AN EFFICIENT CONVOLUTIONAL NEURAL NETWORK FRAMEWORK FOR FUNGUS RECONGNITION AND CLASSIFICATION USING TRANSFER LEARNING

ABSTRACT

Fungus is a vital ecosystem microorganism that plays a crucial part in the decay of organic matter and it also causes harm to human health and food to make them susceptible to infection. To meet the task of classifying and identifying the fungus, both computer vision and conventional methods have been employed. CNN has been utilized in the field of image recognition problems and also powerful in classification. But convolutional neural network requires a plenty of training data, high computational cost and they are quite slow to train. In this framework, transfer learning technique is employed that assists in reducing the time taken for training by reusing the weights of a previously trained source network. Transfer learning is versatile and scalable permitting the pre-trained network models to be used directly as feature extraction and it can be combined with completely new network models. It includes employing models of the neural network trained on some problem as the base point for another relevant problem. In this work, we have built a framework that performs fungus detection and classification based on similar order of fungus using transfer learning. INCEPTIONV3, RESNET50 and VGG16 pretrained network models are used as the source models from where the transfer learning is applied. The dataset used is Mushroom Classification-Common Genus's images taken from Kaggle. The dataset contains nine classes of fungal mushrooms present in the Northern European areas. The proposed model achieves an improve in accuracy of 3.5% compared to the typical CNN model used for fungus detection and classification. Our model gives a maximum accuracy of 98.3% when VGG16 is used as the base source model. From the results, it is observed that by using transfer learning, the time spent in training the model is less compared to training it from scratch.

I. INTRODUCTION

Deep Learning is a sub-field of machine learning, where the algorithms are impressed by the brain's structure and output known as Artificial Neural Networks (ANN) [31]. It's a form of computing that emulates the approach of learning that people in general utilize to realize different kinds of information. Individuals learn by developing an understanding of the situation and environment. Similarly, Deep Learning algorithms develop associate degree abstract illustration of the input data, that permits them to form predictions regarding the data. Deep learning should be seen as the simplest form to automatize predictive analyzing. Deep Learning finds application in computer vision, language process, medical image analysis, recommendation systems, etc.,

Neural Networks are collection of algorithms that supports the brains of human. They consider patterns by deciphering sensory knowledge through machine perception. Numerical vectors which can be used to convert data like pictures, sound, time series or text are the patterns which they find. A neural network sometimes involves an oversized variety of processors running parallelly and organized into layers. The raw input is obtained at the primary layer. Every further layer obtains the output from the previous layer. The output of the system is produced in the final most layer. Industrial applications [30] of those technologies typically concentrate on determining complicated signal process or pattern recognition issues like recognition of handwriting for cheque process, transcription of speech to text, exploration of oil results, climate forecasting and biometric authentication.

A Convolutional Neural Network [26] is a category of DNN, generally used for visual image processing. Multilayer perceptrons are regularized variants of CNNs. It is sometimes called as fully connected neural networks for its completely connected property of neurons linking from previous to next layers. The "fully-connectedness" of neural networks make them vulnerable to data overfitting. Usual methods of regularization incorporates applying to the loss obtained some sort of magnitude measure of weights. CNNs however use a distinct technique for regularization: it takes benefit of the data hierarchical model and assemble a lot of complicated patterns. So, CNNs are at the lower end, on a scale of complexity and connectedness. They (ConvNet) are additionally famed as SIANN, supported its shared-weights design and translation invariance functionality.

Convolutional networks are influenced by evolutionary processes where the pattern of communication between neurons matches the visual cortex of the animal. Human cortical neurons react to stimuli only in confined section of the field of vision called as the receptive field. The receptive fields of various neurons partly overlap so that they seal the whole field of vision. Compared with other image classification algorithms, CNN comparatively uses very little pre-processing. It implies that the neural network knows about the filters which were hand-engineered in ancient algorithms. This independence from previous information and initiative of human in designing of features is a great benefit.

CNN has been flourishing in analyzing visual imagery data. Though CNN is powerful in image classification, it also has some disadvantages. Standardization of hyperparameters is non-trivial in CNN. For efficiency, the scale of the weights of the network and updates of weights is incredibly necessary for evaluation. It won't be a concern if the features are of equivalent form (words,pixels). Nevertheless, if the features are heterogeneous as in several kaggle datasets, updates and weights all can air completely variant scales. So, standardization of inputs has to be done in some way. It needs a big dataset to work on. CNN would require large dataset to produce a great accuracy model with less loss. Convolution is considerably a slow operation than maxpool both reverse and forward. Every network takes for much longer time if the neural network is very big. CNNs have various applications in the fields of video and image recognition, classification and analysis of images, recognition of characters and recommender systems. Since CNN would require large data, transfer learning can be used for smaller datasets. Transfer learning is beneficial when the user has a meagre knowledge for a brand new domain that is to be handled by a neural network and there is an enormous pre-existing knowledge pool which will be transferred to your problem. So when there are only 1,000 images of fungus, by feeding into an existing CNN such as ResNet, trained with quite 1 million images, we can gain tons of mid level and low level feature characteristics. A neural network gains information once trained on a dataset employing a set of hyperparameters, that are parameters whose values are set before the learning method begins. The knowledge gained refers to the weights, connections and other hyperparameters of the neural network, that turn out the required results for the dataset. There will be a myriad set of parameters for every problem, called as the parameter space. Choosing these parameters manually will be toilsome as a result of the parameter space which is usually large and selection is completed by trial and error. The user also will ought to pay hours or maybe days training each selected network, betting on the complexness of the neural network.

Transfer learning is a methodology where the knowledge (weights) gained by a neural network for one task is transferred to a different network that is needed to perform the same task, or maybe completely a different task. Due to the transfer of parameters, the network doesn't have to be compelled to be trained from scratch. The knowledge from the previous task will be reused in some manner for a brand new task. This reduces the time it will take to make the neural network and conjointly the LoC of program.

The pretrained network model is indeed a stored neural network which has been already trained on such a wide dataset, usually on an image categorization task of a big scale. Pretrained network models are utilised as it is or transfer learning is employed for customizing the network model for a specific task. Using the concepts, the prior neural network has learnt to derive useful features from current data. Introducing a new classifier to the pretrained network model which would be trained from start can reuse the features already acquired for dataset. You needn't train the whole network model.

Convolutional base includes features that are generally needful for the classification of images. But, the pretrained network model's final classification component is unique to the original task, consequently unique to the collection of classes upon which network model has been trained. Unfreeze some frozen model base's upper layers and concurrently train the recently introduced classifier layers as well as the base model's final layers. It helps one to fine tune the interpretations of the high order features in base source model to have them most relevant for particular task. Correlations are present between base reference and destination tasks in Transductive Transfer Learning [28], however their underlying realms are totally variant.

