



Aspergillus niger Fungus Detection using Transfer Learning Technique and Modified Backpropagation Algorithm with Inertia and Legendre Polynomial

V Vanitha¹ and M Sornam^{2*}

¹Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research (DU), Porur, Chennai 600 116, Tamil Nadu, India

²Department of Computer Science, University of Madras, Chennai 600 005, Tamil Nadu, India

Received 11 April 2021; revised 03 April 2022; accepted 11 May 2022

Looking at the loss due to health problems from fungal diseases in one hand and the benefits from its industrial/agricultural use, rapid automated fungal species identification is the need of the hour. Hence, proposed a fast identification of fungal species by a 15 minutes staining procedure followed by an artificial-intelligence-based image classification technique. In this modern era, deep architectures have shown a significant performance on computer vision problems. Instead of developing a new model from scratch, the pre-trained convolutional neural network models are available to obtain the appropriate features from input samples using the transfer-learning technique. This work utilizes the transfer-learning approach for feature extraction and classification performed using the proposed modified third-term Backpropagation (BP) algorithm. This proposed algorithm contains Inertia as a third factor in the weight updation rule expanded in the form of the Legendre polynomial to overcome the limitations of the traditional Backpropagation algorithm. The effectiveness of the proposed classifier compared to the results of the existing cutting-edge algorithms namely, Backpropagation algorithm, Backpropagation algorithm using Momentum, and softmax classifier. Compare to the existing models, the proposed model scored a high testing accuracy of 97.27%.

Keywords: Confusion matrix, Fungus classification, Orthogonal polynomial, Pre-trained deep learning models

Introduction

Aspergillus niger fungi belongs to the class called Conidiophores, a type of organism which forms hyphae otherwise known as conidia. The size of fungi can range from 900 to 1600 μm . It has rough spherical conidia measuring 3 to 5 μm . It is a group of filamentous fungus which can be easily cultivated in laboratory conditions. So, it is widely used extensively studied group of fungus.

Over 300 million people worldwide are affected by parasitic fungal infection. From this, 25 million people lose their life or sight. More than 1.5 million individuals die each year from fungal illnesses. In the scientific realm of medicine, the identification of fungal species is a more challenging task as signs and symptoms are asymmetric and non-specific. On an average, the current wet-lab method of fungal identification required at least 72 hours. Losing these golden hours is proven to be precious in clinical settings, generally resulting in the loss of lives. On the other side, most fungi species are more useful for the

industries to produce food enzymes. Early and precise diagnosis allows for rapid antifungal treatment; nevertheless, treatment is frequently delayed or unavailable. This results in mortality, major chronic illness, or blindness. Aspergillosis is the disease caused due to inhalation of *Aspergillus* spores which are present in most indoor as well as outdoor settings. Most people breathe in fungal spores every day but those having weak immune systems get infected. Initially, it starts from the lungs or sinus and spreads to other parts of the body. There are 180 species of *Aspergillus* species identified, from these the most common cause of *Aspergillus* infections is *A. fumigatus*, *A. flavus*, *A. terreus*, and *A. niger*.

Invasive Pulmonary Aspergillosis recognized as a next subsequent disease, the most serious manifestation of coronavirus disease 2019.^(1,2) The WHO reported that over 40 million peoples worldwide got infected by COVID-19 and almost more than 1.1 million people have died. About 40% of hospitalized patients with COVID-19 symptoms develop COVID-19 Associated Pulmonary Aspergillosis (CAPA). This super infection is associated with high

*Author for Correspondence
E-mail: madasamy.sornam@gmail.com

mortality rate. According to the report, 35% of the patients who have invasive pulmonary aspergillosis lead to highest mortality rate.³⁻⁶

This Fungus is commonly found in decaying vegetables or soil and plant. Since it is a type of mould, it is attributed to the cause of pneumonia. This Fungus also forms like a black mould on the outside part of foods such as onion, grapes, apricot and so on.

The genome sequence of *A. niger* is very important to produce citric acid and also for industrial enzymes such as amylases, proteases, and lipases. These enzymes are very useful for industries because it can be transformed into food enzymes. *A. niger* is also utilized for biotransformation as well as waste management. If the *A. niger* cultivated and adapted correctly, then it is very useful for the food industry because of high transfructosylating activity of the enzymes within its surface.⁷

Another important application of *A. niger* is safe dye removal because of the advantage of its bio absorption abilities.⁸ Congo Red dye removal generally requires a certain amount of washing by harmful dye removing substances once the waste water collected and autoclaved, but this can be done easily by simply adding *A. niger* colonies which absorbs the dye.

Polymerase chain reaction, nucleic acid probe technology, and plate count method are some of the classic methods for detecting fungal infection.⁹ These methods are quite costly since they necessitate specialized equipment as well as the use of skilled experts to do the task. This complex analysis was carried out by a scientist using a powerful microscope, and other researchers may employ the electromagnetic spectrum to detect fungus infection. To obtain accurate and reliable prediction, high-end equipment, expert labour and high processing time are required. The screening of fungi requires a large amount of time but in the same duration, the fungi grow enormously. As a result, industries want quick, precise, and instantaneous results, prompting us to seek for a machine learning solution to this challenge.

Many research works were carried out on the identification and classification of fungi in the field of computer vision in the last decade. The sequence research works of Blasco *et al.*¹⁰, Mertens *et al.*¹¹, and Tahir *et al.*¹², is one of the reasons which made the food sector one of the top ten industries in the world. Blasco's and co-author proposed an approach for the food quality control which is utilized by Mehrten for

product classification and it is further used by Tahir for the detection of fungi.

In the year 2013, Asefpour Vakilian & Massah¹³ employed artificial neural network and digital image processing techniques to spot fungal infection in cucumber plants with greater accuracy. In the year 2018, Sutrodhor *et al.*¹⁴, identified the mango leaf ailment using neural network ensemble and support vector machine with accuracy of 87.5%. Pujari *et al.*¹⁵, employed image processing techniques with early signs and symptoms to identify and classify fungal disease affected agriculture and horticulture crops.

