-VGG16 (0.6938) (i) SUPNET-TL (0.8005)

FIGURE 5. Comparative fern evaluation of various neural network architectures using Polygon Area Metric (PAM). The metrics were calculated using the average confusion matrices from Fig. 8. The plots show six key performance indicators: specificity (SP), sensitivity (SE), jaccard index (JI), accuracy (AC), Area Under the Curve (AUC), and F1-score, with PAM representing the blue shaded area inside each polygon. The larger polygon area indicates better overall performance according to PAM.

representing RE/SE, SP, JI, AC, F1, and AUC within a regular hexagon. It is important to note that the regular hexagon consists of 6 sides, each with a length of 1, and the total area of the hexagon is 2.59807. The lengths from the center towards the hexagon vertex correspond to the values of RE/SE, SP, JI, AC, F1, and AUC, respectively, where PA represents the area of the formed polygon. It is important to mention that to normalize the PAM within the [0, 1] range, the PA value is divided by 2.59807.

$$\text{AUC} = \int_0^1 \text{f}(x)\,\text{dx} \tag{13}$$

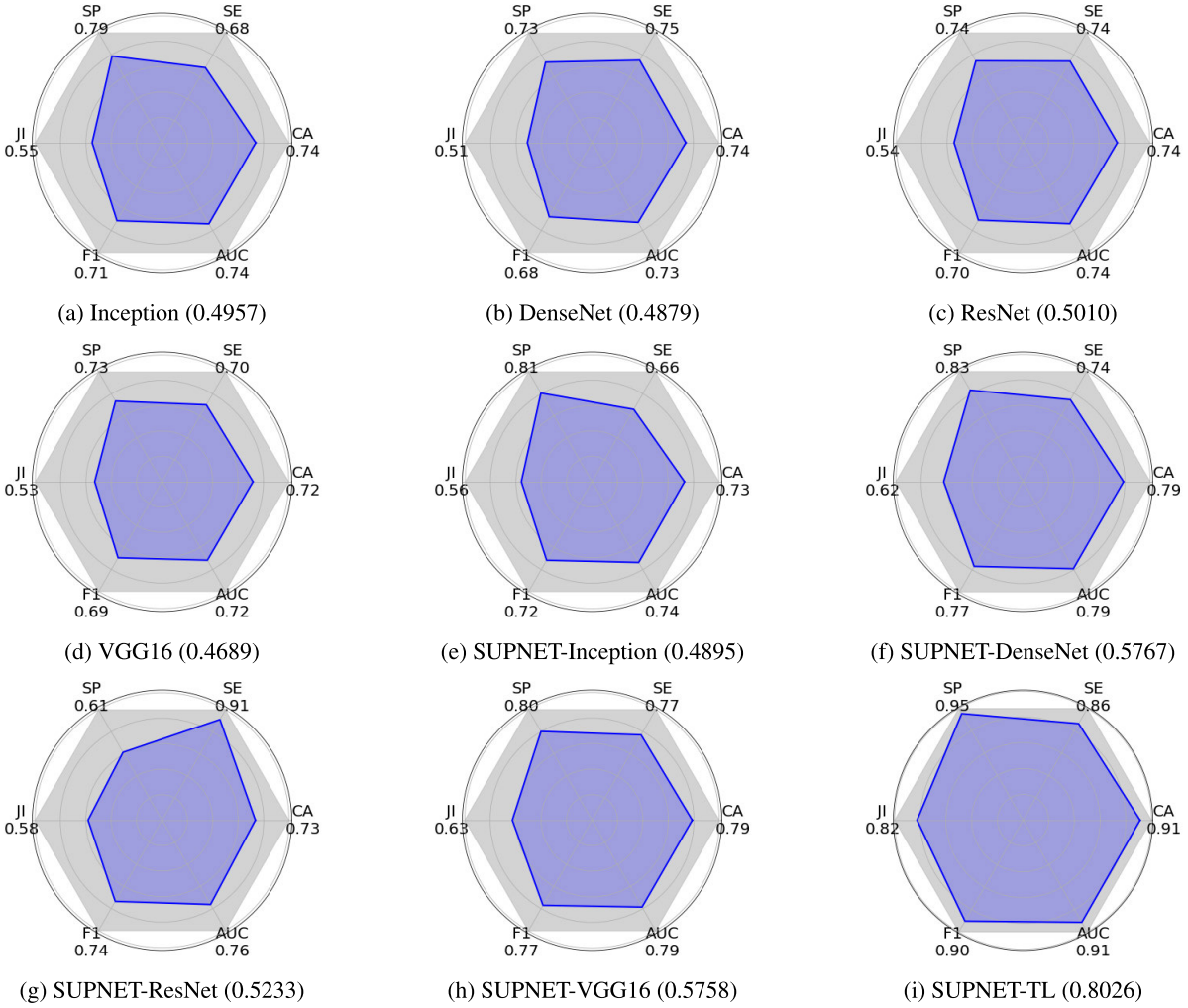
$$\text{PAM} = \frac{\text{PA}}{2.59807} \tag{14}$$

E. CLASSIFICATION RESULT

Experiments were performed using different crop types to evaluate the efficacy of the proposed methodology. For this purpose, we evaluated the tests using two categories of plants (ferns and trees). These crops have particular foliage and stem characteristics. **Tables** 1-2 show the quantitative results of the nine different approaches to plant stress detection. The first four are different convolutional network architectures for classification (Inception [59], DenseNet [60], ResNet50 [61], and VGG-16 [57]).

Supervisory-Net is our supervisory configuration that uses different architectures as classification components (Sup.-N Inception, Sup.-N DenseNet, Sup.-N ResNet50, and Sup.-N VGG-16). In addition, Supervisory-Net+TL uses our proposed learning transfer technique, and we use the VGG-16 architecture [57] as the classification component because of its performance in the experiments (see **Tables** 1-2). In this experiment, we considered the same training synthetic data and the same evaluation dataset for all approaches.

In **Tables** 1-2, the different convolutional networks (Inception [59], DenseNet [60], ResNet50 [61], and VGG-16 [57]) exhibit a lower performance than any other approach. However, the performance of these networks increases when



(a) Inception (0.4957) (b) DenseNet (0.4879) (c) ResNet (0.5010)

(d) VGG16 (0.4689) (e) SUPNET-Inception (0.4895) (f) SUPNET-DenseNet (0.5767)

(g) SUPNET-ResNet (0.5233) (h) SUPNET-VGG16 (0.5758) (i) SUPNET-TL (0.8026)

FIGURE 6. Comparative tree evaluation of various neural network architectures using Polygon Area Metric (PAM). The metrics were calculated using the average confusion matrices from Fig. 7. The plots show six key performance indicators: specificity (SP), sensitivity (SE), jaccard index (JI), accuracy (AC), Area Under the Curve (AUC), and F1-score, with PAM representing the blue shaded area inside each polygon. The larger polygon area indicates better overall performance according to PAM.

TABLE 3. Supervisory-Net+TL results for fern and stressed fern classification. This approach requires 15,984 synthetic training images to surpass 90% F1-Score.

	Training images					
	2,997	5,994	8,991	11,988	14,985	17,982
Precision	0.7529	0.7407	0.7795	0.8395	0.8843	0.9391
Recall	0.8073	0.7876	0.8288	0.8179	0.8578	0.8661
Accuracy	0.7923	0.7787	0.8311	0.8406	0.8635	0.9072
F1-Score	0.7792	0.7634	0.8034	0.8285	0.8708	0.9011

TABLE 4. Supervisory-Net VGG-16 results for fern and stressed fern classification. This approach requires 31,302 synthetic training images to surpass 90% F1-Score.

	Training images					
	2,997	5,994	8,991	17,982	23,976	32,967
Precision	0.7467	0.7462	0.7547	0.8151	0.8458	0.8399
Recall	0.7927	0.7686	0.8102	0.8735	0.9292	0.9698
Accuracy	0.7570	0.7745	0.7845	0.8565	0.8851	0.9024
F1-Score	0.7690	0.7572	0.7815	0.8433	0.8689	0.9002

TABLE 5. Supervisory-Net+TL results for tree and stressed tree classification. This approach requires 54,279 synthetic training images to surpass 90% F1-Score.

