Research article

TRANSFER LEARNING APPROACH FOR CLASSIFICATION OF WIDELY USED SPICES

Arunachalam Sundaram[1], Abdullahi Masud[1], Ali AlMarhoon[1], Bhaskarjit Sarmah[2]

[1] Department of Electrical Engineering, Jubail Industrial College, Kingdom of Saudi Arabia, [2] Data Scientist, BlackRock, India

Keywords: spices, VGG16 model, Artificial Intelligence (AI), Machine Learning (ML), Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN).

https://doi.org/10.53370/001c.35690

Yanbu Journal of Engineering and Science

Vol. 19, Issue 2, 2022

People around the world relish variety of food that are flavourful. Spices add flavours to the food without adding any fat or calories. People have used spices for many centuries and are an integral part of our food. In addition to aroma, spices also have anti-bacterial, anti-inflammatory properties and other health-promoting properties. Recognizing spices from images is a challenging problem for a machine as they come in varying sizes and shapes, different colours, high visual similarity, and texture. The classification of spices presents useful applications in the field of Artificial Intelligence-driven food industry, e-commerce, and health care. In the billion-dollar spice industry, image classification of spices finds applications ranging from receiving, processing, labelling, and packaging them. As there is no dataset currently available for spices, in this work, a Spice10 dataset with 2000 images of spices is first created. This study aims to find out whether the accurate classification of spices is possible using computer vision technology. Instead of building models from scratch, a pre-trained transfer learning approach has been implemented in this work to classify the commonly used spices. The images in the dataset are of different sizes and have to be resized and pre-processed before using it with the transfer learning approach. Few different pre-trained networks are modified and used for the image classification of spices. The best classification average accuracy obtained by the VGG16 model is nearly 93.06% which is better than the other models. The high accuracy of the VGG16 model indicates it can be successfully used for the classification of spices.

1.0. INTRODUCTION

Spices have been used for centuries in diet and have molecular and biochemical properties that can be used to prevent diseases and treat diseases such as cancer, heart diseases, Alzheimer’s to name a few. The healing properties of spices have been investigated in [1]. Cancer-preventing properties are available in spices such as turmeric, black pepper, garlic, cumin, clove, etc.[2] Spices are skillfully used by cooks around the world every day to add flavor to the dish and in addition, adding spices will also help in the fight against diseases. The therapeutic property of spices and remedies for many ailments are well documented in ancient Indian books.[1] Many of these Phytonutrients are not available in fruits and vegetables but are available in spices. Spices contain some unique Phytonutrients and other compounds which have anti-oxidant,[3] anti-microbial,[4] and anti-inflammatory powers.[5] Even though there are numerous spices, this research work concentrates on the classification of the most widely used spices such as turmeric, cumin, pepper, star anise, fenugreek, cardamom, bay leaf, mustard, cinnamon, and cloves.

An investigation of the anticancer property of turmeric which is a common spice in Asian dishes is available in [6]. The medical application of turmeric has been investigated in [7] and it shows turmeric possesses antifungal, antiviral, and anti-venom properties. Cumin is a flowering plant and the dried seeds in the flower are used as a spice by Arabs, Indians, Africans, Americans, Greeks, Egyptians, Romans, and other people in their dishes. The properties of different varieties of cumin are investigated in [8] and the investigation reveals cumin has good antioxidant properties. Different varieties of peppercorns are used as a spice and for seasoning. Black pepper, the most common spice is available in shakers in hotels for seasoning. The nutrient composition of different varieties of pepper is analyzed in [9]. Star anise a spice used for enhancing the flavor of meat and in biryani recipes is also used in perfumes, creams, and pastes in its extracted form. The analysis of the anti-oxygenic property of star-anise is available in [10]. A common ingredient of Indian dishes is cuboid-shaped Fenugreek which is used as a spice to make dal, pickles, sambar (south Indian dish), etc. Therapeutic uses of Fenugreek available in [11,12] reveals that it can be used as a dietary supplement to cure diabetes. Due to its unique flavor, pleasant aroma, and taste, cardamom has been used in food and beverages from ancient times. The Anti-ulcer properties of cardamom have been researched in [13]. Dried bay leaf a spice commonly used in soups, gram masala, and biryani for its unique flavor also has therapeutic effects.[14] One of the most commonly used spices in the United States of America; Asian and European countries are mustard seeds. The use of mustard seed powder for alleviation of bladder cancer is investigated in [15]. Aromatic and flavor-rich cinnamon is used in a variety of dishes such as tea, cereals, sweets, and savories, etc. The properties of cinnamon and its role in the treatment of diabetes is analyzed in [16]. Aromatic cloves are used for culinary, non- culinary, and traditional medicinal purposes. The antimicrobial properties of cloves are analyzed in [17]. The literature review of widely used spices indicates they are used in many forms and also have medicinal properties.

Artificial Intelligence (AI) is a technology that makes a machine learn and simulate human behavior using Machine Learning (ML) algorithms and/or Deep Learning algorithms such as Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) to perceive the environment and take actions.[18] A review of the state-of-the-art Artificial Intelligence (AI) techniques used in revolutionizing the food industry using automation and the advancements made in the agriculture sector towards sustainable food production is available in [19]. Computer vision technology makes a computer mimic a human vision and has been used for a variety of purposes such as image classification, image captioning, object detection, object localization, clustering, video analysis, robotic navigation, and many more.[20] Convolutional Neural Networks (CNN) are widely used for image classification. For image classification, the features extracted by the convolutional layers and pooling layers are passed on to a DNN to classify images. CNN with 60 million parameters and with Graphics Processing Unit (GPU) have classified 15 million images of 1000 categories with very high accuracy.[21] In the food industry, the use of computer vision from harvesting to packing is possible and this technology is being used to maximize profit and quality in the food industry.[22] Food computing is an interdisciplinary study that uses compute vision technology for the acquisition and analysis of food data to address food perception and dietary management of health.[23] A review of dietary assessment based on food recognition and volume estimation in [24] reveals many challenges in food image processing technology. Accuracy-related issues while classifying different sized images by inserting a new layer in CNN are provided in [25]. Food image classification problem using different architectures of CNN has been attempted on standard food image datasets such as UEC100,[26] UEC256,[27] Food101,[28] and Food251[29] in [30]. Food image classification with an accuracy of around 80% is obtained using the Food 101 dataset using Alexnet and VGG16 in [31]. A food classification technique for diabetic patients using CNN with NTUA-Food 2017 dataset with 3248 images is proposed in [32] can classify food into 8 categories based on macronutrient content. CNN based on Residual Network architecture for food image classification using FOOD 475 dataset in [33] reveal CNN can classify a wide variety of food with a high level of accuracy. CNN based on DenseNet structure for classification of Korean food dataset with 18 categorizes in [34] has been implemented using mobile architecture. Classification of Chicken food dishes in Indonesia using CNN in [35] reveals training process is faster when CNN is used with GPU rather than a CPU. Food image classification on the Indian food dataset using CNN is available in [36], Thai foods dataset is available in [37,38], and Chinese foods are available in [39]. Food and beverage image classification using NutriNet is presented in [40]. In addition to recognizing food images, a study in [41] can accurately identify foods in different states such as batons, creamy paste, dice chopped, floured, grated, juiced, julienne, peeled, sliced, wedges and whole. A modified version of Alexnet and GoogLenet has been used to classify images of plants and leaves available in AgriPlant dataset, Folio dataset, and Leafsnap dataset using CNN and bag of visual words in [42].

A survey of deep learning techniques used in the food industry in [43] has revealed that deep learning as a tool in food science and engineering has shown tremendous progress and a lot of scope for future applications. An online collection of food images from 1000 restaurants and 12000 food supplies in China is used in [44] along with image cleaning and CNN model for cloud-based online multiclass prediction of food images such as meat, vegetables for a food supply chain industry. FOODAI developed by Singapore Management University uses deep learning techniques and can recognize food with confidence levels.[45,46] Review of automatic food recognition system by mobile devices and the steps involved in implementing the system are discussed in detail in [47]. Many interesting experiments carried out in [48] to compare the classification accuracy of humans with CNN reveals that the human vision system is better than CNN for a limited number of food images but CNN outperforms human when trained with a larger dataset.

Magnetic food texture sensor along with principal component analysis (PCA) is used in [49] to classify corn snacks, candy, biscuits, and potato snacks based on the texture of the food. Analysis of 1453 food images with 42 categories using K-Nearest Neighbors (KNN) is carried out in [50] have revealed an accuracy of around 80%. But traditional methods based on PCA and feature extraction techniques are limited by hardware performance and have a large error in recognition effect. CNN has outperformed other algorithms such as K-Nearest Neighbors, Fuzzy logic, and Ensemble Decision Trees to accurately classify cherry fruit in [51] and sour lemons in [52].

A detailed literature survey indicates there is no dataset available for the classification of spices and this classification task has not been attempted so far. Moreover, most of the papers in the literature have used python for the classification of images but Matlab also has excellent resources for image classification. Hence in this paper, Matlab has been used for the image classification of spices.

Transfer learning is accomplished using pre-trained models such as Visual Geometry Group (VGG),[53] GoogLeNet,[54] ResNet,[55] Alexnet,[21] etc. These pre-trained models are trained with millions of images available in the Imagenet database[56] and were part of the ImageNet Large-

Scale Visual Recognition Challenge (ILSVRC).[57] These pre-trained models can classify thousands of object categories. Compared to building CNN models from scratch, using a transfer learning approach to classify new objects is easier and faster as these models have extracted informative features from a larger dataset. Few strategies can be used for fine- tuning the model[58] such as

- Retain the architecture and re-training the entire CNN model.
- Train some of the fully connected and feature extraction layers and leave some of the feature extraction layers frozen.
- Retain the architecture and only train the fully connected layer.

