IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 August 2023, accepted 6 September 2023, date of publication 12 September 2023,
date of current version 20 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3314590

RESEARCH ARTICLE

Enhanced Classification of Coffee Leaf Biotic Stress by Synergizing Feature Concatenation and Dimensionality Reduction

MUHAMMAD ARMGHAN LATIF[1], NOOR AFSHAN[2], ZOHAIB MUSHTAQ[3], NABEEL AHMED KHAN[4], MUHAMMAD IRFAN[5], GRZEGORZ NOWAKOWSKI[6], SAMAR M. ALQHTANI[7], SALIM NASAR FARAJ MURSAL[5], AND SERGII TELENYK[6]

[1]Department of Computer and Information System, Cleveland State University, Cleveland, OH 44115, USA
[2]Department of Software Engineering, Faculty of Computer Science, Lahore Garrison University, Lahore 54000, Pakistan
[3]Department of Electrical Engineering, College of Engineering and Technology, University of Sargodha, Sargodha 40100, Pakistan
[4]Department of Electrical Engineering, Riphah International University, Islamabad 44000, Pakistan
[5]Electrical Engineering Department, College of Engineering, Najran University, Najran 61441, Saudi Arabia
[6]Faculty of Electrical and Computer Engineering, Cracow University of Technology, 31-155 Kraków, Poland
[7]Department of Information Systems, College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia

Corresponding author: Zohaib Mushtaq (zohaib.mushtaq@uos.edu.pk)

The APC was funded by the Faculty of Electrical and Computer Engineering, Cracow University of Technology and the Ministry of Science and Higher Education, Republic of Poland (grant no. E-1/2023).

ABSTRACT Significant yield challenges are posed by biotic stress on coffee leaves, which has a negative effect on the revenue generation of this highly utilized commodity. Numerous studies have proposed techniques for the early detection and classification of biotic stress in coffee leaves. In this study, we propose a technique called extracted feature ensemble (EFE) for classifying healthy and infected classes. Transfer learning-based convolutional neural networks (CNNs) and custom-designed features are used to improve classification performance. Under the concept of EFE, three methodologies are proposed for evaluating various extracted feature combinations and determining the effect of dimensionality on the performance of the model. In addition, a semi-segmentation approach is used to guide the extraction of informative foreground details, while non-segmented inputs are used to improve the model's robustness against complex background noise. By improving three open-source datasets for biotic stress categorization in coffee leaves, a new dataset was created and employed. The first proposed method, ECNN, focused on the effective concatenation of five CNNs and obtained a classification accuracy of 93.45% using a decision tree classifier, exceeding the maximum individual accuracy of 86.07% from Mobile-Net v3 features. In addition, the HLGGM method was investigated, which demonstrated an enhanced accuracy of 99.16% by combining dimension-reduced Mobile-Net v3 features with handcrafted features. HLGCM, the final approach represented, aimed at extracting features from dimensionality-reduced handmade and CNN-based data, and ultimately succeeded in accomplishing an accuracy of 99.49 percent by using decision tree model. The obtained results demonstrate the efficacy of feature concatenation in enhancing the classification model's discriminative capabilities and classification accuracy. The appropriate combination of hand-made and CNN-based features gives better accuracy and interesting insights into the effect of feature reduction on model classification efficiency. The article offers dimensionality reduction, directed learning, and feature concatenation techniques for identifying coffee leaf diseases. This work can aid in the development of computationally efficient and accurate disease control and coffee plant sustainability strategies.

INDEX TERMS Features concatenations, transfer learning, coffee leaf, convolutional neural network, biotic stress.

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

I. INTRODUCTION

Plants play a vital role in our ecosystem by being our primary production source, ensuring food sustenance and effectively supply oxygen [1]. They aid in substantial ecological growth by providing habitats, resources, food, well-being for humans and a nurturing bio diversity. Among the myriad diversity of plants, Coffee plants hold an immense significance owing to their globalized consumption and financial impact on society. In United States alone Coffee-related economical activities comprise an approximated 1.6% of total U.S GDP generating nearly 28 billion$ in taxes [2]. In Africa, more than 12 Million households depend on the revenue generating by coffee production [3] On a globalized scale, a raging 125 Million people have their livelihood dependent on Coffee in one way or the other. With the coffee industry having a market value of 102 billion $, its significance is obvious [4], [5].

With a prospective economical overview dependent on the yield, Coffee plants are highly susceptible to biotic stress [6] that may manifest in many forms - resulting in a production drop. These diseases have a tendency to escalate quickly and consume almost entire fields [7]. The coffee plant diseases are categorized as fungal, bacterial, viral, and nematode affecting various regions of a plant [8]. This study focused on the classification of leaf based diseases from the fungal category. These include Leaf Rust, Phoma, Miner and Cercospora [9]. The effect of these diseases can be estimated by the fact that leaf Rust alone can drop the entire crop yield by 75% [10]. These diseases have a tendency to degrade the life of leaf and overall production at various stages of development.

The classification of these leaf diseases poses a significant challenge, when adapting certain traditional or machine learning methodologies. Out of many influencing factors, leaf variability can turn out to be a substantial problem. The fact that leaf from the same branch can show variations in texture, shape, color, and size makes it difficult to learn patterns. In [11] Adding to possible discoloration, lesions or spots identical to other stress classes leads to an ambiguity of models in successfully classifying the symptoms. Other known factors including co-occurrence of various diseases, limited training data, and image quality during acquisition, all of those issues led to the intricacy and complexity of the problem [12]

Many of the traditional, machine and deep learning methodologies have been implemented in recent years for the effective classification of coffee leaf diseases, but none went without a certain set of complications. Most recent studies rely heavily on a transfer learning approach using state-of-the-art convolutional models, because of the limited data [13], [14], [15], [16]. Some modified the pre-trained architectures for both severity estimation and classification [15]. Feature concatenation approaches were also presented for the pre-trained architectures to augment the model's ability to learn on diverse multi-scale features [8]. There were studies that played with the class imbalance conundrum using some traditional or Machine Learning approaches before subjecting the results back to a transfer learning model [14], [16]. In conclusion, many alternate combinations were leveraged to tackle the problems associated with the classification. Some of the significant aspects that were left unattended were the model's ability to deal with the biases, tackling the co-occurrence of diseases to correctly quantify the class probability distribution, the validation data of similar origin causing generalization problems for deep learning models achieving high accuracy and considering accuracy as being the sole benchmark for model's performance. Similarly, computational complexity of pre-trained convolutional models wasn't addressed that is a viable foreteller of the model's deployment capabilities in resource-constrained areas.

This study proposes a unique approach to take on coffee leaf biotic stress classification, using a dimensionality reduction based feature ensemble approach. Usually for smaller datasets, the CNN models might struggle to find meaningful patterns to generalize effectively [17]. To facilitate effective learning, ensemble feature extraction methodologies proposed in this paper can prove to be valuable in complimenting and augmenting the model's abilities. Three methodologies are proposed to evaluate different feature combinations, along with the impact of dimensionality on the model's performance [18]. This contribution of this study are as follows:

1) A methodology is proposed for effective feature concatenation of state-of-the-art transfer learning convolutional neural networks to analyze and evaluate the effect of CNN feature concatenation on model performance.
2) The second proposed methodology aims to explore the impact of a guided learning approach on classification performance by investigating the concatenation of dimensionality-reduced fine-tuned CNN features with handcrafted features.
3) The third proposed methodology is intended to generate an essence vector that captures the most defining properties by efficiently reducing dimensionality and performing feature selection from both the selected convolutional network and handcrafted features. The goal is to create a meaningful comparison of feature quality and length, allowing for an effective evaluation with former methodologies.
4) A semi-segmentation approach was employed to guide the extraction of descriptive foreground details from segmented leaves, the non-segmented leaves added to the robustness of the model against complex background noise.
5) Images for Hand Crafted Feature Extraction were filtered using Gaussian Blur and their brightness was enhanced for prominence of suppressed disease portions.
6) A custom segmentation pipeline was used to subtract the background for emphasized feature extraction of informative portions.
7) A new dataset was proposed by effective refining of three open source datasets [19], [20], [21].

The remaining part of the paper is organized as follows: Section II, the Literature Study is given. In Section III, the methodologies employed are described. The Results and Discussions are given in Section IV respective.

II. LITERATURE REVIEW

Numerous studies have been conducted to investigate the use of deep learning techniques in the categorization and detection of coffee leaf diseases. This literature review looks at a few of these research, noting the methodology used and the corresponding accuracy as can be shown in Table 1. Novtahaning et al. [13] in their study used 1300 images from [20]. A bagging ensemble of three individual convolutional neural networks i.e., VGG-16, Inception-v3 and ResNet-50 was used to effectively aggregate the output of three models for enhanced accuracy. The accuracy achieved through this approach was 97.13%. A similar study conducted by Francis Jesmer et al. [9] on 4675 images achieved 95.98% accuracy using the stage-wise ensemble of EffcientNet0, DenseNet-121 and VGG-16 models.

By combining the strength of different models, the stage wise ensemble was able to achieve improved accuracy and classification performance Some studies focused specifically on targeting class imbalances, such as the work of Hasan et al. [14]. In their study, the authors used a kernel density estimation (KDE) approach for automated clustering and classification. 2700 images from the dataset [20] were used to cluster the images into groups followed by a ResNet-50 architecture to classify the images. The accuracy achieved through this approach is 98%. Furthermore, comparative studies were also following up for traditional and deep learning feature extraction techniques. In this regard, Boa Sorte et al. [16], developed a framework for coffee leaves biotic stress classification using statistical and texture attributes along with a separate model, Alex Net to observe the performance. The convolutional method outranked the traditional technique on a dataset [20] of 750 images by yielding 98% on Kappa Statistics.

Work on transfer learning models was extended for enhanced stress classification results. A study by Abuhayi and Mossa [8] explored the feature concatenation of pre-trained Google-Net and ResNet features on a private dataset of 3288 images in an attempt to enhance the classification accuracy. The resultant accuracy from a concatenation of diverse set of features was 99.07%. Tassis and Krohling [22], In their respective study, employed few shot learning approach to classify the stress class of 1685 images from [20]. The purpose of using few shot learning was to address the limitations of traditional deep learning networks which need a large dataset for better generalization. The resultant accuracy obtained through this approach was 96%. Identical to classical transfer learning approaches, Paulos and Woldeyohannis [23] in their study used pre-trained Mobile Net and ResNet-50 models for classification of coffee leaf disease from 1120 images belonging to a private dataset. The accuracy reached a staggering 97.01% and 99.89% accuracy on 4 classes respectively. Alongside classification of biotic stress, some studies extended their approach to severity classification. One such study by Esgario et al. [15] proposed a deep learning based end-to-end biotic stress classification and severity estimation model. The model was trained on dataset using a modified version of ResNet 50 and was successful in achieving an accuracy of 95.24%.

While the above studies apprise us of the significant progress in biotic stress classification for coffee leaves, several research gaps still persist. These include exploring novel architectures, optimizing some data augmentation strategies, developing interpretable and explainable models, and inspecting and evaluating transfer learning variations. Additionally, class imbalance poses a significant problem in this domain owing to an occurrence frequency of some stress types and rarity of others. Addressing this challenge is also a force to be reckoned with. Overall, the reviewed studies demonstrate the effectiveness of deep learning techniques in the classification and recognition of coffee leaf diseases. The achieved accuracies range from 95.98% to 99.89% for varying classes and datasets, indicating the potential of these models for practical applications in the agricultural sector. Table 1. Illustrates the recent existing studies implemented on the different Coffee plant disease datasets.

III. MATERIALS AND METHODS

A. DATASET DESCRIPTION

The dataset used in this experimental work was a combination of Bracol [20], JMUBEN [19] and PDCMD [20] datasets. The new dataset is a collection of open source images available on [20] for coffee leaf disease containing high quality images of four classes of coffee leaf diseases namely Cercospora, Phoma, Rust and Healthy taken under controlled conditions. The [18] Dataset is a collection of open-source images available on [19] along with the associated information focusing on coffee leaf diseases. It is specifically designed to aid in the analysis and diagnosis of coffee plant health. The dataset [19] includes a diverse range of high-quality images featuring various types of coffee leaf diseases such as Cercospora, Rust, Phoma, Miner and Healthy. Description for each class is given in Table 2.