	Training images					
	20,313	26,973	33,633	40,293	46,953	54,612
Precision	0.8261	0.8330	0.8555	0.8952	0.9232	0.9465
Recall	0.7875	0.7836	0.8347	0.8425	0.8250	0.8593
Accuracy	0.8301	0.8379	0.8501	0.8596	0.8768	0.9076
F1-Score	0.8063	0.8075	0.8450	0.8681	0.8713	0.9008

TABLE 6. Supervisory-Net VGG-16 results for tree and stressed tree classification. This approach requires 86,580 synthetic training images to surpass 90% F1-Score.

	Training images					
	20,313	26,973	33,633	54,612	73,593	86,913
Precision	0.6142	0.6471	0.6853	0.7705	0.8679	0.9247
Recall	0.6380	0.6769	0.7241	0.7517	0.8143	0.8768
Accuracy	0.6781	0.6794	0.7200	0.7776	0.8462	0.9110
F1-Score	0.6259	0.6617	0.7042	0.7610	0.8402	0.9001

these networks use the proposed configuration. For example, there is a considerable increase compared to the proposed configuration with the VGG-16 (Sup.-N VGG-16) versus the VGG-16 [57] architecture. This configuration allowed an increase of 5.2% in precision and 5.27% in recall (**Table** 1).

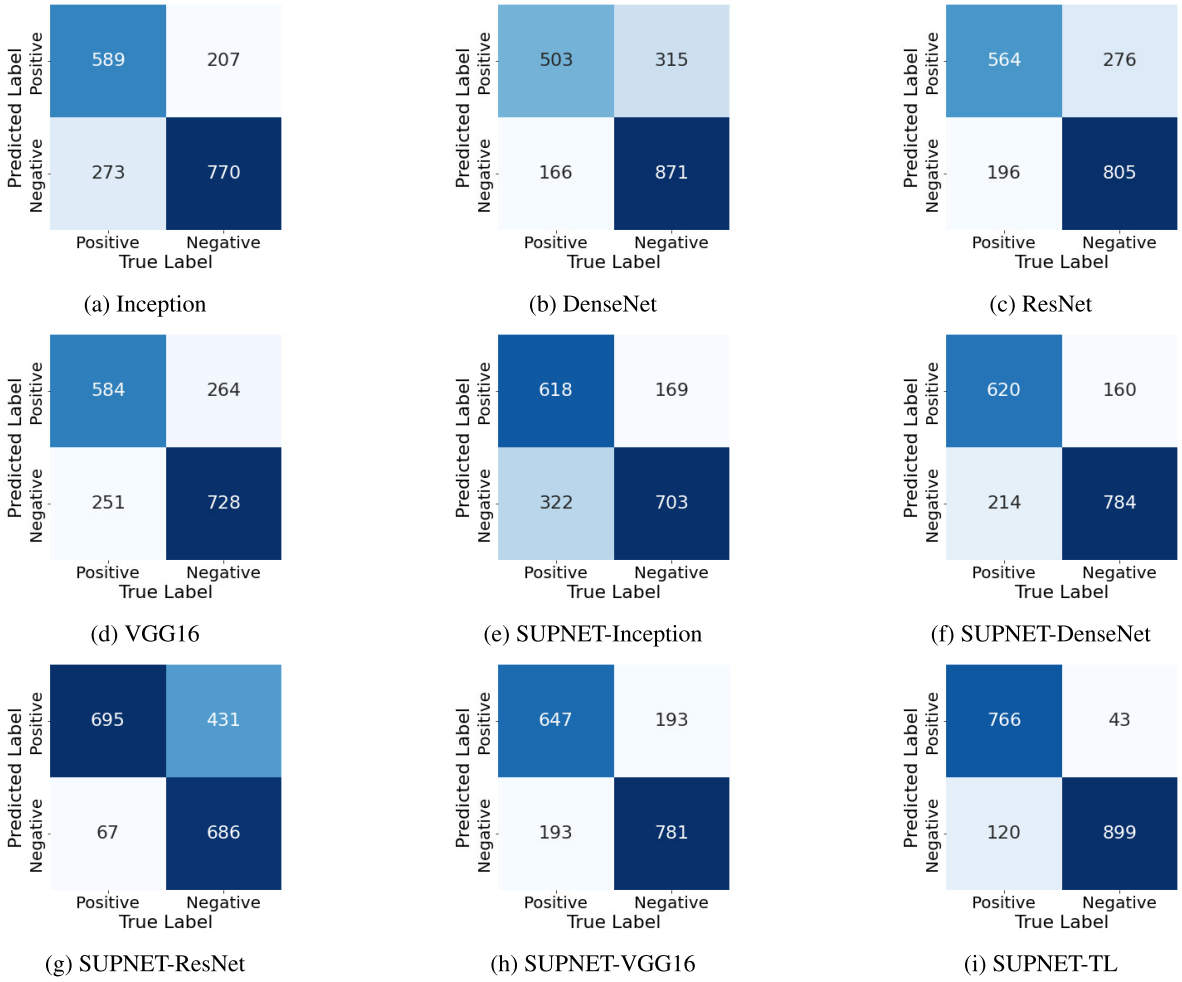


FIGURE 7. Average confusion matrices for tree classification obtained from 5-fold cross-validation using different convolutional neural network architectures.

On the other hand, comparing our proposal (Sup.-Net+TL) with Sup.-N VGG, the performance increased by 18.65% in the average precision and 5.53% in the average recall; that is, we correctly recognized 18.65% more in the classification and 5.53% more in the ground truth.

In all comparisons, the proposed configuration required fewer training images than conventional classification architectures to achieve the desired performance. This improvement is possible because we used the generalized convolutional features of ImageNet [58] with the classical transfer learning technique and the proposed transfer technique allows us to obtain a positive classification trend with few training images. In addition, we analyzed the statistical significance of the proposed classification approach. For this purpose, we used ANOVA for multiple comparison tests. A statistically significant difference tells you whether approaches (Incep [59], Dense [60], ResNet [61], VGG [57], Sup.-N Incep, Sup.-N Dense, Sup.-N ResNet, Sup.-N VGG) are substantially different from the proposal (Supervisory-Net+TL). In **Table** 1, the † symbol indicates a significant difference between the classification approach and the proposed method in the F-score 0.

Table 2 presents additional classification metrics for trees and ferns, including specificity, jaccard index, accuracy, AUC, and PAM. These metrics are essential for calculating the PAM score, which provides a general metric of the performance of each approach compared to basic precision, recall, and F1-score metrics. Our proposed approach, Supervisory-Net+TL, demonstrates significant improvements across these metrics. For tree classification, Supervisory-Net+TL achieved the highest specificity 0.9538, jaccard index 0.8246, accuracy 0.9106, and AUC 0.9133, outperforming other models. For ferns classification, Supervisory-Net+TL similarly showed maximum values in specificity 0.9446, jaccard index 0.8258, accuracy 0.9084, and AUC 0.9098. When averaged both categories, Supervisory-Net+TL reached an average PAM of 0.8010, surpassing the performance of other architectures.

Figures 5 and 6 show different neural network evaluations for classifying ferns and trees using the graphic

