ISSN: 2834-7706

Research Article

Journal of Mathematical Techniques and Computational Mathematics

A Mathematical Analysis of Feature Extraction Methodologies with Application on Leaf Disease Detection

Akshat Kishore*

*The Shri Ram School, Aravali Campus, India

*Corresponding Author
Akshat Kishore , The Shri Ram School, Aravali Campus, India.

Submitted: 2023, Nov 20; **Accepted:** 2023, Dec 22; **Published:** 2023, Dec 30

Citation: Kishore, A. (2023). A Mathematical Analysis of Feature Extraction Methodologies with Application on Leaf Disease Detection. *J Math Techniques Comput Math, 2*(12), 575-583.

Abstract

This study compares novel and established feature extraction techniques for classifying leaf diseases. Conventional methods like Transfer Learning and DCT are contrasted with novel techniques like Segmentation using a heterogeneous data set of healthy and diseased leaf images. With little training data, transfer learning is beneficial, but with large data, more efficient algorithms may be more suitable, such as the custom segmentation model described in this work. The development of efficient plant disease detection systems using the information from these discoveries would improve agricultural practises and increase food security.

Keywords: Feature Extraction, Image Transforms, Leaf Disease

1. Introduction

According to the Food and Agriculture Organisation of the United Nations, the number of starving people has been increasing since 2014; these reports show that over 690 million people are hungry, which constitutes over 8.9% of the world population. Also, more than 90% of the world relies of agriculture, but more than 50% of crop production is lost crop disease. Therefore, recognising crop disease is essential and urgent.

Plant diseases have had massive impacts on crops. A plant disease is a disturbance of a plant's natural condition that prevents or alters the plant's essential functioning. All plant species, whether wild and domesticated, are susceptible to illness. Despite the fact that each species is prone to certain illnesses, they are always relatively uncommon. Depending on the presence of the pathogen, the surrounding circumstances, the crops and kinds farmed, and the frequency and prevalence of plant diseases change every season. While certain plant kinds are more susceptible to disease outbreaks than others are, others are not [1].

Several diseases can drastically limit crop yield, which poses a severe threat to food security. So, it is crucial to identify plant diseases precisely. Conventional categorisation techniques, such eye-only observation and lab testing, have several drawbacks, including the fact that they are time-consuming and subjective, and require expert skill and previous experience. Convolutional neural network (CNN)-based deep learning (DL) techniques, in particular, are currently finding considerable use in the categorisation of plant diseases. They are state-of-the-art technologies in this industry and have all or some of the issues with conventional categorisation methods resolved. In this work, we have explored novel combinations of feature extraction methods and deep learning models.

Moreover, the identification and discovery of various illnesses may be accomplished with the use of microscope and DNA sequencing techniques. Despite the fact that many farmers worldwide lack access to these diagnostic instruments, the great majority of them own a cell phone. Hence, a smartphone-based application that aids in crop disease diagnosis based on taking and automatically analysing a photo of a plant leaf is a viable option. Artificial intelligence technology advancements have cleared the door for the creation of automated systems that can diagnose illnesses more quickly and accurately.

Previous work in this field focuses mainly on the use of large deep neural networks using trainable parameters [2]. The majority of conventional machine learning algorithms were developed in laboratories, and therefore lack the robustness required for real-world agricultural applications. However, training millions of these parameters is time consuming. Therefore, to improve the efficiency of performing this task, we have proposed the use of a custom model which achieves the same accuracy as in the previous studies, using feature extraction methods.

In this paper, we aim to prove the hypothesis that the usage of efficient feature extraction algorithms can eliminate the need of training models with hundreds of layers and millions of parameters. This paper focuses on the comparative analysis of novel and standard feature extraction algorithms, which are then passed into the neural network as an input one by one, comparing their time efficiency as well as accuracy.

2. Materials and Methods

In this section, we present the leaf disease datasets used for experimentation in the problem and the proposed methodologies used to solve the task.

2.1. Dataset

In this work, we use the publicly available leaf disease classification dataset. This dataset consists of 70,344 images of 38 classes. These include 14 different plant species, with healthy and diseased classes for each. Each of the 38 classes contain 2,000 images each. The data already consists of augmented images, therefore no augmentation was explicitly done in code. Further, we consider the 14 different classes of leafs, assuming that the input consists of one of these.