Transfer Learning varies from conventional machine learning because it uses the pre-trained models trained on a dataset to be utilized for some process or problem. Transfer learning is more dependent on the dataset we use. Knowledge from source models can be used to boost learning in the target task [29]. Apart from providing capabilities to reprocess already-built models, transfer learning may assist learning the target task in the following ways:

- Improved baseline performance: Increasing the information gained of the isolated learner (also known as an ignorant learner) with information gained from a source model, the baseline output and regularization would possibly improve due to this knowledge transfer.
- Model-development time: Utilizing knowledge from a source model may also facilitate in totally learning the target task, as compared to a target model that learns from scratch. This, in turn ends up in enhancements of overall time taken to develop/learn a model.
- Improved final performance: Higher final performance may well be earned by leveraging transfer learning.

Transfer learning leads to reduction in LoC. Proposing a model to classify fungal mushrooms based on their characteristics using transfer learning would be useful in medical applications. So, transfer learning would make image classification visibly higher than time taking convolutional neural networks, in the view of accuracy gain and elapsed time. This was the central motivation in developing a framework that performs transfer learning by choosing the most similar source model.

Fungi perform an important part in ecosystem harmony. They colonize much of the earth's ecosystems, preferring deep and humid environment. Many fungi kingdom members, however rise on woodland surface wherever the deep and moist surroundings are wealthy in rotting debris created from animals and vegetation. Fungi play a key role in such ecosystems as symbiotic organisms and fabricators, allowing members of the other realms to be provided with sufficient nutrition. On the other side, fungus can cause diseases too. Most fungi aren't dangerous, however some sorts are often harmful to health. It can cause common skin disease like rash, lung infections and some bloodstream infections that are often deadly. Therefore, we tend to detect and classify similar fungal mushrooms using transfer learning which can be used in medical applications for finding fungi causing same type of diseases. CNN facilitate in image recognition and classification, but they are extremely time-consuming. Though CNN is a widespread topic, reducing the time consumption would be a significant achievement.

In medical research, the fungus indications and signs are non specific for incredibly large fields resulting in difficult and daunting task of fungal identification and classification. The aim is to establish a transfer learning method based on CNN to identify fungus and recognize various types of fungus by identifying the similar mushrooms based on their order to which it belongs to and compare the metrics. A lot of time is spent in classification task when CNN is employed. Therefore we use the concept of transfer learning where the knowledge learned in one task is reused in another task in the same domain. Transfer learning improves the training time and also regularizes better than CNN. Hence, in this work we address the use of transfer learning for fungus classification.

The paper is organized as follows. Section II consists of a survey of papers that are related to the topics discussed above. Section III describes the proposed system that briefly delineates the architecture of the transfer learning framework. Section V explains the methodology in the framework and analyses the performance of the framework after evaluation. Section V outlines conclusion regarding our work and future work that could be undertaken to enhance the work done.

II. RELATED WORKS

An elaborate study of various literatures pertaining to CNN based approach, Pre-trained model method and Develop model method in Transfer Learning, Multi class classification are explained in this section. Also, we discuss about CNN based approach followed by different transfer learning approaches. An overview of various methods of performing transfer learning are also discussed.

CNN is a sort of ANN and is a deep learning approach, a CNN based architecture which [1] comprises of various layers such as pooling, convolution, fully connected layers, and functions for activation are built for the customized fungus dataset so as to advance the state of the art in the classification of fungus. Different pictures of complicated microbes are present in this dataset by extricating sample of images from defiled foods, documents and colonies of fungus incubated in the laboratory. Those pictures contains images of fungus spores and dirt of five kinds. Optical sensor system are likewise used for acquiring these images. 40,800 images are utilized to develop fungus data set for fungus detection and fungus classification. Finally the proposed CNN based approach demonstrated the promising result of 94.8% accuracy. A CNN model is designed [2] with autoencoders, sparsecoding and RBM, which contains pooling, convolutional, dense layers with RELU as an activation function. The dataset contains 159 coffee leaf images of 12.1MP. The dataset is previously examined by the expert and the results of the trained CNN is evaluated and tested. The dice coefficient with CNN resulted an average and median of 0.59 and 0.61 respectively. Morphological erosion is used with CNN to get an average and median of 0.79 and 0.81 respectively. In this process, overfitting is prevented by using large dataset. Dice coefficient is employed to compare the similarity of two images. After crossing all the layers, the network gets reduced to two neurons, thus determines whether diseased or normal.

The authors [3] have analyzed many emerging patterns of transfer learning. Transfer learning is categorized into three various settings: unsupervised transfer learning, transductive transfer learning and inductive transfer learning. In addition, each methodology of transfer learning could be categorized into four contexts on the basis of "what to transfer" in learning. They incorporate instance-transfer approach, the feature-representation-transfer technique, the parameter-transfer technique, and relational-knowledge transfer technique. The strategies presume the destination domain is linked to the chosen reference domain.

Pre trained approach is a transfer learning approach in which pre-trained model is used as the baseline point for another problem. Some pre-trained network models are VGG16, RESNET50, INCEPTION V3, RESNET V2. The researchers presented one efficient method which uses transfer learning for plant recognition and leaf classification. This [4] uses network architectures which are pre-trained ones. The custom leaf data undergo image augmentation using augmentation techniques such as flipping, multiplying, transform and contrast normalization. This pre-trained model is used directly in the proposed work to learn features of the leaf. The training process is done using 6 different CNN models InceptionV3, InceptionV4, Resnet-Inception, ResnetV1, ResnetV2, Mobile net. The networks are processed with two datasets called Flavia which has 32 classes, leafsnap which has 184 classes shows 99.6% and 90.54% accuracy respectively.

An adversarial language transfer learning platform for speech recognition with low resources is proposed [5] in which the acoustic model is trained by SHL model (Shared Hidden Model) to guarantee common layers can acquire language functionality via an adversarial method of learning. As the shared features may degrade the performance, the methodologies like cross-lingual knowledge transfer learning, multilingual training and SHL model are used. The advantage is that all destination models result in better exposure of the features and better instantiation by transfer learning. The datasets used are IARPA Babel datasets and by utilizing the features transferred from SHL network there is an improved performance of 10.1%.

Usage of a smart method of fault diagnosis, equipped with labeled data failed to identify unlabeled data due to disparity in data distribution. So using the transfer learning, author proposed a new intelligent method [6] called as deep convolution transfer learning neural network which includes two parts: they are adaptation of domain and recognition of condition. The methodologies used are feature extraction in which the number of features can be reduced and the module condition recognition has been attained by 1D CNN. In domain adaptation, domain classifying and discrimination discrepancy is done. The proposed network attained an accuracy of 89%. The effectiveness of the above proposed method is then verified.

A CNN based method for faster recognition of fruits [7] using transfer learning is proposed to overcome long training time and over fitting. Over fitting means the fluctuations and noise in the data used for training, to overcome this, transfer learning technique has been utilized. Global depth wise convolution is adapted to enhance the improvement of performance. Fruits-360 data set containing 55244 images across 81 classes is used. For thinner factor 0.5, the performance recorded by vanilla model is 0.9598 and performance produced by the proposed model is 0.9806. Therefore accuracy obtained is higher than the accuracy obtained by vanilla model.