Out of the above approaches, the most popular and the one used in this study is to retain the architecture by making changes only to the last fully connected layer and to re-train the entire CNN model. This article proposes a transfer learning approach to classify widely used spices. Three widely used pre-trained models such as Alexnet, VGG16, and GoogLeNet are employed for the classification of spices. Hyperparameter tuning is also employed to optimize the value of the most common hyper-parameters. The performance of each transfer learning approach is compared based on the accuracy of the classification of spices.

2.0. DESCRIPTION OF THE SPICE10 DATASET

One of the important requirements for machine learning is the availability of the dataset to train the machine. For supervised learning, a vast amount of images along with labels are necessary. To identify spices using deep learning models a spice dataset consisting of images of spices and their labels is the most important requirement and at present, the spice dataset is not available. The vast increase of websites, social media, youtube channels related to food content has helped many to share recipes and food images. Some of the ingredients used in these recipes are the spices and these spice images are downloaded from various websites to create the Spice10 dataset with 2000 images consisting of ten different categories of Spices namely bay leaf, cardamom, cinnamon, cloves, cumin, fenugreek, mustard, pepper, star anise, and turmeric. Each category has 200 images. More than 90% of the training images in the Spice10 dataset are low- resolution color images and have a size of less than 20kb. The Spice10 dataset occupies 37MB of space on the hard disk. To test the accuracy of the trained model on unseen images of spices a test folder consisting of 20 images with 2 images for each class is created. This test folder occupies 3.7 MB on the hard disk.

A dataset in a Matlab is referred to as "Datastore". An imageDatastore in Matlab contains a collection of images and Matlab has the necessary function which also helps us to perform operations such as reading and processing images stored in multiple folders on a disk or in a remote location. Creating labeled data is easy in Matlab as it allows folder name as the label name for the images stored in the folder by using 'LabelSource', 'foldernames' as name-value pairs when creating the ImageDatastore object. Hence ten subfolders with the names bayleaf ,cardamom, cinnamon, cloves, cumin, fenugreek, mustard, pepper, staranise, and turmeric were created in a folder Spice10. Sample images in the Spice10 dataset are shown in Figure 1.

The first row of Figure 1 shows a sample of bay leaves in the dataset. Fresh bay leaf, dried bay leaf, or grounded forms of bay leaves are commonly used in cooking for their unique aromatic flavors. All these forms of bay leaf are available in the dataset. It is widely used for preparing biryani, gram masala powder, soups, stews, menudo, adobo, massaman curry, and also as a tea infuser. The texture of the bay leaf available in one country is different from other countries. Most often fresh bay leaves are shiny and olive green in color and dried bay leaves are in matte olive green in color.

The second row of Figure 1 shows a sample of cardamom in the dataset. Cardamoms are mostly available in green and black colors are used for flavoring during cooking and also in drinks. Cardamoms are spindle-shaped and when triangular-shaped paper-like pods have opened the seeds inside are in black. Cardamom in its oil form has been used to relieve various stomach issues. Cardamom powders are used in cooking and also in baking. The cardamom with pods, seeds of cardamom, powdered form of cardamom, and cardamom oil are all available in the dataset. The third row of Figure 1 shows a sample of cinnamon in the dataset. Cinnamon is available as a dried bark, as powder, as oil, and as a paste. Cinnamons are used in the preparation of stews, sweets, savories, tea, and also in preparation of medicines. Cinnamon has a medium shade of brown color and is a widely used condiment around the world.

The fourth row of Figure 1 shows a sample of cloves in the dataset. Cloves are aromatic and used in cooking as buds or as powder. Cloves are also used in baking and as pomander by inserting them in oranges. Cloves and clove powder are dark brown with a reddish tinge. Clove available in various forms is part of the dataset. The fifth row of fig 1 shows a sample of cumin in the dataset. Cumin is a dried seed of a plant and is widely used as a whole or in powdered form. Cumin's are available in brown and black color. Cumin has a distinctive flavor and smell. It is used as part of masala powders, in preparation of some cheese as well as loaves of bread, in soups, and also as a decoction. Cumin as part of decoction, brown cumin, black cumin, powdered form of cumin is all part of the dataset.

The sixth row of fig 1 shows a sample of fenugreek in the dataset. Fenugreek leaves and seeds are used in a variety of cooking. Fenugreek in powdered form is used as a medicine. Leaves are green in color, seeds are angular and brownish-yellow color. The dataset contains the leaves, fenugreek seeds, and powdered form of fenugreek. The seventh row of Figure 1 shows a sample of mustard in the dataset. Mustard is widely used as a condiment to enhance the flavor of the dish. Mustard is used in a variety of forms. Mustard seeds are available in yellowish color to black color are all part of the dataset. Mustard in leafy form, oil form, paste form, and powdered form are all part of the dataset.

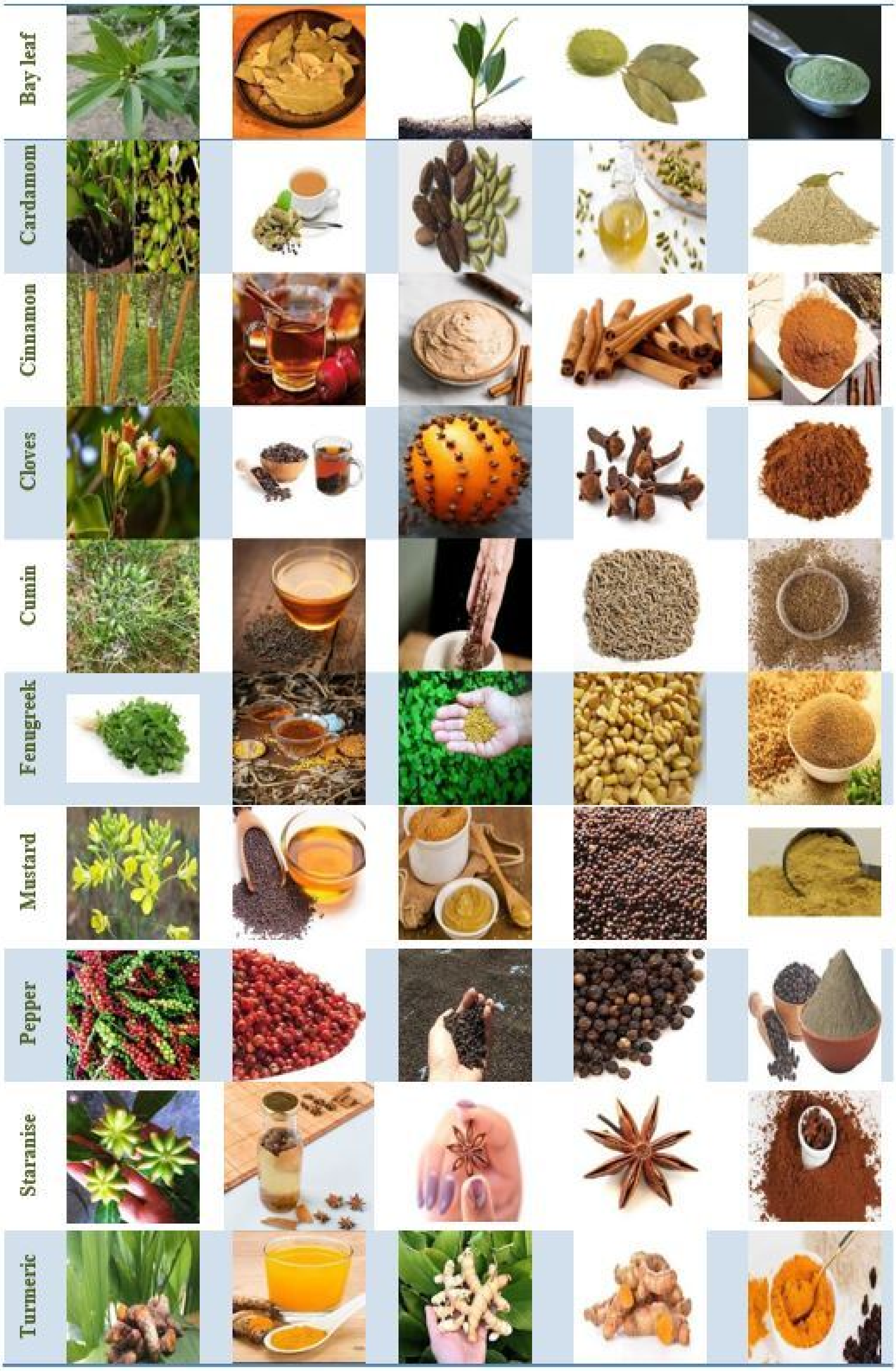


Figure 1. Sample images in each category of the Spice10 dataset

The eighth row of Figure 1 shows a sample of pepper in the dataset. Pepper is available in different shapes, forms, and colors. Peppers are used in soups, sauces, stews, marinades, and many other dishes. Peppers in all shapes, forms, and colors are all part of the dataset. The ninth row of fig 1 shows a sample of star anise in the dataset. Star anise is used as a whole or grounded form in cooking and is part of the dataset. The last row of Figure 1 shows a sample of turmeric in the dataset.

Turmeric powders are used in the preparation of a variety of curries and also as food colors. Turmeric rhizome, turmeric powder, turmeric solution are all part of the dataset. From the sample images, it can be inferred that the classification of spices is very complicated as spices in addition to the naturally occurring forms are available in different other forms such as leafy form, powdered or grounded form, as oils, and also as a paste. The machine has to learn the shape of each spice, the various texture of each spice, the different colors of each spice, and must also differentiate a spice that occurs in many forms. The textures of the powdered form of most spices are similar and hence the machine has to differentiate by its color. A small round black mustard and black peppercorn look very similar. The powdered form of cinnamon, star anise, and cloves are all different shades of brown color. Due to all these reasons, it is not easy for a machine to learn all the spices.