In ILSVRC2014 challenge, Szegedy *et al.*¹⁶ won a challenge by proposing the convolutional neural network called GoogleNet. The premier scope of this network is to limit the number of parameters in the network by introducing Inception module which replace fully connected layer by utilized the average pooling layer at the top of the convolution network. The runner up of the ILSVRC2014⁽¹⁷⁾ proposed the VGGNet and the objective of this network is to enhance the performance by increasing the depth of the convolution network. The architecture of the network consists of 16 fully connected convolution layer, the size of pooling and convolution is 2×2 and 3×3 respectively. It is very expensive to implement since it deals with a large number of parameters which require more memory.

The following research works focused on the classification of microbes by employing convolutional neural network. The work of Dong *et al.*¹⁸, utilizing a wide-field microscope and recorded larval zebra fish brain Z-stack pictures, suggested a method to identify cells that contain Tyrosine Hydroxylase. A supervised max pool convolutional neural network was utilized to recognize cellular pixels that had been preselected by a support vector machine classifier in this study.

Malon *et al.*¹⁹, utilized already learned features of convolutional neural network along with manually designed nuclear features to identify all mitotic figures from the hematoxylin and eosin-stained tissues. Tahir *et al.*²⁰ suggested a CNN model to classify five species of fungi, including *Versicolor*, *Eurotium*, *Cladosporium*, *Penicillioides*, *Restrictus* with a 94.8% accuracy.

The main benefaction of this work is to propose an efficient method for recognition of *Aspergillus niger* fungus, which is based upon pre-trained model such as VGG16Net and proposed algorithm named as Modified Backpropagation Algorithm using Inertia

and Legendre Polynomial (MBPI – LP). It demonstrated the effectiveness of Principal Components by applying PCA for global features determined from fully connected layers. The experimentation was done on features derived at various levels of fully connected layer and showed that FC6 determined features best for classification when compared to FC7 and FC8. The rest of the paper contains materials and methods followed by results and conclusion.

Materials and Methods

Image Acquisition

This section elaborates the category of fungus targeted in this research and how its images were collected. The setup of the system is also elaborated here.

A thorough study was conducted to know about different type of fungus infections which includes the study about its origin, colonies, nomenclature and occurrence of particular types of fungus. The fungi are classified as kingdom because fungi are microbes and are eukaryotes. Because of its unique characteristics, fungi have been placed separate from animal and plant kingdom. The statistics shows that in this world 7 00 000 to 5 million types of fungi are available.²¹ This work abundantly focused on detection of *A. niger* because of its need in multipurpose industrial applications and threats to the human life due to the disease causing potential.

Lyophilized culture of *A. niger* (MTCC ID: 8652) was obtained from Microbial Type Culture Collection, Chandigarh, India. These lyophilized cultures were dissolved in sterile distilled water and were cultivated as broth culture in Potato Dextrose Broth and recovered after the prescribed incubation period of 7 days. After 7 days at 25°C, the cultivated cultures were inoculated on Agar plates containing 10 different medium viz. CzapekDox, Potato Dextrose, Sabouraud Dextrose, V8 Juice, Wickerham, Lignocellulose, 2% YES medium, 0.7% YES, Malt extract, and Emmon's culture.²² Fungal growth requires a slightly acidic environment. Hence, the medium were tested for pH and were adjusted accordingly with 1N HCl. After 7 to 11 days of incubation, the mycelia formed on the plate were taken on a glass slide and stained with Lactophenol Cotton Blue dye, for observation and documentation. Binocular research microscope (Labomed, Vision-2000, India) equipped with trinocular head fitted with digital camera was used for photo documentation of micromorphology of

the fungus at 400 X resolution.²³ Totally 1500 sample images were obtained at 400 X resolution. Some of the sample fungi images used for classification are shown in Fig. 1(a) – (c).

Principal Component Analysis (PCA)

PCA is a statistical technique for feature extraction and most powerful unsupervised technique for dimensionality reduction. It is a procedure used to transform the dependent correlated variables into independent uncorrelated variables and it is termed as Principal Components. Generally, the first component covers the maximum variance of the data, followed by the succeeding components which account the remaining variance of the variables as well. The procedure for PCA is as follows:

Step 1: Normalize the dataset by subtracting the means of each and every attribute

Step 2: Compute the Covariance matrix

Step 3: Compute the Eigen values and Eigen vector

Step 4: Find the diagonal matrix whose diagonal elements are square root of the Eigen values.

Step 5: Compute the factor structure

Step 6: Compute the factor score where each and every column of the output matrix are called principal components

Deep CNN Features

To obtain CNN features, first, select the suitable CNN model. The robust model widely used in much visual recognition task is VGG16Net. Since the VGG16Net model showed its promising result in many computer vision applications, this work adopted the deep VGG16Net model and interestingly the domain is not trained by fungi image dataset.

The architecture of VGG16Net model consists of 5 blocks of convolutional layer and each convolution followed by max-pooling layer. There are 3 fully connected layers. The convolution layers generate features by convolving the input using the bank of linear filters and it is followed by rectification layer used to apply non-linearity on the feature maps. The

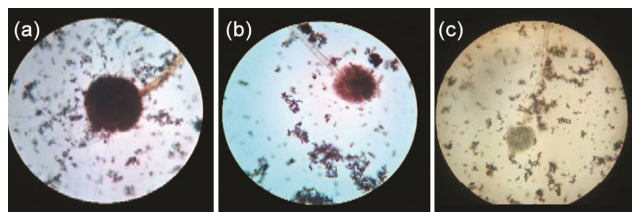


Fig. 1 — Sample images of *A. niger* (a–c)

feature maps of last convolution layer are smaller in size such as 14×14 and larger in number such as a number of features are 512 whereas the first convolution layer are larger in size 224×224 and smaller in a number of feature such as 64. Finally, the last three fully connected layers process the features with linear operation followed by ReLU activation function. The first two FC layers return 4096-dimensional vector while last FC layers yield a vector of dimension 2622. The convolution layers yield low, medium and high-level features along with their spatial information while the fully connected layers return global features which discard spatial information.