A predictive modeling problem is solved by develop model approach. In this approach a skillful model is developed which is better than the naive model.

In [8], the authors projected feature based transfer learning supported distribution similarity. The decrease of distribution discrepancy is achieved because of the options of two domains area unit reweighed with the distribution similarity between supply and target domains, as this aims at the partial overlap between two totally different domains. The methods used are unit kullback–leibler divergence, feature completion on similarity. The projected methodology is capable of effectively enhancing the prediction operate's accuracy. The above method has obtained a precision of 0.8113 and this methodology is performed on facebook and sina micro blog for predicting the effectiveness of the work done.

Image acquisition technique based on customized sketch is proposed with deep convolution neural network [9] employing transfer learning to give a conceptual featured fine-grain image. The general methods used for feature extraction are clustering, euclidean distance method. The user history feedback is obtained with hand drawn picture which is given as input for the proposed model. By using this model, the features are tuned which leads the network model to learn the customized conceptual knowledge also. For feature extraction, eigen vectors are used. The acquired generic method has attained significantly higher accuracy of mean. The experimental results state that the model on the Flickr15 K dataset has a high generalization capability with the mean average accuracy as 0.64.

A neural network framework [10] for extremely precise system malfunction diagnosis is developed. By wavelet transformation the sensor data are converted to obtain time frequency distributions and machine health monitoring is also done. The methodologies used are time-frequency imaging, short time fourier transform and continuous wavelet transform. The machine learning fault diagnosis is done using pipeline and generalization on three datasets with gear boxes and bearings whose dataset sizes are of 6000, 9000, 5000 respectively and the output obtained with gear box dataset achieved a great improvement from 94.8% to 99.64%.

Transfer learning network for system diagnosis based on sparse auto enoder is suggested. Sparse auto encoder which has three layers [11] extracts raw data features and to reduce the discrepancy penalty of training and testing features, the technique called maximum mean discrepancy is applied. The testing dataset is motor bearing dataset. Among prediction accuracy and the standard deviation, a clear linear relation is observed. This model achieved a great improvement on the dataset and the accuracy achieved is high as 99.82% .

Transfer learning is classified into three categories based on the source and target domain such as Inductive, Transductive and Unsupervised transfer learning [22][23]. Inductive transfer learning [22] happens while source and the destination tasks are variant regardless of any domain. When an abundance of labeled data are available, inductive learning becomes similar to Multitask learning. Multitask learning attempts to acquire the base and destination task at the same time even in the case of them being different. Self-taught learning is another one when there are no availability of labeled data. It uses the unlabeled data from another base task to improve predictions in destination task. Transductive transfer learning [22] happens when the base and destination tasks are same where their domains are different. More source data with no target data will be available in this case. Translated learning is a type of Transductive learning when feature space of base data and destination data are different and are linked to migrate the knowledge. Unsupervised transfer learning [22] is said to happen when knowledge is transferred from multiple different source tasks to multiple similar target tasks. With no labeled data in destination and source domain, this setting aims to solve unsupervised issues like clustering and reduction of dimension. The knowledge is transferred from multiple different source tasks to multiple similar target tasks in unsupervised transfer learning.

Using unsupervised transfer learning, a multi-scale convolution sparse coding model is proposed [12] which can be used for biomedical applications. The methodologies adapted are feature extraction and color decomposition. This MSCSC gives less feature redundancy with best performance and this adds to improved scale-specific filters and decreased replication across scales. This MSCSC approach effectively studies filter banks at various scales along with enforced scale-specificity of learned patterns and offers an unsupervised solution for acquiring exchangeable basic information by fine-tuning it to specific tasks and obtaining the successful result.

CNN architectures such as CifarNet, AlexNet, GoogleNet architectures are used for ILD classification. The ImageNet-trained models [13] are used which yields early promising results. Multiclass categorization is used because the volume of training data classified is extremely inadequate, as compared to binary class categorization of the lymph node dataset. When adding ImageNet pre-trained AlexNet to PASCAL dataset, the performance of semantic 20-class object recognition and optimization functions increases dramatically over existing methods which do not use deep CNNs.

The author uses active inquiry techniques by [14] developing a system which incorporates active learning along with transfer learning, by doing so, a great improvement in generalization ability is seen in classification models. The methodologies used are MCLU-enhanced clustering based diversity margin and sampling-closest support vector. This active learning can cut redundancy in samples and improve the accuracy of the classifier.

A network that uses multisource transfer learning with CNN for analysis of lungs condition is proposed by transferring information from a common grouping of domains. Lung tissue data is fine-tuned on the networks [15] which are pre trained on six publicly available datasets. The resulting networks are combined in an ensemble and the training method improves the CNN's great precision and reliability in the classification of lung tissue structure tasks. High performance gain is achieved by several transfer of information from six generic textures databases. The proposed solution resulted in an efficiency benefit of around 2 per cent contrasted with the CNN.

III.PROPOSED WORK

CNN is most widely used in image classification. Though CNN is used for many classification tasks, it is time and data consuming. We proposed a model which can classify similar fungal mushrooms by order using transfer learning. This section includes Transfer learning module which explains the work performed by transfer learning from base models, Similarity finding phase that finds the equivalent mushrooms by misclassification rate and finally Classification is performed based on the similarity obtained from the previous phase.

A. TRANSFER LEARNING

The dataset is segregated into training data and testing data. New model is constructed by transfer learning from predefined models like VGG16, RESNET50 and INCEPTIONV3. Best model is selected from the constructed models. Similarity between mushrooms are to be found using the best model. Mushroom species are then grouped by their order and classification of mushroom species are then obtained. Figure 3.1 depicts the overall architecture of the work.

The classifier used are fully connected layers. The idea is to include a pile of fully-connected layers. This is led by a softmax activated layer. The distribution over every possible class label is the result of the softmax layer and then we tend to categorize the picture by the most likely class. Since the outputs of a softmax function can be interpreted as a probability, softmax layer is employed as the final most layer in the model. Number of classes present is given as the parameter for the softmax function.