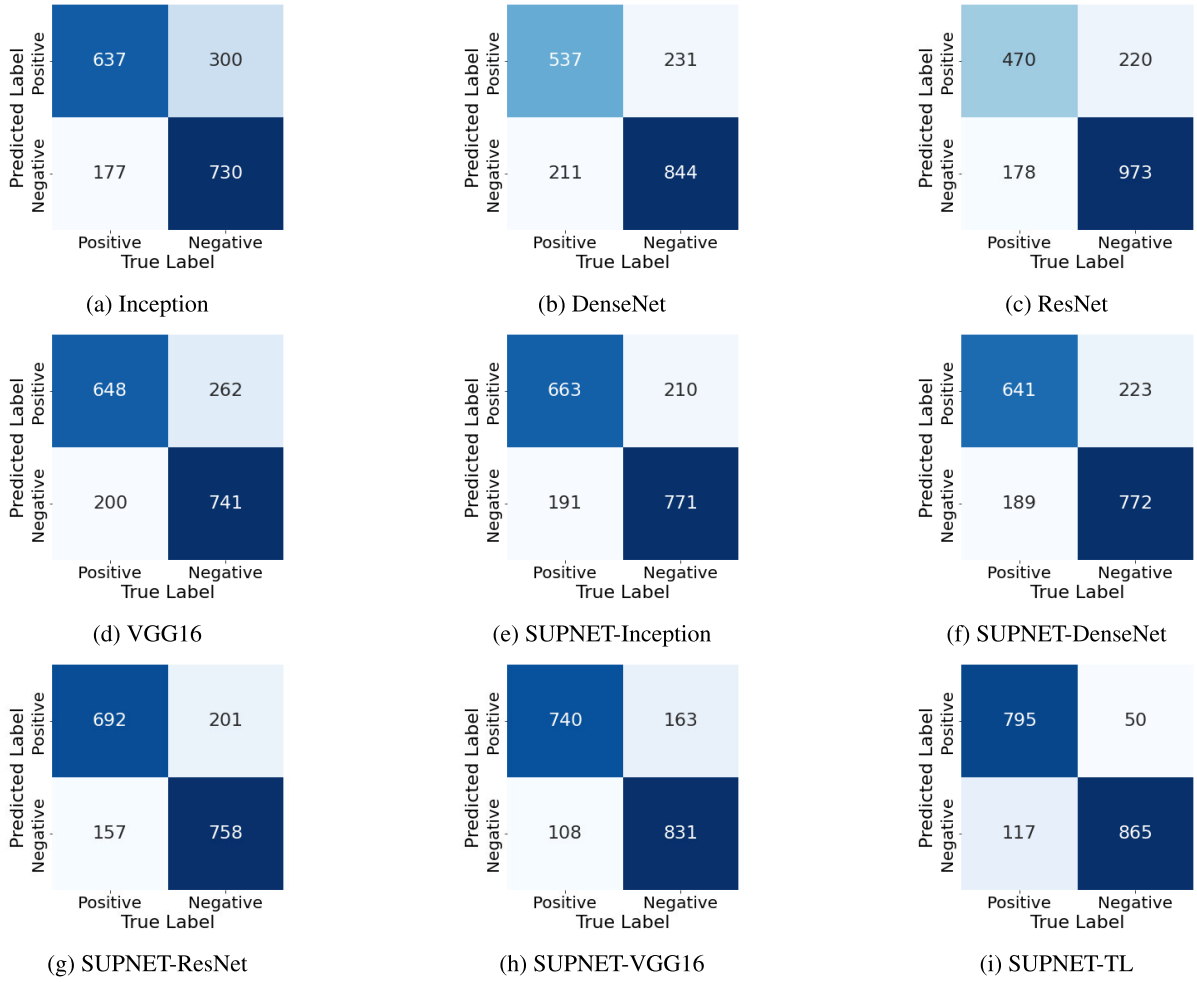


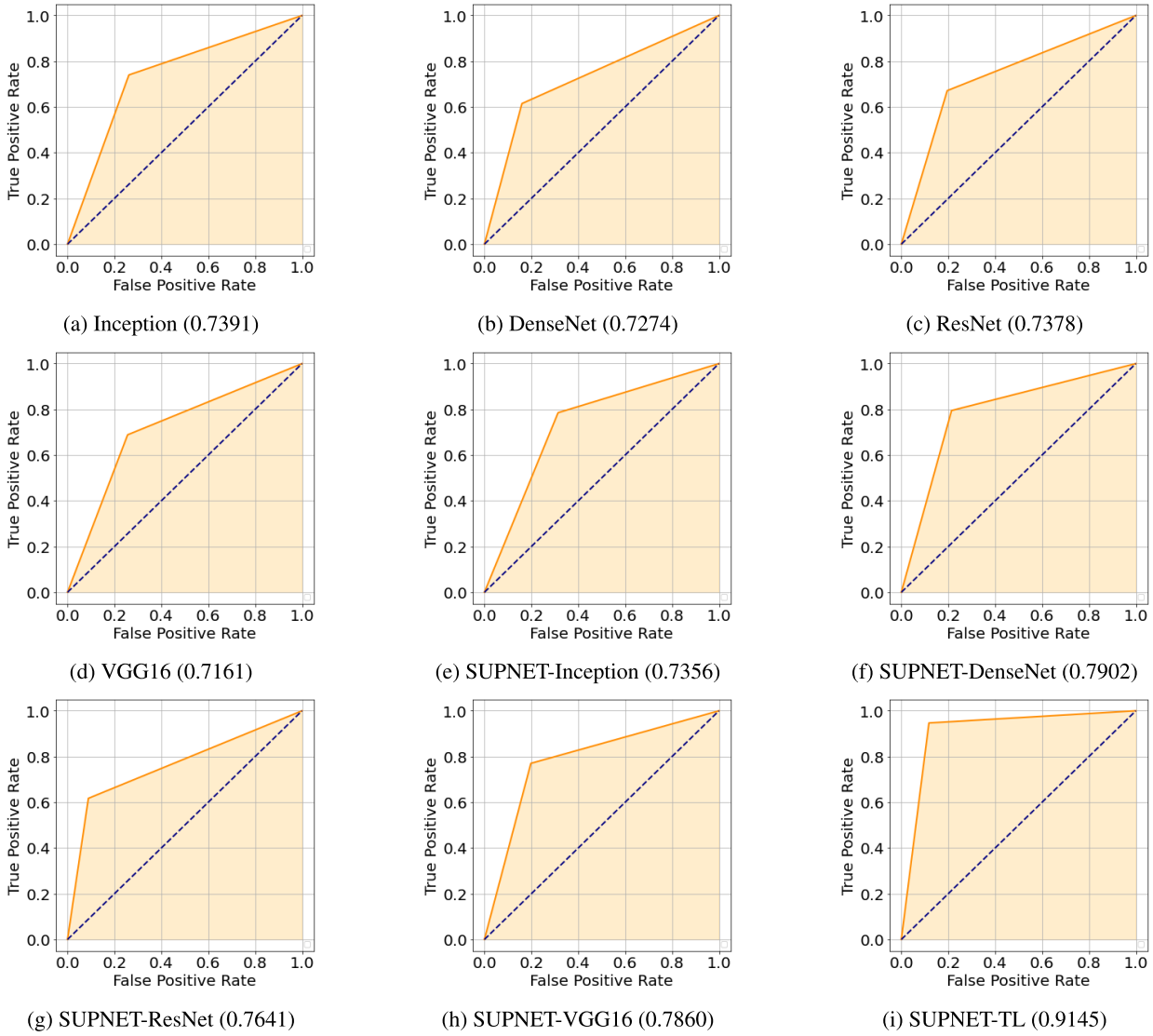
FIGURE 8. Average confusion matrices for fern classification obtained from 5-fold cross-validation using different convolutional neural network architectures.

PAM. This metric represents an overall measure of performance, calculated from six key indicators: specificity, sensitivity, jaccard index, accuracy, AUC, and F1-score. Each plot shows a polygon where the blue-shaded area represents the PAM score. A larger shaded area indicates a better general performance. In these experiments, the Supervisory-Net+TL model exhibits the largest shaded area, achieving PAM scores of 0.8005 for ferns and 0.8026 for trees. Other models (Supervisory-Net VGG16 and Supervisory-Net DenseNet) show acceptable performance, although lower. Architectures such as Inception and DenseNet exhibit smaller areas, reflecting lower performance. These results underline the effectiveness of Supervisory-Net+TL in classification tasks, demonstrating balanced and superior performance compared to other architectures.

F. SUPERVISORY-NET+TL VS SUPERVISORY-NET VGG-16

We compared the best classification performance using the supervisory configuration (Supervisory-Net VGG-16) and our configuration with the proposed transfer learning technique (Supervisory-Net+TL). These comparisons (**Tables** 3- 6) allow us to appreciate the behavior of the feedback loops. **Tables** 3 and 4 show the quantitative results for Supervisory-Net+TL and Supervisory-Net VGG-16, respectively. In these tables, we analyze the fern and stressed fern classifications. In this case, our methodology with transfer learning only needs half of the data to surpass 90% in the F1-score metric; that is, an approach that does not use the proposed transfer learning technique requires double the amount of data to achieve the required performance. Considering 17,984 images, Supervisory-Net+TL exhibited the best performance in precision, accuracy, and F1-score metrics. In the case of the recall metric, the performance was similar, with a variation of 0.74%.