Inertia

Inertia’s law states that,

- 1) The body or object at rest remains at rest.
- 2) If an object moves in a straight line at a uniform speed, then keep moving in a straight line.

Inertia’s momentum is computed using the following formula,

$$L = 1/2 \times mr^2 \quad \dots (1)$$

where, L is inertia, m is mass and r is the distance between the object and the axis of rotation.

Legendre Polynomial

In the field of Mathematics, Legendre Polynomial is a system of complete and orthogonal polynomial that has a vast number of properties useful for many applications. In the field of Artificial Intelligence, Legendre Polynomial based Neural Network was proposed for function approximation problem because of its recurrence relationship. It involves less computation when compared with trigonometric equations. The Legendre Polynomial are defined by $L_n(x)$ where n is the order and the Value of x lies between -1 to +1 is the argument of the Polynomial. The first few Legendre Polynomials are:

$$\begin{aligned} L_0(x) &= 1 \\ L_1(x) &= x \\ L_2(x) &= \frac{1}{2} (3x^2 - 1) \\ L_3(x) &= \frac{1}{2} (5x^3 - 3x) \quad \dots (2) \end{aligned}$$

The recursive formula is used to generate higher order Legendre Polynomials are expressed as

$$(2n + 1) \times L_n(x) = (n + 1)L_{n+1}(x) + nL_{n-1}(x) \quad \dots (3)$$

Existing State-of-the-art Techniques

Softmax Classifier

In general, hinge loss and squared hinge loss are used commonly when trained machine learning or deep

learning models. But nowadays, there is widely used another loss function and classifier in deep learning models are cross-entropy loss and softmax classifier. It allows quantifying how much good or bad the classification function correctly classified the data points in the dataset. The last layer in the deep learning models are softmax classifier that returns the probability scores for each class label.

Step 1: Input (a): Feed each and every input vector to network a_i from the dataset A where A belongs to $R^{m \times n}$.

Step 2: Intermediate layer output: compute linear operation between layers and apply activation function to find the output of the layers. For each $g = 2, 3, \dots, G$
Calculate

$$h^g = W^g s^{g-1} + b^g \quad \dots (4)$$

where, h is the intermediate layers, W is the weight matrix and G is the number of layers.

Step 3: Apply activation function to get the output of the intermediate layers

$$s^g = \sigma(h^g) \quad \dots (5)$$

where, σ is the relu activation function.

Step 4: Output Error: calculate the vector,

$$\delta^G = \omega_h G \odot \sigma'(S^G) \quad \dots (6)$$

Step 5: Backpropagate the error:

For each $g = G - 1, G - 2, \dots, 2$ compute

$$\delta^g = W^T \delta^{g+1} \odot \sigma'(S^g) \quad \dots (7)$$

Step 6: Output: Loss functions gradient is computed by

$$\frac{\partial G}{\partial W_{jk}^g} = s_k^{g-1} \delta_j^g \quad \dots (8)$$

Backpropagation (BP) Algorithm

The backpropagation algorithm is the widely used algorithm to implement supervised learning tasks. The algorithm approximates a uni-directional mapping from r-dimensional input space to s-dimensional output space where r and s are the numbers of input and output variables. Learning is accomplished by updating the weight connection between the layers using error feedback from the training samples.

BP Algorithm with the Single Term

In general, the BP algorithm contains a single term in the weight updation rule called Learning rate. The learning rate is used to fasten the convergence speed of the network. The number of epochs taken to reach

the minimum error depends upon the value of the learning rate. If it is maximum, then there is a possibility of missing the global minima value. Network learning becomes very slow if it is a small value. Finding the optimum value for the learning rate is the most important task in the learning process.

BP algorithm weight updation rule using the single term,

$$U(t + 1) = U(t) + \alpha \times \delta_j \times x_i \quad \dots (9)$$

where, U is the weight vector, α is the learning rate, δ_j is the gradient, and x is the input vector.

BP Algorithm with the Two Terms

In a two-term BP algorithm, the first term is the Learning rate and the second term is the Momentum. Momentum is used to accelerate the speed of the network convergence and increase its generalization capability. The addition of momentum term smooths out the oscillatory behavior and increases the speed of the network.

Weight updation rule of BP algorithm with learning rate and momentum as,

$$U(t + 1) = U(t) + \alpha \times \delta_j \times x_i + \beta \times U(t - 1) \quad \dots (10)$$

where, α is the learning rate and β is the momentum factor.

Proposed Method: BP Algorithm with the Three Terms

The major limitation of the BP algorithm with one and two terms is the presence of temporary, local minima resulting from the saturation behavior of activation function and slow convergence rate. The third term of Legendre Polynomial given in Eq. (2) substituted in Eq. (1), given the following equation,

$$Inertia = (1/2 \times (1/2(3\gamma^2 - 1)) \times (U(t) - U(t - 1))^2 \dots (11)$$

where, γ is the inertia term expanded in the form of Legendre Polynomial third term given in Eq. (2) and r in Eq. (1) represented as the difference between current weight and old weight of the hidden layer respectively. As a third term in the weight updation rule between input layers to hidden layer, Eq. 11 is used.

Proposed Algorithm

The set of algorithms for the modified VGG16Net model using PCA and Legendre Polynomial are described in the following section:

Algorithm 1: Feature Extraction using VGG16Net

Input:

Deep model used for feature extraction is VGG16Net. Input images from Gallery with class

labels $((I_1, C_1), (I_2, C_2), (I_3, C_3), \dots, (I_n, C_n))$, where I is the input image and C is the corresponding label
Output:

Features for Prediction

Process:

1. For $k=1$ to N :
2. Model = VGG16Layers()
3. Features = extract (I_k , layer = FC6 or FC7 or FC8)
4. R_k = vectorize (features)
5. $R = R + R_k$

where, N is the total number of Input images, R is the matrix of input vectors for principal component analysis.