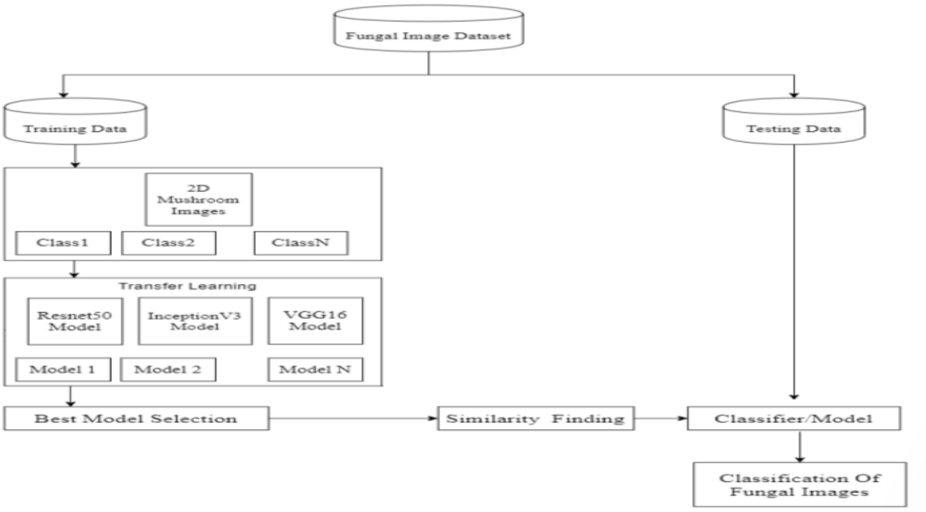

Figure 3.1 Classification based on similarity using transfer learning – Overall architecture

1) Architecture of Transfer Learning

Transfer learning solely works in deep learning if the features of the model learned from the first task are common. The base network is trained in transfer learning technique on a base dataset and task to reuse or pass the learned features to a second target neural network to be trained on a new dataset and task. These methods will tend to work if the features are common and general which means suitable for both target and source tasks rather than unique to the source task.

The reference or the pre-trained neural network models are already being trained on ImageNet dataset, which is a broad visual database developed to be used in software research for visual object recognition. ImageNet dataset consists of more than 14 million images. This was designed by academics known for computer vision research. Figure 3.2 depicts the architecture of transfer learning.

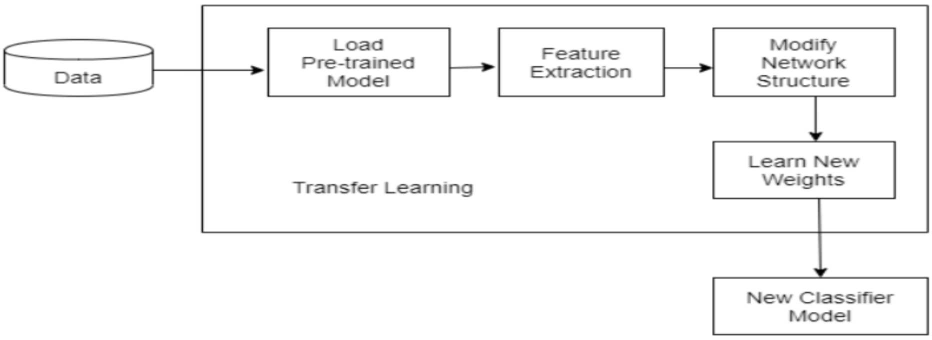

Figure 3.2 Architecture of transfer learning

Each CNN model has a convolutional base and classification part. Convolutional base of the pre-trained models are extracted and transferred while the classification phase is left. The new model will hold the convolutional base of the reference networks and further, dense layers and batch normalisation layers are added to it. The final most dense layer has the softmax activation function which gives over all other classes the probability that the fungal image belonging to its target class. By this way, the predefined neural networks are modified and the model is designed to be used specifically for the dataset.

2) Fine tuning

The higher level layers of the pre-trained models can be reused through fine tuning for the new dataset to increase the performance. For this, weights are calibrated in such a way that the model learns data specific high level features to the dataset. Typically this approach is suggested when the training dataset is broad and very close to the source dataset on which the pre-trained neural network was being

trained. A brief description of various strategies of fine tuning is given. Figure 3.3 explains the fine tuning performed to the base models.
Train the complete model: The design of the pre-trained neural network model is employed and training is performed on the dataset. The model is being learnt from scratch, so big dataset is needed and also a great deal of processing power is required.

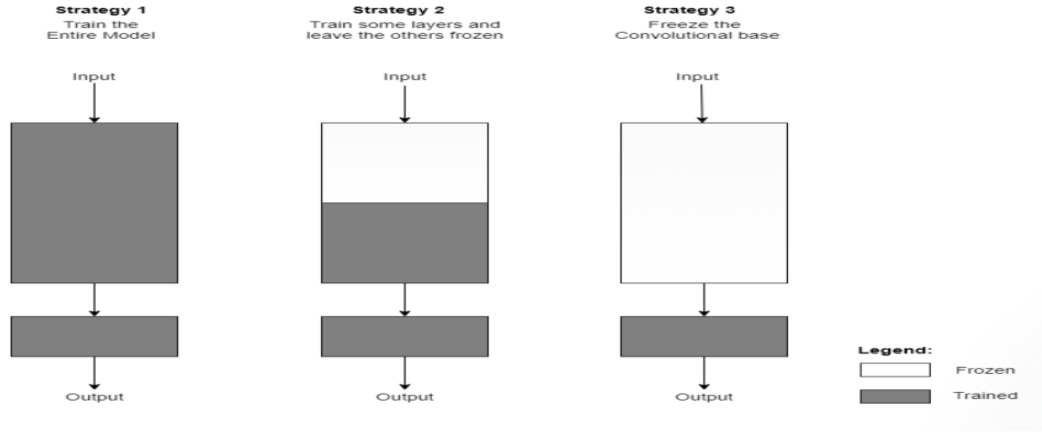

Figure 3.3 Fine tuning the base models

Train certain layers and leave the others frozen: Lower layers refer to general characteristics, whereas higher layers refer to different unique characteristics. Selecting what proportion we would like to regulate the weights of the network plays a serious role. Usually, if the dataset is small and a huge number of parameters are given, a lot of layers is left frozen to prevent the problem of overfitting. By comparison, if it is a wide dataset and the number of parameters is less, the model may be improved by training more number of layers for the brand new task as overfitting will not be an problem.

Freeze the convolutional core: This case leads to an severe train/freeze trade-off scenario. The key task is to make the original form of convolutional structure stay and its outputs are used to feed the classifier. The pre-trained model is employed as a fixed feature extraction mechanism, that will be useful if the computational power is brief, the dataset is limited, or pre-trained model solves the same problem as the one we need to resolve.

The aim is to freeze the convolutional core (Strategy 3) of the predefined models and add dense layers, batch normalization layers to the network. By doing this, the number of parameters to be trained is less than when typical CNN is used. When the number of trainable parameters are less, in turn the time taken to train the model becomes less.

3) Feature Extraction

In building a CNN model, specific algorithms are used for feature extraction. Since we use transfer learning technique, the models itself can be used as fixed feature extractors. Reference networks trained on ImageNet dataset have a lot of convolutional layers and pooling layers thus making the neural network deep. Depending upon the classification task, the network can be modified for better accuracy and results. The convolutional base are used as such from the reference network and the new model is built on top of it. Lower layers will recognize lower level features whereas the latter layers will identify and recognize specific features. Figure 3.4 explains the process of feature extraction in the model.