The dataset is a comprehensive collection of images open-sourced available at [21] representing various plant diseases. This dataset is specifically curated for the task of plant disease classification using machine learning and computer vision techniques. The number of diseased coffee leaf classes in all three datasets are given in Table 3. Furthermore, the illustration of the stress types considered in this study is in Figure 1. The figure shows the afflicted regions of the coffee leaves along with the stress typing.

B. DATA PREPROCESSING

1) REFINING

Assuming the datasets are represented by D_1, D_2 and D_3 having m,n and o samples for each of the respective biotic

TABLE 1. Related literature on similar datasets.

Reference	Year	Data Set	Methodology	Results
[13]	2022	BraCol Dataset with 1300 images. 250 for each class	Bagging Ensemble of three individual CNN models VGG16, ResNet50, Inception V3.	97.31%
[14]	2023	BraCol Dataset with 2700 samples	The approach was based on an extended kernel density estimation (KDE) method and a ResNet50 architecture.	98%
[15]	2020	BraCol Dataset	Multi task system based on ResNet 50	95.24%
[16]	2019	Private Dataset 750 Images Each Class	Statistical and Texture attributes and Convolutional Features using Alex Net.	98.0%
[8]	2023	Private Dataset of 3288 images	Feature Concatenation of GoogleNet and ResNet Features	99.08%
[9]	2022	4675 Images belonging to RoCole, BraCol and LiCole Datasets	Stage-wise ensemble of EfficientNet0, DenseNet 121 and VGG16 features	95.98%
[23]	2022	1685 Images from Arabica Dataset	Few Shot Learning using Mobile Net v2	96%
[22]	2022	1120 Images from Private Dataset	Transfer Learning Mobile Net and ResNet-50 models used for classification.	97.01 and 99.89%

TABLE 2. Biotic stress description.

Biotic Stress	Description
Cercospora	Fungal pathogen that causes leaf spot. Appears as small circular or angular spots on leaves.
Miner (Peri Leucoptera coffee)	Insect larva that tunnels within leaves of coffee plant. Causes Serpentine trails or blotchy patterns.
Phoma (Phoma costaricensis)	Genus of Fungi, causes development of lesions spots on leaves, stems or fruits.
Healthy	Free from visible signs of damage.
Rust (Hemileia vastatrix)	Fungal disease that causes formation of reddish brown/ orange pustules on leaves, stems or fruits.

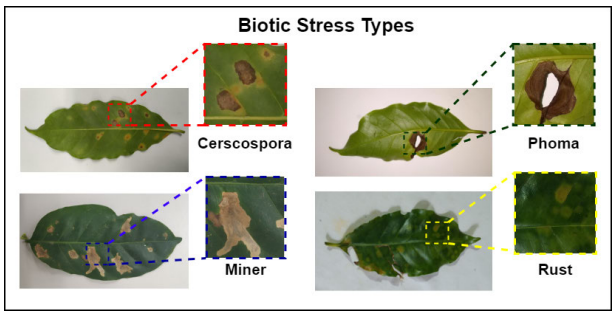


FIGURE 1. Biotic stress types for coffee leaves.

stress class. The steps to obtain new refined dataset D' are given in Algorithm 1.

2) TRAIN AND VALIDATION SPLIT

The newly refined dataset D' given in Table 2. is subjected to a stratified split resulting in a training dataset D_t and Validation dataset D_v with a split ratio q of 80 : 20 is given.

$$D_t = q \times size(D') \tag{1}$$
$$D_v = (1 - q) \times size(D') \tag{2}$$

where,
q = Desired proportion (0 < q < 1)
D_t = Training split
D_v = Validation split
D' = Parent Dataset

Algorithm 1 Algorithm for Initial Dataset Refining

```
    Input: Datasets list [D1, D2, D3]
    Output: Refined Dataset D'
1   Unique Set → Create an empty set to store unique samples
2   for set in Input // Loop over each dataset
3       for images in Sample set // Looping over images in datasets
4           Instance check → Check multiple instance of each image file
5           Append → Add only a single instance to Unique Set
6       End
7   End
8   Class Segregation → segregate images in Unique Set based on class name
9   Instances → Count image instances in each class
10  Sample Size → Define under-sampling size
11  New Dir (D')→ Create an empty directory with all class sub-directories
12  for image in each class // Loop over images in each segregated class
13      if(i < sample size) // run the loop till the defined under-sample length
14          Append → Add samples to identical empty class directory in D'
15          Increment → increment i by one after each append
16  End
```

Training Dataset D_t is subjected again for another stratified split, where the cardinality of the D_t split is given by the size of D_s and D'_s referring to segmented and non-segmented segregation given by

$$D_s = p \times size(D_t) \tag{3}$$
$$D'_s = (1 - p) \times size(D_t) \tag{4}$$

where,
p = Proportion of data to be allocated ranging (0 < p < 1)
D_t = Training dataset
D_s = Data split for segmentation pipeline
D'_s = Non-segmented dataset

3) SEGMENTATION

The second split of D_t is designed to emphasize the foreground for efficient feature extraction through segmented inputs and improve the model's resilience to background noise using non-segmented data [24]. The samples within the D_s split are segmented using a series of image processing algorithms to subtract the background region and retain only the inherent leaves and their corresponding biotic stress characteristics.

TABLE 3. Dataset description.

Biotic Stress	Raw Data			Refined Data	Memory On Disk		Splits		Channel
Type	Bracol	JMuBEN	PDCMD	RCD	Avg. File	Class	Train	Val	Type
Phoma	430	6571	–	576	0.05MB	38.2MB	453	113	RGB
Rust	360	8336	442	600	0.5MB	681MB	480	120	RGB
Miner	–	17000	–	614	0.45MB	675MB	491	123	RGB
Healthy	467	19000	439	600	0.23MB	292MB	480	120	RGB
Cercospora	490	7681	55	600	0.1MB	61.4MB	480	120	RGB

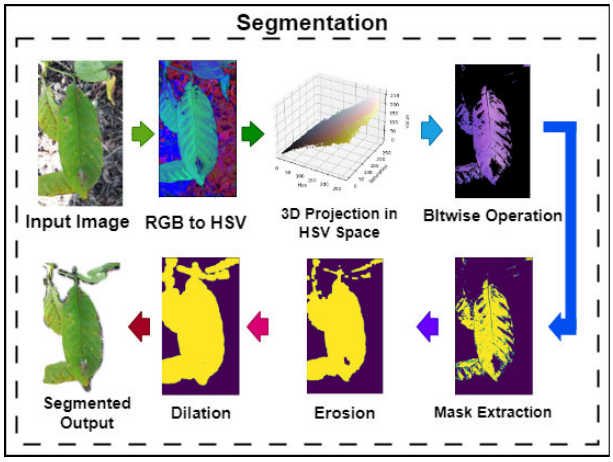


FIGURE 2. Image segmentation pipeline.

For a sample $D_s i$ in the set D_s with pixel values in (x,y) plane, the segmentation process is initiated with the conversion of the respective sample in HSV color space [25] and extracting required Hue, Saturation and Value components, using a threshold t_h followed by Bit-wise AND operation to create a mask. Morphology operations are performed with a structure element to constrict and extend the mask as needed as seen in the Figure 2. The step-wise implementation of the background subtraction pipeline in given in Algorithm 2. For a few image samples with significant background complexity, guided segmentation approach was employed using Orbit Background Removal tool [25]. Later the segmented set D_s is merged with the un-segmented counterpart D'_s.

$$D_f = D_s + D'_s \tag{5}$$

where,
D_f = Proposed dataset for feature extraction

C. FEATURE EXTRACTION

1) PRE-PROCESSING FOR HAND CRAFTED AND CNN FEATURE EXTRACTION

The dataset D_f is then propagated to a feature extraction pipeline, consisting of hand-crafted techniques and pre-trained convolutional neural networks. For the hand-crafted feature extraction techniques, Histogram of Oriented Gradients (HOG) [26], Gray-Level Co-occurrence Matrix(GLCM) [27], Local Binary Patterns (LBP) [28] and Gabor Filters [29] are employed. Whereas for CNNs based feature extraction, state-of-the-art techniques inclusive of but not limited to MobileNet V3 – Large [30], Xception [31], VGG-16 [32] and the likes are used.

Prior to being subjected for feature extraction, the dataset D_f is processed for each of the respective feature extraction method such that the samples are gray-scaled and re-sized to 128×256 for hand crafting based extraction and 224×224 for CNN based techniques. The specified size ratio of 1:2 is preferred for Hand Crafted Extractions owing to a similar input's size ratio in HOG research paper [26] while the 224×224 input resolution for CNN feature extraction provides an optimal trade-off between computational cost and feature representation and is generically standardized.

The dataset D_f intended for hand-crafted features extraction under-goes a Gaussian filtering [33] and brightening stage given as under.

$$G_{(x,y)} = \frac{1}{2\Pi\sigma^2 e^{\frac{-(x^2+y^2)}{(2\sigma^2)}}} \tag{6}$$

where,
$G_{(x,y)}$ = Gaussian filter value at (x,y)
σ = standard deviation

$$B_{o(x,y)} = B_{i(x,y)} \times k \tag{7}$$

here,
$B_{o(x,y)}$ = Output Image's pixel intensity at coordinates (x,y)
$B_{i(x,y)}$ = Input Image's pixel intensity at position (x,y)
k = Brightness scaling factor

2) HAND CRAFTED FEATURE EXTRACTION

Hand Crafted Feature Extraction involves carefully extracting or designing specific visual features from the input data, most often from raw images. By meticulously analyzing the patterns and characteristics from the input data, features of interest and defining nature are derived. These features may involve edges, textures, colors, shapes or other visually relevant properties. Each technique extracts unique characteristics that prove vital in robust discerning of the object vaguely illustrated in Figure 4.

a: HISTOGRAM OF ORIENTED GRADIENTS (HOG)

For proposed dataset D_f having n samples and size (M × N), where each sample I has pixels values positioned at (x,y), the mathematical representation for extraction of HOG

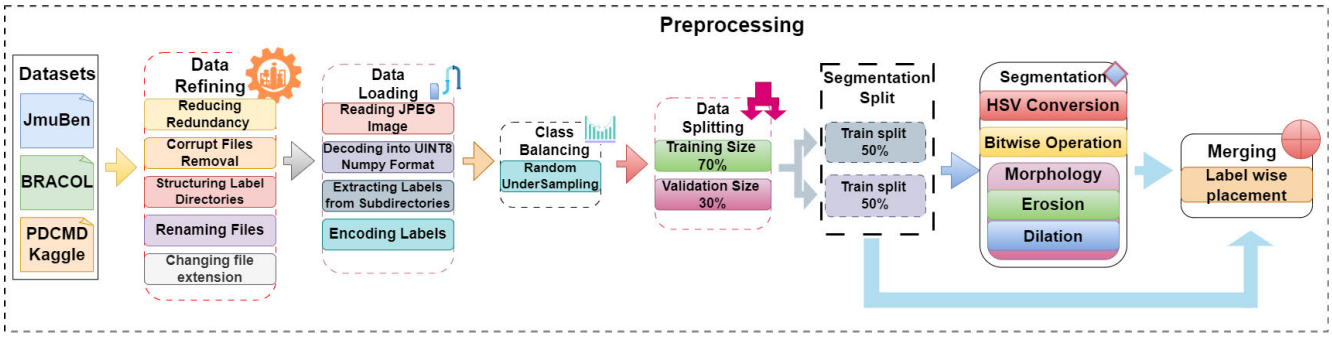


FIGURE 3. Data pre-processing pipeline.