2.2. Preprocessing

Many images of the dataset have low contrast, are slightly blurry and have indistinguishable features. To prevent this from being an issue during the learning phase, two main filters were applied to each image in the dataset before training:

• CLAHE Contrast (Contrast Limited Adaptive Histogram Equalisation Contrast): This filter enhances the local contrast of each image. Instead of using the entire image, CLAHE operates on portions of the image called tiles. To eliminate the erroneous borders, the adjacent tiles are blended using bilinear interpolation. The filter was applied on the luminance channel of the image.

• Sharpening: This was applied to make the edges more prominent to aid feature extraction algorithms. Sharpening was done by overlaying a filter kernel over the image pixels. The kernel acts according to the equation below:

$g(x, y) = \omega * f(x, y)$

Where $g(x,y)$ is the output matrix, ω is the kernel matrix and $f(x,y)$ is the original matrix

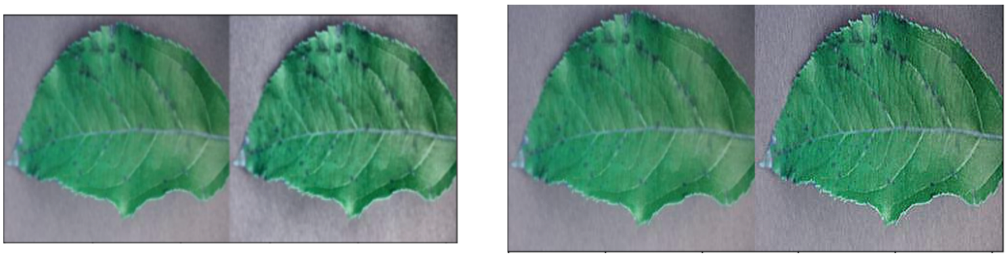

Figure 1: Images before and after sharpening and CLAHE Contrast processing.

2.3. Feature Extraction

The dimensionality reduction method, which divides and condenses a starting set of raw data into smaller, easier-to-manage groupings, is called feature extraction. Features such as texture and shape play an important role in the classification of leaf disease. In this section, we consider and explain the working of 4 types of feature extraction algorithms: DCT, DFT, Transfer learning, Segmentation followed by shape, colour and texture extraction

2.3.1. DCT- Discrete Cosine Transform

With regard to the visual quality of the picture, the discrete cosine transform (DCT) assists in dividing the image into components (or spectral subbands) of varying significance. The DCT transfers a signal or picture from the spatial domain to the frequency domain, much like the discrete Fourier transform does. That is, transforming signals from the spatial representation into the frequency representation is the fundamental purpose of DCT [3]. The DCT Matrix can be computed using this formula:

$$F = C.f.C^t$$

where C is defined below, for $u = 0$ and for $0 < u < N-1$ or $0 < v < N-1$ respectively.

$$C = \sqrt{\frac{1}{N}}$$

$$C = \sqrt{\frac{2}{N}} cos[\frac{(2v+1)\pi u}{2N}]$$

The sum of each matrix produced above results in the required DCT Matrix for the image. In the above matrix equation, represents the original 8x8 block. Each 8x8 of these blocks result in a 64 element DCT matrix (also 8x8), in which the first element corresponds to low frequencies from the original block. As we move away from this element in all directions, each element corresponds to a higher and higher frequency of the original image.

In our work, we used the following algorithm to extract features using DCT:

1. Split the image into blocks of size 8x8
2. Convert each value in the 8x8 block to be in the range of (-128,128) (3)
3. Apply DCT of this 8x8 block. Once this DCT is applied, obtain 64 DCT coefficients.
4. This DCT Matrix consists of both low frequency and high frequencydata of the image block. Now, from this matrix we

extract the lower frequency elements, which is what the human eye is sensitive to.
Therefore, through this algorithm, we effectively extract only the significant features present in the image. The algorithm is summarised in the figure 2.

2.3.2. DFT- Discrete Fourier Transform

A Fourier Transform pair is often written $f(x) \leftrightarrow F(x)$, or $F(f(x)) = F(\omega)$ where F is the Fourier transform operator. If $f(x)$ is thought of as a signal (i.e. input data) then we call $F(\omega)$ the signal's spectrum. In other words, if f is thought of as the impulse response of a filter (which operates on input data to produce output data) then we call F the filter's frequency response. The Fourier transform can be calculated through the formula below [4]:

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx$$

Through the formulae above, it is clearly visible that the Fourier transform breaks a periodic function as a weighted sum of infinite sinusoids of different frequencies.