The features learned by the model trained on a ImageNet are reused for the fungal dataset. This has been achieved by initializing the pre-trained model and adding on top of it a fully-connected classifier. The pre-trained model is "frozen" and solely the classifier’s weights are adjusted throughout training. In such instance, the convolutional core extracts the characteristics and features related to every image and simply training a classifier which decides the class of the image is sufficiently enough for classification.

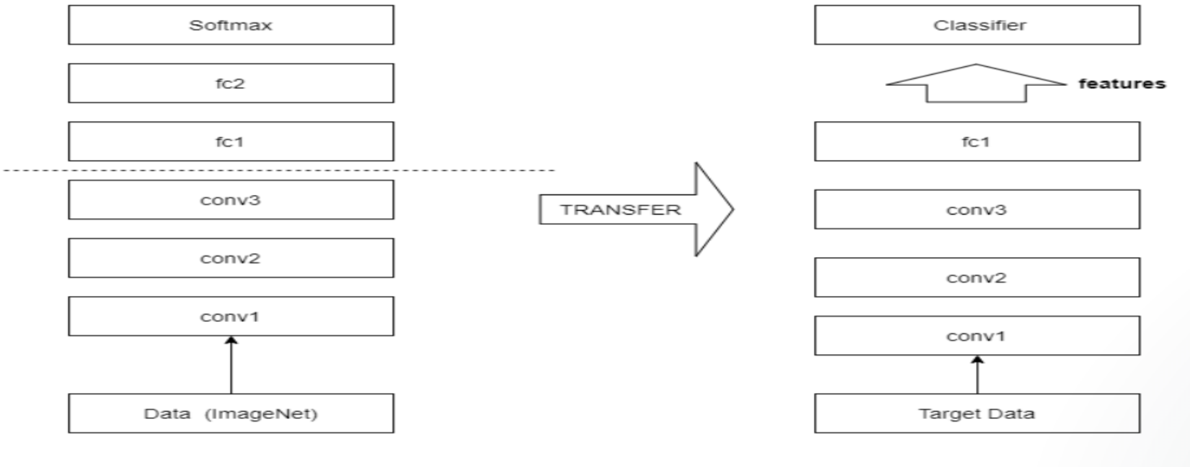

Figure 3.4 Feature Extraction-Architecture

B. SIMILARITY FINDING

The similar match for mushrooms have to be found to group them by order. The mushroom species for which the similar match has to be found is made the testing data. All the other species are made as the training data. The base predefined network is selected. The preprocessed fungal images are trained and validated. The non trained class of images are tested and predicted. The class which has the highest prediction rate will be the most similar match for the tested mushroom species. Confusion matrix and classification report is obtained for the same. Misclassification rate is also found for the testing class. Therefore the class with least misclassification rate and highest prediction rate will be more similar. Algorithm 3.1 explains the same. The same process is repeated for other base networks. Further, the similar species for all fungal classes are found. The same model is used for all process. Hence there is no need of constructing new model once again. The time for constructing the model is then reduced.

Algorithm 3.1:Similarity finding algorithm for the non trained class 'Suillus'

Input:{Hygrocybe, Russula, Agaricus, Boletus, Entoloma, Amanita, Lactarius, Cortinarius}. Mushroom dataset consisting of eight classes.

Output:Finding similar match class for the class ‘Suillus’

Procedure:

1: begin

2: for each model in list of predefined models

3: Images are preprocessed to the format of predefined models

4: Neural network built by transfer learning is trained and validated

5: Find prediction of unknown class ‘Suillus’ images

6: Obtain the classification report of all classes

7: Obtain the confusion matrix of all classes

8: Find the misclassification rate for class 'Suillus' over other classes

9: Class of minimum misclassification rate is found

10: end

C. CLASSIFICATION BASED ON SIMILARITY

Founded similar species are grouped by order. Each order possess specific characteristics and has several species. The nine species present in the dataset are categorized into three order of classes. The three classes are Agaricales, Boletales and Russulales. Only a part of species belonging to the same order are given as training data and the remaining species is given as testing data.

The model will find the correct order for testing species. By this way, with only less data for training, the model can find species belonging to the same order. Normally, the testing data will have the images of the same class of training data. Here, the model instead of finding new images, it is able to find new species which belongs to the class of training data. This model can be used for finding mushroom species having equivalent nutritional content and also the species that cause dangerous diseases which can be used in medical applications.

IV. EXPERIMENT AND RESULTS

The overall work is implemented in four phases with results analyzed, first phase explains how the transfer learning is performed on the dataset, second phase demonstrates the implementation of finding similarity between fungal mushrooms, the third phase describes how the classification based on similarity is performed on the dataset and the final phase gives the overall analysis of the used base models.

A. DATASET AND TRANSFER LEARNING

1) Experiment Dataset

The dataset used is Mushroom Classification-Common Genus’s images taken from Kaggle. Nine folders of fungal mushroom images found in common Northern European areas are present inside this dataset. Each folder consists of 300 to 1500 selected images of mushrooms genuses. A sample of images present in the dataset are given in Figure 4.1. Mushroom Classification- Common Genus’s images Dataset, https://www.kaggle.com/maysee/mushrooms-classification-common-genuss-images, last accessed on April 3,2020.

Figure 4.1 Part of fungal mushroom images in the dataset

Base Models are the source reference networks from where the base architecture is transferred. The different base networks used are INCEPTIONV3, VGG16 and RESNET50. The evaluation metrics of performing transfer learning from the base models on the dataset is given in Table 5.1.

Table 5.1 Performance of various predefined models on the dataset

S. No	MODEL	IMAGE SIZE	ACCURACY	LOSS
1	VGG16	224*224	96%	0.1
2	RESNET50	224*224	76%	0.6
3	INCEPTIONV3	299*299	86%	1.3

All the images in the dataset are resized to 224 by 224 for RESNET50, VGG16 models and 299 by 299 for INCEPTIONV3 model. Data is segregated into two parts: 80% for training data and 20% for testing data. ImageDataGenerator is used to create real-time data batches with augmentation techniques during training. INCEPTIONV3 (RESNET50,VGG16)'s ‘preprocess_input’ function is used to ensure that we preprocess our input training images the same way the images originally used to train base model were preprocessed.

‘validation_split’ parameter will automatically set aside a specified fraction of images for validation.