Tables 5-6 show the quantitative results for Supervisory-Net+TL and Supervisory-Net VGG-16, respectively. In these tables, we analyze the tree and stressed tree classifications. In this case, our methodology with transfer learning requires 60% less data to surpass 90% in the F1-score metric; that is,



(a) Inception (0.7391) (b) DenseNet (0.7274) (c) ResNet (0.7378)

(d) VGG16 (0.7161) (e) SUPNET-Inception (0.7356) (f) SUPNET-DenseNet (0.7902)

(g) SUPNET-ResNet (0.7641) (h) SUPNET-VGG16 (0.7860) (i) SUPNET-TL (0.9145)

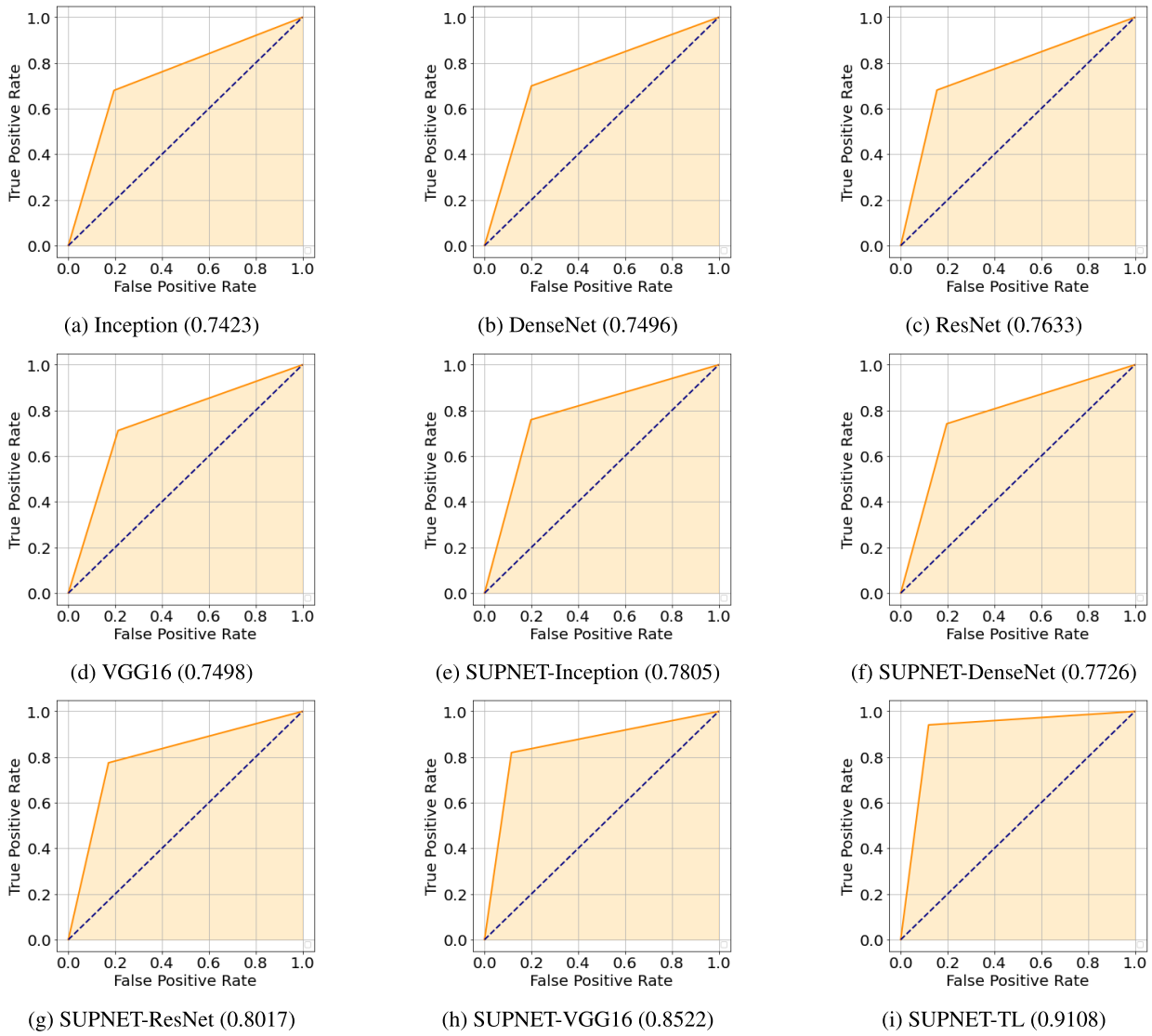
FIGURE 9. ROC Curves and corresponding AUC values for tree classification using various convolutional neural network architectures. The metrics were calculated using the average confusion matrices from Fig. 7. Each plot shows the True Positive Rate (TPR) against the False Positive Rate (FPR), with the diagonal line representing a random guess. The area under each curve (AUC) reflects the model's ability to distinguish between classes.

the approach that does not use the proposed transfer learning technique needs 32,301 additional images. Considering 54,279 images, Supervisory-Net+TL exhibited the best performance in precision, recall, accuracy, and F1-score metrics. In the two experiments (**Tables** 3 - 6), we observed an improvement when considering the proposed transfer learning technique. In addition, the proposed configuration uses the VGG-16 architecture [57] as the classification network due to its performance (**Table** 1). **Tables** 5 and 6 show that the increments of the synthetic training images are multiples of 333. These increments of 333 correspond to the image limit of the feedback loop (**Section** III-C2).

The model enables the establishment of classification performance (desired performance ξ), which provides an efficient method for stressed crop classification using fewer images for training. Second, our methodology simplifies the often complex tasks of dataset generation and training, offering a pathway to automate the processes for classifying stressed plants. Thus, it reduces the manual effort required for data preparation and model training, making the entire pipeline more efficient and scalable. Finally, the potential to automate these tasks can lead to significant improvements in the efficiency of training classification networks for the detection of stressed crops. These contributions are critical to advancing agricultural technology in computer vision.

V. CONCLUSION

This study introduced a new supervisory configuration for detecting stressed plants and generating synthetic databases.



(a) Inception (0.7423) (b) DenseNet (0.7496) (c) ResNet (0.7633)

(d) VGG16 (0.7498) (e) SUPNET-Inception (0.7805) (f) SUPNET-DenseNet (0.7726)

(g) SUPNET-ResNet (0.8017) (h) SUPNET-VGG16 (0.8522) (i) SUPNET-TL (0.9108)

FIGURE 10. ROC Curves and corresponding AUC values for fern classification using various convolutional neural network architectures. The metrics were calculated using the average confusion matrices from Fig. 8. Each plot shows the True Positive Rate (TPR) against the False Positive Rate (FPR), with the diagonal line representing a random guess. The area under each curve (AUC) reflects the model's ability to distinguish between classes.

Our strategy was to integrate the generalization capacity of deep learning with the automatic generation of synthetic information. This methodology comprises three phases. First, we collected a small set of Internet images of the stressed crops. Second, the process involves final layer training of the image generation model by introducing a new node into the network. Finally, we supervised the generative model using a classification neural network and a feedback loop. This supervision increases the quality of the generated images.

We used synthetic and real image datasets (Internet images) that provide different crops (ferns and trees) and two crop labels (stressed and unstressed). The proposed configuration showed a 0.238 increase in average precision and a 0.108 increase in average recall compared with traditional classification architectures using the same synthetic dataset; that is, we correctly recognized 23.85% more in the classification and 10.8% more in the ground truth. Although we trained our model using synthetic images, we evaluated the efficiency of this supervisory configuration by using real image datasets. These results demonstrated the feasibility of this configuration for the classification of stressed crops using synthetic datasets. We used synthetic images to train the network because the state-of-the-art does not have sufficient stressed crop images; that is, the datasets available in the state of the art are insufficient for training the proposed methodology.