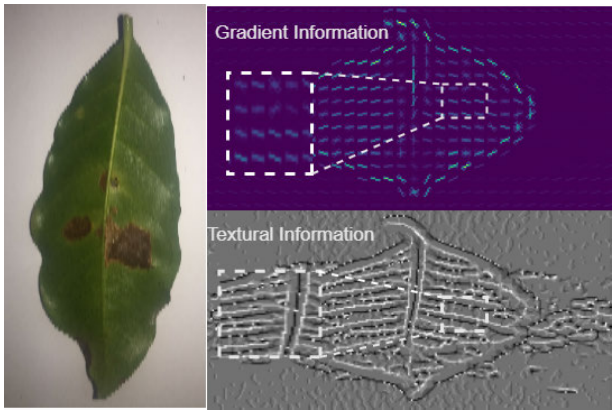


FIGURE 4. Hand crafted gradient and textural features.

descriptors for I is given as.

$$G_{mag(x,y)} = \sqrt{(I_{(x+1,y)} - I^2_{(x-1,y)})} + \sqrt{(I_{(x,y+1)} - I^2_{(x,y-1)})} \quad (8)$$

The gradient orientation given gradient magnitudes (G_x, G_y) is represented below.

$$\Theta = tan^{-1}2\frac{G_y}{G_x} \quad (9)$$

here,
$G_{mag(x,y)}$ = Gradient magnitudes for pixel at (x,y)
G_x, G_y = Magnitudes at x and y axis
$tan^{-1}2$ = Arc-tangent function
The normalization factor N_f is given as

$$S_{(mag)} = \sum_{i=0}^{Bx}\sum_{j=0}^{By} G_{mag(i,j)^2} \quad (10)$$

$$h = \sqrt{S_{mag} + \epsilon} \quad (11)$$

S_{mag} = square of the magnitude of gradients
ϵ = value added for preventing undefined values
The final vector of feature descriptors is.

$$H_1 = [h_{(1,1)}, h_{(1,2)}, \ldots, h_{(y,z)}] \quad (12)$$

Algorithm 2 Algorithm for Image Segmentation

```
Input: Image from Data-split Ds
Output: Segmented Image for Dataset Ds
RGB to HSV Convert image from RGB color space to HSV color space
Components Extract Hue Saturation and Value Components from image
HSV Projection Project HSV space in 3D to determine threshold values
Threshold Define upper and lower threshold for each channel
for (x,y) in HSV_image // loop over each pixel in HSV image sample
    if ((x,y) in threshold range) // check for the pixel value in threshold range
        Set value to 1 // If in range set value to 1
end
Mask Create empty mask of size HSV_image
for (x,y) in HSV_image // loop over post-threshold HSV_image
    if (x,y) and Mask(x,y) is not 0 // check if (x,y) for both is not zero
        Mask(x,y) = 255 // set empty mask pixel to max value
end
Structure Element // Define a structure element i.e., square, circle
//Morphology Operation On Mask
//Erosion and Dilation
for (x,y) in Mask // Iterate over mask pixels
    for (i,j) in Structure Element // iterate over pixels in structure element
    // Erosion
        if (x,y) is 1 // check if pixel in mask is 1
            if neighbor pixel equals to 1 // check if neighbor pixel has value 1
                set (x,y) to 1 else set to 0 // set the mask value to 1
    // Dilation
        if (x+i, y+j) in range(size(Mask)) // check if in bound of Mask
            set (x+i, y+j) to 1 // set the respective pixel to 1
    end
end
// Applying mask on original image
Segment Create a empty image of size(HSV_image) for segmented output
for (x,y) in RGB_image // loop over original RGB image
    if (x,y) in Mask is 1 // check if pixel in Mask is at value 1
        set segmented_image(x,y) to RGB_image(x,y)
    else
        set segmented_image(x,y) to 255 // set 255 for white background
end
```

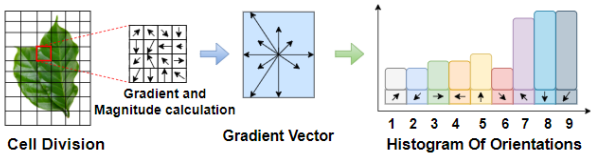


FIGURE 5. Histogram of oriented gradients (HOG) based feature extraction.

b: GRAY LEVEL CO-OCCURRENCE MATRIX(GLCM)

Similarly, the Dataset Df duly pre-processed for the hand-crafted feature extraction is passed through a GLCM based

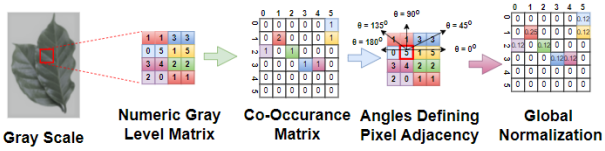


FIGURE 6. Gray level co-occurrence matrix (GLCM) based feature extraction.

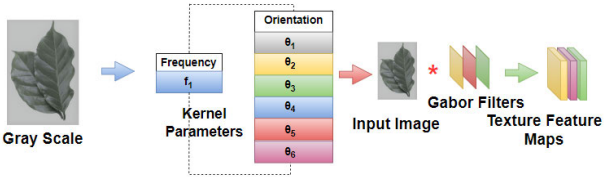


FIGURE 7. Gabor Filter Based feature Extraction.

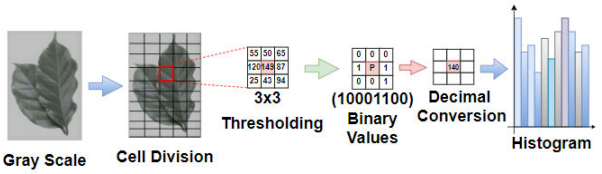


FIGURE 8. Local binary patterns (LBP) based feature extraction.

feature extraction scheme to extract the features. For an image sample I with pixels values in gray-scale given by (x,y) having size (M x N), the mathematical representation shown as.

$$G(I_{(x,y)}, I_{(x+d_x,y+d_y)}) = G(I_{(x,y)}, I_{(x+d_x,y+d_y)}) + 1 \quad (13)$$

The normalized matrix G_{norm} is given as

$$G_{norm(i,j)} = \frac{G_{(i,j)}}{T} \quad (14)$$

T = Number of valid pixel pairs

Normalized Matrix G_{norm} serves as a foundation to compute various descriptive features such as Entropy, Correlation, Energy, Homogeneity and Contrast.

c: GABOR FILTERS

For Gabor feature extraction from Input Image $I \rightarrow D_f$ prior pre-processed for hand-craft techniques, having pixel values positioned at (x,y).

$$G(x, y) = e^{\frac{-(x'-x^o)^2+(y'-y^o)^2)}{(2\sigma^2)}} cos(2\Pi f(x' - x^o) \quad (15)$$

where,
(x', y') = coordinates in spatial domain
(x_o, y_o) = center of Gabor filter
σ = width control for Gabor envelope
f = frequency of sinusoidal plane wave

Assuming we have a gray scale Image I with pixel values positioned at (x,y), the convolution operation C(x,y) between the image and the Gabor filter is given ∋

$$C(x, y) = \sum\sum I(u, v)G(x - u, y - v) \quad (16)$$

where,
(u,v) = coordinates within the neighborhood around the pixel

For each filter response (x,y), the magnitude M(x,y) is given as

$$M(x, y) = \sqrt{G(x, y)^2 + jG(x, y)^2} \quad (17)$$

where,
jG(x,y) = imaginary part of filter at coordinates (x,y)

By leveraging the magnitude, we can compute meaningful features such as Energy E_n and Entropy E

d: LOCAL BINARY PATTERNS(LBP)

For the fourth hand-crafted feature extraction technique, Local Binary Pattern (LBP) is employed. The post-processed dataset D_f, having sample Image I with pixel coordinates at (x,y) is subjected to LBP.

$$B_i = \begin{cases} 1 & \text{if } p_i < c \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

where,
B_i = Binary value of i^{th} neighboring pixel
P_i = intensity value of i^{th} neighboring pixel
c = central pixel

The direction (clockwise/anti-clockwise) invariant value of LBP for the pixel(x,y) is calculated as.

$$LBP_{(x,y)} = \sum_{i=0}^{N-1} B_i \times 2^i \quad (19)$$

N = number of neighboring pixels

Assuming we have K number of pixels within the image, the histogram H(c) for the LBP codes for each pixel is calculated by a bin increment of 1.

$$H(c) = H(c) + 1 \quad (20)$$

The normalized value of Histogram H_{norm}(c) is calculated as given

$$H_{norm}(c) = \frac{H(c)}{K} \quad (21)$$

3) CNN FEATURE EXTRACTION

Following the hand crafted feature extraction, the pre-processed Dataset D_f re-sized and normalized is fed to each of the below mentioned networks for feature extraction.

a: MOBILE-Net v3 LARGE

The stem block of MobileNet V3 [30] consists of a convolutional layer denoted as conv. Here the learnable filters capture the spatial information from input I→D_f. The conv

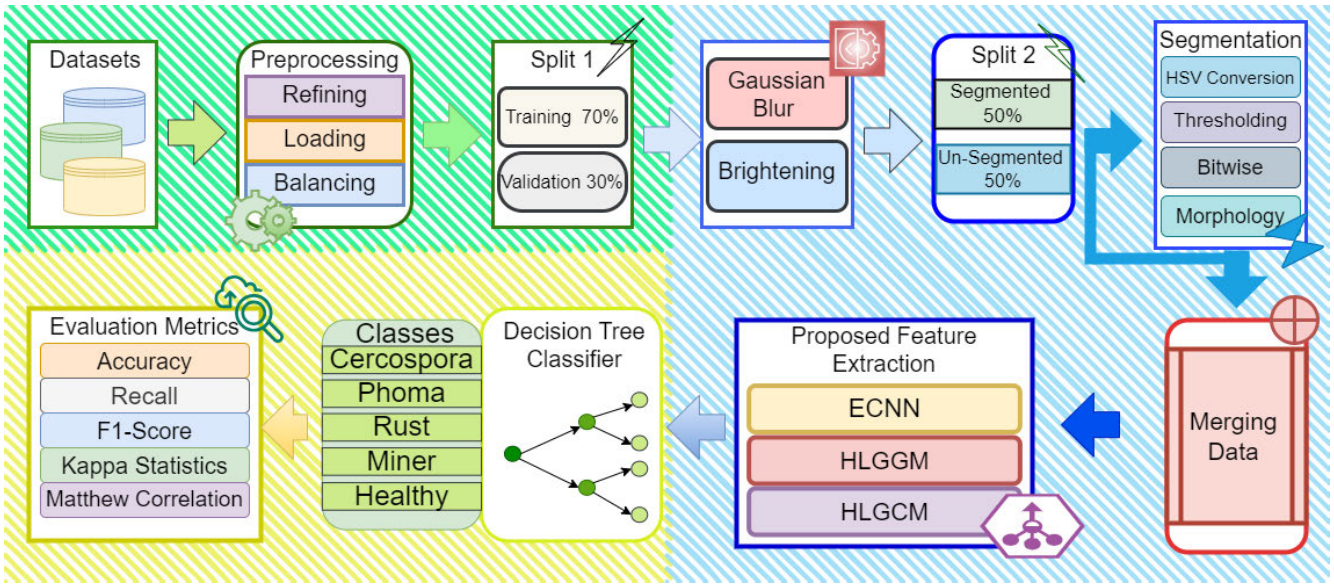


FIGURE 9. General block diagram inclusive for biotic stress classification.

layer is followed by a batch normalization (BN) layer shortly succeeding a ReLU activation layer.

$$Block_{stem} = Conv([B_{norm}[ReLU\,Conv(I)]]) \quad (22)$$

Based On the concept of depth-wise separable convolutions, the large blocks of the model also referred to as inverted residual blocks consist of separable convolutions, squeeze-and-excitation operations followed by a linear bottleneck given as.

$$Block_{inv-res} = L_{bottleneck}([Depth_{sep}[X_{stem}]]) \quad (23)$$

The feature maps obtained are pooled using G.Avg Pooling operation to compute the average of each feature map in the last layer for a latent feature vector.

$$Pool_{(avg.global)}(Block_{(res-inv)}) \quad (24)$$

The feature vector obtained from MobileNet V3 is given as under.

$$C_1 = [m_1, m_2, \ldots, m_n] \quad (25)$$

b: XCEPTION

These entry blocks also termed as separation block in Xception [31] consist of depth-wise separable convolutions, batch normalization, and ReLU activation. Residual or skip connections are used to connect the inputs of each block to the outputs. For the input sample $I \rightarrow D_f$, the Separation block is given by.

$$Block_{sep} = I + Depth_{sep}([B_{norm}[ReLU[Depth_{sep}(I)]]]) \quad (26)$$

The middle blocks consist of repeated separable convolution operations with residual connections.

$$Block_{middle} = Block_{sep} + Sep_{conv}([B_{norm} \times [ReLU[Sep_{conv}(Block_{sep})]]]) \quad (27)$$