From number theory, $e^{i\theta} = isin\theta + cos\theta$. Note that the difference between DCT and DFT is the presence of complex terms in the DFT result. Therefore, the Fourier coefficient for each x is a complex number and is not real. This is because it needs to capture the frequency as well as the phase of the component sinusoidal wave. The Fourier transform results in the sum of the frequencies which build up the image.

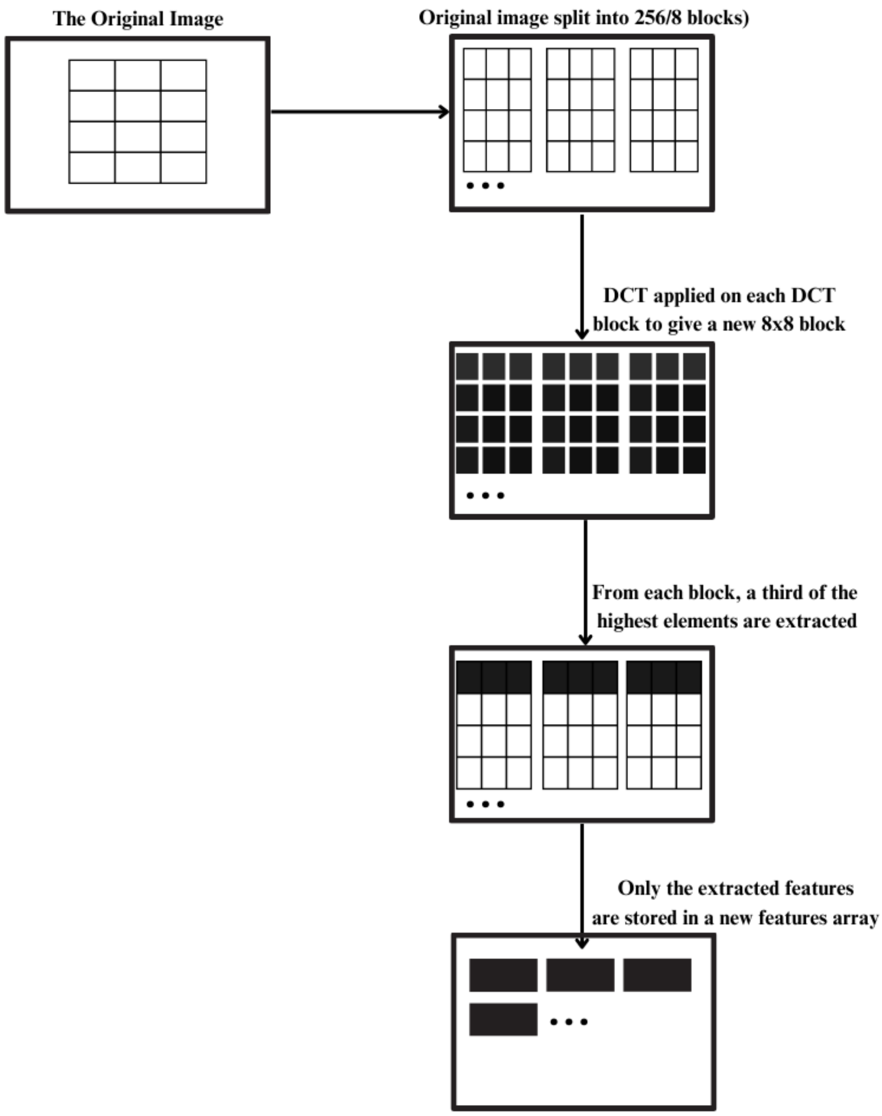

Figure 2: Graphical represntation of the Feature extraction using DCT algorithm.

Now, the Discrete Fourier transform is used to calculate the Fourier Transform when $f(x)$ is periodic, instead of this continuous Fourier Transform. DFT can be computed as below:

$$F = T.f.T^t$$

This formula can be used to compute the DFT of one single block of the image. Here, the Fourier Matrix T is given as below:

$$W_N^{nk} = [e^{-j2\pi}]^{\frac{nk}{N}}$$

All the F obtained through these formulae, that is for each block, can be summed to produce the desired Fourier transform.
We can use the DFT to compute the Fourier transform in case of images, since images represent periodic signals.