3.0. TRANSFER LEARNING APPROACH

Various deep learning networks have been trained to classify millions of images consisting of thousands of categories. These pre-trained models are generic models that contain vast knowledge of various low and high-level features of the classified images. The transfer learning approach in machine learning uses the knowledge gained while classifying thousands of categories to classify other related images in a new dataset. This is made possible because deep learning can be applied to any domain; features are automatically learned according to the dataset, and due to their property of generalization. Matlab allows fine-tuning the architectures of many pre-trained networks such as Alexnet, VGG16, VGG19, and GoogLeNet. Due to this fine-tuning feature available in Matlab, most of the architecture can be retained and only a few layers may be modified according to the requirement. In this work, Alexnet, GoogLeNet, and VGG 16 have been implemented to classify the images of the spices. The methodology employed in Matlab for image classification of Spice10 dataset is shown in Figure 2.

The pre-trained Convolution Neural Network (CNN) is many layers deep and comprises manly layers such as convolutional layers, pooling layers, and fully connected layers. Convolutional layers use filter functions to extract features from the images. Convolution is a mathematical transformation of a given function into a form that is more useful than the original function and in image processing convolution is a specialized linear operation while involves scanning the whole image using filters to produce feature maps. CNN works on volumes maintaining spatial structures. Each neuron in CNN receives several inputs and the weighted sum is passed over non-linear activation function such as the rectified linear unit (ReLU), or hyperbolic tangent function. The whole network has a loss function that is optimized using a backpropagation algorithm. Pooling layers are used to remove redundant or low-level features. During pooling, a small feature of the feature map is taken and in that small region, the direction of the strongest gradient (Max Pooling) or the average of the gradient (Average Pooling) is found.

3.1. ALEXNET

Alexnet is an eight-layer deep Convolution Neural Network (CNN) pre-trained on the ImageNet database.[57] Alexnet can classify millions of images into thousands of categories. Alexnet can be installed in Matlab[59] and directly applied to classification problems. In this work, the only last three layers of Alexnet are modified according to the Spice10 dataset. The remaining architecture is retained as it is. The number of channels in the last fully connected layer is modified to 10 to match the number of classes of classification in the Spice10 dataset. The modified architecture of Alexnet is shown in Table 1.

The image size for Alexnet is a 227x227x3 input and so spice images have to be resized to 227x227x3 to be used with Alexnet. This is made possible in Matlab by using augmentedImageDatastore function available in Matlab. Alexnet uses five convolutional layers and three fully connected layers. All convolutional layers and the first two fully connected layers use a piecewise linear function called Rectified Linear Unit (ReLU) activation function which outputs a positive value for any positive input otherwise it makes the output zero. The last fully-connected layer uses a normalized exponential function called Softmax and a cross-entropy loss function for multidimensional classification.

Normalization layers, pooling layers, and dropout techniques are also part of Alexnet architecture as seen from Table 1. Cross channel normalization is usually used with the ReLU activation function which is unbounded and detects big neuron response while damping uniformly large responses in the local neighborhood. The dropout technique is used to ignore randomly selected neurons during training to avoid complex co-adaptations during training. Neurons that are randomly selected during dropout do not contribute during the forward propagation of information and backward propagation of error during training the CNN. Pooling is used in Convolutional Neural Networks to remove redundant features. All these techniques are used in Alexnet to learn robust features, improve convergence time, avoid the problems of vanishing gradients, reducing over-fitting, improving generalization, and stabilized training.

3.2. VGG16

One of the best models available today to classify images is the CNN model developed by the Visual Geometry Group (VGG) of Oxford University[53] called VGG16 or OxfordNet.

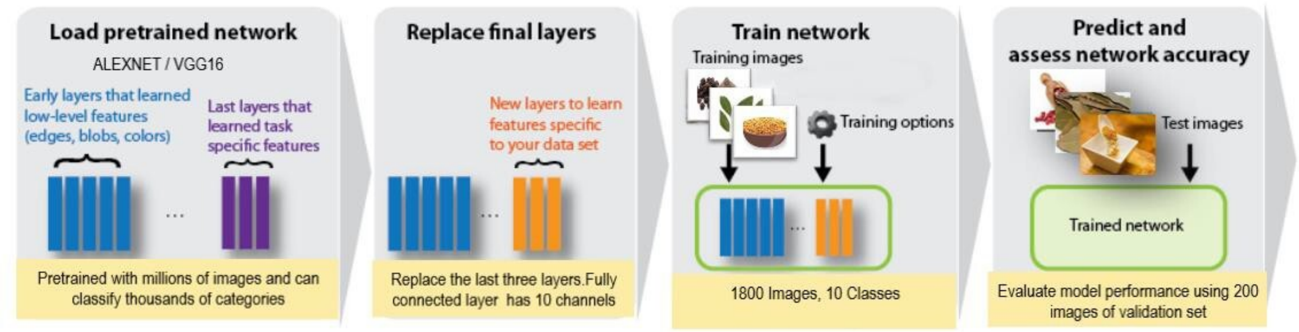

Load pretrained network
ALEXNET / VGG16
Early layers that learned low-level features (edges, blobs, colors)
Last layers that learned task specific features
Pretrained with millions of images and can classify thousands of categories
Replace final layers
New layers to learn features specific to your data set
Replace the last three layers.Fully connected layer has 10 channels
Train network
Training images
Training options
1800 Images, 10 Classes
Predict and assess network accuracy
Test images
Trained network
Evaluate model performance using 200 images of validation set

Figure 2. Transfer learning approach used for classification of spices

VGG16 can be installed in Matlab[60] and directly applied to classification problems. VGG16 is a huge neural network that has 138 million parameters when compared to 60 million parameters of Alexnet. In this work, only the last three layers of VGG16 are modified according to the Spice10 dataset. The remaining layers are retained as it is. The number of channels in the last fully connected layer is modified to 10 to match the number of classes of classification in the Spice10 dataset. The modified architecture of VGG16 that has sixteen layers is shown in Table 2. The image size for VGG16 is a 224x224x3 input and all images of spices have to be resized to 224x224x3 to be used with VGG16 using augmentedImageDatastore function available in Matlab. All convolutional layers have a 3x3 filter with a stride of 1 and all max-pooling layers

3.3. GOOGLENET

To overcome the problem of overfitting and exploding gradients of architectures with a very high number of layers the researchers in Google used inception modules to develop the GoogLeNet architecture which follows judicious dimension reduction. GoogLeNet architecture consists of 27 layers including 9 inception modules and interested readers can refer to [54,61] for the detailed description. The input image size for GoogLeNet architecture is 224x224x3 input. 1x1, 3x3, and 5x5 convolutional filters are used in the inception modules to retain important spatial information of the images. A dropout layer with a 40% dropout is used before the linear layer. The linear layer consists of 1000 neurons to classify images into 1000 categories. But, since the Spice10 dataset has only 10 classes the number of neurons is changed to 10 and the entire architecture is retained as it is. The architecture of GoogLeNet is shown in Figure 3.

4.0. TRAINING AND TESTING OF THE MODELS

A transfer learning approach using Alenet, VGG16, and GoogLeNet was implemented in Matlab release 2018b. The training was carried out with HP Pavilion Laptop with Intel (R) Core (TM) i7-8550U CPU @ 1.8 GHz with 16 GB Ram. The first step in the training process is to resize the images to suit the requirements of the pre-trained architecture. This can be easily performed using the imageDataAugmenter object in Matlab. In addition, various other options are also available to rotate, translate, and reflect the images to combat over-fitting. The modified images need not be stored in the computer as the datastore in Matlab which augments the images during training according to the options specified in the imageDataAugmenter object and then discards the images after the training process is complete.

4.1. HYPER PARAMETER TUNNING WITH ALEXNET ARCHITECTURE

Tuning hyperparameters in machine learning is a very important task as it affects the learning rate of the algorithm, the time complexity of the algorithm, and improves the performance of the algorithm. Normally used approaches for tunning hyperparameters are grid search and random search techniques. But considering the computational complexity involved with very high numbers of hyperparameters to be tunned, in this work coarse-to-fine tunning is adopted to find the optimal set of hyperparameters. In this technique, a random search is carried out to find the promising areas in the search space. This is followed by a deeper search around the promising search space to find the optimal values for hyperparameters. Out of all available well-known architectures, Alexnet has the least number of layers and trainable parameters. Hence the hyperparameter tunning for the Spice10 dataset is carried out in this work using Alexnet architecture.

The most common optimization algorithms used are the adaptive moment estimator (ADAM), stochastic gradient descent (SGD), and root-mean-square propagation (RMSprop). By experimentation it was found SGD with a momentum factor of 0.9 was able to perform better than ADAM and RMSprop. The mini-batch was set as 16 because of limited GPU memory capability and a bigger value caused problems in the convergence of the algorithm. Using the coarse-to-fine technique the optimal values of the hyper-parameters were found as maximum epochs as 15, validation frequency of 50, and validation patience as 5.