2) Proposed Network Architecture Summary

INCEPTIONV3,RESNET50 and VGG16 are kept as the base models and transfer learning is performed. Pre-trained weights are transferred and the fully-connected output layers are skipped by setting the parameter ‘include_top=False’.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 512)               14714688
_________________________________________________________________
batch_normalization_1 (Batch (None, 512)               2048
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
batch_normalization_2 (Batch (None, 256)               1024
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
batch_normalization_3 (Batch (None, 128)               512
_________________________________________________________________
dense_3 (Dense)              (None, 3)                 387
=================================================================
Total params: 14,882,883
Trainable params: 166,403
Non-trainable params: 14,716,480
_________________________________________________________________
```

Figure 4.2 Modified model transferred from VGG16

The extra layers added are batch normalization layers and three dense layers. To decrease the variance, computation complexity, and to extract lower level features from the previous layers, pooling operation is performed. Average pooling operation is used. It takes into account the average values of features and feeds it to next layer making the output, a generalized computation. The model summary of the network when transfer learning is performed from RESNET50, INCEPTIONV3 and VGG16 is shown in Figure 4.3, Figure 4.4, Figure 4.2 respectively.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
batch_normalization_1 (Batch (None, 2048)              8192
_________________________________________________________________
dense_1 (Dense)              (None, 256)               524544
_________________________________________________________________
batch_normalization_2 (Batch (None, 256)               1024
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
batch_normalization_3 (Batch (None, 128)               512
_________________________________________________________________
dense_3 (Dense)              (None, 3)                 387
=================================================================
Total params: 24,155,267
Trainable params: 562,691
Non-trainable params: 23,592,576
_________________________________________________________________
```