Algorithm 2: Extracting Principal Components

Input:

Input obtained by applying Algorithm 1

Output:

Principal Components

Process:

1. Normalize the input vector R

$$R' = \frac{r - \bar{r}}{\sigma} \quad \dots (12)$$

where, σ denotes standard deviation and \bar{r} is the mean.

2. Compute Correlation Matrix

$$cor(r_1, r_2) = \frac{cov(r_1, r_2)}{\sigma_{r_1} \sigma_{r_2}} \quad \dots (13)$$

where, $cov(r_1, r_2)$ is the covariance matrix

3. Calculate Eigen Value

$$Det|A - \lambda I| = 0 \quad \dots (14)$$

Compute Eigen Vector

$$[A - \lambda I] [V] = 0 \quad \dots (15)$$

Compute diagonal matrix whose diagonal factors of a diagonal matrix are the square roots of Eigen Values

4. Find the factor structure using the formula,

$$S = VL \quad \dots (16)$$

Where V is the Eigen Vector and L is the Diagonal Matrix

5. Find Coefficient Matrix B ,

$$B = VL^{-1} \quad \dots (17)$$

Compute factor score by

$$F = ZB \quad \dots (18)$$

where, Z is the Standard Score for the Dataset input vector X and obtained by using the formula,

$$Z_i = \frac{R_i - M}{SD} \quad \dots (19)$$

where, M and SD denotes mean and Standard Deviation respectively. From matrix Z_i keep m principal component analysis which covers 95% of variance.

Algorithm 3: Modified Backpropagation Algorithm using Inertia and Legendre Polynomial

Input:

Principal Components in the form of Input vector obtained using Algorithm 2.

Output:

Classification score

Process:

1. Divide the dataset into samples for training (70%) and testing (30%).
2. Feed the input vectors to the input layer from each training pattern set $T = \{R_i, O_i\}$ where R_i is the input vector and O_i is the output vector.
3. Compute net input for each hidden node using

$$h_j = Net_j = \sum_{i=1}^n u_{ij} \times r_i, j = 1, 2, \dots, q \quad \dots (20)$$

4. Calculate each hidden node's output and the sigmoid activation function $g(\cdot)$ expressed as,

$$g(h_j) = \frac{1}{(1 + \exp(-h_j))} \quad \dots (21)$$

Output node's net input can be calculated as,

$$Z_k = Net_k = \sum_{j=1}^m v_{jk} \times g(h_j), k = 1, 2, \dots, p \quad \dots (22)$$

Apply the activation function to nodes in the output layer to calculate its output.

$$g(Z_k) = \frac{1}{(1 + \exp(-Z_k))} \quad \dots (23)$$

Determine the output nodes gradient,

$$g'(Z_k) = g(Z_k) \times (1 - g(Z_k)) \quad \dots (24)$$

Compute the following error at the output nodes

$$\delta_k = (O_k - g(Z_k)) \times g'(Z_k) \quad \dots (25)$$

The error at a hidden neuron can be estimated using the formula,

$$\delta_j = g'(Z_k) \times \sum_{k=1}^m \delta_k \times v_{jk} \quad \dots (26)$$

Weight modification of the Output layer:

5. Updating the weight with the momentum term,

$$V(t+1) = V(t) + \alpha \times \delta_k \times h_j + \beta \times V(t-1) \quad \dots (27)$$

where, α represents learning rate and β represents the momentum factor.

Proposed weight updation for Hidden layer:

6. The proposed weight updation with the term's momentum and inertia:

$$U(t+1) = U(t) + \alpha \times \delta_j \times x_i + \beta \times U(t-1) + (1/2(3\gamma^2 - 1) \times (U(t) - U(t-1))^2) \quad \dots (28)$$

where, α and β represents the learning rate and momentum factor respectively. The new term γ is inertia's mass term extended by Legendre polynomial using Eq. (2), assigned with a tiny positive value and r is the distance between old weight $U(t-1)$ and new weight $U(t)$.

Steps 3 to 11 should be repeated until the accuracy of the testing samples is 98–100%.

Algorithm 1 is used to extract the features from fully connected layers in VGG16Net model such as FC6, FC7, and FC8 respectively. From this large number of features, Algorithm 2 is used to extract Principal Components which gives significant features more relevant for prediction. Since the input dimension reduces, the computational complexity is also getting minimized. For classification, Algorithm 3 represents a proposed algorithm called Modified Third Term Backpropagation employing Inertia and Legendre Polynomial. The training continues to bring down the Mean Square Error as 0.01.

Results and Discussion

On a PC with an Intel core I5, 7th Generation CPU @2.50 GHz and 4 GB RAM, the intended system was constructed using Python (version 3.5) in the Spyder environment. The experiment was conducted on real time collected 1500 samples of *Aspergillus* Fungus images with the help of biological experts. The performance of the proposed system is reported based on accuracy, precision, recall and F1 score.

The VGG16Net architecture has three completely connected layers: FC6, FC7, and FC8, with FC6 and FC7 returning 4096 features respectively. There are 2622 features in FC8 respectively.

The number of principal components obtained for features FC6, FC7, and FC8 respectively that covers 95% of variance of the dataset given in Figs 2–4. After applying PCA, the dimension of FC6, FC7 and FC8 dataset reduced to 763, 954, and 242 respectively.

The performance of the existing and proposed algorithm in terms of Epoch and time to reach the targeted mean square error is given in Table 1. By observing the value given in the Table 1, proposed algorithm takes least minimum number of epochs such as 8, 11, and 27 to reaches the targeted mean

square error when compared with the other existing algorithms.

The Deep Convolutional neural network with MBPI-LP gave better accuracy for the different level of features obtained from VGG16Net architecture and it is clearly depicted in Fig. 5 (a-c). The learning curve of proposed algorithm reaches the global minimum without any oscillation. Since the term

Inertia added as a third term, it keeps the convergence of the network in the same direction without getting stuck in the local minima. Additionally, the term stretched as Legendre orthogonal polynomial removes the correlation and makes the weight vector as an independent one that better accelerates the network for speed convergence.