The most important hyperparameter is the learning rate. To obtain the best learning rate the learning rate was changed from $1e^{-3}\ to\ 1e^{-5}$ to find the performance of the algorithm. 90% of the images in the dataset (1800 images) were used for training and 10% of the images (200 images) in the dataset were used for validation. To find the most important learning rate hyperparameter several experiments

Table 1. The modified architecture of Alexnet used for classification of spices in Spice10 dataset

	Name	Type	Specification	Feature Maps	Stride	Padding
Input						
1	'data'	Image Input	227x227x3 images with 'zerocenter' normalization	-	-	-
Convolution Layer 1						
2	'conv1'	Convolution	11x11x3 convolutions	96	[4 4]	[0 0 0 0]
3	'relu1'	Activation Function	ReLU	-	-	-
4	'norm1'	Normalization	cross channel normalization with 5 channels per element	96	-	-
5	'pool1'	Pooling	3x3 max pooling	96	[2 2]	[0 0 0 0]
Convolution Layer 2						
6	'conv2'	Convolution	5x5x48 convolutions	256	[1 1]	[2 2 2 2]
7	'relu2'	Activation Function	ReLU	-	-	-
8	'norm2'	Normalization	cross channel normalization with 5 channels per element	256	-	-
9	'pool2'	Pooling	3x3 max pooling	256	[2 2]	[0 0 0 0]
Convolution Layer 3						
10	'conv3'	Convolution	3x3x256 convolutions	384	[1 1]	[1 1 1 1]
11	'relu3'	Activation Function	ReLU	-	-	-
Convolution Layer 4						
12	'conv4'	Convolution	3x3x192 convolutions	384	[1 1]	[1 1 1 1]
13	'relu4'	Activation Function	ReLU	-	-	-
Convolutional Layer 5						
14	'conv5'	Convolution	3x3x192 convolutions	256	[1 1]	[1 1 1 1]
15	'relu5'	Activation Function	ReLU	-	-	-
16	'pool5'	Pooling	3x3 max pooling	256	[2 2]	[0 0 0 0]
Fully Connected Layer 1						
17	'fc6'	Fully Connected	4096 fully connected layer	-	-	-
18	'relu6'	Activation Function	ReLU	-	-	-
19	'drop6'	Dropout	50% dropout	-	-	-
Fully Connected Layer 2						
20	'fc7'	Fully Connected	4096 fully connected layer	-	-	-
21	'relu7'	Activation Function	ReLU	-	-	-
22	'drop7'	Dropout	50% dropout	-	-	-
Fully Connected Layer 3						
23	'fc8'	Fully Connected	10 fully connected layer	-	-	-
24	'prob'	Activation Function	softmax	-	-	-
Output Layer						
25	'output'	Classification	crossentropyex	-	-	-

	Name	Type	Specification	Feature Maps	Stride	Padding
		Output				

were conducted and the results are summarized in Table 3.It was found the network did not learn when the learning rate was set at $1e^{-3}$ which is a case of under-fitting. From the results obtained shown in Table 3, the best learning rate was chosen as $1e^{-4}$ as it was able to produce a better average validation accuracy of 85.5% with a 1.32% standard deviation and lesser training time when compared to a learning rate of $1e^{-5}$.

For the results obtained in Table 3, the number of neurons in the fully connected 'fc7' layer shown in Table 1 was 4096. Experiments were also conducted by varying the neurons in the fully connected layer 'fc7' shown in Table 1. Experiments were conducted with 2048 and 1024 neurons using the learning rate as $1e^{-4}$. The results of the experiments are shown in Table 4. The results indicate decreasing the number of neurons to 2048 and 1024 does not improve the validation accuracy of the Alexnet model above 85.5% as shown in Table 3 and hence the number of neurons is not changed.

4.2. RESULTS OBTAINED WITH ALEXNET

Alexnet architecture is loaded into Matlab and the last three layers are modified to match with the Spice10 dataset as shown in Table 1. The Alexnet requires an input size of 227x227x3 but input images in the Spice10 dataset have different sizes and hence augmented image datastore available in Matlab is used to automatically resize the training and validation images to 227x227x3. In addition, the images are rotated by an angle of 20 degrees and also translated by 3 pixels horizontally as well as vertically to avoid over-fitting. The Alexnet was trained using the optimized hyperparameters. The best result obtained during training using optimized hyperparameters is summarized in Table 5.

From Figure 4 it can be inferred that when the validation accuracy does not significantly change for 5 epochs the training stops automatically as validation patience was specified as 5 during training. This prevents over-fitting of the model and the validation accuracy of 88.5% indicates that the model will perform at an accuracy of at least 88.5% on the unseen dataset during implementation.

Figure 5 shows the confusion chart obtained for the validation set. As seen from the Figure the validation accuracy is 88.5%. The diagonal elements of the confusion chart show the images correctly identified by the Alexnet model. Out of 20 images of Cinnamon in the validation data, the model can identify all of them correctly but for cumin, it can identify only 15 out of the 20 images correctly. Two images of cumin were wrongly identified as cinnamon, two images were wrongly identified as mustard, and one image was wrongly identified as clove.

4.3. RESULTS OBTAINED WITH VGG16

VGG16 architecture is loaded into Matlab and the last three layers are modified to match with the Spice10 dataset as shown in table 2. The VGG16 requires an input size of 224x224x3 but input images in the Spice10 dataset have different sizes and hence augmented image datastore available in Matlab is used to automatically resize the training and validation images to 224x224x3. In addition, the images are rotated by an angle of 25 degrees and also translated by 3 pixels horizontally as well as vertically to avoid over-fitting. 90% of the images in the dataset were used for training and 10% of the images in the dataset were used for validation. The VGG16 was also trained using the SGD algorithm with the optimized hyperparameters. Validation accuracy obtained during the five runs of the algorithm is summarized in Table 6. TheVGG16 architecture can produce an average validation accuracy of 93.06% which is far higher than Alexnet but has an average time for training is 1 hour 12 minutes which is very high when compared to Alexnet. This time for training is high since the number of trainable parameters is far more when compared to Alexnet.

Figure 6 shows the training progress for the best results obtained by VGG16 architecture which is 94% and it can be inferred that when the validation accuracy does not significantly change for 5 epochs the training stops automatically as validation patience was specified as 5 during training. This prevents over-fitting of the model and the validation accuracy of 94 % indicates that the model will perform at an accuracy of at least 94% on the unseen dataset during implementation. Figure 7 shows the confusion matrix obtained for the validation set. As seen from the Figure the validation accuracy is 94%. The diagonal elements of the confusion chart show the images correctly identified by the VGG16 model. The VGG16 model can correctly identify all 20 images of bay leaf, cinnamon, cumin, fenugreek, and pepper available in the validation dataset.

4.4. RESULTS OBTAINED WITH GOOGLENET

After loading the architecture of GooLeNet in Matlab the images sizes were automatically modified to 224x224x3 using the image datastore function. In addition, the images are rotated by an angle of 25 degrees and also translated by 3 pixels horizontally as well as vertically to avoid overfitting. 90% of the images in the dataset were used for training and 10% of the images in the dataset were used for validation. The GooLeNet was also trained using the SGD algorithm with the optimized hyperparameters. Validation accuracy obtained during the five runs of the algorithm is summarized in Table 7. Even though GoogLeNet can produce an average validation accuracy of 90.2% it is lesser than the average validation accuracy of the VGG16

Table 2. The modified architecture of VGG16 used for the classification of spices

	Name	Type	Specification	Feature Maps	Stride	Padding
Input						
1	'input'	Image Input	224x224x3 images with 'zerocenter' normalization	-	-	-
Convolution Layer 1						
2	'conv1_1'	Convolution	3x3x3 convolutions	64	[1 1]	[1 1 1 1]
3	'relu1_1'	Activation Function	ReLU	-	-	-
Convolution Layer 2						
4	'conv1_2'	Convolution	3x3x64 convolutions	64	[1 1]	[1 1 1 1]
5	'relu1_2'	Activation Function	ReLU	-	-	-
6	'pool1'	Pooling	2x2 max pooling	64	[2 2]	[0 0 0 0]
Convolution Layer 3						
7	'conv2_1'	Convolution	3x3x64 convolutions	128	[1 1]	[1 1 1 1]
8	'relu2_1'	Activation Function	ReLU	-	-	-
Convolution Layer 4						
9	'conv2_2'	Convolution	3x3x128 convolutions	128	[1 1]	[1 1 1 1]
10	'relu2_2'	Activation Function	ReLU	-	-	-
11	'pool2'	Pooling	2x2 max pooling	128	[2 2]	[0 0 0 0]
Convolution Layer 5						
12	'conv3_1'	Convolution	3x3x128 convolutions	256	[1 1]	[1 1 1 1]
13	'relu3_1'	Activation Function	ReLU	-	-	-
Convolution Layer 6						
14	'conv3_2'	Convolution	3x3x256 convolutions	256	[1 1]	[1 1 1 1]
15	'relu3_2'	Activation Function	ReLU	-	-	-
Convolution Layer 7						
16	'conv3_3'	Convolution	3x3x256 convolutions	256	[1 1]	[1 1 1 1]
17	'relu3_3'	Activation Function	ReLU	-	-	-
18	'pool3'	Pooling	2x2 max pooling	256	[2 2]	[0 0 0 0]
Convolution Layer 8						
19	'conv4_1'	Convolution	3x3x256 convolutions	512	[1 1]	[1 1 1 1]
20	'relu4_1'	Activation Function	ReLU	-	-	-
Convolution Layer 9						
21	'conv4_2'	Convolution	3x3x512 convolutions	512	[1 1]	[1 1 1 1]
22	'relu4_2'	Activation Function	ReLU	-	-	-
Convolution Layer 10						
23	'conv4_3'	Convolution	3x3x512 convolutions	512	[1 1]	[1 1 1 1]
24	'relu4_3'	Activation Function	ReLU	-	-	-
25	'pool4'	Pooling	2x2 max pooling	512	[2 2]	[0 0 0 0]
Convolution Layer 11						

	Name	Type	Specification	Feature Maps	Stride	Padding
26	'conv5_1'	Convolution	3x3x512 convolutions	512	[1 1]	[1 1 1 1]
27	'relu5_1'	Activation Function	ReLU	-	-	-
Convolution Layer 12						
28	'conv5_2'	Convolution	3x3x512 convolutions	512	[1 1]	[1 1 1 1]
29	'relu5_2'	Activation Function	ReLU	-	-	-
Convolution Layer 13						
30	'conv5_3'	Convolution	3x3x512 convolutions	512	[1 1]	[1 1 1 1]
31	'relu5_3'	Activation Function	ReLU	-	-	-
32	'pool5'	Pooling	2x2 max pooling	512	[2 2]	[0 0 0 0]
Fully Connected Layer 1						
33	'fc6'	Fully Connected	4096 fully connected layer	-	-	-
34	'relu6'	Activation Function	ReLU	-	-	-
35	'drop6'	Dropout	50% dropout	-	-	-
Fully Connected Layer 2						
36	'fc7'	Fully Connected	4096 fully connected layer	-	-	-
37	'relu7'	Activation Function	ReLU	-	-	-
38	'drop7'	Dropout	50% dropout	-	-	-
Fully Connected Layer 3						
39	fc8'	Fully Connected	10 fully connected layer	-	-	-
40	prob'	Activation Function	softmax	-	-	-
Output Layer						
41	output'	Classification Output	crossentropyex	-	-	-

model. The confusion chart obtained for the best results of GoogleNet is shown in Figure 8.