In conclusion, the above experiments demonstrated the feasibility of proposing a configuration of deep learning networks for synthetic data generation and crop classification. We studied both stressed and unstressed plants

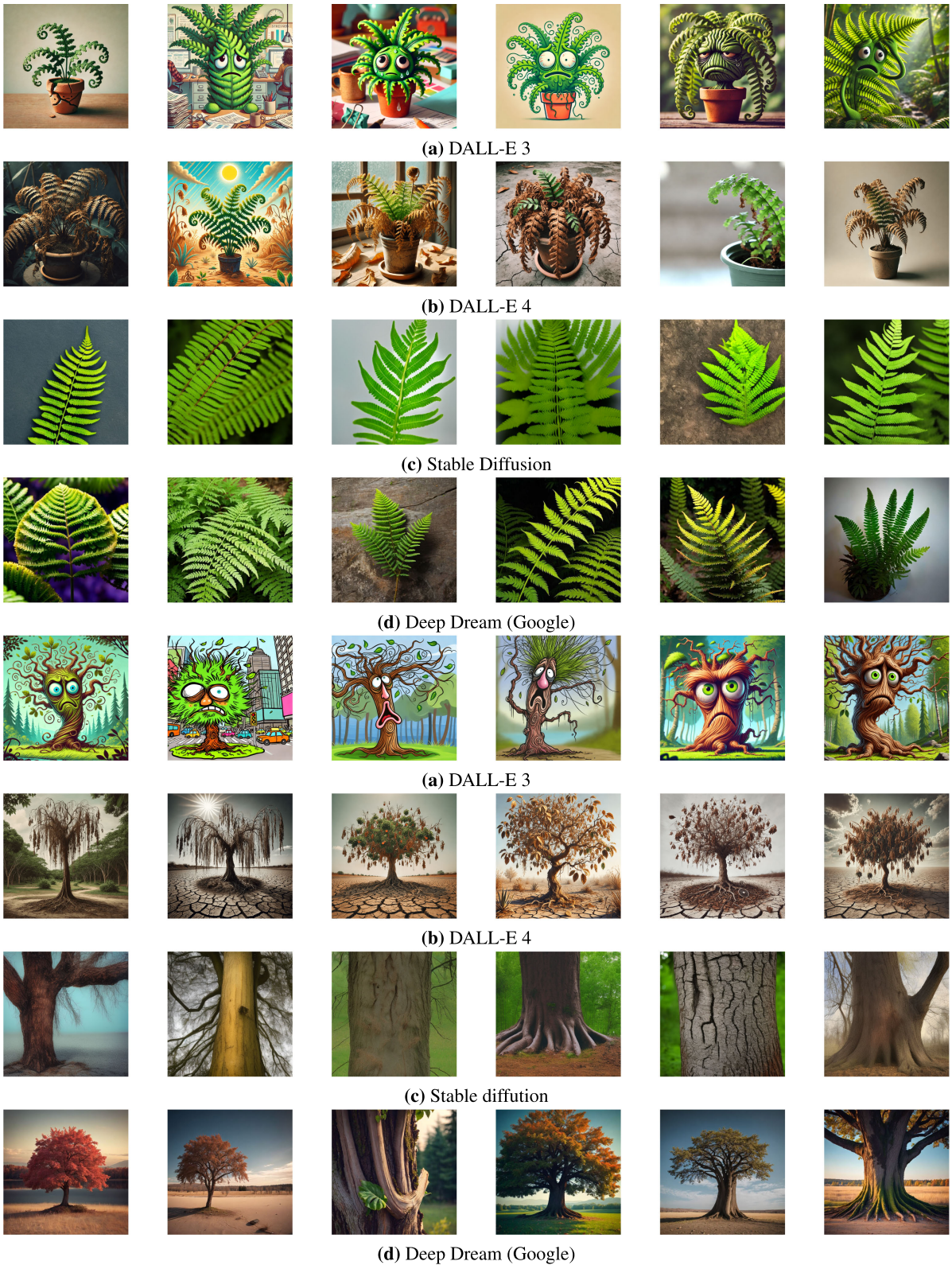
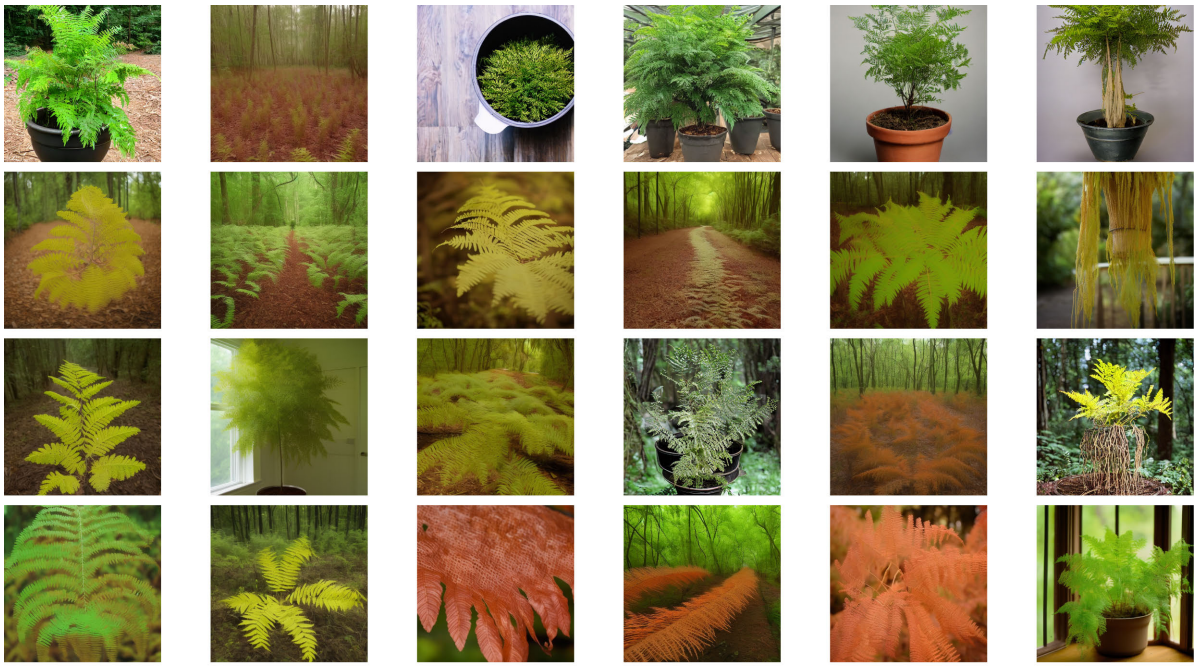
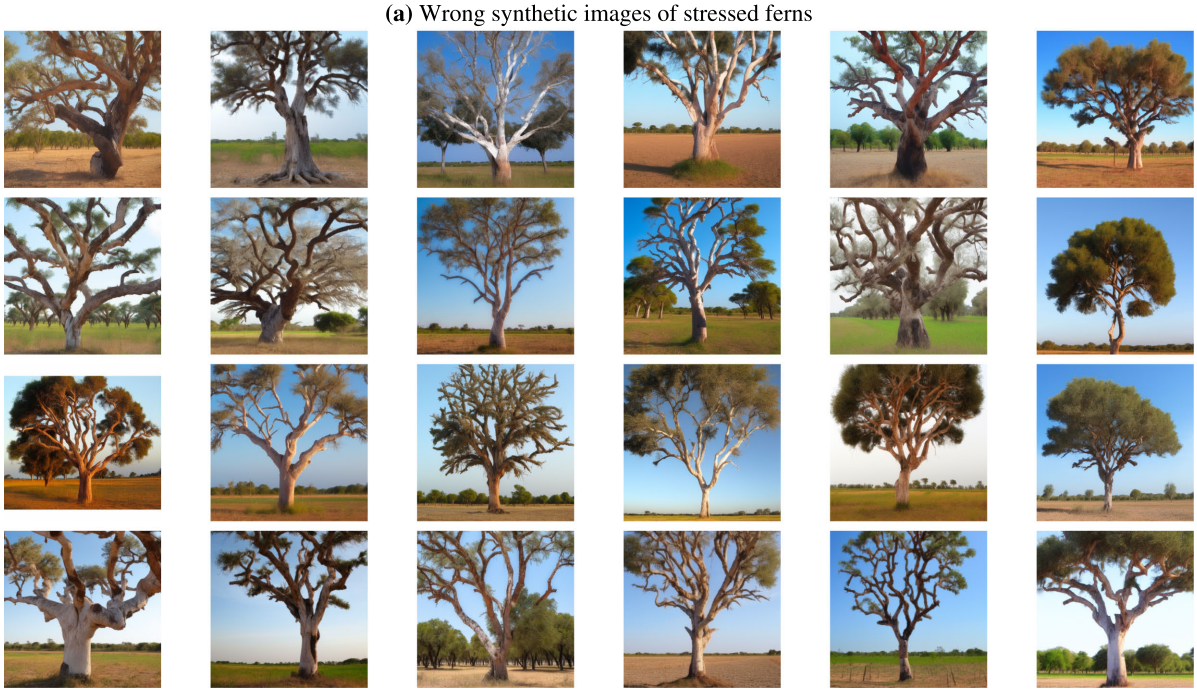