Figure 4.3 Modified model transferred from RESNET50

Optimizers are employed for loss reduction by changing the parameters and adjusting the weights. The optimizer used here is Adam. Adam holds the advantages of both Adagrad and RMSprop that works well with thin gradients and in on-line settings. Adam optimizer takes comparatively low memory and also works well with standardization of hyperparameters. The rate at which the weights are updated or adjusted during training is called learning rate. Learning rate is adjusted to 0.001. The chosen learning rate gives a model with lower loss. Batch normalization is utilized to avoid the network from being overfitted. It decreases the internal covariance shift, regularizes the model and reduces the need for dropout. It also allows the use of saturating non-linearities and good learning rates. In the next layer, dense layer links every neuron from the previous layer to each other. Because of this property, dense layers are also known as fully connected layers.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
inception_v3 (Model)         (None, 2048)              21802784  
_________________________________________________________________
batch_normalization_95 (Batc (None, 2048)              8192      
_________________________________________________________________
dense_1 (Dense)              (None, 256)               524544    
_________________________________________________________________
batch_normalization_96 (Batc (None, 256)               1024      
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896     
_________________________________________________________________
batch_normalization_97 (Batc (None, 128)               512       
_________________________________________________________________
dense_3 (Dense)              (None, 3)                 387       
=================================================================
Total params: 22,370,339
Trainable params: 562,691
Non-trainable params: 21,807,648
_________________________________________________________________
```

Figure 4.4 Modified model transferred from INCEPTIONV3

The activation function used for the last dense layer is softmax. This function considers entirely different events as to the probability distribution of the case over ‘n’. Softmax function will quantify each target category’s probabilities over all the other possible target categories. Higher learning rates can also be used because batch normalization ensures no activation that has gone extremely low or high. As a result of this, things which have not been trained in previously, will start to train. It adds few noise to every activations of hidden layers similar to dropout. Therefore, batch normalization and softmax function are used which is a good thing because lot of information is not going to be lost. The best model is to be selected.

3)Evaluation of best model

Through experimental analysis, VGG16 network behaves well for the dataset with great accuracy and lesser loss. The model obtains a highest accuracy of 98.3% when VGG16 is used as the base reference model. It is chosen as the best model. Figure 5.1 depicts the comparison of training accuracy vs validation accuracy for VGG16 on the dataset. Figure 5.2 shows the comparison of training loss vs validation loss for VGG16 on the dataset.

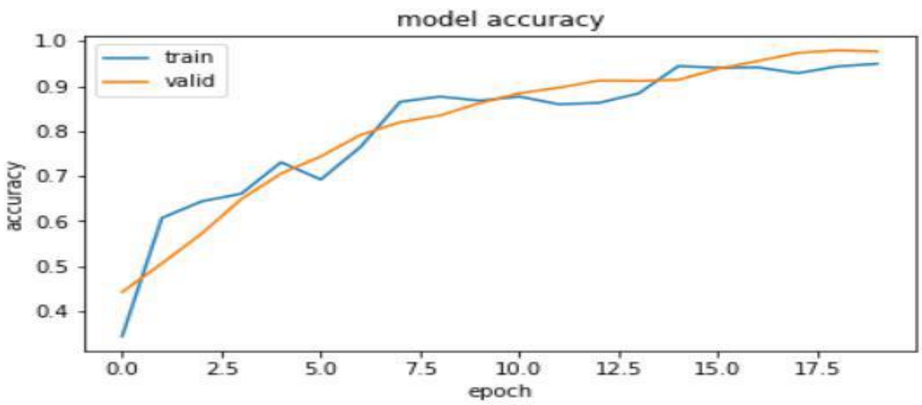

Figure 5.1 Comparison of training accuracy and validation accuracy of the best model.

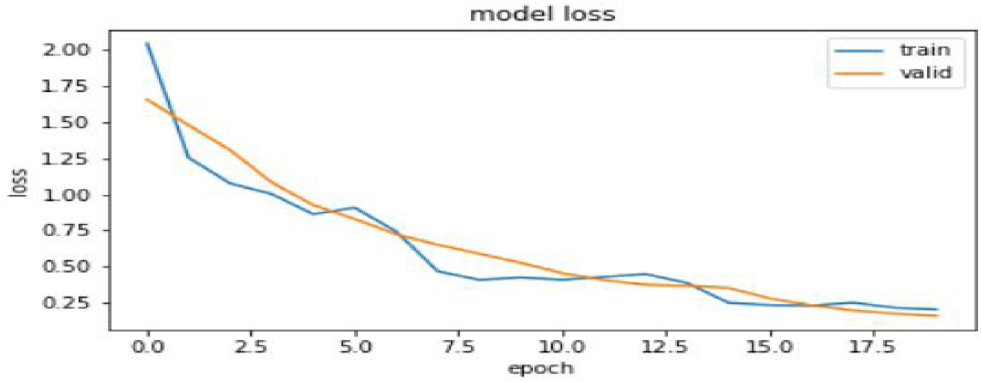

Figure 5.2 Comparison of training loss and validation loss of the best model

B. SIMILARITY FINDING

Neural network models for a set of classes are created using transfer learning from predefined models. Similarity has to be found for the classes so that they can be grouped by their order for further analysis. The task is to find the similar match among the classes present. Misclassification rates for different classes are found to find the similar match.

1) Evaluation of Misclassification rate

The training data consists of the classes namely Hygrocybe, Russula, Agaricus, Boletus, Entoloma, Amanita, Lactarius, Cortinarius. Misclassification rate obtained when the testing class 'Suillus' is given to every predefined models is given in Table 5.2.

Table 5.2 Evaluation of misclassification rate for the testing class

CLASS	VGG16	RESNET50	INCEPTIONV3
Agaricus	0.72	0.89	0.99
Amanita	0.87	0.91	0.99
Boletus	0.21	0.52	0.99
Cortinarius	0.90	0.977	0.91
Entoloma	0.90	0.88	0.81
Hygrocybe	0.83	0.98	0.99
Lactarius	0.89	0.93	0.97
Russula	0.93	0.80	0.99

Through analysis, it is seen that the class 'Boletus' has the least misclassification rate when the fungal mushroom images of class 'Suillus' is tested. Therefore, class 'Suillus' is similar to class 'Boletus'. Similar mushrooms are found and they are grouped by the order. Similar order mushrooms possess same and equivalent characteristics. Mushrooms possess nutritional content and also some mushrooms cause notorious diseases to human health. They are found through misclassification rate and prediction rate.

Figure 4.5 Predicted/Actual/Loss/Probability of actual class of mushroom images using RESNET50 model

Figure 4.6 Predicted/Actual/Loss/Probability of actual class of mushroom images using VGG16 model

The actual class, predicted class, loss and probability of actual class of the mushroom images when VGG16 and RESNET50 used as the base model are shown in Figure 4.6, Figure 4.5. It is seen that predicted loss is less when VGG16 is used as the base model compared to INCEPTIONV3 and RESNET50. The model behaves well for the neural network of 16 layers deep. So, VGG16 is chosen as the best model.

C. CLASSIFICATION BASED ON SIMILARITY

The data is divided into three classes according to order of mushrooms based on similarity. With only less number of species for each order, the aim is to develop a model which can find new species belonging to the same order.

1) Classification for best model

The data is divided into three classes namely Agaricales, Boletales and Russulales.

Training Data:

Agaricales: It is composed of 500 images of four species namely Cortinarius, Hygrocybe, Entoloma and Agaricus

Boletales: It is composed of 500 images of species Boletus.

Russulales: It is composed of 500 images of species Russula.

Testing Data:

Three new species namely Amanita, Suillus and Lactarius each 50 images are taken as testing data and are given to the best model VGG16. The Table 5.3 shows the prediction rate assessment for the test data.

Table 5.3 Prediction rate evaluation for the test data

	AGARICALES	BOLETALES	RUSSULALES
LACTARIUS	15.52	8.65	25.81
SUILLUS	12.76	25.58	11.64
AMANITA	26.10	12.24	11.65

From the Table 5.3, it is found that the mushroom classes Lactarius, Suillus and Amanita has the high probability of belonging to the order Russulales, Boletales and Agaricales respectively.

D. OVERALL MODELS ANALYSIS - METRICS

The accuracy, the number of parameters and loss are compared for the predefined models VGG16, RESNET50 and INCEPTIONV3.

1) Models comparison by parameters

The number of parameters to be trained, non-trainable parameters for the predefined models are listed in Table 5.4.

Table 5.4 Comparison of models by parameters on the dataset

Model	Image Size	Number of layers	Total parameters	Trainable parameters	Non trainable parameters
VGG16	224*224	16	14,882,883	1,66,403	14,716,480
INCEPTIONV3	229*229	48	22,370,339	5,62,691	21,807,648
RESNET50	224*224	50	24,55,267	5,62,691	23,592,576

2) Models comparison by accuracy

Figure 5.3 depicts the accuracy of the dataset when the base models VGG16, RESNET50 and INCEPTIONV3 are used for fungus classification based on similarity by transfer learning.

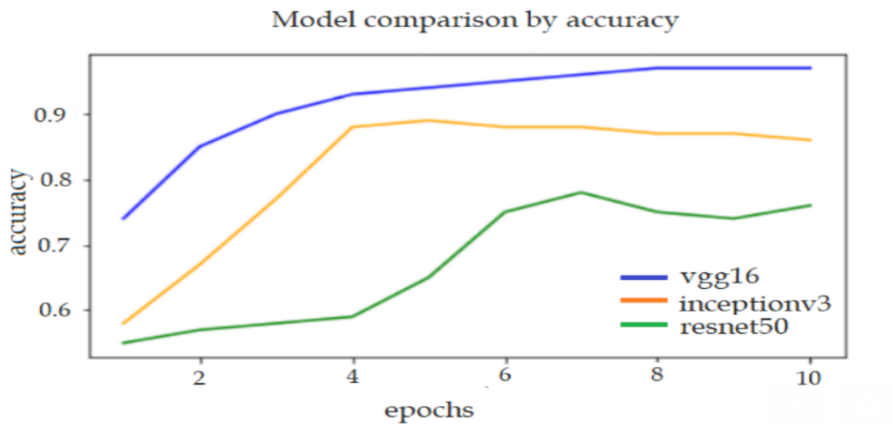

Figure 5.3 Comparison of models by accuracy on the dataset.

3) Models comparison by loss

Figure 5.4 depicts the loss of the dataset when the base models VGG16, RESNET50 and INCEPTIONV3 are used for fungus classification based on similarity by transfer learning.

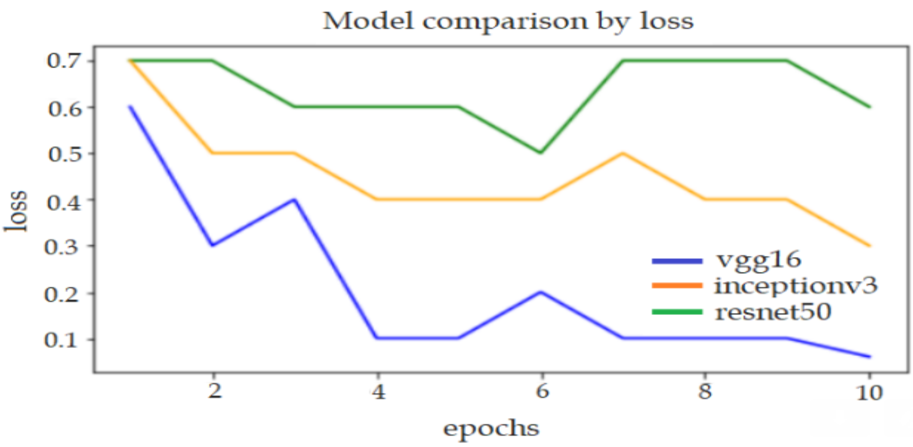

Figure 5.4 Comparison of models by loss on the dataset

V. CONCLUSION