4.5. COMPARISON OF RESULTS

The summary of the best results obtained with Alexnet, VGG16, and GoogLeNet are summarized in the best table 8. The average validation accuracy obtained by VGG16 is 93.06% which is greater than the average accuracy obtained by Alexnet which is 85.5% and GoogLeNet which is 90.2%. Hence for this Spice10 dataset VGG16 model is preferred over the other two models

To further check the performance of the VGG16 model on unseen data, 20 images in the test folder shown in Figure 9 were used for classification. Out of twenty images, the model was correctly able to identify 19 images with a testing accuracy of 95%. The confusion chart obtained during testing of the model with an unseen dataset is shown in Figure 10. Only the powdered form of cloves shown in the second row of Figure 8 was wrongly identified as cinnamon by the VGG16 model. This is due to the high similarity in color between cinnamon and cloves.

4.6. FEATURE MAP

Features help us to understand the different features learned by the model. The feature map for the fully connected layer obtained for the 10 classes available in the Spice10 dataset is shown in Figure 11 to visualize the features learned for each spice by the VGG16 model. From the feature map, we can infer the model has correctly learned the color, texture, size, and other features of the spices.

6.0. CONCLUSION

A Spice10 dataset which consists of 2000 images of ten different spices is created in this research work to examine if a machine can classify those images using the transfer learning approach. The classification of spices is a challenging job for a machine as spices are available in many forms,

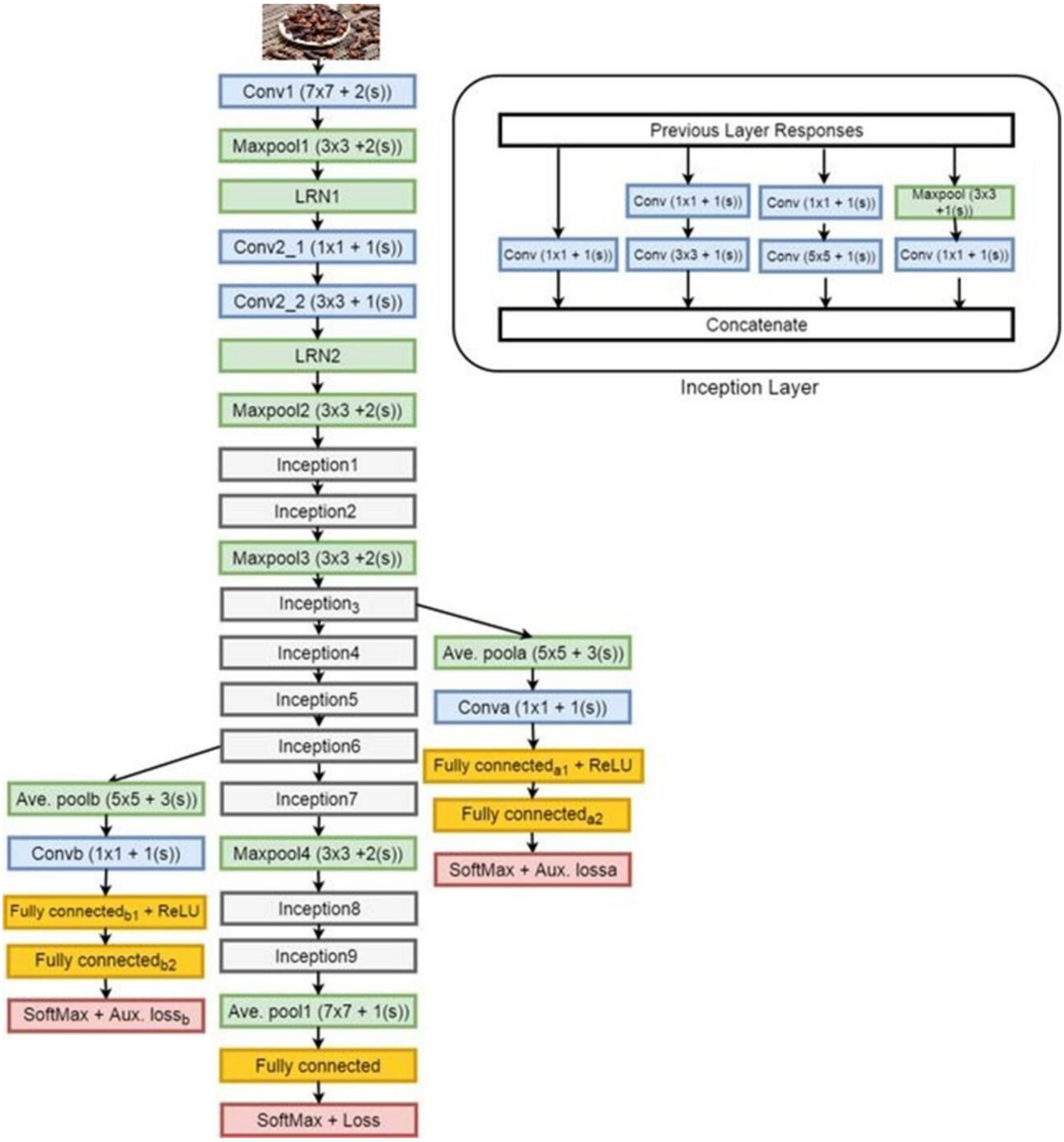


Figure 3. Architecture of GooLeNet

Table 3. Validation accuracy obtained with various learning rates

Learning rate	Validation accuracy	Time	Learning rate	Validation accuracy	Time
$1e^{-4}$	85%	7 min 32 sec	$1e^{-5}$	81%	9 min 34 sec
	84%	5 min 20 sec		79.5%	5 min 39 sec
	85%	7 min 52 sec		88%	9 min 7 sec
	87.5%	6 min 39 sec		82%	9 min 47 sec
	86%	6 min 42 sec		84.5%	6 min 56 sec
Mean	85.5%	6 min 49 sec	Mean	83%	8 min 12 sec
Std.Dev.	1.32%	58.92 sec	Std.Dev.	3.33%	1 min 49 sec

Table 4. Validation accuracy obtained by the number of neurons in the fully connected layer ‘fc7’

Number of Neurons	**Validation Accuracy**	**Time**	**Number of Neurons**	**ValidationAccuracy**	**Time**
2048	81%	5 min 47 sec	1024	84%	6 min 24 sec
	81.5%	5 min 31 sec		79%	7 min 50 sec
	81.5%	10 min 36 sec		83%	9 min 57 sec
	78.5%	6 min 56 sec		80.5%	7 min 14 sec
	83%	6 min 56 sec		84.5%	11 min 53 sec
Mean	81.1%	6 min 29 sec	**Mean**	82.2%	8 min 40 sec
Std. Dev.	1.63%	2 min 1 sec	**Std. Dev.**	2.36%	2 min 13 sec

Table 5. Summary of results obtained with Alexnet architecture

Validation Loss	**Validation Accuracy**	**Training Accuracy**	**Training Time**	**Epoch**	**Iterations**
0.3890	88.5%	81.25%	8 min 29 sec	15	1680

have many similarities in color, shape, and texture. By fine-tuning Alexnet, VGG16, and GoogLeNet architecture, the classification of spices was successfully implemented. The average validation accuracy obtained by the VGG16 model is 93.06% and it can outperform Alexnet and GoogLeNet models. Further, when the VGG16 model was tested on an unseen dataset it was able to produce a testing accuracy of 95%. The high accuracy of the VGG16 model indicates it can be successfully used for the classification of spices.

7.0. FUTURE WORK

These datasets can be combined with the existing food dataset to form a large-scale food dataset which will be a vital resource for the development of image classification and recognition algorithm. A larger dataset with many other spices would be trained on a GPU and using the feature maps a mobile application for real-time classification of spices would be developed. An extension of the research work would be to add more spices and research with other pre-trained models.

ACKNOWLEDGMENT

The first author would like to thank the University of Texas at Austin for offering AI and ML courses through Great Learning. The authors would like to acknowledge all support and resources received from the Royal Commission of Jubail and Yanbu for the successful completion of this work. We extend our appreciation to Mrs. Sornalakshmi Arunachalam for her efforts in the preparation of the dataset.