FIGURE 11. Comparison of different image generation approaches. We present images of stressed crops generated with different image-generation networks using the same prompt (DALL-E 3 (a), DALL-E 4 (b), Stable Diffusion (c), Deep Dream Google (d)).

under various conditions. Unlike previous studies, the proposed approach enables us to determine the classification performance when applying this configuration. In addition, our methodology facilitates the generation of datasets and



(a) Wrong synthetic images of stressed ferns

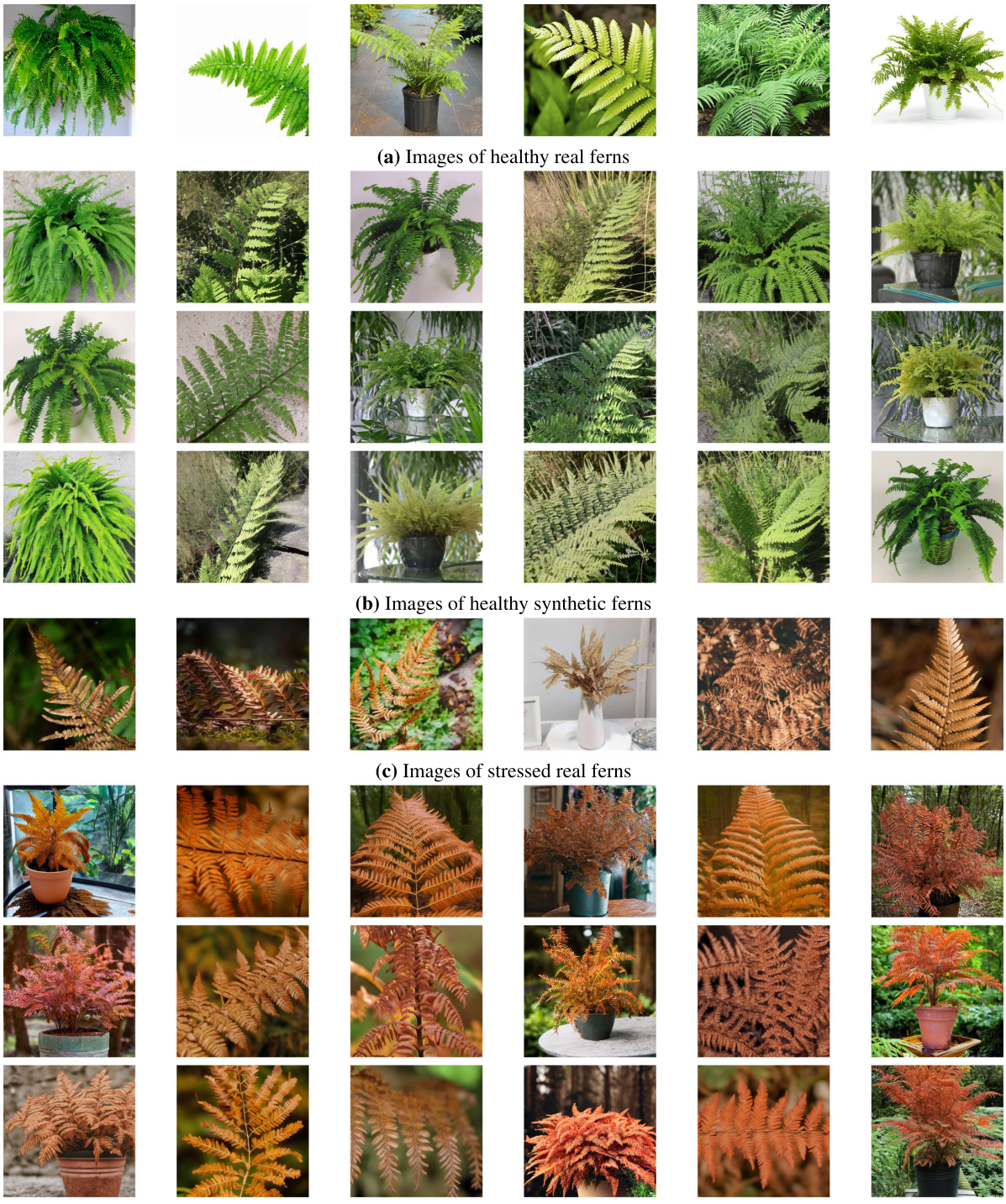


(b) Wrong synthetic images of stressed trees

FIGURE 12. **Examples of synthetic images discarded. Section (a) shows synthetic images of stressed ferns that our methodology discards. Section (b) presents synthetic images of stressed trees that our methodology discards. Discarding these images, we can create synthetic datasets that accurately represent the characteristics and morphology of stressed crops.**

training; that is, this study provides an approximation to automate the training process and data generation for stressed plant classification. Thus, tools for monitoring and caring for crops under stress can be developed with precise performance. Finally, the automation of these processes could potentially boost productivity and efficiency in agricultural work.

The main limitations of the proposed method lie in 3 principal aspects. First, only two crops are being used, ferns and trees, which restricts the generalizability of the results to other plant species. In addition, the type of stress assessed must be visible since our method can recognize stress in images where the plant shows physical signs of stress. Finally, an evaluation dataset (environment in the real



(a) Images of healthy real ferns

(b) Images of healthy synthetic ferns

(c) Images of stressed real ferns

(d) Images of stressed synthetic ferns

FIGURE 13. Comparison of real and synthetic fern images under different health conditions. (a) Healthy real ferns. (b) Healthy synthetic ferns generated by the proposed method. (c) Stressed real ferns. (d) Stressed synthetic ferns with real stress patterns.

world) is required with different conditions and variations, representing a challenge regarding data availability and collection.

Future work involves further investigation of the improvement of the proposed supervisory configuration. There is also an open question about what steps in our method can be replaced by a single module (deep module). In addition, we consider exploring the implementation of parallel processing or reconfigurable computing to accelerate the supervisory configuration.

(a) Images of healthy real trees

(b) Images of healthy synthetic trees

(c) Images of stressed real trees

(d) Images of stressed synthetic trees

FIGURE 14. Comparison of real and synthetic tree images under different health conditions. (a) Healthy real ferns. (b) Healthy synthetic ferns generated by the proposed method. (c) Stressed real ferns. (d) Stressed synthetic ferns with real stress patterns.

APPENDIX

A. TREE CONFUSION MATRICES

See Figure. 7.

B. FERN CONFUSION MATRICES

See Figure. 8.

C. TREE ROC-AUC

See Figure. 9.

D. FERN ROC-AUC

See Figure. 10.

E. STRESSED IMAGES OF DIFFERENT GENERATING NETWORKS

See Figure. 11.

F. WRONG SYNTHETIC IMAGES OF STRESSED CROPS

See Figure. 12.

G. REAL AND SYNTHETIC FERN IMAGES WITH AND WITHOUT STRESS

See Figure. 13.

H. REAL AND SYNTHETIC TREE IMAGES WITH AND WITHOUT STRESS

See Figure. 14.

REFERENCES

[1] Z. Gao, Z. Luo, W. Zhang, Z. Lv, and Y. Xu, "Deep learning application in plant stress imaging: A review," *AgriEngineering*, vol. 2, no. 3, pp. 430–446, Jul. 2020.

[2] L. Jia, L. Liu, Y. Zhang, W. Fu, X. Liu, Q. Wang, M. Tanveer, and L. Huang, "Microplastic stress in plants: Effects on plant growth and their remediations," *Frontiers Plant Sci.*, vol. 14, Aug. 2023, Art. no. 1226484.

[3] R. Xue, C. Zhang, H. Yan, J. Li, J. Ren, M. Akhlaq, M. U. Hameed, and K. N. Disasa, "Physiological response of tomato and cucumber plants to micro-spray in high-temperature environment: A scientific and effective means of alleviating crop heat stress," *Agronomy*, vol. 13, no. 11, p. 2798, Nov. 2023.

[4] A. Abbas, Z. Zhang, H. Zheng, M. M. Alami, A. F. Alrefaei, Q. Abbas, S. A. H. Naqvi, M. J. Rao, W. F. A. Mosa, Q. Abbas, A. Hussain, M. Z. Hassan, and L. Zhou, "Drones in plant disease assessment, efficient monitoring, and detection: A way forward to smart agriculture," *Agronomy*, vol. 13, no. 6, p. 1524, May 2023.