Confusion Matrix

In machine learning classification problems, the table which is used to show the performance of the classification is called confusion matrix. In statistical classification problems the confusion matrix is also called error matrix. From the name it depicts that it is used to identify confusion in between the classes. It is the summary of prediction results of classification problems. The four parameters can be derived from confusion matrix such as:

True Positive (TP) - Both observation and prediction to be positive.

False Negative (FN) - Actual value positive but predicted it as negative.

True Negative (TN) - Both observation and prediction as negative.

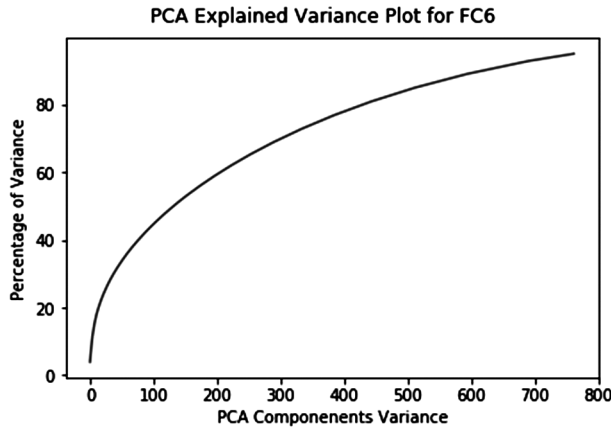


Fig. 2 — PCA Plot for FC6

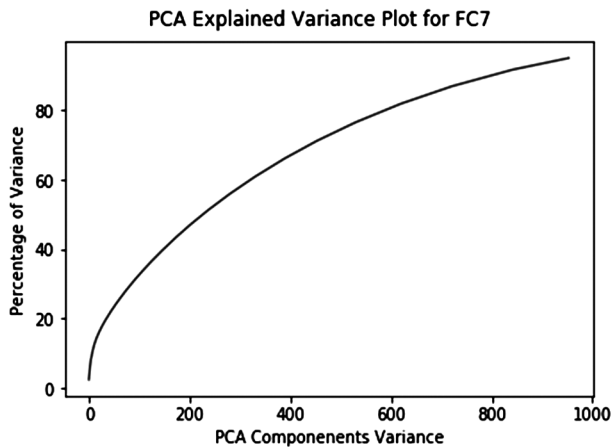


Fig. 3 — PCA Plot for FC7 Features

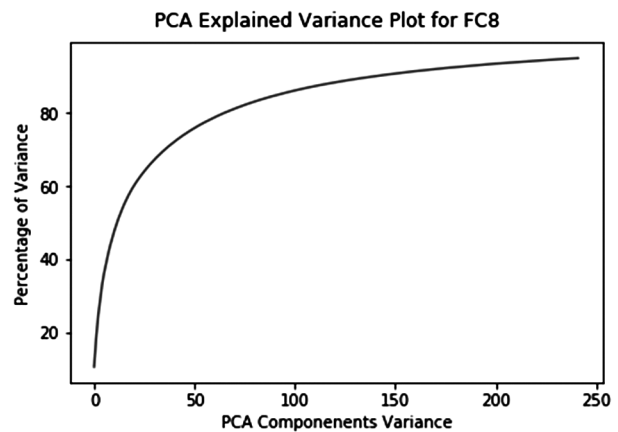


Fig. 4 — PCA Plot for FC8 Features

Table 1 — Simulation Results

Deep Networks	Layer/No. of features	Epoch	MSE	Time
VGG16Net-SBP	FC6/763	78	0.01	3 min 54 sec
VGG16Net-SBPM	FC6/763	96	0.01	4 min 48 sec
VGG16Net-Softmax	FC6/763	22	0.01	1 min 6 sec
Proposed VGG16Net-MBPI-LP	FC6/763	8	0.01	24 sec
VGG16Net-SBP	FC7/954	62	0.01	3 min 6 sec
VGG16Net-SBPM	FC7/954	60	0.01	3 min
VGG16Net-Softmax	FC7/954	83	0.01	4 min 9 sec
Proposed VGG16Net-MBPI-LP	FC7/954	11	0.01	33 sec
VGG16Net-SBP	FC8/242	112	0.01	5 min 36 sec
VGG16Net-SBPM	FC8/242	143	0.01	7 min 18 sec
VGG16Net-Softmax	FC8/242	33	0.01	1 min 39 sec
Proposed VGG16Net-MBPI-LP	FC8/242	27	0.01	1 min 18 sec

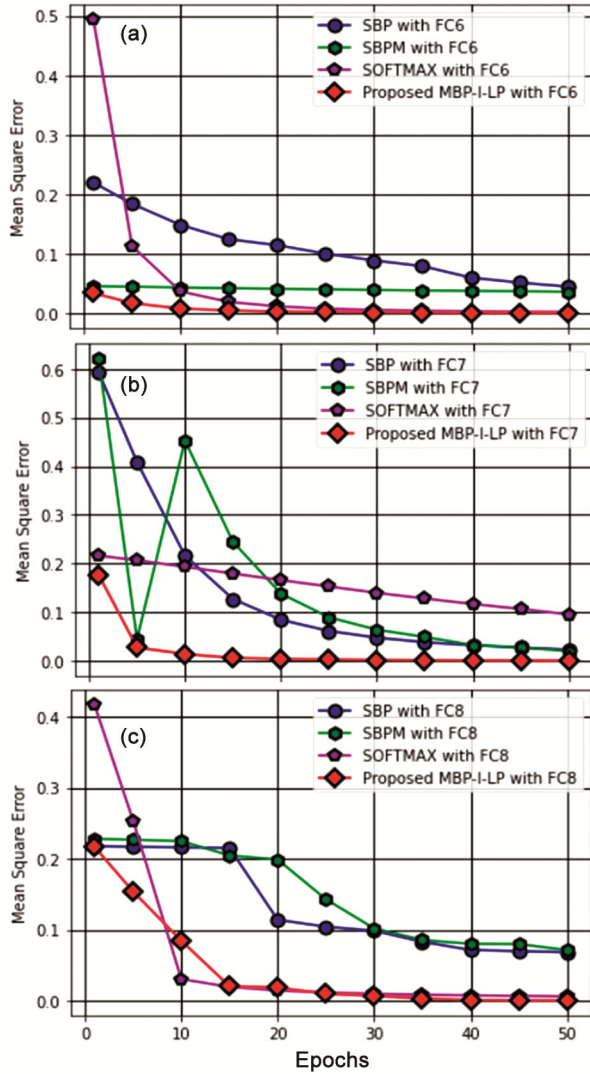


Fig. 5 — Learning Curve Plot for Proposed Vs Existing Algorithm for (a) FC6 Features (b) FC7 Features (c) FC8 Features