Submitted: February 14, 2022 AST, Accepted: May 15, 2022 AST

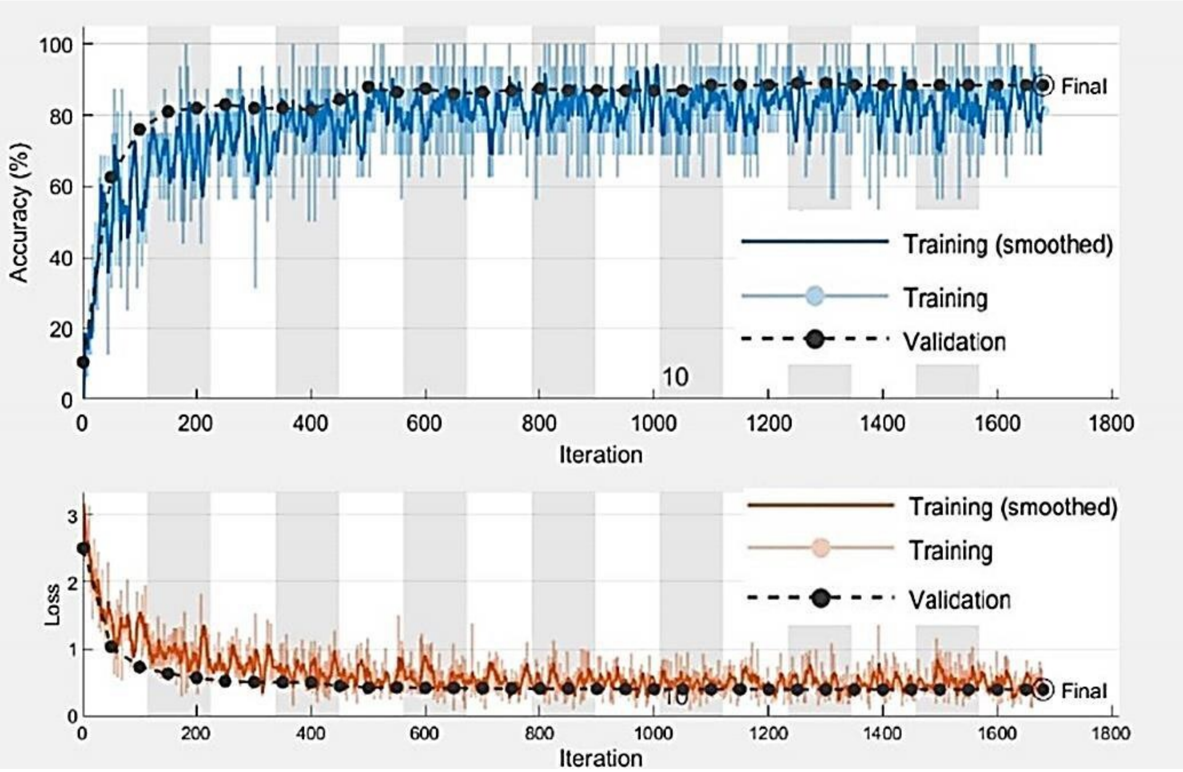
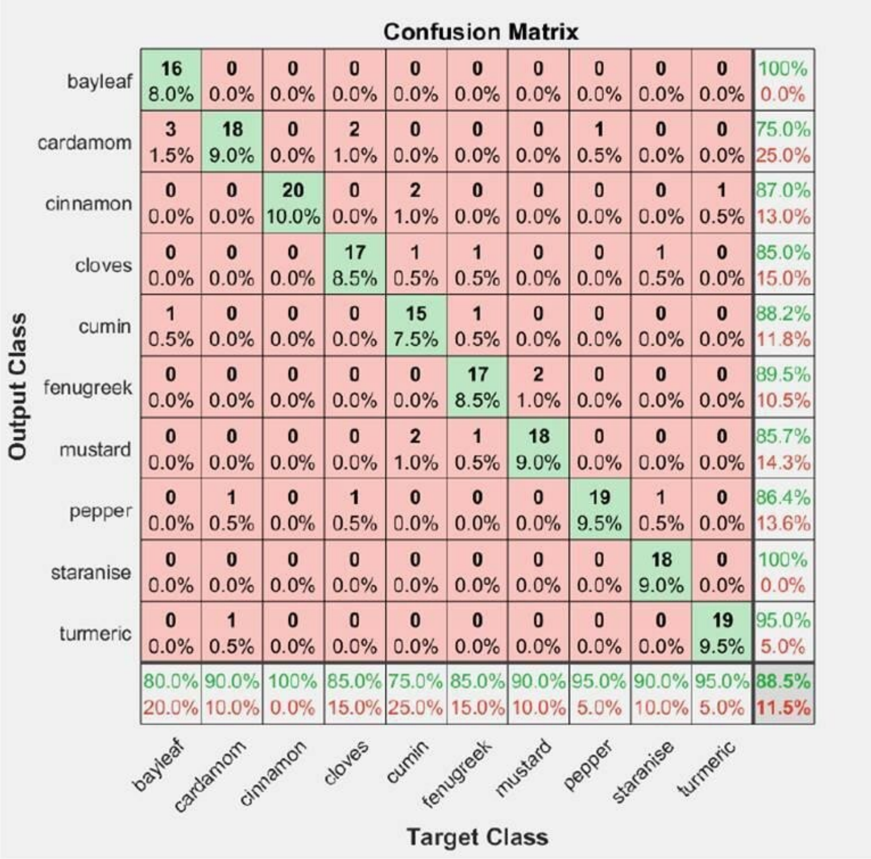


Figure 4. Training Progress for the ALEXNET architecture


Confusion Matrix
Output Class
Target Class

	bayleaf	cardamom	cinnamon	cloves	cumin	fenugreek	mustard	pepper	staranise	turmeric	
bayleaf	16 8.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
cardamom	3 1.5%	18 9.0%	0 0.0%	2 1.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	75.0% 25.0%
cinnamon	0 0.0%	0 0.0%	20 10.0%	0 0.0%	2 1.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	87.0% 13.0%
cloves	0 0.0%	0 0.0%	0 0.0%	17 8.5%	1 0.5%	1 0.5%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	85.0% 15.0%
cumin	1 0.5%	0 0.0%	0 0.0%	0 0.0%	15 7.5%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	88.2% 11.8%
fenugreek	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	17 8.5%	2 1.0%	0 0.0%	0 0.0%	0 0.0%	89.5% 10.5%
mustard	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 1.0%	1 0.5%	18 9.0%	0 0.0%	0 0.0%	0 0.0%	85.7% 14.3%
pepper	0 0.0%	1 0.5%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	19 9.5%	1 0.5%	0 0.0%	86.4% 13.6%
staranise	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	18 9.0%	0 0.0%	100% 0.0%
turmeric	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	19 9.5%	95.0% 5.0%
	80.0% 20.0%	90.0% 10.0%	100% 0.0%	85.0% 15.0%	75.0% 25.0%	85.0% 15.0%	90.0% 10.0%	95.0% 5.0%	90.0% 10.0%	95.0% 5.0%	88.5% 11.5%

Figure 5. Confusion Chart obtained for the ALEXNET model with validation data

Table 6. Summary of results obtained with VGG16 architecture

Learning Rate	Validation Accuracy	Time
$1e^{-4}$	91.50%	62 min 6 sec
	94.00%	72 min 49 sec
	93.95%	74 min 7 sec
	92.95%	86 min 33 sec
	92.93%	68 min 58 sec
Mean	93.06%	72 min 54 sec
Std.Dev.	1.02%	8 min 56 sec

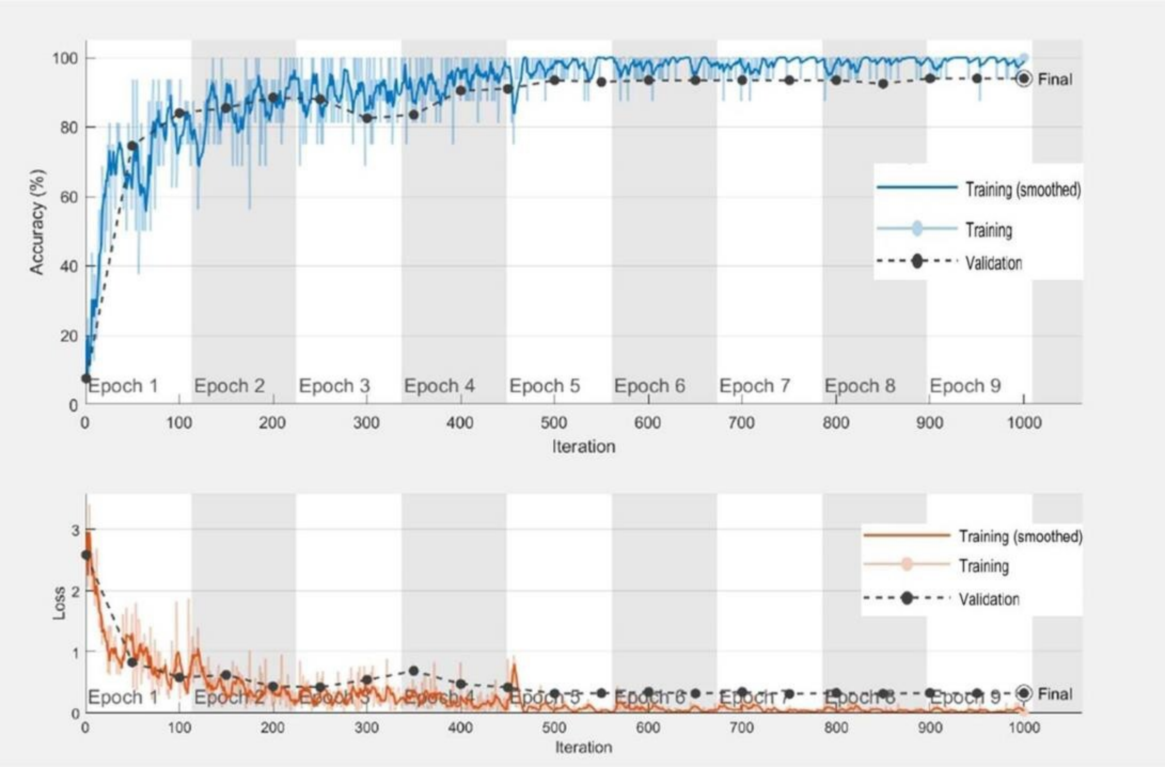
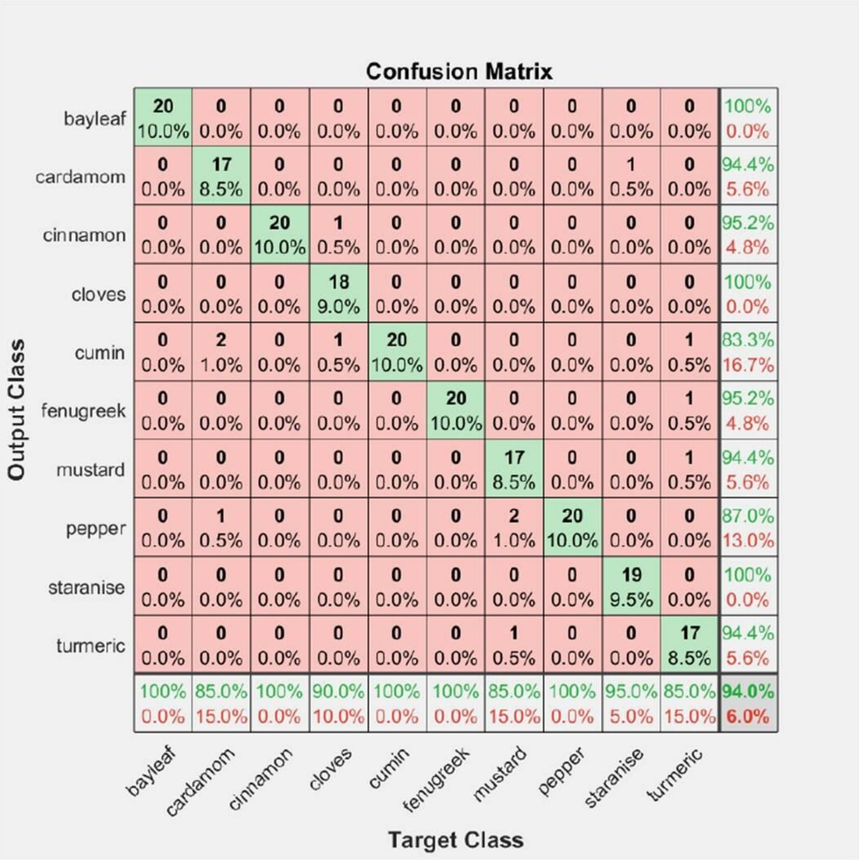