Similar to the entry blocks, the exit blocks consist of depth-wise separable convolutions, batch normalization, and ReLU activation represented as

$$Block_{exit} = Block_{middle} + Depth_{sep}(B_{norm}[ReLU[Depth_{sep}(Block_{middle})]]) \quad (28)$$

The head block consists of a G.Avg Pooling Operation to compute the average of each feature map for a latent vector given as

$$Pool_{(avg.global)}(Block_{exit}) \quad (29)$$

The feature vector obtained from Xception is given by

$$C_2 = [x_1, x_2, \ldots, x_3] \quad (30)$$

c: DenseNet-169

The dense block in DenseNet-169 [34] consists of a dense stack of bottleneck layers consisting of a 1×1 convolution followed by a 3×3 convolution. Each dense block's output is concatenated with the input and fed into the next dense block. For a sample input image $I \rightarrow D_f$, the dense block is given by

$$Block_{dense} = Conv([(I), Layer_{dense}[Conv[Conv(I)]]]) \quad (31)$$

The transition block followed by a dense block is intended to reduce the spatial dimensions and number of channels using a 1×1 convolution followed by average pooling.

$$Block_{trans} = Pool_{avg.global}([Conv(Block_{dense})]) \quad (32)$$

The Head block consists of a G.Avg Pooling operation to compute the average of each feature map and output a latent feature vector.

$$Pool_{(avg.global)}(Block_{trans}) \quad (33)$$

The feature vector obtained from DenseNet-169 is given by

$$C_3 = [d_1, d_2, \ldots, d_3] \quad (34)$$

d: VGG-16

Convolutional blocks in a VGG-16 [32] architecture consist of convolutional layers with 3×3 filters, followed by a batch normalization and ReLU activation function. After every two subsequent convolutions a max pooling is performed to reduce the spatial dimension. For a sample image $I \rightarrow D_f$, the convolutional block is given as

$$Block_{conv} = Pool_{max}([ReLU[B_{norm}[Conv([ReLU \times [B_{norm}Conv(I)]])]]]) \quad (35)$$

A G.Avg Pooling is performed on the feature maps obtained to output a latent feature vector for classification $\in$

$$Pool_{(avg.global)}(Block_{conv}) \quad (36)$$

The feature vector obtained from VGG-16 is given by

$$C_4 = [v_1, v_2, v_3, \ldots, v_4] \quad (37)$$

e: ResNet - 152

The ResNet-152 [35] as the name suggests is built upon the notion of residual connections to mitigate the vanishing gradients originating within the deep networks. The skip connection or residual block consist of convolutional layers with batch normalization and ReLU activation. Each residual block has a skip connection that directly adds the input to the outputs. For the dataset D_f, an input image sample I is subject to the network such that.

$$Block_{(residual)} = I + Conv([B_{norm}[ReLU[Conv(I)]]]) \quad (38)$$

The feature maps are the pooled via G.Avg Pooling to compute a latent representation of features given as

$$Pool_{(avg.global)}(Block_{residual}) \quad (39)$$

The feature vector obtained from ResNet-152 is given by

$$C_5 = [r_1, r_2, r_3, ..r_n] \quad (40)$$

4) PROPOSED METHODOLOGY

Convolutional Neural Network learns hierarchical representation of the data through a series of convolutions, pooling and non-linear activations. These networks au-tomatically extract complex features and abstractions from raw data without much pro-cessing, adding to the generalizing capability of these powerful models. On the other side, hand-crafted features are engineered to capture certain local details and patterns including shape, texture, color and statistical features giving us the ability to extract and interpret specific characteristics of data. Convolutional Neural Networks may not explicitly capture certain local variations and details as they excel in gathering a global context [36], so by incorporating hand-crafted features along with the convolutional features in an ensemble – we can ensure comprehensive representation of the data and a more explainable ap-proach towards the classification process [37].

The ensemble of features is not confined to an interoperability of CNN and Hand Crafted techniques, rather it can be extended to an information transfer between features obtained through varying convolutional architectures. Hence, by leveraging each model's hierarchical feature representations and varying levels of abstractions, an interplay between different convolutional network features can result in a robust classification model [38]. Inspired by the concept of ensemble features, we proposed three methodologies to enhance the classification performance and robustness of our model as defined in Figure 3. These tabular description of proposed feature extraction methodologies and their combinations are given in Table 4 and 5 respectively.

The Table 4 represents the statistical descriptions for each proposed feature extraction technique. Each statistical descriptive characteristic namely Mean, St. Deviation, Min and Max is computed for features extracted from an individual image instance and then cross computed for all instances of the images. This approach is repeated for each of the three feature extraction methodologies proposed.

Similarly, the feature count for each technique is given in Table 5 representing the number of features taken from each of the corresponding CNN and Hand crafted techniques. Finally the total feature count for each proposed methodology is show in the respective table.

a: ECNN

The first proposed methodology solely relies on the feature vectors obtained via Convolutional Neural Networks. The feature vectors extracted by processing the dataset D_f through each of the five state-of-the-art transfer learning models are concatenated to form a final feature vector. The concatenated features result in an enhanced representation power by capturing diverse patterns. Moreover, the ensemble of feature extraction adds to the discriminative power of the feature representation while enhancing the robustness to the variations in the input data. The finally acquired ensemble feature vector is classified using a decision tree based classifier as show in Figure 4.The mathematical expression for the ensembled MobileNet v3-Large, Xception, DenseNet-169, ResNet-152 and VGG-16's feature vector is given.

$$F = \sum_{i=1}^{5} C_i \quad (41)$$

where,

F = Concatenated feature vector

f_i = Feature vector obtained through each of the five CNN models.

b: HLGGM

The second proposed methodology incorporates an ensemble of hand crafted features and dimensionality reduced MobileNet features. Prior to hand crafted features extraction from the respective dataset D_f, the data was denoised using Gaussian filter [36] with a succeeding brightening stage

TABLE 4. Aggregate feature description for all proposed methodologies.

Aggregate Feature Description					
Method	Description for single image features	Description for all Image Instances			
		Mean	St. Deviation	Min	Max
HLGGM	Mean	2.951374e+00	9.372308	-13.728154	25.784746
	Std.	2.712132e+01	70.717053	119.780772	146.363182
	Min	-4.88393e-14	0.000354	-1191.5015	0.001000
	Max	2.699146e+02	688.346125	19.541667	1191.502457
HLGM	Mean	1.273070e-01	0.163618	-0.089744	0.792408
	Std.	2.085468e+00	5.453015	9.224174	11.415857
	Min	-9.76636e-14	0.051247	-1191.5015	0.354345
	Max	2.699146e+02	688.346125	19.541667	1191.502457
ECNN	Mean	1.933298	1.798352	0.003508	14.011202
	Std.	15.368003	9.379808	0.249384	42.975283
	Min	0.000	0.00	-0.288517	0.00
	Max	1191.50	688.345843	13.628227	2383.0

TABLE 5. Feature combination for each proposed methodology.

Proposed Methods	CNN Features					Hand Crafted Features				Processed Features		Feature Count
	C_1	C_2	C_3	C_4	C_5	F_1	F_2	F_3	F_4	P_M.N	H_C	
ECNN	✓	✓	✓	✓	✓							7232
HLGGM						✓	✓	✓	✓	✓		16780
HLGCM							✓	✓	✓	✓	✓	99
	960	2048	1664	512	2048	16761	5	2	2	10	80	

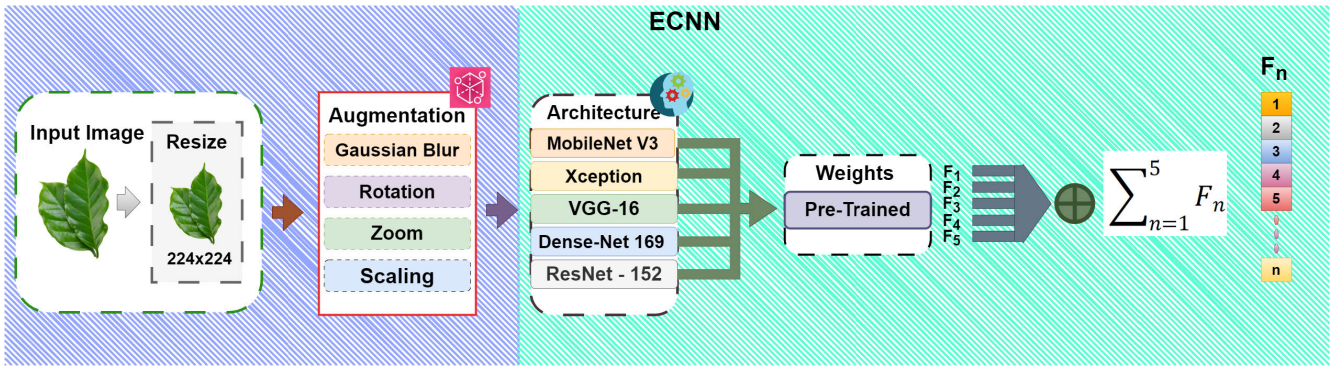


FIGURE 10. Block diagram of first proposed methodology (ECNN).

for added prominence. Hand Crafted feature extraction was then carried using Histogram Of Oriented Gradients (HOG), Linear Binary Patterns (LBP), Gray Level Co-occurrence Matrix(GLCM) and Gabor Filters in an attempt to extract the defining and descriptive properties of the data.

For the feature extraction using Convolutional Neural Network (CNN) a state-of-the-art fine-tuned model Mobile-Net v3 Large was used. The extracted feature vector from the CNN based architecture was reduced using a Principal Component Analysis(PCA) based approach resulting in a lower dimensional representation. The motive was to alleviate the curse of dimensionality and improved computational efficiency, by retaining only the most informative components. The final feature vector was then propagated towards a decision tree classifier for obtaining prediction values as illustrated via Figure 5. The mathematical expression of the aforementioned methodology is given as:

$$F_1 = [h_1, h_2, h_3, \ldots, h_n] \tag{42}$$
$$F_2 = [g_1, g_2, g_3, \ldots, g_n] \tag{43}$$
$$F_3 = [gr_1, gr_2, gr_3, \ldots, gr_n] \tag{44}$$
$$F_4 = [l_1, l_2, l_3, \ldots, l_n] \tag{45}$$
$$H = \sum_{i=1}^{4} F_i \tag{46}$$

where,
H = Concatenated Hand Crafted feature vector.
F_1 = HOG features
F_2 = GLCM features
F_3 = Gabor features
F_4 = LBP features
Similarly the Principal Component Analysis based Dimensionality reduced Mobile-Net v3 features are given by:

$$M.N = [n_1, n_2, n_3, \ldots, n_n] \tag{47}$$
$$X_c = X - \mu \tag{48}$$
$$\sum = \frac{1}{nX_c^T X_c} \tag{49}$$
$$\sum = V \times D \times V^T \tag{50}$$

$$P = [p_1, p_2, p_3, p_4, \ldots, p_n] \tag{51}$$

$$P_{(M.N)} = X_c P \tag{52}$$

Algorithm 3 Algorithm for ECNN Feature Extraction

```
    Input: Image → Df
    Output: Concatenated Feature Vector F
1   Resize Resize the input image to 224 × 224
2   Normalize Normalize the Pixel Values between 0 and 1
3   Feature Vector F Initialize an Empty Feature Vector [ ]
4   Extraction Use 5 Pre-trained CNN Models for feature extraction
5   for (i =0 to 4) // loop over all feature vectors obtained by 5 CNN models
6       Append(CNN model feature[i]) to Feature Vector F
7   End
8   Classification : Use Decision Tree for final Classification
```

where,
M.N = Mobile Net features
μ = mean vector of dataset D_f
X_c = Centered feature matrix
$\sum$ = co-variance matrix of the dataset
V = Eigen vector matrix
D = Diagonal Eigen Value matrix
P_{MN} = Projection Matrix of Eigen values
The resultant feature matrix of this proposed methodology is given as.

$$FF = H + P_{M.N} \tag{53}$$

Algorithm 4 Algorithm for HLGGM Feature Extraction

```
    Input: Image → Df
    Output: Concatenated Feature Vector FF
1   Resize 1 Resize the input image to 224 × 224 for Mobile-Net v3
2   Normalize Normalize the Pixel Values between 0 and 1
3   Fine Tune: Freeze top 150 layers of MobileNet for retaining weights
4   Extraction Use fine-tuned Mobile-Net for feature extraction
5   PCA Vector PMN Initialize feature vector for PCA reduced vector M.Net
6   PCA Apply PCA on Mobile Net Features to reduce the features to 10
7   Append(PCA) to Feature Vector PMN
8   Resize 2 Resize input image to 128 × 256 for hand crafted feature extraction
9   ProcessApply Gaussian filter and enhance brightness of image
10  Vector H Initialize an empty vector H for storing hand crafted features
11  Vector FF Initialize an empty vector FF for final concatenated vector
11  for ( i in all Hand Crafted Techniques)//Iterate over hand crafted features
12      Append(hand crafted features[i]) to vector H
13  End
14  Concatenate[H , PMN] and Append to FF
15  ClassificationClassify final vector FF with Decision Tree Classifier.
```

c: HLGCM

The third proposed methodology centers around the computational efficiency and the dimensionality curse of the feature vectors obtained from both the hand crafted and the convolutional methods. The proposed approach tends to reduce the dimensionality of HOG descriptors and MobileNet features using an effective feature selection approach. Using a Gaussian filter followed by a brightness enhancement stage, the dataset D_f is passed through a hand crafted feature extraction pipeline where techniques like Histogram of Oriented Gradients (HOG), Linear Binary Patterns (LBP), Gray Level Co-occurrence Matrix (GLCM) and Gabor Filters are used. Prior to being concatenated with hand crafted features obtained from other aforementioned techniques, the dimensionality of the HOG descriptors is considerably reduced using a chi square test while the features of fine-tuned Mobile-Net v3 were reduced using Principal Component Analysis (PCA).