[5] A. P. Rosa, L. Barão, L. Chambel, C. Cruz, and M. M. Santana, "Early identification of plant drought stress responses: Changes in leaf reflectance and plant growth promoting rhizobacteria selection—The case study of tomato plants," *Agronomy*, vol. 13, no. 1, p. 183, Jan. 2023.

[6] G. A. Mesías-Ruiz, M. Pérez-Ortiz, J. Dorado, A. I. de Castro, and J. M. Peña, "Boosting precision crop protection towards agriculture 5.0 via machine learning and emerging technologies: A contextual review," *Frontiers Plant Sci.*, vol. 14, Mar. 2023, Art. no. 1143326.

[7] M. Safdar, M. A. Shahid, A. Sarwar, F. Rasul, M. D. Majeed, and R. M. Sabir, "Crop water stress detection using remote sensing techniques," *Environ. Sci. Proc.*, vol. 25, no. 1, p. 20, 2023, doi: 10.3390/ECWS-7-14198.

[8] H. S. Naik, J. Zhang, A. Lofquist, T. Assefa, S. Sarkar, D. Ackerman, A. Singh, A. K. Singh, and B. Ganapathysubramanian, "A real-time phenotyping framework using machine learning for plant stress severity rating in soybean," *Plant Methods*, vol. 13, no. 1, pp. 1–12, Dec. 2017.

[9] M. Janni, M. Gullì, E. Maestri, M. Marmiroli, B. Valliyodan, H. T. Nguyen, and N. Marmiroli, "Molecular and genetic bases of heat stress responses in crop plants and breeding for increased resilience and productivity," *J. Experim. Botany*, vol. 71, no. 13, pp. 3780–3802, Jun. 2020.

[10] N. S. Chandel, Y. A. Rajwade, K. Dubey, A. K. Chandel, A. Subeesh, and M. K. Tiwari, "Water stress identification of winter wheat crop with state-of-the-art AI techniques and high-resolution thermal-RGB imagery," *Plants*, vol. 11, no. 23, p. 3344, Dec. 2022, doi: 10.3390/plants11233344.

[11] M. B. Riley, M. R. Williamson, and O. Maloy, "Plant disease diagnosis," *Plant Health Instructor*, vol. 10, 2016.

[12] R. Prasad, K. R. Ranjan, and A. K. Sinha, "AMRAPALIKA: An expert system for the diagnosis of pests, diseases, and disorders in Indian mango," *Knowl.-Based Syst.*, vol. 19, no. 1, pp. 9–21, Mar. 2006.

[13] W. Liu, L. Meng, X. Liu, C. Liu, and W. Jin, "Establishment of an ELISA method for quantitative detection of PAT/pat in GM crops," *Agriculture*, vol. 12, no. 9, p. 1400, Sep. 2022.

[14] D. Rotenberg, T. S. Thompson, T. L. German, and D. K. Willis, "Methods for effective real-time RT-PCR analysis of virus-induced gene silencing," *J. Virological Methods*, vol. 138, nos. 1–2, pp. 49–59, Dec. 2006.

[15] A. I. Cardos, A. Maghiar, D. C. Zaha, O. Pop, L. Fritea, F. M. Groza, and S. Cavalu, "Evolution of diagnostic methods for *Helicobacter pylori* infections: From traditional tests to high technology, advanced sensitivity and discrimination tools," *Diagnostics*, vol. 12, no. 2, p. 508, Feb. 2022, doi: 10.3390/diagnostics12020508.

[16] M. Venbrux, S. Crauwels, and H. Rediers, "Current and emerging trends in techniques for plant pathogen detection," *Frontiers Plant Sci.*, vol. 14, May 2023, Art. no. 1120968.

[17] J. J. Olas, F. Fichtner, and F. Apelt, "All roads lead to growth: Imaging-based and biochemical methods to measure plant growth," *J. Experim. Botany*, vol. 71, no. 1, pp. 11–21, Jan. 2020, doi: 10.1093/jxb/erz406.

[18] A. K. Singh, B. Ganapathysubramanian, S. Sarkar, and A. Singh, "Deep learning for plant stress phenotyping: Trends and future perspectives," *Trends Plant Sci.*, vol. 23, no. 10, pp. 883–898, Oct. 2018.

[19] A. Singh, B. Ganapathysubramanian, A. K. Singh, and S. Sarkar, "Machine learning for high-throughput stress phenotyping in plants," *Trends Plant Sci.*, vol. 21, no. 2, pp. 110–124, Feb. 2016.

[20] M. Ouhami, A. Hafiane, Y. Es-Saady, M. El Hajji, and R. Canals, "Computer vision, IoT and data fusion for crop disease detection using machine learning: A survey and ongoing research," *Remote Sens.*, vol. 13, no. 13, p. 2486, Jun. 2021, doi: 10.3390/rs13132486.

[21] M. Zekiwos and A. Bruck, "Deep learning-based image processing for cotton leaf disease and pest diagnosis," *J. Electr. Comput. Eng.*, vol. 2021, pp. 1–10, Jun. 2021.

[22] Y. Long and M. Ma, "Recognition of drought stress state of tomato seedling based on chlorophyll fluorescence imaging," *IEEE Access*, vol. 10, pp. 48633–48642, 2022.

[23] R. Deng, M. Tao, H. Xing, X. Yang, C. Liu, K. Liao, and L. Qi, "Automatic diagnosis of Rice diseases using deep learning," *Frontiers Plant Sci.*, vol. 12, Aug. 2021, Art. no. 701038.

[24] Y.-Y. Zheng, J.-L. Kong, X.-B. Jin, X.-Y. Wang, T.-L. Su, and M. Zuo, "CropDeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture," *Sensors*, vol. 19, no. 5, p. 1058, Mar. 2019, doi: 10.3390/s19051058.

[25] Y.-H. Wang and W.-H. Su, "Convolutional neural networks in computer vision for grain crop phenotyping: A review," *Agronomy*, vol. 12, no. 11, p. 2659, Oct. 2022, doi: 10.3390/agronomy12112659.

[26] W. Yang, C. Yang, Z. Hao, C. Xie, and M. Li, "Diagnosis of plant cold damage based on hyperspectral imaging and convolutional neural network," *IEEE Access*, vol. 7, pp. 118239–118248, 2019.

[27] W.-J. Hu, J. Fan, Y.-X. Du, B.-S. Li, N. Xiong, and E. Bekkering, "MDFC-ResNet: An agricultural IoT system to accurately recognize crop diseases," *IEEE Access*, vol. 8, pp. 115287–115298, 2020.

[28] Z. Huang, Z. Pan, and B. Lei, "Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data," *Remote Sens.*, vol. 9, no. 9, p. 907, Aug. 2017, doi: 10.3390/rs9090907.

[29] Y. Yuan, L. Chen, Y. Ren, S. Wang, and Y. Li, "Impact of dataset on the study of crop disease image recognition," *Int. J. Agricult. Biol. Eng.*, vol. 15, no. 5, pp. 181–186, 2022.

[30] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. S. Albahri, B. S. N. Al-dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy, Y. Duan, A. Abdullah, L. Farhan, Y. Lu, A. Gupta, F. Albu, A. Abbosh, and Y. Gu, "A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications," *J. Big Data*, vol. 10, no. 1, p. 46, Apr. 2023.

[31] D. Elavarasan and P. M. D. Vincent, "Crop yield prediction using deep reinforcement learning model for sustainable agrarian applications," *IEEE Access*, vol. 8, pp. 86886–86901, 2020.

[32] J. Lucas, G. Tucker, R. B. Grosse, and M. Norouzi, "Don't blame the ELBO! A linear VAE perspective on posterior collapse," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[33] R. Bhalodia, I. Lee, and S. Elhabian, "dpVAEs: Fixing sample generation for regularized VAEs," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–18.

[34] R. Pastrana, "Disentangling variational autoencoders," 2022, *arXiv:2211.07700*.

[35] Y. Lu, D. Chen, E. Olaniyi, and Y. Huang, "Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review," *Comput. Electron. Agricult.*, vol. 200, Sep. 2022, Art. no. 107208.