Figure 6. Training progress for the VGG16 architecture.

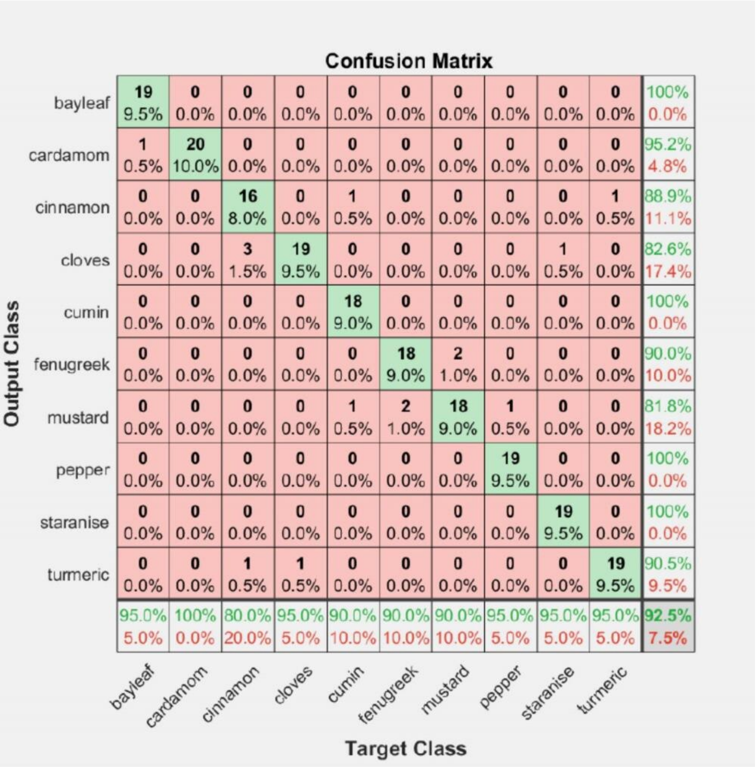


	bayleaf	cardamom	cinnamon	cloves	cumin	fenugreek	mustard	pepper	staranise	turmeric	
bayleaf	20 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
cardamom	0 0.0%	17 8.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	94.4% 5.6%
cinnamon	0 0.0%	0 0.0%	20 10.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.2% 4.8%
cloves	0 0.0%	0 0.0%	0 0.0%	18 9.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
cumin	0 0.0%	2 1.0%	0 0.0%	1 0.5%	20 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	83.3% 16.7%
fenugreek	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 10.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	95.2% 4.8%
mustard	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	17 8.5%	0 0.0%	0 0.0%	1 0.5%	94.4% 5.6%
pepper	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 1.0%	20 10.0%	0 0.0%	0 0.0%	87.0% 13.0%
staranise	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	19 9.5%	0 0.0%	100% 0.0%
turmeric	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	17 8.5%	94.4% 5.6%
	100% 0.0%	85.0% 15.0%	100% 0.0%	90.0% 10.0%	100% 0.0%	100% 0.0%	85.0% 15.0%	100% 0.0%	95.0% 5.0%	85.0% 15.0%	94.0% 6.0%

Figure 7. Confusion Chart obtained for the VGG16 model

Table 7. Summary of results obtained with GoogLeNet architecture

Learning Rate	Validation Accuracy	Time
$1e^{-4}$	90%	12 min 46 sec
	92.5%	22 min 52 sec
	89.0%	23 min 50 sec
	91.0%	17 min 18 sec
	88.5%	48 min 28 sec
Mean	90.2%	25 min 2 sec
Std.Dev.	1.43%	12 min 22 sec



	bayleaf	cardamom	cinnamon	cloves	cumin	fenugreek	mustard	pepper	staranise	turmeric	
bayleaf	19 9.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
cardamom	1 0.5%	20 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.2% 4.8%
cinnamon	0 0.0%	0 0.0%	16 8.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	88.9% 11.1%
cloves	0 0.0%	0 0.0%	3 1.5%	19 9.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	82.6% 17.4%
cumin	0 0.0%	0 0.0%	0 0.0%	0 0.0%	18 9.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
fenugreek	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	18 9.0%	2 1.0%	0 0.0%	0 0.0%	0 0.0%	90.0% 10.0%
mustard	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	2 1.0%	18 9.0%	1 0.5%	0 0.0%	0 0.0%	81.8% 18.2%
pepper	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	19 9.5%	0 0.0%	0 0.0%	100% 0.0%
staranise	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	19 9.5%	0 0.0%	100% 0.0%
turmeric	0 0.0%	0 0.0%	1 0.5%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	19 9.5%	90.5% 9.5%
	95.0% 5.0%	100% 0.0%	80.0% 20.0%	95.0% 5.0%	90.0% 10.0%	90.0% 10.0%	90.0% 10.0%	95.0% 5.0%	95.0% 5.0%	95.0% 5.0%	92.5% 7.5%

Figure 8. Confusion Chart obtained for the GoogLeNet model

Table 8. Comparison of the best results obtained by various architectures

Architecture	Learning rate	Validation Loss	Training Accuracy	Validation Accuracy	Training Time	Epoch	Iterations
Alexnet	$\mathbf{1e^{-4}}$	0.3890	81.25%	88.5%	8 min 29 sec	15	1680
VGG16	$\mathbf{1e^{-4}}$	**0.3164**	**100%**	**94.00%**	**72 min 49 sec**	**9**	**1000**
GoogLeNet	$1e^{-4}$	0.2550	93.75%	92.5%	22 min 52 sec	14	1550

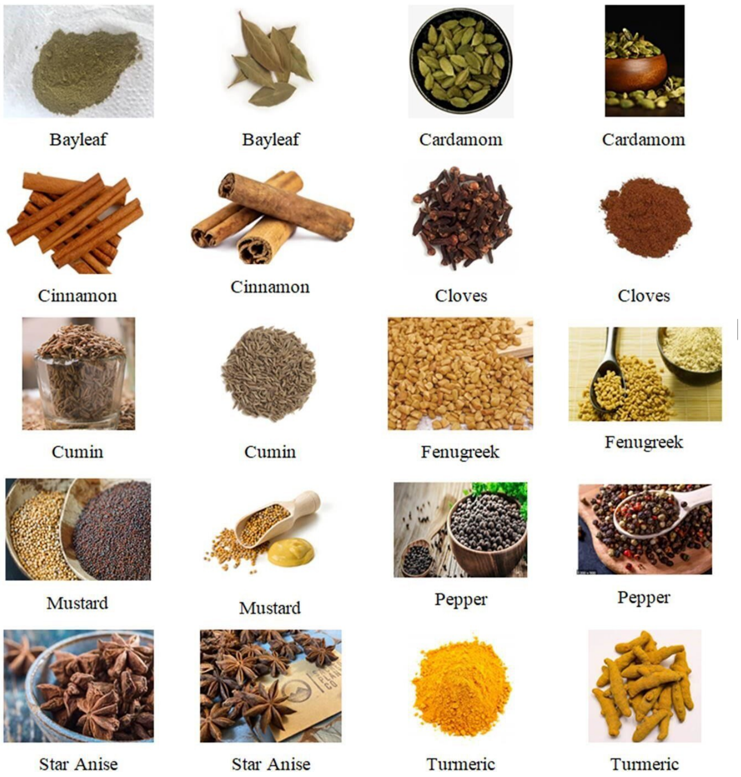
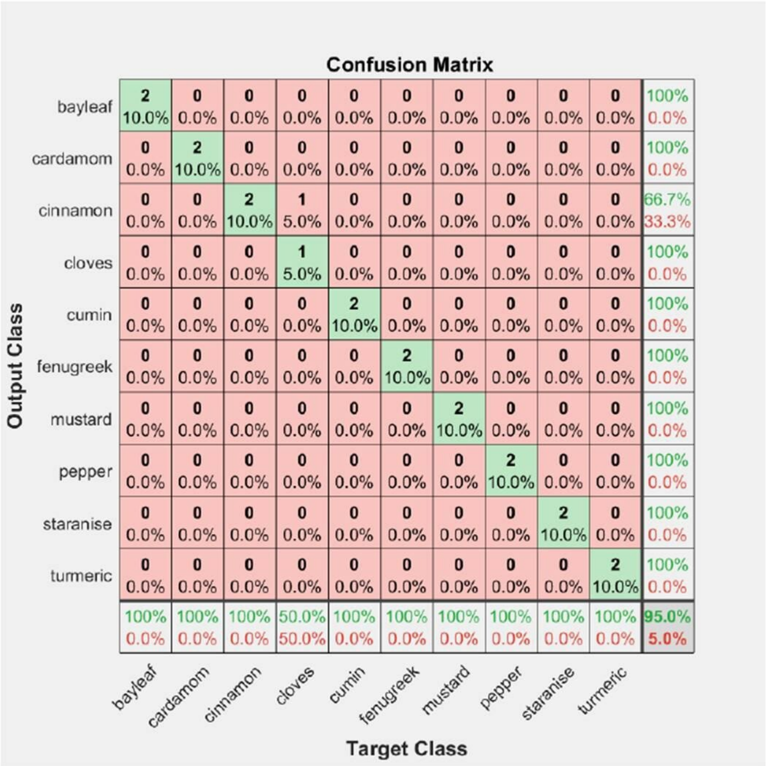