The resultant feature vector was then fed into a decision tree based classifier for the estimation of class probability values as shown in Figure 6. The aforementioned feature reduction substantially decreased the computational costs associated with classifying HOG descriptors and Mobile-Net features by only selecting the most informative components. This dimensionality reduced ensemble of hand-crafted and convolutional features offer a novel approach towards the classification process at hand by surpassing previous state-of-the art convolutional architectures in terms of both the computational efficiency and the classification score. The chi squared based feature selection of HOG descriptors is given by

$$chi^2(x, y) = \sum \frac{(F_1 - E)^2}{E} \tag{54}$$

$$F_1' = F_1[:, K_{top}] \tag{55}$$

where,
F_1 = Hog descriptor from eq (28)
Y = class labels
E = Expected frequencies based on F_1 and Y observed frequencies.
K_{top} = Indices of largest elements
F_1' = Top K features obtained from HOG descriptors after chi2 statistics
After chi2 statistics, the final hand crafted feature ensemble is given as

$$H_c = [\sum_{i=2}^{4} F_i] + F_1' \tag{56}$$

The final feature vector is given by

$$FF' = P_{M.N} + H_c \tag{57}$$

where,
$P_{M.N}$ = PCA based MobileNet v3 features taken from eq (52)
H_c = chi square based ensemble of hand crafted features
FF' = Final feature vector consisting of CNN and Hand Crafted ensemble.

5) EVALUATION METRICS

The evaluation metrics are used to assess the performance and efficacy of many ma-chine and deep learning models. In the context of multi-class classification of various diseases, evaluation metrics provide a deep and thorough insight on not only the general performance but also class-wise performance. Helping in assessing the model's inclining disposition and bias towards certain classes over others. Some of the metrics considered for assessing the performance of aforementioned techniques are given below:

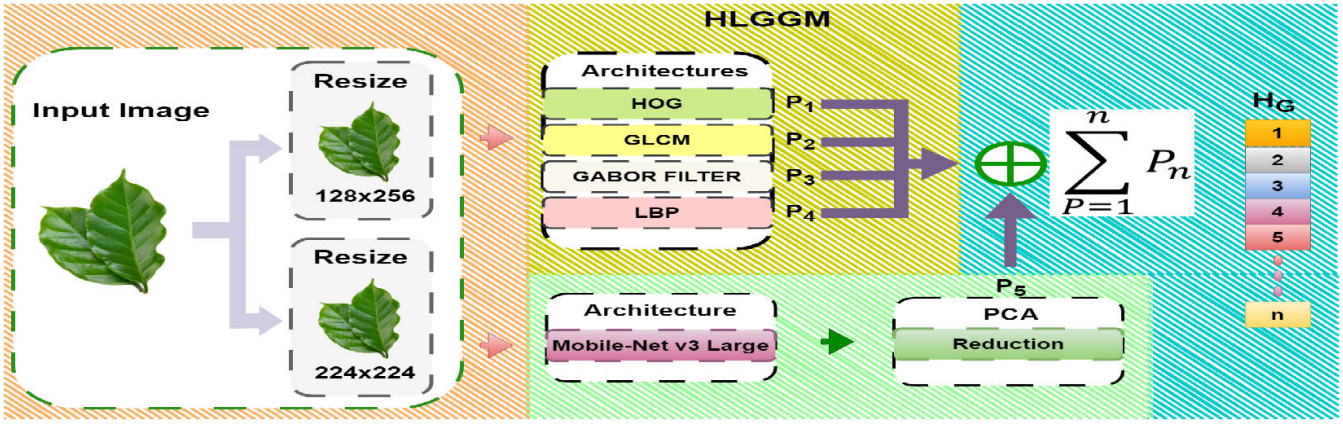


FIGURE 11. Block diagram of second proposed methodology (HLGGM).

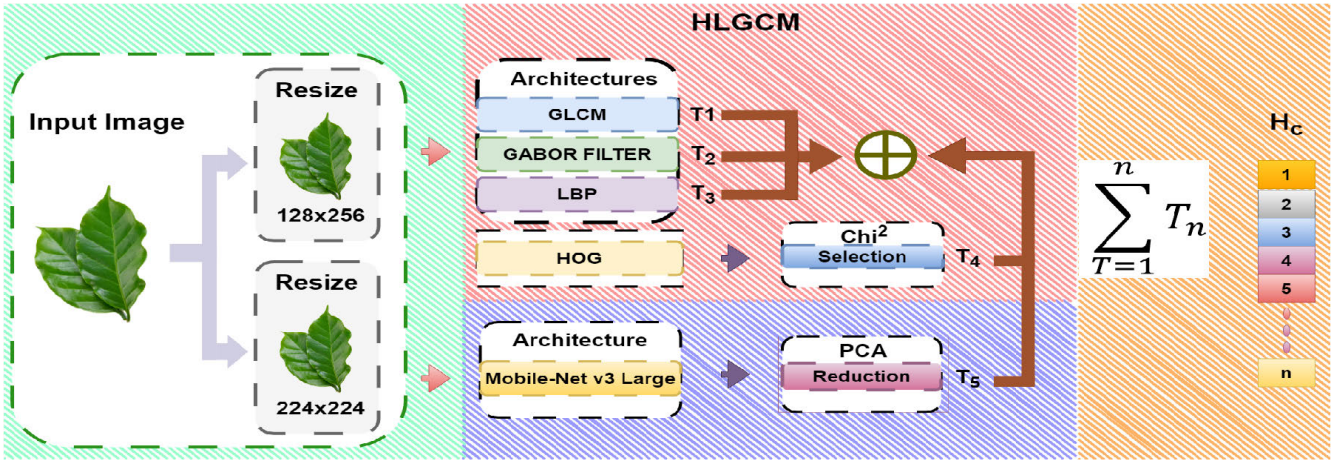


FIGURE 12. Block diagram of third proposed methodology (HLGCM).

Algorithm 5 Algorithm for HLGCM Feature Extraction

Input: Image → Df
Output: Concatenated Feature Vector FF'
Resize 1 Resize the input image to 224 × 224 for Mobile-Net v3
Normalize Normalize the Pixel Values between 0 and 1
Fine Tune: Freeze top 150 layers of MobileNet for retaining weights
Extraction Use fine-tuned Mobile-Net for feature extraction
PCA Vector PMN Initialize feature vector for PCA reduced vector M.Net
PCA Apply PCA on Mobile Net Features to reduce the features to 10
Append(PCA reduced components) to Feature Vector PMN
Resize 2 Resize input image to 128 × 256 for hand crafted feature extraction
ProcessApply Gaussian and brightness filter
Vector Hc Initialize an empty vector Hc for storing hand crafted features
Vector F1'Initialize an empty vector for storing Chi2 Based HOG Features
Vector FF' Initialize an empty vector FF' for final concatenated vector
Chi Square Apply Chi2 test on HOG features to reduce to 80 components
for (i in all Hand Crafted Techniques)//Iterate over hand crafted features
    Append(hand crafted features[i]) to vector Hc
End
Concatenation: Concatenate[Hc, PMN] and Append to FF
ClassificationClassify final vector FF with Decision Tree Classifier.

a: PRECISION

Precision provides the insight about the instances that were correctly predicted i.e., (True Positives) against all predictions marked as Positive i.e., (True Positives + False Positives) given mathematically as

$$Precision(PR) = \frac{T_{Positive}}{T_{Positives} + F_{Positives}} \times 100 \quad (58)$$

b: RECALL

Recall evaluates the model's ability to avoid False Negatives. Recall is given by a ratio of True Positives out of all positively marked instances i.e., (False Positives + True Positives). Mathematically Recall is given as.

$$Recall(RC) = \frac{T_{Positive}}{(T_{Positives} + F_{Positives})} \times 100 \quad (59)$$

c: ACCURACY

The most basic of evaluation matrices is Accuracy, as it determines the correctly classified instances over a total of all instances. Accuracy can be represented as.

$$Accuracy(AC) = \frac{T_{Pos} + TNeg}{T_{Pos} + TNeg + F_{Pos} + FNeg} \times 100 \quad (60)$$

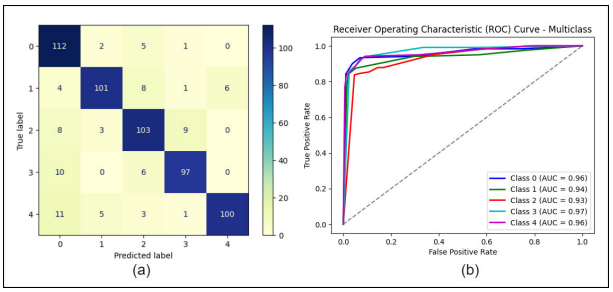


FIGURE 13. (a) Confusion matrix for Mobile-Net v3-Large (b) ROC curve for Mobile-Net v3-large.

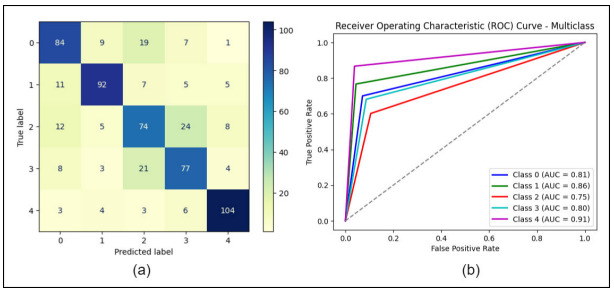


FIGURE 14. (a) Confusion matrix for Xception (b) ROC Curve For Xception.

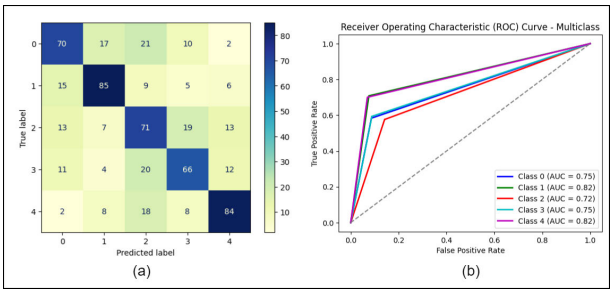


FIGURE 15. (a) Confusion matrix for Dense-Net 169 (b) ROC curve for Dense-Net 169.

d: F1-SCORE

F1 Score provides a quantification of the balance between the precision and recall. In general, F1 Score can be described as the Harmonic mean of the two metrics. It proves useful in evaluating the performance of a class imbalanced data. It is given as.

$$F1 - Score(FS) = \frac{2(PR*RC)}{(PR + RC)} \times 100 \tag{61}$$

e: KAPPA STATISTICS

The Kappa Statistics also referred to as Cohen's Kappa Co-efficient is a statistical measure of the agreement between actual and predicted class labels. It accounts for the possibility of a occurrence by chance or a random occurrence. Mathematical representation is given by

$$Kappa_{stats}(KS) = \frac{(p_o - p_e)}{(1 - p_e)} \tag{62}$$

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. EVALUATION OF INDIVIDUAL CNN FEATURE EXTRACTION METHODS

The assessment of individual feature vectors obtained through each of the five state-of-the-art transfer learning based convolutional neural network was carried out on a decision tree classifier for the respective data set D_f. The pre-trained models used for feature extraction were Mobile Net v3 Large, Xception, Dense Net 169, Res Net 152 and VGG-16 respectively. The lowest accuracy obtained from the respective pre-trained networks was of Xception, lying around 63.08 % while the highest performing pre-trained model on the same dataset D_f in terms of accuracy was Mobile Net v3 Large giving an accuracy value of 86.07%. The varying degree of results obtained on the feature vectors of the size 960, 2048, 1664, 512 and 2048 suggest the network results not solely relying on the number of features obtained but also on the quality of the descriptors. Mobile-Net v3, ranking second to last among the aforementioned methodologies in terms of feature count, achieved the highest accuracy.