[36] S. Modak, J. Heil, and A. Stein, "Pansharpening low-altitude multispectral images of potato plants using a generative adversarial network," *Remote Sens.*, vol. 16, no. 5, p. 874, Mar. 2024, doi: 10.3390/rs16050874.

[37] Q. Dai, X. Cheng, Y. Qiao, and Y. Zhang, "Crop leaf disease image super-resolution and identification with dual attention and topology fusion generative adversarial network," *IEEE Access*, vol. 8, pp. 55724–55735, 2020.

[38] H. Madokoro, K. Takahashi, S. Yamamoto, S. Nix, S. Chiyonobu, K. Saruta, T. K. Saito, Y. Nishimura, and K. Sato, "Semantic segmentation of agricultural images based on style transfer using conditional and unconditional generative adversarial networks," *Appl. Sci.*, vol. 12, no. 15, p. 7785, Aug. 2022, doi: 10.3390/app12157785.

[39] A. E. Blanchard, C. Stanley, and D. Bhowmik, "Using GANs with adaptive training data to search for new molecules," *J. Cheminformatics*, vol. 13, no. 1, pp. 1–8, Dec. 2021.

[40] L. Jiang, B. Dai, W. Wu, and C. C. Loy, "Deceive D: Adaptive pseudo augmentation for GAN training with limited data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 21655–21667.

[41] R. Sauber-Cole and T. M. Khoshgoftaar, "The use of generative adversarial networks to alleviate class imbalance in tabular data: A survey," *J. Big Data*, vol. 9, no. 1, p. 98, Aug. 2022.

[42] M. Di Cicco, C. Potena, G. Grisetti, and A. Pretto, "Automatic model based dataset generation for fast and accurate crop and weeds detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5188–5195.

[43] R. R. Patil and S. Kumar, "Rice transformer: A novel integrated management system for controlling Rice diseases," *IEEE Access*, vol. 10, pp. 87698–87714, 2022.

[44] B. Zhuang, J. Liu, Z. Pan, H. He, Y. Weng, and C. Shen, "A survey on efficient training of transformers," 2023, *arXiv:2302.01107*.

[45] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo, "StyleSwin: Transformer-based GAN for high-resolution image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11294–11304.

[46] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Switzerland: Springer, 2020, pp. 213–229.

[47] B. Thokala and S. Doraikannan, "Detection and classification of plant stress using hybrid deep convolution neural networks: A multi-scale vision transformer approach," *Traitement du Signal*, vol. 40, no. 6, pp. 2635–2647, Dec. 2023.

[48] A. Kumar Patra, A. Varshney, and L. Sahoo, "An explainable vision transformer with transfer learning combined with support vector machine based efficient drought stress identification," 2024, *arXiv:2407.21666*.

[49] K. Lehouel, C. Saber, M. Bouziani, and R. Yaagoubi, "Remote sensing crop water stress determination using CNN-ViT architecture," *AI*, vol. 5, no. 2, pp. 618–634, May 2024.

[50] J. Xie, J. Hua, S. Chen, P. Wu, P. Gao, D. Sun, Z. Lyu, S. Lyu, X. Xue, and J. Lu, "HyperSFormer: A transformer-based end-to-end hyperspectral image classification method for crop classification," *Remote Sens.*, vol. 15, no. 14, p. 3491, Jul. 2023, doi: 10.3390/rs15143491.

[51] A. Khan, Z. Rauf, A. Sohail, A. Rehman, H. Asif, A. Asif, and U. Farooq, "A survey of the vision transformers and their CNN-transformer based variants," 2023, *arXiv:2305.09880*.

[52] M. Pérez-Patricio, J. Camas-Anzueto, A. Sanchez-Alegría, A. Aguilar-González, F. Gutiérrez-Miceli, E. Escobar-Gómez, Y. Voisin, C. Rios-Rojas, and R. Grajales-Coutiño, "Optical method for estimating the chlorophyll contents in plant leaves," *Sensors*, vol. 18, no. 2, p. 650, Feb. 2018, doi: 10.3390/s18020650.

[53] A. Lowe, N. Harrison, and A. P. French, "Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress," *Plant Methods*, vol. 13, no. 1, p. 80, Dec. 2017.

[54] G. Ríos-Toledo, M. Pérez-Patricio, L. Á. Cundapí-López, J. L. Camas-Anzueto, N. A. Morales-Navarro, and J. D. J. Osuna-Coutiño, "Plant stress recognition using deep learning and 3D reconstruction," in *Proc. Mexican Conf. Pattern Recognit.* Springer, 2023, pp. 114–124, doi: 10.1007/978-3-031-33783-3_11.

[55] K. Omasa, F. Hosoi, and A. Konishi, "3D LiDAR imaging for detecting and understanding plant responses and canopy structure," *J. Experim. Botany*, vol. 58, no. 4, pp. 881–898, Nov. 2006.

[56] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[58] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[59] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[60] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[62] O. Aydemir, "A new performance evaluation metric for classifiers: Polygon area metric," *J. Classification*, vol. 38, no. 1, pp. 16–26, Apr. 2021.

J. RENÁN VELÁZQUEZ-GONZÁLEZ received the B.Eng. degree in computer systems from the Instituto Tecnológico de Tuxtla Gutiérrez, Tuxtla Gutiérrez, Chiapas, Mexico, in July 2017, for confirming his commitment and passion to technological advancement, and the M.Sc. degree (Hons.) in computer science from the renowned National Institute of Astrophysics, Optics and Electronics (INAOE), Puebla, Mexico, in December 2020, for demonstrating exceptional academic prowess. He is currently pursuing the Ph.D. degree in engineering science with the Instituto Tecnológico de Tuxtla Gutiérrez. His research interests include deep learning, image processing and analysis, and the innovative applications of semantic segmentation.

MADAIN PÉREZ-PATRICIO received the Ph.D. degree in automation and industrial computing from Université Lille 1, Sciences et Technologies, Villeneuve-d'Ascq, France, in 2005. Since September 1997, he has been a Research Professor with the Department of Postgraduate Research, National Technological Institute of Mexico, Tuxtla Gutiérrez, Mexico.

J. A. DE JESÚS OSUNA-COUTIÑO received the M.Sc. degree (Hons.) in mechatronics engineering from the National Technological Institute of Mexico, Tuxtla Gutiérrez, Mexico, in June 2015, and the Ph.D. degree in computer science from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Puebla, Mexico, in August 2020. Since September 2022, he has been a Postdoctoral Researcher with the Department of Science, National Technological Institute of Mexico. His research interests include computer vision, the development of deep learning configurations, dataset automation, and 3D reconstruction.

JORGE LUIS CAMAS-ANZUETO received the M.Sc. and Ph.D. degrees in optics with a major in optoelectronic from the National Institute of Astrophysics, Optics and Electronics, Puebla, Mexico, in 2000 and 2004, respectively. He is currently a Researcher with the Postgraduate Department, National Technological Institute of Mexico, Tuxtla Gutiérrez, Mexico. He is also a member of the National System of Researchers. His current research interests include fiber sensors, optical sensors, and optical metrology.

ABIEL AGUILAR-GONZÁLEZ received the Ph.D. degree in computer science from the National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico, and the Ph.D. degree in electronic systems from the Pascal Institute, Université Clermont Auvergne (UCA), France, in June 2019. His Ph.D. thesis was on embedded hardware architectures (FPGA/CUDA) for SLAM applications. He is currently a Software Architect at PolyWorks Mexico, where he works on computer vision algorithms and unity implementations suitable for augmented reality systems under 3D metrology applications.

N. A. MORALES-NAVARRO received the B.Eng. degree in computer systems engineering and the M.Sc. degree in mechatronics engineering from the Instituto Tecnológico de Tuxtla Gutiérrez and the Ph.D. degree from Universidad de Ciencia y Tecnología Descartes, in 2017. He has been a Researcher at the Instituto Tecnológico de Tuxtla Gutiérrez, Chiapas, Mexico, since 2012. His research interests include computer vision, deep learning, and automation in mechatronic systems.

CARLOS A. HERNÁNDEZ-GUTIÉRREZ is an Entrepreneur and Pioneering Researcher in GaN-based devices in Mexico. Currently, he works for the Electrical and Electronics Department, TecNM Campus Tuxtla Gutiérrez. His research interests include solar cells, photodetectors, nanostructured devices, and circuit design.