In this framework, CNN based transfer learning method is developed that can classify new fungal mushroom species belonging to the same order. Transfer Learning with CNN achieves a high accuracy rate in the fungus classification model. The project work has been implemented in a Kaggle Kernel with 14GB RAM. The proposed model achieves an improve in accuracy of 3.5% compared to the typical CNN model [1] used for fungus detection and classification. Our model gives a maximum accuracy of 98.3% when VGG16 is used as the base source model.

To deal with the challenges posed by over-fitting and long training time of CNN, we adapted transfer learning methods and used batch normalization layers. Another important aspect to note is that average pooling with transfer learning performs well than min pooling or max pooling operation. The weights in the models constructed using transfer learning can also be used in some other problem task. So the model built can be reused for some other classification task in the same domain. Since the pre-trained networks like RESNET50, VGG16 and INCEPTIONV3 are already trained on ImageNet dataset, there is no need of training all the whole model, rather transferring the weights is sufficient. Any user can easily plug-in the constructed models and use within their framework easily.

Transfer learning is versatile and pre-trained models can be used directly as fixed feature extractors and can be embedded into entirely new models. The advantage of transferring knowledge is the reduced number of integrated LoCs. Fungi can cause various number of human diseases such as poisonings, infections due to parasites, allergies and certain fungi also have nutritional and medicinal significance. Agaricales mushrooms have the properties of antidiabetic and anticancer. Intake of order Boletales mushrooms in combination with ethanol will lead to a typical syndrome of antabuse. Russulales order may cause diseases of silverleaf. The model can be used to identify the fungal mushroom species which causes similar types of diseases in medical applications. Since the model architecture is reused for similarity finding and classification, there is no need of building separate neural network model for each task. Transfer learning from predefined networks leads to the less number of parameters to be trained.

In the future work, the experiment can be carried out in GPU to parallalize the transfer learning task of different base models. Transfer learning performs well in case of small datasets, for large datasets new methods can be proposed. Furthermore, the experiment can be extended to give a description of the fungal image about its characteristics, features, nutritional value, toxic substances. Ensemble learning methods can also be combined with transfer learning using different classifiers so as to build a combined model. This framework can be extended to classify fungi which are free-living in soil or water and those present in food items.

REFERENCES

1. M. W. Tahir, N. A. Zaidi, A. A. Rao, R. Blank, M. J. Vellekoop and W. Lang, "A Fungus Spores Dataset and a Convolutional Neural Network Based Approach for Fungus Detection," in *IEEE Transactions on NanoBioscience*, vol. 17, no. 3, pp. 281-290, July 2018.
2. A. P. Marcos, N. L. Silva Rodovalho and A. R. Backes, "Coffee Leaf Rust Detection Using Convolutional Neural Network," *2019 XV Workshop de Visão Computacional (WVC)*, pp. 38-42, São Bernardo do Campo, Brazil, 2019.
3. Sinno Jialin Pan,Qiang Yang,"A Survey on Transfer Learning" *IEEE Transactions on Knowledge and Data Engineering*,vol.22,issue.10,October 2010.
4. A. Beikmohammadi and K. Faez, "Leaf Classification for Plant Recognition with Deep Transfer Learning," *2018 4th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 21-26,*Tehran, Iran*, 2018.

5. J. Yi, J. Tao, Z. Wen and Y. Bai, "Language-Adversarial Transfer Learning for Low-Resource Speech Recognition," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 3, pp. 621-630, March 2019.
6. L. Guo, Y. Lei, S. Xing, T. Yan and N. Li, "Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines With Unlabeled Data," in *IEEE Transactions on Industrial Electronics*, vol. 66, no. 9, pp. 7316-7325, Sept. 2019.
7. Z. Huang, Y. Cao and T. Wang, "Transfer Learning with Efficient Convolutional Neural Networks for Fruit Recognition," *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 358-362,Chengdu, China, 2019.
8. X. Zhong, S. Guo, H. Shan, L. Gao, D. Xue and N. Zhao, "Feature-Based Transfer Learning Based on Distribution Similarity," in *IEEE Access*, vol. 6, pp. 35551-35557, 2018.
9. Q. Qi, Q. Huo, J. Wang, H. Sun, Y. Cao and J. Liao, "Personalized Sketch-Based Image Retrieval by Convolutional Neural Network and Deep Transfer Learning," in *IEEEAccess*,vol.7,pp.16537-16549,2019.
10. S. Shao, S. McAleer, R. Yan and P. Baldi, "Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning," in *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2446-2455, April 2019.
11. L. Wen, L. Gao and X. Li, "A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 136-144, Jan. 2019.
12. H. Chang, J. Han, C. Zhong, A. M. Snijders and J. Mao, "Unsupervised Transfer Learning via Multi-Scale Convolutional Sparse Coding for Biomedical Applications," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1182-1194, 1 May 2018.
13. H. Shin *et al*., "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," in *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285-1298, May 2016.
14. J. Yang, S. Li and W. Xu, "Active Learning for Visual Image Classification Method Based on Transfer Learning," in *IEEE Access*, vol. 6, pp. 187-198, 2018.
15. S. Christodoulidis, M. Anthimopoulos, L. Ebner, A. Christe and S. Mougiakakou, "Multisource Transfer Learning With Convolutional Neural Networks for Lung Pattern Analysis," in *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 76-84, Jan. 2017.
16. Abadi M, Agarwal A, Barham P, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”, CoRR abs/1603.04467, 2016.
17. Chollet, Francois, “Keras: Deep learning library for theano and tensorflow”, *https://keras.io* , Last accessed on April 3,2020.
18. Numpy: *https://numpy.org/* , Last updated on July 26,2019.
19. Matplotlib : *https://matplotlib.org/*, Last updated on March 19,2020.
20. Pandas : *https://pandas.pydata.org/* , Last updated on March 18,2020.
21. Pan S.J. and Yang Q, “A survey on transfer learning”, *IEEE Transactions on knowledge and data engineering*, Vol. 10, pp.1345- 1359, 2010.
22. Patricia N. and Caputo B, “Learning to learn, from transfer learning to domain adaptation: A unifying perspective”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1442- 1449, 2014.
23. Mushroom Classification-Common Genus’s images Dataset, *https://www.kaggle.com/maysee/mushrooms-classification-common-genuss-images* , Last accessed on April 3,2020.
24. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay, “Scikit-learn: Machine Learning in Python”, *JMLR 12,* pp. 2825-2830, 2011.
25. SciPy : *https://scipy.org/scipylib/* , Last accessed on April 3,2020.
26. S.Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, pp. 1-6, 2017,.
27. J.Yang and J. Li, "Application of deep convolution neural network," *2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*,pp. 229-232,Chengdu, 2017,.

28. Arnold, R. Nallapati and W. W. Cohen, "A Comparative Study of Methods for Transductive Transfer Learning," *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*,pp. 77-82 ,Omaha, NE, 2007.
29. B.D. Gürkaynak and N. Arica, "A case study on transfer learning in convolutional neural networks," *2018 26th Signal Processing and Communications Applications Conference (SIU),* pp. 1-4, Izmir, 2018.
30. E.Vonk, L. C. Jain and L. P. J. Veelenturf, "Neural network applications," *Proceedings Electronic Technology Directions to the Year 2000,* Adelaide, SA, pp. 63-67,Australia, 1995.
31. M. Mishra and M. Srivastava, "A view of Artificial Neural Network," *2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)*,pp. 1-3,Unnao, 2014.
32. Chetlur S, Woolley C, Vandermersh P, Cohen J, Tran J, Catanzaro B, and Shelhamer E, “cuDNN: Efficient Primitives for deep learning”, *CoRR abs/1410.0759,* 2014.