It can therefore be concluded that the inherent composition of a CNN architecture plays a crucial role in capturing the defining properties of the input. Adding to it, the play of feature count in determining the classification performance cannot be ruled out, considering a high number of descriptive features that may result in enhancing the model's performance albeit with some shortcomings i.e., computational costs as seen in Table 6 and susceptibility to over-fitting. The parametric configuration of the decision tree classifier for this experiment's setting is defined in the Table 7. The Figure 13 - 17, explicitly illustrate the performance of each of the above mentioned methods.

The results from the confusion matrices and ROC curves from each of the aforementioned feature extraction techniques suggest the difficulty of the models in distinguishing the different classes. Due to significant similarity of statistical, texture and color features of each disease class, the convolutional models appear to struggle. As seen in the confusion matrices, left diagonal depict the values that were truly predicted or in other term True Positives for a particular class. The values in above figures, seem to have a high misclassification rate as seen in the number of mis-classified instances lying in blocks apart from the diagonal. Similarly, the ROC curve above suggests the individual classification accuracy for each class by defining the Sensitivity or True Positive rate along the y-axis and False Positive rate along the x-axis. The classification performance of each CNN models for each class can be evaluated by observing the ROC curve as it treads away from the vertical axis. Apart from Mob‘‘ileNet, the Area Under Curve (AUC) for other CNN models is observed to have a lower value depicting the inefficient discerning capacities of the individual models.

The Table 6,illustrates the performance evaluation measures of each feature extraction technique. There are various

TABLE 6. Performance evaluation metrics for all implemented and proposed methodologies.

Feature-Extraction Methods	Evaluation Metrics							
	Precision	Recall	Accuracy	Gr.Matthew Corr.	F1 Score	Kappa Stats.	Computational Cost	
							Tr. Time	Pr. Time
MobileNet-V3 Large	0.8674	0.8607	0.8607	0.8272	0.8615	0.8258	0.1354	0.0107
ResNet 152	0.7247	0.7231	0.7231	0.6540	0.7236	0.6539	5.8525	0.0019
Dense-Net 169	0.7782	0.7734	0.7734	0.7173	0.7747	0.7168	4.6341	0.0018
Xception	0.6342	0.6308	0.6308	0.5387	0.6318	0.5383	2.4747	0.0021
VGG 16	0.6882	0.6845	0.6848	0.6060	0.6853	0.6055	1.0566	0.0009
HLGCM	0.9950	0.9949	0.9949	0.9937	0.9979	0.9937	0.0915	0.0022
HLGGM	0.9919	0.9916	0.9916	0.9895	0.9916	0.9895	20.5649	0.2446
ECNN	0.9348	0.9345	0.9345	0.9182	0.9346	0.9181	1.0657	0.0652

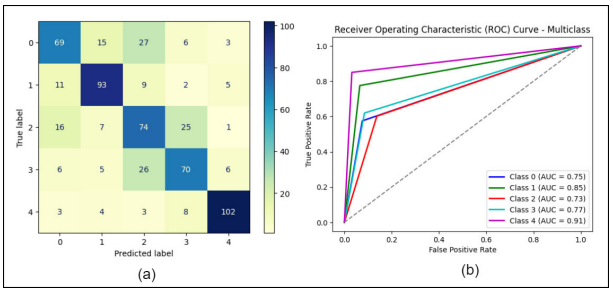


FIGURE 16. (a) Confusion matrix for VGG-16 (b) ROC curve for VGG-16.

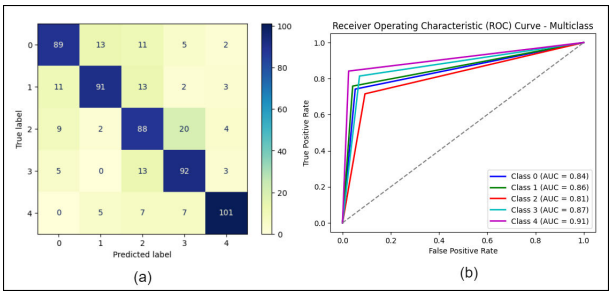


FIGURE 17. (a) Confusion matrix for ResNet-152 (b) ROC curve for ResNet-152.

evaluation metrics against which the performance was tested. Most individual CNN models didn't perform well with the least accuracy, precision and recall value at 63.4%,63.08% and 63.08% by Xception. Whereas, MobileNet landed at a testing 86.07% accuracy with 86.74% precision and 86.07% recall. Similarly, the computational cost for the MobileNet was 0.13 seconds for the training and 0.01 seconds for prediction – slightly lower then our proposed method HLGCM training time. The ECNN method achieved 93.45% in terms of accuracy while for Recall, Precision and F1 Score the results are 93.45%, 93.48% and 93.46% respectively. For HLGGM features, the precision was at 99.19%, accuracy, recall and F1-score at 99.16%. Lastly HLGCM features stood at the top of the evaluation metrics with the precision at a staggering 99.50%, accuracy and recall at 99.49% with the F1 Score and Kappa Statistics at 99.79% and 99.37%.

B. EVALUATION OF ECNN FEATURES

The Evaluation of feature vectors obtained through an effective concatenation of five state-of-the-art convolutional neural networks provide an optimistic overview of the technique. The model selection ensured different architectures are picked based on their varied feature extraction mechanisms including the depth-wise separable and residual connection based hierarchy. These variations in different architectures allowed a diversity of features. The individual classification for each of the respective CNN based features in the previous approach, stood at the highest accuracy of 86.07% derived from the Mobile-Net v3 features. Meanwhile, the concatenated feature approach namely ECNN soared the accuracy value from a model 86.07% highest to a new benchmark value of 93.45% on the corresponding decision tree classifier with a depth value of 250. The parametric definitions of the respective classifier is given in the Table 7. The significant increase in approx. 7.38% accuracy proved the efficacy of feature concatenation in enhancing the classification and discerning capabilities of the model. The performance increase may be accounted to the fact that the higher number of concatenated features added more descriptive details that a single architecture wasn't able to capture. Although the increase in the number of features resulted in the computational complexity, this technique can be significantly effective under resource-abundant setups where accuracy is of utmost priority. In crux, a final concatenated vector holding essential definitive properties of various scales helped the model better distinguish between the classes. The descriptive illustration of the evaluation metrics including the ROC curve is given in the Figure 18. Moreover, the metrics details inclusive of precision, sensitivity and other evaluation criterion is provided in the Table 6.

TABLE 7. Parametric configuration for decision tree classifier.

Decision Tree Classifier Parameters					
Methods	Random State	Max Depth	Criterion	Splitter	n-estimators
CNN-Individual	47	350	gini	random	500
ECNN	75	250	entropy	best	600
HLGGM	75	250	entropy	best	600
HLGCM	75	250	entropy	best	600

The Confusion Matrix for the ECNN shows an improved performance then the individual CNN counterparts. There are lower mis-classified instances as seen in Figure 18(a). Con-fusion Matrix. Similarly, the Individual classification performance for each class is better as illustrated in the ROC Curve. Class 2 has comparatively lower AUC score due to a slightly higher False Positive Rate as seen in Fig.18(b) ROC. Overall performance of the ECNN approach is satisfactory.

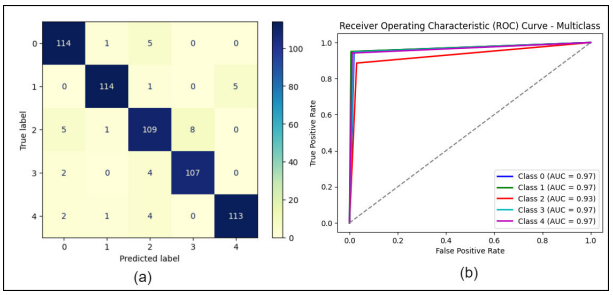


FIGURE 18. (a) Confusion matrix for ECNN (b) ROC curve for ECNN.

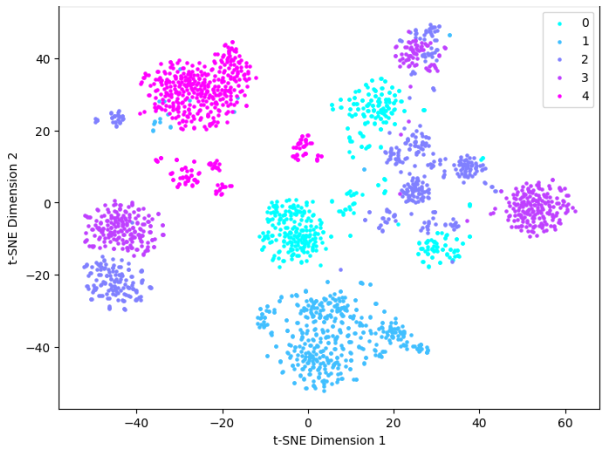


FIGURE 19. T-SNE based visualization of ECNN features.

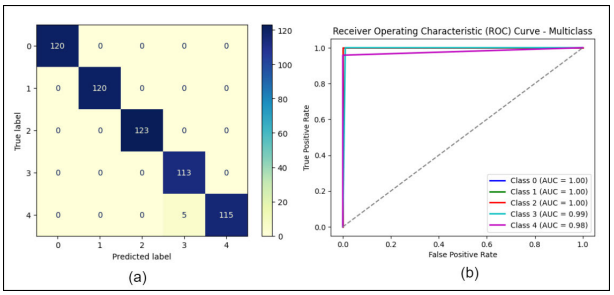


FIGURE 20. (a) Confusion matrix for HLGGM (b) ROC curve for HLGGM.

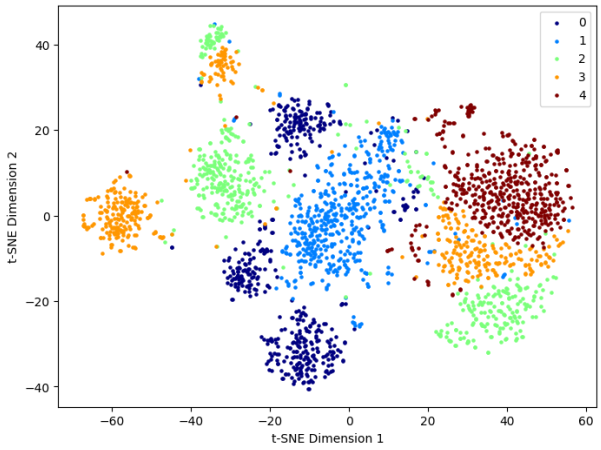


FIGURE 21. T-SNE based visualization of HLGGM features.

In Figure 19, the t-SNE plot shows the projection of the class distribution for ECNN features. For a detailed examination of the relationships between coffee leaves with varying stress types, the perplexity parameter has been set to a lower value. Meanwhile, the segregation of clusters in the above plot signifies the lower-dimensional representation of distinct patterns in the disease features. Each cluster represents a group of instances that share similar characteristics or exhibit similar patterns in the feature space. Different separations between clusters suggests that there are clear boundaries between different types or categories of coffee leaf diseases based on the ECNN features. Furthermore, the proximity of points within each cluster indicates similarities in the feature representations and related disease characteristics. Points that are closer to each other in above t-SNE plot are likely to have more similar features and share common attributes. This observation implies that instances within the same cluster are more likely to belong to the same or closely related disease category.