Further, the Fast Fourier Transform is an algorithm for calculating DFT, which is nothing but a generalisation of results obtained through the DFT formula. To compute the DFT of an N-point sequence using DFT would take $O(n^2)$ multiplies and adds. The FFT algorithm computes the DFT using $O(nlogn)$ multiplies and adds. Therefore, FFT is an algorithm to compute DFT in a computationally efficient manner. In our project, we used the following algorithm to calculate the DFT and IDFT:
1. Calculate the DFT of the image using FFT
2. Extract the magnitude from the result which is initially a complexoutput
3. Create a circular mask to extract either low frequency or high frequency(high frequency corresponds to edges and low frequency corresponds to objects).
4. Overlay the mask on the DFT image
5. Apply IDFT on this to convert the image back to spatial domain (forvisualisation).

2.3.3. Transfer Learning

A model created for one task being used as the basis for another is referred to as transfer learning. Pre-trained models are frequently utilised as the foundation for deep learning tasks in computer vision and natural language processing because they save both time and resources compared to developing neural network models from scratch and because they perform far better on related tasks.

Considering a source model with domain D_s and task T_s and target domain D_t with corresponding task T_t, transfer learning is the process used to improve the predictive function of the target task using related information from D_s and T_s [5]. Machine learning uses transfer learning as a key approach to address the fundamental issue of inadequate training data. By easing the requirement that the training data and the test data must be i.i.d (independent and identically distributed), it attempts to transfer knowledge from the source domain to the target domain [6]. In our work, we used 3 models as the base model for transfer learning. A brief introduction to each is given below:
• ResNet 50: ResNet stands for Residual Network and is a specific type of convolutional neural network (CNN). The ResNet50 architecture which took first place in the ILSVRC-2015 competition, was developed to address the degradation issue and the issue of multiple non-linear layers not learning identity maps [7]. A network in network design called ResNet50 is based on numerous stacked residual units. The network is constructed using residual units as its building elements. Layers for convolution and pooling make up these units. This architecture accepts input images with a size of (224, 224) pixels and employs 3x3 filters like in VGG16.
• VGG-16: A CNN architecture called VGG16, which won the ILSVRC2014, has over 138 million parameters [8]. The most notable aspect of this architecture is that it never changes the convolution layers, which use 3x3 filters with stride 1, and the padding and maximum pooling layers, which always use 2x2 filters with stride 2. This convolution and maximum pooling layers configuration is followed continuously throughout the VGG16 architecture. The last three FC layers are the last with Softmax activation function and the first two with ReLU. The input layer in this 16-layer design accepts 224 x 224 pixel pictures. • Inception V3: The third version of Google's Deep Learning Evolutionary Architectures is called Inception V3 [9]. Szegedy created the Inception V1 architecture, and Inception V2 implemented batch normalisation. Then, Inception V3 added the concept of factorisation. Reducing the amount of connections and parameters without sacrificing

Figure 3: The workflow used to implement transfer learning.

the network's effectiveness is the major goal of factorisation. Convolutions, average pooling, maximum pooling, concatenations, dropouts, and fully connected layers are some of the symmetrical and asymmetrical building components that make up the model itself. The input layer of the Inception V3 architecture, which incorporates the Softmax function in the last layer, has a total of 42 layers and accepts images with a 299x299 pixel resolution.

For each of the 3 pre-existing models, we used the following procedure to extract features from the images in the leaf dataset:
1. After obtaining the required model, each trainable parameter, or weight,of the model is set to be non-trainable. That is, the weights of the models are set to those that were used to train the model on the imagenet dataset.
2. Then, the head or top of the model, so that only the layers involved infeature extraction are left.
3. Using this, predictions are made on the leaf dataset and reshaped tobe inputted into the Keras model.

2.3.4 Segmentation

Most of the aforementioned methods of feature extraction result in feature matrices which contain ten thousands of elements for each image. Even though this is a reduction from the 256x256x3 size of images, it is still slow to train a model with 70,000 images of 10,000 features each. The process is yet slower on even larger datasets. The proposed segmentation and consequent feature extraction, however, results in less than 100 features per image.

Segmentation is the process of dividing an image into its constituting parts. The diseases are often present as orange or red spots on the leafs and thus, we decided to use colour-based segmentation to separate the diseased parts from the healthy parts of the leaf.