False Postive (FP) - Prediction is positive but actual value is negative

From these four parameters, three metrics can be calculated such as Precision, Recall and F1 Score.

From the Table 2, the recall value obtained by using VGG16Net-MBPI with LP is around 1 for all the test dataset. It showed that the *A. niger* positive and negative samples is at most classified correctly. This leads small number of FN in results. The precision value of VGG16Net-MBPI with LP is almost 1 for all of the experimental dataset. It demonstrates that positive samples labeled as positive are truly positive; implying that there is less FP. The F1 measure of the VGG16Net-MBPI with LP is around one for all the test dataset showed that the proposed classifier is flawless.

Table 2 — Average measures obtained from confusion matrix

Layer	Algorithm	Precision	Recall	F1-Score
FC6	VGG16Net-SBP	0.80	0.90	0.85
	VGG16Net-SBPM	0.83	0.93	0.88
	VGG16Net-Softmax	0.96	0.95	0.95
	Proposed VGG16	0.97	1.00	0.97
	Net-MBPI with LP			
FC7	VGG16Net-SBP	0.79	0.89	0.83
	VGG16Net-SBPM	0.81	0.91	0.85
	VGG16Net-Softmax	0.81	0.90	0.82
	Proposed VGG16	0.93	0.91	0.85
	Net-MBPI with LP			
FC8	VGG16Net-SBP	0.72	0.82	0.77
	VGG16Net-SBPM	0.68	0.78	0.73
	VGG16Net-Softmax	0.76	0.87	0.81
	Proposed VGG16	0.96	0.94	0.95
	Net-MBPI with LP			

Table 3 — McNemar’s Test

Classifier	McNemar’s Test	
	Statistics	P- Value
Prop. MBPI-LP Vs SBP	0.0000	0.0001
Prop. MBPI-LP Vs SBPM	0.00003	0.00051
Prop. MBPI-LP VsSoftmax	0.00004	0.0023

McNemars Test

The performance of two distinct classifiers on a single dataset can be analyzed using McNemar's test. The p-value of the proposed hybrid models lesser than 0.05 compared to the existing pre-trained models showed that there is a significant difference in terms of proportion of error exist in between the models. The results of the McNemar’s test are given in Table 3.

The p-value given in the Table 3 showed that there is a different proportional error in between the two different classifiers. Since the p-value of proposed MBPI-LP is lesser than 0.05 when compared with SBP, SBPM, and Softmax classifier. It showed the better performance of the Proposed MBPI-LP.

Comparative Analysis

The Backpropagation (BP) algorithm is the most popular supervised machine learning algorithm. The major problem reported about the BP algorithm is slow convergence rate and entrapment of local minima. Many researchers proposed the different techniques to overcome these problems. The following is the comparative study about the existing modified versions of the Backpropagation algorithm in the literature along with the proposed MBPI-LP.

Burse *et al.*²⁴ proposed the improved backpropagation algorithm to overcome the drawbacks of the existing traditional BP algorithm such as slow convergence speed and local minima

problem. They introduced the third factor called a proportional factor to overcome above stated issues. They tested the algorithm on benchmark binary classification dataset and showed that it takes 1000 epochs to reach the MSE of 0.0001 that is advanced than traditional BP.

Subavathi *et al.*²⁵ proposed the adaptive modified backpropagation algorithm to increase the convergence speed and minimize the error. The author proposed the optimized learning rate between input layer to hidden layer and hidden layer to output layer respectively. The value of the learning rate is determined by computing the differential linear and nonlinear errors of output layer and hidden layer separately. It has done experiment on the proposed method on XOR and Parity checking problem.

Amiri *et al.*²⁶ proposed the adaptive learning rate to overcome the limitation of backpropagation algorithm. Learning rate is the important parameter that decides how many epochs the model required to take to reach the global minimum point. Inappropriate value to the learning rate leads to the poor convergence rate. Hence author proposed the technique to adaptively set the value to the learning rate during training and it is depending on error curve changes. Three datasets have been used for the experimentation such as seeds, pima Indian diabetes and Glass. The training procedure terminates at its 100th epoch by reached its target mean square error.

Naga Ratnam *et al.*²⁷ optimized the traditional backpropagation algorithm by introduced the adaptive learning rate. The value of the learning rate can be computed as follows. The immediate and non-linear oversights for each neuron in the yielding layer are duplicated with a subordinate of the relating neuron's institution work. Thereafter, separated to get the flexible learning rate for the yielding layer.²⁷ The proposed technique applied and tested on diabetes and heart disease prediction dataset. They showed 2000 epochs for convergence.

The authors proposed the optimized backpropagation algorithm by fine-tuning the hyper-parameters such as learning rate and weight vector. The weight vector optimized during forward pass by introducing the Nguyen-Widrow method. The value of the learning rate getting increased or decreased during backward pass based on the direction of error surface.²⁸ They performed the experiment on the proposed technique to classify the file. Experimentation was done on 19 number of PDF file and it took 80 epochs for prediction.

Sornam *et al.*²⁹ proposed the modified backpropagation algorithm by introducing the new factor Inertia as a part of weight updation. Inertia has been expanded in the form of Chebyshev orthogonal polynomial. They applied the proposed algorithm in the standard benchmark binary and multi-class classification problems. They showed proposed algorithm effectiveness by comparing it with the SBP and SBPM.