C. EVALUATION OF HLGGM FEATURES

The ensemble of hand-crafted and convolutional neural network based features showed promising outcomes when classified using the Decision Tree Classifier under the parametric settings defined within the Table 7. The experiment was an endeavor towards determining the extent to which a feature concatenation would enhance the model's performance. The concatenation of hand crafted features and a PCA based dimensionality reduced Mobile-Net v3 feature vector generated a classification accuracy of 99.16% on identical dataset. Outperforming the previous ECNN methodology's result of 93.45% by a margin of 5.71%. The final HLGGM feature vector held a size of 16780 inclusive of 10 features from PCA based fine-tuned Mobile-Net v3 architecture. The hand-crafted feature descriptors played a crucial part along with the condensed Mobile-Net v3 feature vector in determining the characteristics of the input images. The nature of input required a low level extraction of definitive characteristics for meticulous distinction of disease classes, which was successfully accomplished by the use of hand crafted techniques and an added fine-tuned Mobile-Net architecture. More details regarding the evaluation matrices is given in the Table 6 and the Figure 20 - 21.

The Confusion Matrix for HLGGM is yet again depicting the model's capabilities in terms of individual classification. Moreover, The ROC Curve shows the exceptional AUC scores for each class with just a scarce amount of miss-classified instances. In terms of performance, the model surpassed the former two approaches namely ECNN and the individual CNN models.

The t-SNE projection for HLGGM features, shown in Figure 21, provides a visualization of the relationships between different leaf disease features. With perplexity

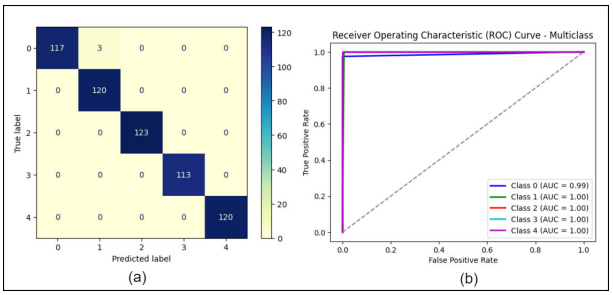


FIGURE 22. (a) Confusion matrix for HLGCM (b) ROC curve for HLGCM.

parameter set to a lower value, the t-SNE algorithm captures local structures and nearby similarities among the data points. This projection demonstrates, how the various disease features of the leaves are related to each other based on the HLGGM features. The projection further reveals that the relationships between the disease characteristics are not straightforward and simple, but a complex and densely intricate pattern. This complexity is shown by a slight overlap of the disease characteristics in the visualization. The overlapping clusters suggest that there are similarities and shared characteristics between different types of leaf diseases. This emphasizes the intricacy and inter-dependencies among the various leaf disease features captured by the HLGGM features which cannot all be expressed by a lower dimensional representation.

D. EVALUATION OF HLGCM FEATURES

The derivation of HLGCM features from dimensionality reduced hand-crafted and convolution network was a step towards determining the impact of feature reduction on the model's efficiency. The method attained an accuracy of 99.49% on the decision tree classifier, outperforming the ECNN and HLGGM approaches having accuracy of 93.45 and 99.16% respectively with the feature count of HLGCM lying at a bare 99, inclusive of chi^2 based hog features, LBP,GLCM, GABOR and PCA based MobileNet Features. The exceptional results further clarified the co-relation between the feature vector size and the quality of descriptors. It was found after the HLGGM classification results that the HOG based descriptors along with other Hand Crafted and CNN Techniques captured the essence of distinct classes providing a highest achieved accuracy of 99.16%, however, the size of the final vector alone eluded to a redundancy of features that can be reduced after a careful feature selection. Hence an exposure to a chi^2 based feature reduction for HOG descriptors to a mere 80 length vector successfully captured the most definitive properties. The concatenation of the reduced vector with other respective features not only reduced the final vector size and computational cost but also enhanced the final classification accuracy by a factor of 0.33%. The evaluation results are given in Table 6 and an illustration of the ROC curve is given in the Figure 22. The performance of HLGCM in terms of truly predicted values and false positive rates can

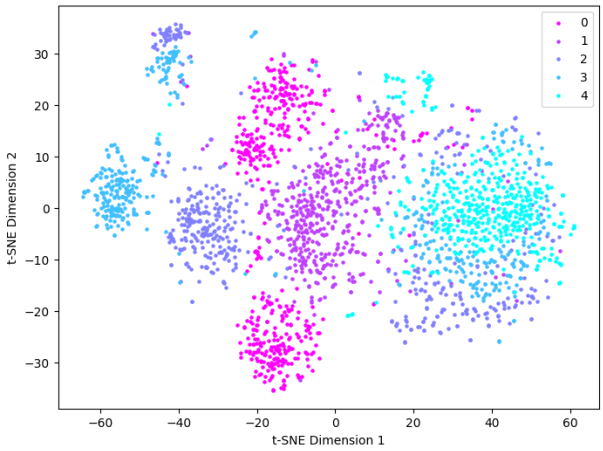


FIGURE 23. T-SNE based visualization of HLGCM features.

be inferred from the Confusion Matrix and the ROC Curve. As observed, the phenomenal AUC score for each class instance depicted in ROC Curve shed light on the model's abilities to distinguish each class. Moreover, a few instances of mis-classification shown in Confusion Matrix, themselves define the extent to which the HLGCM technique subdues the performance of previous methods. Thus making it more effective in classification of our given problem.

In Figure 23, the t-SNE projection is applied to the HLGCM features with a lower perplexity. Consequently, the projection reveals feature overlapping, indicative of the class similarities.The overlapping, alludes that certain classes are semantically identical with shared patterns and characteristics, which are reflected in their HLGCM feature representations.

This overlapping behavior further highlights the presence of inter-class similarities, where data points from different classes are closer to each other in the lower-dimensional t-SNE space due to their shared semantic attributes.

Overall, the t-SNE projection using the HLGCM features at a lower perplexity value effectively captures the similarity between classes, as evidenced by the overlapping of clustered points. This provides insights into the semantic relationships and shared characteristics among the different classes in the dataset.

E. COMPARISON WITH EXISTING STUDIES

Ground breaking research has been ongoing in the field of biotic stress classification for Coffee Leaves, however, densely intricate texture and patterns of various diseases makes it difficult to classify them using straightforward approaches. Moreover, the inherent textural and color similarities between varying disease types make many models susceptible to mis-classification. A couple of recent research studies analyzed for their methodologies, preferred convolutional architectures for classification of varying stress types as mentioned in Table 8.

Almost all of these studies showed inclination for a transfer learning based approach, due to limited data. Approaches

TABLE 8. Brief comparison of proposed methodologies with existing studies.

Reference	Year	Dataset	Features Extraction Methods	No. of Classes	Accuracy	Classifier
[13]	2022	BRACOL	Efficient Net B0 ResNet-152 VGG-16	Healthy Miner Cercospora Phoma Rust	97.31%	Softmax
[14]	2023	BRACOL	ResNet-50	Miner Rust Cercospora Phoma Healthy	98%	Softmax
[15]	2020	BRACOL	ResNet-50	Miner Rust Cercospora Phoma Healthy	95.24%	Softmax
[16]	2019	Private Dataset	GLCM LBP AlexNet	Cercospora Rust Healthy	98.0%	Softmax
[8]	2023	Private Dataset	GoogLeNet ResNet	Cercospora Rust Healthy Phoma	99.08%	Softmax
[9]	2022	Private Dataset	Efficient Net V0 Dense Net -121 VGG 16	Mold CLR CBD CWD Miner Rust Mold Red Spider Mites	95.98%	Softmax
[23]	2022	Private Dataset	Mobile Net ResNet-50	Cercospora Phoma Brown Eye Spot Rust Wilt Healthy	97.01% 99.89%	Softmax
[22]	2022	Arabica Coffee Dataset	Custom CNN Few Shot Learning	Miner Rust Cercospora Phoma Healthy	96%	Softmax
Our Study	2023	RCD	ECNN		93.45%	
			HLGGM	Miner Rust Cercospora Phoma Healthy	99.16%	Decision Tree
			HLGCM		99.49%	

including Few Shot Learning, modifications to existing state-of-the-art architectures, ensemble learning and guided learning were used to achieve high accuracy on different datasets. While [14] worked on a deeper analysis of class overlapping and their segregation using a Kernel Density approach prior to using a convolutional network for a guided take on imbalanced classes, others focused entirely on pushing the capabilities of the convolutional networks achieving a maximum accuracies of 99.07 and 99.87% using concatenated and transfer learning models for 5 and 4 class

classification. The computational costs and limited control over feature extraction pose significant challenges.While convolutions are indeed a powerful mechanism for feature extraction, hand crafted features provide a complementary interpretation of complex features and textures for better explainability of underlying decision making process if used alongside CNN. Along with the performance, the computational complexity can be greatly mitigated using only the selective highly descriptive features from both convolutional networks and hand crafted techniques. As is evident by our HLGCM method which attained a state-of-the-art accuracy of 99.49% for 5 classes along with an exceptional computational cost reduction compared with other state of the art architectures as given in Table 8.

V. CONCLUSION

This paper addressed various challenges posed by biotic stress on coffee leaves. These challenges have a fatal footprint and have a detrimental impact on the yield. This paper contributes by proposing three methodologies using a feature ensemble technique employing state-of-the-art CNN architectures and hand-crafted features to enhance the classification performance. Different combinations of feature ensemble were inspected and analyzed to assess their impact on model's performance. Moreover, semi-segmentation approach was considered for an effective guided extraction of foreground details and a deliberate openness to background noise for enhanced robustness. Additionally, a newly re-fined dataset was utilized for stress classification. The competency of the proposed technique was evaluated through various metrics. It was found that the concatenation approach worked better then the non-concatenated approaches for this setting. The proposed ensemble of CNN and handcrafted features named HLGGM attained superior performance than the ensemble of only CNN based features named ECNN by attaining 99.16% accuracy. Similarly, the dimensionality reduction further enhanced the performance of CNN and handcrafted techniques by attaining 99.49% accuracy. Revealing further insights about the role of feature reduction in augmenting model's efficiency. The challenges pertaining to plant disease classification and the effective novel approaches proposed in this research aim to contribute towards the deployable and computationally inexpensive solutions. Moreover, the proposed approaches offer a potentially improved disease classification for timely interventions and nurturing a sustainable coffee production. Despite attaining significant results, the following study faces some limitations. One of which is the validation and real world deployment testing. The evaluation methodologies employed in this paper focus mostly on the classification accuracy for this particular dataset, however the experimentation should be extended to tests under various environmental conditions and for different sets of data. Moreover, the proposed methodologies are complex in their composition and require segregated feature extraction and processing before classification albeit having a significantly lower decision time. Therefore, further combinations of features should be explored for a computational friendly and low complexity methodology. In future, the study will further be extended to a low complexity and computationally efficient approach along with an automated severity estimation for an effective diagnosis of the extent of biotic stress. Moreover, additional work on the deployment on edge devices will be explored for effective integration in real world scenarios.

ACKNOWLEDGMENT

The authors acknowledge the support from the Deanship of Scientific Research, Najran University. Kingdom of Saudi Arabia, for funding this work under the Research Groups funding program grant code number (NU/RG/SERC/12/9).