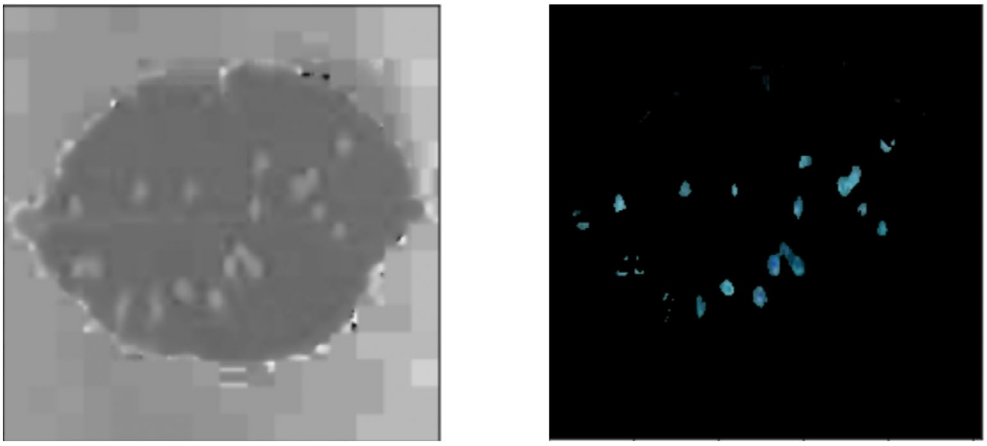

Figure 4: Visual representation of hue based segmentation. On the left is a black and white leaf image with boosted contrast (CLAHE).

This is simply implemented in code using boolean masks of hue values to separate out the different hues of the image, as showed in the figure 4.
This results in a 256x256 image with the diseased part of the leaf. Now, we can apply the following extraction methods on this segmented image:
1. Colour Feature Extraction: from each segmented image we can extractthe colour information it stores. In our work, we used the following colour parameters to obtain this colour information: (a) Mean of red channel (b) Mean of green channel (c) Mean of blue channel (d) Standard Deviation of red channel (e) Standard Deviation of green channel (f) Standard Deviation of blue channel (f) Mean of hue channel (g) Mean of saturation channel (h) Mean of value channel (i) Standard.
Deviation of value channel [10]
All of this information was extracted by accessing each channel of the image in different colour spaces
2. Texture Feature Extraction using GLCM: The grey-level co-occurrencematrix (GLCM) is a statistical way of assessing texture that analyses the spatial relationship of pixels, also known as the grey-level spatial dependency matrix. The GLCM functions define the texture of an image by calculating how often pairs of pixel with given values and in a specified spatial relationship occur in an image, building a GLCM, and then extracting statistical measures from this matrix. GLCM has shown to be a popular statistical approach of extracting textural characteristic from photos [11].

Consider a situation where we want to calculate how often the pixels surrounding an image are a particular value. It is to get this probability that a GLCM is used. That is, the (0,0) element in the GLCM shows the probability of how often the pixel of value 0 has another pixel of value 0 below it. Similarly, the (0,1) element holds the probability of how often a pixel of value 0 has a pixel of value 1 below it, and so on. However, if we want to find the same for the right or left direction, we simply need to change the positional operator used in code, which tells the code which pixel to check for. Now, using our GLCM matrix, a group of properties can be extracted which represent the texture information of an image. In each of the following equations below, P refers to the GLCM matrix. These properties can be considered to quantify the GLCM matrix. These are defined as below:

$$contrast = \sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$$

$$dissimilarity = \sum_{i,j=0}^{levels-1} P_{i,j}|i-j|$$

$$homogeneity = \sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2}$$

$$ASM = \sum_{i,j=0}^{levels-1} P_{i,j}^2$$

$$energy = \sqrt{ASM}$$

3. Shape Feature Extraction: To represent the features about the shapeof the diseased areas, we used Hu Moments. Image moments are often used in computer vision and image processing to describe the geometry of an item in an image. These snapshots record essential details like an object's area, centroid (i.e., its x, y coordinates), orientation, and other desired characteristics. A real-valued feature vector with 7 values is what the Hu Moments descriptor produces. The form of the item in a picture is quantified and captured by these 7 parameters. Mathematically, moments are simply a statistical expectations of a random variable. Moments are the weighted average of all image pixel intensity values, while taking into consideration the location of the pixel. The regular moment of shape in a binary image is represented as below:

$$M_{i,j} = \sum_x \sum_y x^i y^j I(x, y)$$

However, the relative moments which are centred about the centroid (x,y) are:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y)$$

These relative moments are translation invariant, that is wherever the shape is present in the image, the relative moment will be the same. However, these relative moments are not scale and rotation invariant, which is where Hu Moments provide an advantage. Hu moments, on the other hand, break these moments into 7 features which finally form our feature vector and represent our shape in a quantifiable manner [12].