Existing Technique 1⁽²⁴⁾ — Introduced the third term called Proportional factor

Existing Technique 2⁽²⁵⁾ — Adaptive learning rate-based error in between the layers.

Existing Technique 2⁽²⁶⁾ — Adaptive learning rate based on changes in the error curve.

Existing Technique 2⁽²⁷⁾ — Adaptive learning rate based on non-linear sighs on each neuron.

Existing Technique 2⁽²⁸⁾ — optimize the weight vector using Nguyen-Widrow method and adaptive learning rate based on the direction of error surface.

Existing Technique 2⁽²⁹⁾ — Proposed third term in the weight updation rule.

The comparison between the existing and proposed modifications in the SBP is shown in Fig. 6. The comparison depends on the number of features for predictions and epochs for convergence. Existing techniques were applied to the dataset that contains a finite number of features for prediction. The proposed MBPI-LP was tested on a real-time fungi image dataset holding enormous features for prediction. But the number of epochs taken by the MBPI-LP is minimal compared with the existing techniques. Hence the proposed algorithm performed better than the existing techniques.

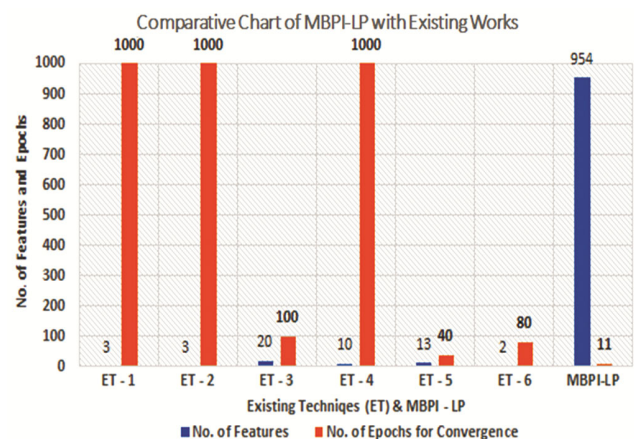


Fig. 6 — Comparative chart of Proposed Vs Existing modifications in SBP

Time Complexity

The worst-case time complexity of the proposed MBPI algorithm is obtained in terms of number of iterations performed.

$$O (EP \times \max (n \times q, m \times q, m \times p)) \quad \dots (29)$$

where n, q, and m represents the number of input, hidden, and output neurons respectively.

It is the time complexity for the proposed algorithm where (n × q) and (m × p) are the number of operations performed to compute the net input for hidden and output neurons respectively, whereas (m × q) is the number of times to compute the error at hidden neurons. EP is the total number of epochs taken during training for convergence. In existing softmax classifier also, the same sequence of operations are performed such as computing net input for hidden and output neurons by applying linear operation, activation function, calculate cross entropy loss, and backpropagate the error.

Hence, the time complexity of the existing softmax classifier is also same for the proposed one. But, when the proposed and existing algorithms based is compared on the epochs, the number of epochs taken by existing classifiers and proposed MBPI-LP can be computed as,

$$EP_{SBP} = \sum_{i=1}^3 EP_{SBP(i)} \quad \dots (30)$$

$$EP_{SBPM} = \sum_{i=1}^3 EP_{SBPM(i)} \quad \dots (31)$$

$$EP_{Softmax} = \sum_{i=1}^3 EP_{Softmax(i)} \quad \dots (32)$$

$$EP_{MBPI-LP} = \sum_{i=1}^3 EP_{MBPI-LP(i)} \quad \dots (33)$$

where, i = 1, 2, 3 represents FC6, FC7, and FC8 level of features from VGG16net architecture respectively.

According to the table,

$$EP_{SBP} = 252$$

$$EP_{SBPM} = 299$$

$$EP_{Softmax} = 138$$

$$EP_{MBPI-LP} = 46$$

Comparative time complexity of Proposed MBPI-LP with Existing classifiers:

$$\begin{aligned} O_{Existing} & (EP_{Existing} \times \max(n \times q, m \times q, m \times p)) \propto \\ O_{MBPI-LP} & (EP_{MBPI-LP} \times \max(n \times q, m \times q, m \times p)) \quad \dots (34) \end{aligned}$$

By substituting corresponding epoch values on both sides of the above equation returns the following conclusion,

$$O_{SBP} = 5.48(O_{MBPI-LP})$$

$$O_{SBPM} = 6.5(O_{MBPI-LP})$$

$$O_{Softmax} = 3(O_{MBPI-LP})$$

Hence, the proposed algorithm is 5.48 times faster than SBP, 6.5 times faster than SBPM, and 3 times faster than Softmax classifier respectively.

Conclusions

In this work, we proposed a deep learning based *A. niger* fungus detection. This work utilized a pre-trained CNN model for the extraction of features and proposed MBPI-LP used as a classifier. By considering the wealth of information returned from the activations of convolution layer, discriminative features were identified by employing an efficient dimensionality reduction technique called Principal Component Analysis. Through extensive experiment, here Modified Backpropagation Algorithm was employed using Inertia and Legendre Polynomial and its results are compared with existing softmax classifier of VGG16Net Architecture. The existing VGG16Net showed better accuracy only for the features obtained from the activation of FC6 layer whereas the MBPI-LP showed its superior performance for all the features returned from FC6, FC7 and FC8, respectively. These findings may find their application to various types of microbes available in the fungi kingdom.