REFERENCES

[1] E. Elgin. *Native Plants and Ecosystem Services*. Accessed: Jun. 28, 2023. [Online]. Available: https://www.canr.msu.edu/nativeplants/
[2] *The Economic Impact of the Coffee Industry*. Accessed: Jun. 28, 2023. [Online]. Available: https://www.ncausa.org/Research-Trends/Economic-Impact
[3] *Promoting Domestic Coffee Consumption in Africa - CABI.org*. Accessed: Sep. 13, 2023. [Online]. Available: https://www.cabi.org/projects/promoting-domestic-coffee-consumption-in-africa/
[4] C. Nab and M. Maslin, "Life cycle assessment synthesis of the carbon footprint of Arabica coffee: Case study of Brazil and Vietnam conventional and sustainable coffee production and export to the United Kingdom," *Geo, Geography Environ.*, vol. 7, no. 2, Jul. 2020, doi: 10.1002/GEO2.96.
[5] C. Bunn, P. Läderach, O. Ovalle Rivera, and D. Kirschke, "A bitter cup: Climate change profile of global production of arabica and robusta coffee," *Climatic Change*, vol. 129, nos. 1–2, pp. 89–101, Mar. 2015, doi: 10.1007/S10584-014-1306-X.
[6] *Coffee | Diseases and Pests, Description, Uses, Propagation*. Accessed: Jun. 28, 2023. [Online]. Available: https://plantvillage.psu.edu/topics/coffee/infos
[7] *TECA—Technologies and Practices for Small Agricultural Producers*. Accessed: Jun. 28, 2023. [Online]. Available: https://teca.apps.fao.org/teca/es/technologies/4541
[8] B. M. Abuhayi and A. A. Mossa, "Coffee disease classification using convolutional neural network based on feature concatenation," *Informat. Med. Unlocked*, vol. 39, Feb. 2023, Art. no. 101245, doi: 10.1016/j.imu.2023.101245.
[9] F. J. P. Montalbo, "Automated diagnosis of diverse coffee leaf images through a stage-wise aggregated triple deep convolutional neural network," *Mach. Vis. Appl.*, vol. 33, no. 1, p. 19, Jan. 2022, doi: 10.1007/s00138-022-01277-y.
[10] E. Gichuru, G. Alwora, J. Gimase, and C. Kathurima, "Coffee leaf rust (Hemileia vastatrix) in Kenya—A review," *Agronomy*, vol. 11, no. 12, p. 2590, Dec. 2021, doi: 10.3390/agronomy11122590.
[11] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems Eng.*, vol. 144, pp. 52–60, Apr. 2016, doi: 10.1016/j.biosystemseng.2016.01.017.
[12] P. Sahu, A. Chug, A. Singh, D. Singh, and R. Singh, "Challenges and issues in plant disease detection using deep learning," 2021, doi: 10.4018/978-1-7998-3299-7.ch004.
[13] D. Novtahaning, H. A. Shah, and J.-M. Kang, "Deep learning ensemble-based automated and high-performing recognition of coffee leaf disease," *Agriculture*, vol. 12, no. 11, p. 1909, Nov. 2022, doi: 10.3390/agriculture12111909.
[14] R. I. Hasan, S. M. Yusuf, M. S. Mohd Rahim, and L. Alzubaidi, "Automatic clustering and classification of coffee leaf diseases based on an extended kernel density estimation approach," *Plants*, vol. 12, no. 8, p. 1603, Apr. 2023, doi: 10.3390/plants12081603.
[15] J. G. M. Esgario, R. A. Krohling, and J. A. Ventura, "Deep learning for classification and severity estimation of coffee leaf biotic stress," *Comput. Electron. Agricult.*, vol. 169, Feb. 2020, Art. no. 105162, doi: 10.1016/j.compag.2019.105162.

[16] L. X. Boa Sorte, C. T. Ferraz, F. Fambrini, R. D. R. Goulart, and J. H. Saito, "Coffee leaf disease recognition based on deep learning and texture attributes," *Proc. Comput. Sci.*, vol. 159, pp. 135–144, 2019, doi: 10.1016/j.procs.2019.09.168.

[17] H. L. Dawson, O. Dubrule, and C. M. John, "Impact of dataset size and convolutional neural network architecture on transfer learning for carbonate rock classification," *Comput. Geosci.*, vol. 171, Feb. 2023, Art. no. 105284, doi: 10.1016/j.cageo.2022.105284.

[18] E. Cengil and A. Çinar, "The effect of deep feature concatenation in the classification problem: An approach on COVID-19 disease detection," *Int. J. Imag. Syst. Technol.*, vol. 32, no. 1, pp. 26–40, 2022, doi: 10.1002/ima.22659.

[19] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, "Arabica coffee leaf images dataset for coffee leaf disease detection and classification," *Data Brief*, vol. 36, Jun. 2021, Art. no. 107142, doi: 10.1016/j.dib.2021.107142.

[20] R. Krohling, J. E. T. de Souza, and L. M. Tassis, "BRACOT—A Brazilian Arabica Coffee Tree images dataset for instance segmentation of coffee leaves," Mendeley Data, V1, 2021, doi: 10.17632/pmkbyjpf6k.1.

[21] *Plant Disease Classification Merged Dataset | Kaggle*. Accessed: Jun. 28, 2023. [Online]. Available: https://www.kaggle.com/datasets/alinedobrovsky/plant-disease-classification-merged-dataset

[22] L. M. Tassis and R. A. Krohling, "Few-shot learning for biotic stress classification of coffee leaves," *Artif. Intell. Agricult.*, vol. 6, pp. 55–67, 2022, doi: 10.1016/j.aiia.2022.04.001.

[23] E. B. Paulos and M. M. Woldeyohannis, "Detection and classification of coffee leaf disease using deep learning," in *Proc. Int. Conf. Inf. Commun. Technol. Develop. Afr. (ICTDA)*, Nov. 2022, pp. 1–6, doi: 10.1109/ICT4DA56482.2022.9971300.

[24] K. Kc, Z. Yin, D. Li, and Z. Wu, "Impacts of background removal on convolutional neural networks for plant disease classification in-situ," *Agriculture*, vol. 11, no. 9, p. 827, Aug. 2021, doi: 10.3390/agriculture11090827.

[25] D. D. Burdescu, M. Brezovan, E. Ganea, and L. Stanescu, "A new method for segmentation of images represented in a HSV color space BT—Advanced concepts for intelligent vision systems," in *Advanced Concepts for Intelligent Vision Systems* (Lecture Notes in Computer Science), J. Blanc-Talon, W. Philips, D. Popescu, and P. Scheunders, Eds. Berlin, Germany: Springer, 2009, pp. 606–617.

[26] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2005, pp. 94–102, doi: 10.1109/CVPR.2005.177.

[27] T. Löfstedt, P. Brynolfsson, T. Asklund, T. Nyholm, and A. Garpebring, "Gray-level invariant Haralick texture features," *PLoS ONE*, vol. 14, no. 2, pp. 1–18, 2019, doi: 10.1371/journal.pone.0212110.

[28] A. Hadid, "The local binary pattern approach and its applications to face analysis," in *Proc. 1st Workshops Image Process. Theory, Tools Appl.*, 2008, pp. 1–9, doi: 10.1109/IPTA.2008.4743795.

[29] A. Malhotra, A. Sankaran, A. Mittal, M. Vatsa, and R. Singh, *Fingerphoto Authentication Using Smartphone Camera Captured Under Varying Environmental Conditions*, 1st ed. Amsterdam, The Netherlands: Elsevier, 2017, doi: 10.1016/B978-0-08-100705-1.00006-3.

[30] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[31] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[33] Z. Sheng, M. Wang, X. Li, and X. Qin, "A new image denoising method based on the Gaussian filter," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, no. 1, Mar. 2008, pp. 929–932, doi: 10.1109/ICASSP.2008.4517763.

[34] 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, doi: 10.1109/CVPR.2017.243.

[35] 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, doi: 10.1109/CVPR.2016.90.

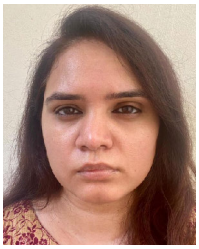
[36] H. Hosseini, B. Xiao, M. Jaiswal, and R. Poovendran, "On the limitation of convolutional neural networks in recognizing negative images," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 1–7, doi: 10.1109/ICMLA.2017.0-136.

[37] I. U. Haq, H. Ali, H. Y. Wang, C. Lei, and H. Ali, "Feature fusion and ensemble learning-based CNN model for mammographic image classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3310–3318, Jun. 2022, doi: 10.1016/j.jksuci.2022.03.023.

[38] G. Bogacsovics, J. Toth, A. Hajdu, and B. Harangi, "Enhancing CNNs through the use of hand-crafted features in automated fundus image classification," *Biomed. Signal Process. Control*, vol. 76, Jul. 2022, Art. no. 103685, doi: 10.1016/j.bspc.2022.103685.

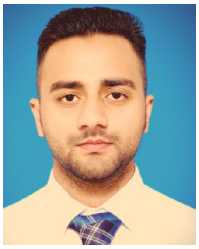
[39] A. Simon-Gruita, M. D. Pojoga, N. Constantin, and G. Duta-Cornescu, *Genetic Engineering in Coffee*. Amsterdam, The Netherlands: Elsevier, 2019, doi: 10.1016/B978-0-12-815864-7.00014-3.

MUHAMMAD ARMGHAN LATIF received the B.Sc. degree from the University of Management and Technology, Sialkot, Pakistan, and the M.S. degree from Cleveland State University, Cleveland, OH, USA. His research interests include neural networks, machine learning, deep learning, computer vision, and data science.



NOOR AFSHAN received the B.S. degree in software engineering and the M.S. degree in computer sciences from Superior University, Lahore, Pakistan, in 2013 and 2016, respectively. She is currently a Lecturer of computer science at Lahore Garrison University, where her research focuses on data analytics and mining techniques of various data sets related to medical, metrological, and speech. She is currently working on research projects aimed at exploring big data and data science techniques that can be useful to maximize medical science and other crucial fields of research work.

ZOHAIB MUSHTAQ received the B.Sc. degree from Islamia University, the M.S. degree from the Government College University, Lahore, and the Ph.D. degree in electrical engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan. He is currently an Assistant Professor of electrical engineering at the University of Sargodha, Sargodha, Pakistan. Previously, he was with Riphah International University as an Assistant Professor. He has published research in IEEE and other reputable journals. His research interests include neural networks, machine learning, feature engineering, deep learning, computer vision, and data science.



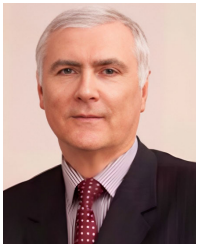
NABEEL AHMED KHAN received the degree in electrical engineering from Riphah International University, Islamabad. His research interests include deep learning, signal processing, computer vision, machine learning, and data science.

MUHAMMAD IRFAN received the Ph.D. degree in electrical and electronic engineering from Universiti Teknologi PETRONAS, Malaysia, in 2016. He has two years of industry experience (October 2009 to October 2011) and six years of academic experience (since January 2017) in teaching and research. Currently, he is working as an Associate Professor at the Electrical Engineering Department, Najran University, Saudi Arabia. He has authored more than 200 research articles in reputed journals, books, and conference proceedings (Google Scholar Citations 2500, H-index 23). His main research interests include automation and process control, energy efficiency, condition monitoring, vibration analysis, artificial intelligence, the Internet of Things (IoT), big data analytics, smart cities, and smart healthcare.

GRZEGORZ NOWAKOWSKI is currently an Assistant Professor in computer science at the Department of Automatics and Informatics, Cracow University of Technology, Poland. His scientific interests include information systems, computational intelligence, databases, big data, cloud computing, soft computing methods, fuzzy database queries, and their application in big data analysis.

SAMAR M. ALQHTANI received the Ph.D. degree in information technology. She has more than five years of academic experience in teaching and research. Currently, she is working as an Associate Professor with the College of Computer Science and Information Systems, Najran University, Saudi Arabia. She has authored more than 40 research articles in reputed journals, books, and conference proceedings. Her main research interests include artificial intelligence, the Internet of Things (IoT), big data analytics, smart cities, and smart healthcare.

SALIM NASAR FARAJ MURSAL received the Ph.D. degree in electronic and communication engineering. He has more than ten years of academic experience in teaching and research. Currently, he is working as an Assistant Professor with the College of Engineering, Najran University, Saudi Arabia. He has authored research articles in reputed journals, books, and conference proceedings.



SERGII TELENYK received the M.Sc.Eng., Ph.D., and D.Sc. (Higher Doctorate) degrees in information technologies from the Igor Sikorsky Kyiv Polytechnic Institute, Ukraine, in 1975, 1982, and 2000, respectively. He currently holds the position of Professor at the Cracow University of Technology, where he has been involved in scientific and educational activities since 2016. In addition, he is also a Professor at the Igor Sikorsky Kyiv Polytechnic Institute. His research interests encompass various areas, including IT infrastructure management, software engineering, mathematical logic, computational linguistics, artificial intelligence, control theory, information systems design, methodologies related to the support of the life cycle of services in information systems, the design and implementation of services, the preparation and provision of services, and the development of services.