Therefore, we successfully extracted 3 sets of features from each image: the colour features, texture features and shape features. All these 3 sets are appended into a single array and passed into the learning model. The feature set consists of a matrix with 34 elements per image, as opposed to the other algorithms which created more than 10,000 elements in the matrix of each image.

2.4. Deep Learning

Each of the features extracted in the previous stage of the process were one by one passed to a basic, custom Keras model with 17 fundamental layers. The model can be visualised schematically through the figure 5. Note that the input sizes change for every feature pool fed into the model, the one shown in the figure below shows the input sizes particularly for the DCT feature pool, i.e. 1024x3x8. The figure can be used to understand every layer in the custom model.

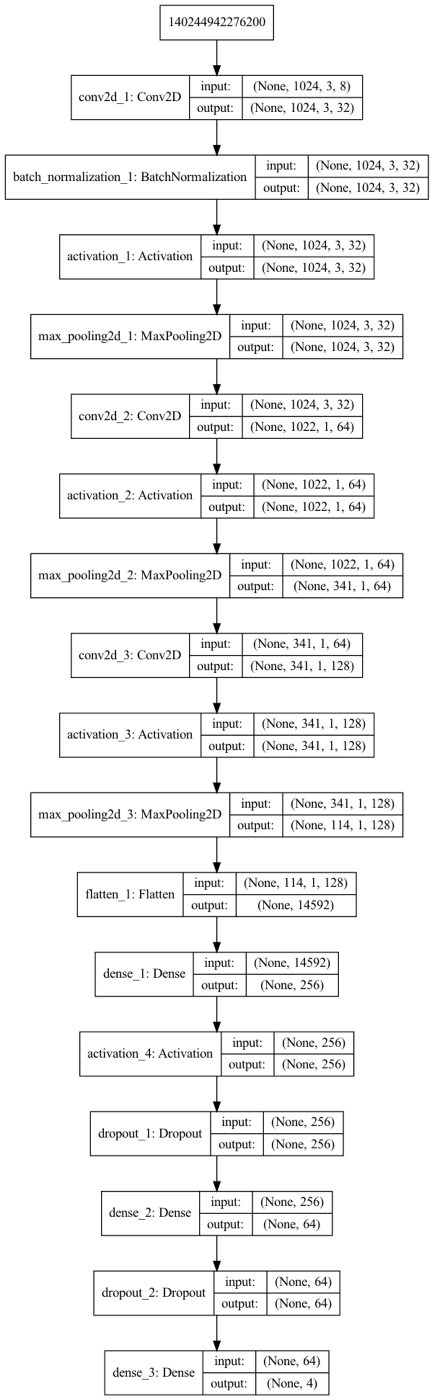

Figure 5: The full deep learning model architecture. The input sizes shown here are for the DCT method of feature extraction.

2.5. Model Evaluation

In our work, we used K-Fold Cross validation to evaluate the efficiency of the model in case of each feature pool.

The main characteristic of cross-validation is that every sample in our dataset has the chance to be tested. In k-fold cross-validation, we loop through a dataset set k times. We divide the dataset into k parts for each round, using one part for validation and merging the remaining k − 1 parts to create a training subset for model assessment.

Five separate models will be fitted in 5-fold cross-validation; these models will be fitted to unique but partially overlapping training sets and assessed on non-overlapping validation sets. The cross-validation performance is ultimately calculated as the arithmetic mean across the k performance estimations from the validation sets.

$$Accuracy = \frac{1}{N}\sum_{i=1}^{k} accuracy_i$$

	DCT	DFT	Transfer Learning	Segmentation
Accuracy	0.816	0.615	0.945, 0.964. 0.766	0.823
Output Size	(256,256)	(256,256)	(8,8,2048), (6,6,2048), (8,8,512)	(21,2,1)

Table 1: The performance of different feature extraction methods on a leaf disease dataset

The accuracy of the segmentation method increases greatly with increase in size of data, mainly because it treats very specific parts of each image and thus needs for data in general.