References

- 1 Dimopoulos G, Almyroudi M P, Myrianthefts P & Rello J, COVID-19-associated pulmonary aspergillosis (CAPA), *J Intensive Care*, **1(02)** (2021) 71–80.
- 2 Gouzien L, Cocherie T, Eloy O, Legriel S, Bedos J P, Simon C, Marque-Juillet S, Ferré A & Bruneel F, Invasive aspergillosis associated with Covid-19: A word of caution, *Infect Dis*, **51(4)** (2021) 383–386.
- 3 Thompson III G R, Oliver A C, Peter G P, Thomas F P, Martin H, Jeffrey D J, Cornelius J C & Hong N M, Invasive aspergillosis as an under-recognized superinfection in COVID-19, *Open Forum Infect Dis*, **7(7)** (2020).
- 4 Machado M, Valerio M, Álvarez-Uría A, Olmedo M, Veintimilla C, Padilla B, De la Villa S, Guinea J, Escribano P, Ruiz-Serrano M J & Reigadas E, Invasive pulmonary aspergillosis in the COVID-19 era: An expected new entity, *Mycoses*, **64(2)** (2021) 132–143.
- 5 Lai C C & Yu W L, COVID-19 associated with pulmonary aspergillosis: A literature review, *J Microbiol Immunol Infect*, **54(1)** (2021) 46–53.
- 6 Dellière S, Dudoignon E, Fodil S, Voicu S, Collet M, Oillie P A, Salmona M, Dépret F, Ghelfenstein-Ferreira T, Plaud B & Chousterman B, Risk factors associated with COVID-19-associated pulmonary aspergillosis in ICU patients: a French multicentric retrospective cohort, *Clin Microbiol Infect*, **27(5)** (2021) 790.e1–790.e5

- 7 Schuster E, Dunn-Coleman N, Frisvad J C & Van Dijck P W, On the safety of *Aspergillus niger* – a review, *Appl Microbiol Biotechnol*, **59(4)** (2002) 426–435.
- 8 Asses N, Ayed L, Hkiri N & Hamdi M, Congo red decolorization and detoxification by *Aspergillus niger*: removal mechanisms and dye degradation pathway, *Biomed Res Int* (2018), <https://doi.org/10.1155/2018/3049686>.
- 9 Dighton J & White J F, *Classical Methods and Modern Analysis for Studying Fungal Diversity in the Fungal Community*, CRC Press, (2005) 213–234.
- 10 Blasco J, Aleixos N & Moltó E, Computer vision detection of peel defects in citrus by means of a region oriented segmentation algorithm, *J Food Eng*, **81(3)** (2007) 535–543.
- 11 Mertens K, De Ketelaere B, Kamers B, Bamelis F R, Kemp B J, Verhoelst E M, De Baerdemaeker J G & Decuypere E M, Dirt detection on brown eggs by means of color computer vision, *Poult*, **84(10)** (2005) 1653–1659.
- 12 Tahir M W, Zaidi N A, Blank R, Vinayaka P P, Vellekoop M J & Lang W, Detection of fungus through an optical sensor system using the histogram of oriented gradients, *IEEE Sens J* (2016) 1–3.
- 13 AsefpourVakilian K & Massah J, An artificial neural network approach to identify fungal diseases of cucumber (*Cucumis sativus* L.) plants using digital image processing, *Arch Phytopathol Pflanzenschutz*, **46(13)** (2013) 1580–1588.
- 14 Sutrodhor N, Hussein M R, Mridha Md F, Karmokar P & Nur T, Mango leaf ailment detection using neural network ensemble and support vector machine, *Int J Comput Appl*, **181(13)** (2018) 31–36.
- 15 Pujari J D, Yakkundimath R & Byadgi A S, Identification and classification of fungal disease affected on agriculture/horticulture crops using image processing techniques, *IEEE Int Conf Comput Intell Comput Res* (2014) 1–4.
- 16 Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V & Rabinovich A, Going deeper with convolutions, *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*, (2015) 1–9.
- 17 Simonyan K & Zisserman A, Very deep convolutional networks for large-scale image recognition, *ArXiv org*, (2014) 1409–1556.
- 18 Dong B, Shao L, Da Costa M, Bandmann O & Frangi A F, Deep learning for automatic cell detection in wide-field microscopy zebrafish images, *Proc Int Symp Biomed Imaging* (2015) 772–776.
- 19 Malon C D & Cosatto E, Classification of mitotic figures with convolutional neural networks and seeded blob features, *J Pathol Inform*, **4** (2013).
- 20 Tahir M W, Zaidi N A, Rao A A, Blank R, Vellekoop M J & Lang W, A fungus spores dataset and a convolutional neural network based approach for fungus detection, *IEEE Trans NanoBiosci*, **17(3)** (2018) 281–290.
- 21 Gautam A K, Verma R K, Avasthi S, Bohra Y, Devadatha B, Niranjana M & Suwannarach N, Current insight into traditional and modern methods in fungal diversity estimates, *J Fungus*, **8(3)** (2022) 226.
- 22 Atlas R M, *Handbook of Microbiological Media* (CRC press) 2004.
- 23 James C & Natalie S, *Microbiology A Laboratory Manual* (Pearson Education) 2014.
- 24 Burse K, Manoria M & Kiran V P S, Improved back propagation algorithm to avoid local minima in multiplicative neuron model, *Adv Inf Commun Technol*, (2011) 67–73
- 25 Subavathi S J & Kathirvalavakumar T, Adaptive modified backpropagation algorithm based on differential errors, *Int J Comput Sci Eng Appl*, **1(5)** (2011) 21.
- 26 Amiri Z, Hassanpour H, Khan N M & Khan M H M, Improving the performance of multilayer backpropagation neural networks with adaptive learning rate, *Int Conf Adv Big Data Comput Data Commun Syst*, 2018, 1–4
- 27 Naga Ratnam A, Narayana G, Yamini G, Veera Giri Nivas N, A predictive analysis of heart diseases and diabetes using adaptive modified backpropagation, *Int J Eng Technol*, **7(3.6)** (2018) 352–357.
- 28 Andayani U, Nababan E B, Siregar B, Muchtar M A, Nasution T H & Siregar I, Optimization backpropagation algorithm based on Nguyen-Widrom adaptive weight and adaptive learning rate, *Int Conf Ind Eng Appl*, 2017, 363–367
- 29 Sornam M & Vanitha V, An improved third term backpropagation algorithm–inertia expanded Chebyshev orthogonal polynomial, *J Intell Fuzzy Syst*, **37(3)** (2019) 3795–3804.