The working of the complete methodology can be summarised in the figure 6.

3. Conclusion

With the introduction of various modern deep learning techniques, the process of leaf disease classification can be simplified to a great extent. In this work, we proposed the usage of multiple novel and standard feature extraction methods, going into the working of each, to dimensionally reduce the size of the input data, therefore reducing training times exponentially. We proved that the usage of feature extraction in combination with a custom,

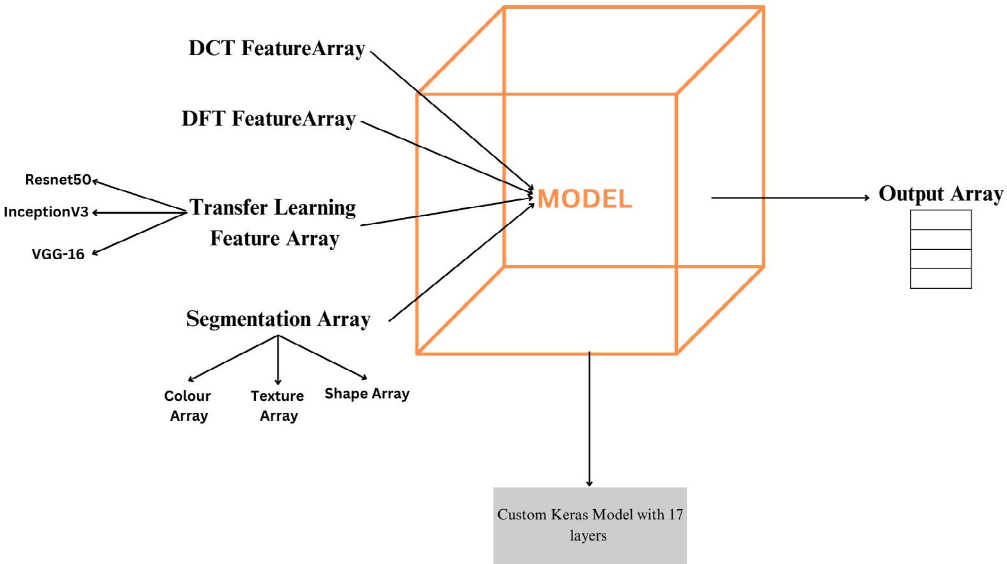

Figure 6: The entire methodology discussed in this paper is depicted in this figure.

small deep learning model can achieve great efficiency, therefore eliminating the need of training the model using the entire image data as an input.

4. Acknowledgments

Research conducted under guidance of Dr. Naveen Kumar, Professor at University of Delhi and Dr. Ankit Rajpal, Assistant Professor at University of Delhi.

References

1. Lu, J., Tan, L., & Jiang, H. (2021). Review on convolutional neural network (CNN) applied to plant leaf disease classification. Agriculture, 11(8), 707.
2. Atila, Ü., Uçar, M., Akyol, K., & Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model. Ecological Informatics, 61, 101182.
3. Mustafa, W. A., Yazid, H., Khairunizam, W., Jamlos, M. A., Zunaidi, I., Razlan, Z. M., & Shahriman, A. B. (2019, June). Image enhancement based on discrete cosine transforms (DCT) and discrete wavelet transform (DWT): a review. In IOP Conference Series: Materials Science and Engineering (Vol. 557, No. 1, p. 012027). IOP Publishing.
4. Oberst, U. (2007). The fast Fourier transform. SIAM journal on control and optimization, 46(2), 496-540.
5. Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big data, 3(1), 1-40.
6. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27 (pp. 270-279). Springer International Publishing.

7. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
8. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
9. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. Inception-ResNet and the impact of residual connections on learning. arXiv 2016. arXiv preprint arXiv:1602.07261.
10. Azim, M. A., Islam, M. K., Rahman, M. M., & Jahan, F. (2021). An effective feature extraction method for rice leaf disease classification. Telkomnika (Telecommunication Computing Electronics and Control), 19(2), 463-470.
11. Mohanaiah, P., Sathyanarayana, P., & GuruKumar, L. (2013). Image texture feature extraction using GLCM approach. International journal of scientific and research publications, 3(5), 1-5.
12. Hu, M. K. (1962). Visual pattern recognition by moment invariants. IRE transactions on information theory, 8(2), 179-187.