Figure 9. Images in the test folder of Spice10 dataset



	bayleaf	cardamom	cinnamon	cloves	cumin	fenugreek	mustard	pepper	staranise	turmeric	
bayleaf	2 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
cardamom	0 0.0%	2 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
cinnamon	0 0.0%	0 0.0%	2 10.0%	1 5.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	66.7% 33.3%
cloves	0 0.0%	0 0.0%	0 0.0%	1 5.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
cumin	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
fenugreek	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
mustard	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 10.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
pepper	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 10.0%	0 0.0%	0 0.0%	100% 0.0%
staranise	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 10.0%	0 0.0%	100% 0.0%
turmeric	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 10.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	50.0% 50.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	95.0% 5.0%

Figure 10. Confusion Chart obtained for the VGG16 model with the test dataset

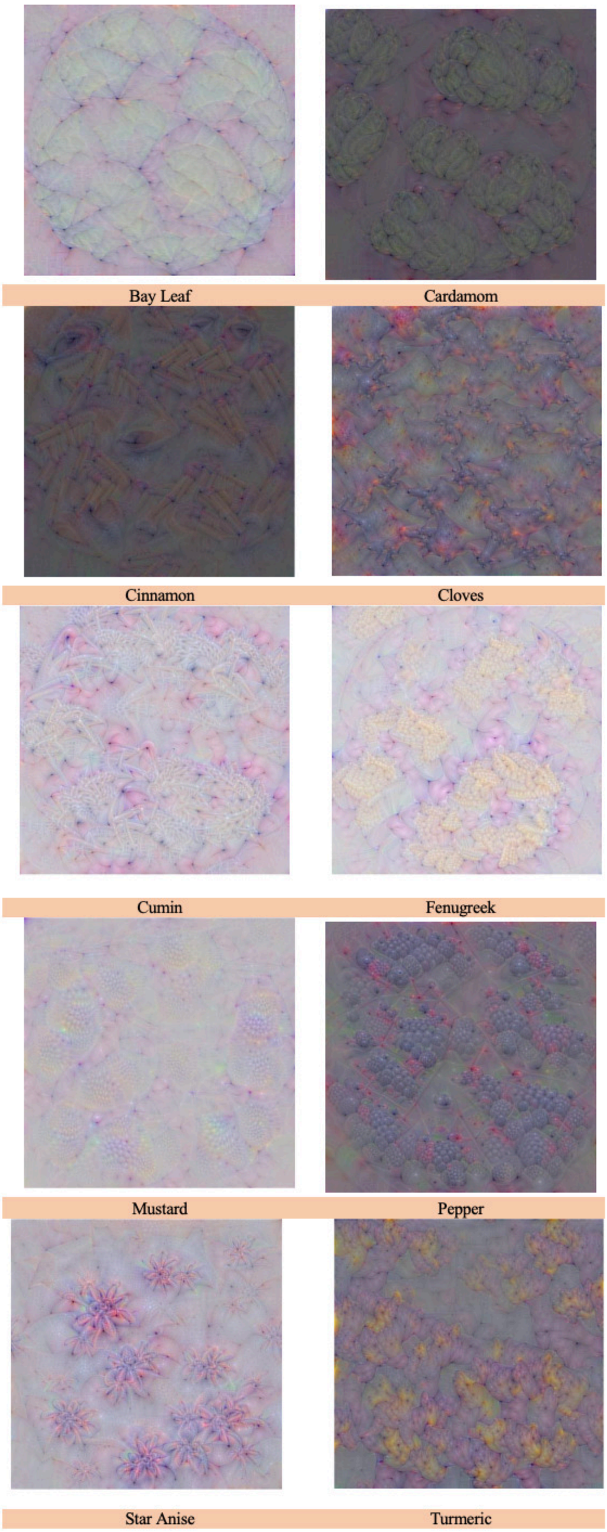


Figure 11. Feature maps of the spices using the VGG16 model

REFERENCES

1. Aggarwal B, Yost D. *Healing Spices: How to Use 50 Everyday and Exotic Spices to Boost Health and Beat Disease*. Sterling Publishing Co., Inc; 2011.

2. Imam H, Riaz Z, Alam I. Cancer preventing spices. *J Cancer Metastasis Treat*. 2015;1:41-42. doi:10.4103/2394-4722.154131

3. Lindberg Madsen H, Bertelsen G. Spices as antioxidants. *Trends Food Sci Technol*. 1995;6(8):271-277. doi:10.1016/s0924-2244(00)89112-8

4. Arora DS, Kaur J. Antimicrobial activity of spices. *Int J Antimicrob Agents*. 1999;12(3):257-262. doi:10.1016/s0924-8579(99)00074-6

5. Surh YJ. Anti-tumor promoting potential of selected spice ingredients with antioxidative and anti-inflammatory activities: a short review. *Food Chem Toxicol*. 2002;40(8):1091-1097. doi:10.1016/s0278-6915(02)00037-6

6. Kuttan R, Bhanumathy P, Nirmala K, George MC. Potential anticancer activity of turmeric (Curcuma longa). *Cancer Lett*. 1985;29(2):197-202. doi:10.1016/0304-3835(85)90159-4

7. Chattopadhyay I, Biswas K, Bandyopadhyay U, Banerjee RK. Turmeric and curcumin: biological actions and medicinal applications. doi:10.2307/24107978

8. Thippeswamy NB, Naidu KA. Antioxidant potency of cumin varieties—cumin, black cumin and bitter cumin—on antioxidant systems. *Eur Food Res Technol*. 2005;220(5-6):472-476. doi:10.1007/s00217-004-1087-y

9. Guil-Guerrero JL, Martínez-Guirado C, Del Mar Rebolloso-Fuentes M, Carrique-Pérez A. Nutrient composition and antioxidant activity of 10 pepper (Capsicum annuun) varieties. *Eur Food Res Technol*. 2006;224(1):1-9. doi:10.1007/s00217-006-0281-5

10. Padmashree A, Roopa N, Semwal AD, Sharma GK, Agathian G, Bawa AS. Star-anise (Illicium verum) and black caraway (Carum nigrum) as natural antioxidants. *Food Chem*. 2007;104(1):59-66. doi:10.1016/j.foodchem.2006.10.074

11. Basch E, Ulbricht C, Kuo G, Szapary P, Smith M. Therapeutic Applications of Fenugreek. *Oxford Handb Transcranial Stimul*. 2012;8.

12. Geberemeskel GA, Debebe YG, Nguse NA. Antidiabetic Effect of Fenugreek Seed Powder Solution (*Trigonella foenum-graecum L.*) on Hyperlipidemia in Diabetic Patients. *J Diabetes Res*. 2019;2019:1-8. doi:10.1155/2019/8507453

13. Jamal A, Javed K, Aslam M, Jafri MA. Gastroprotective effect of cardamom, Elettaria cardamomum Maton. fruits in rats. *J Ethnopharmacol*. 2006;103(2):149-153. doi:10.1016/j.jep.2005.07.016

14. Fang F, Sang S, Chen KY, Gosslau A, Ho CT, Rosen RT. Isolation and identification of cytotoxic compounds from Bay leaf (Laurus nobilis). *Food Chem*. 2005;93(3):497-501. doi:10.1016/j.foodchem.2004.10.029

15. Bhattacharya A, Li Y, Wade KL, Paonessa JD, Fahey JW, Zhang Y. Allyl isothiocyanate-rich mustard seed powder inhibits bladder cancer growth and muscle invasion. *Carcinogenesis*. 2010;31(12):2105-2110. doi:10.1093/carcin/bgq202

16. Gruenwald J, Freder J, Armbruester N. Cinnamon and health. *Crit Rev Food Sci Nutr*. 2010;50(9):822-834. doi:10.1080/10408390902773052

17. Joshi B, Prasad Sah G, Bahadur Basnet B, et al. Phytochemical extraction and antimicrobial properties of different medicinal plants: Ocimum sanctum (Tulsi), Eugenia caryophyllata (Clove), Achyranthes bidentata (Datiwan) and Azadirachta indica (Neem). *J Microbiol Antimicrob*. 2011;3:1-7.

18. Russell S, Peter N. *Artificial Intelligence: A Modern Approach*. Pearson; 2020.

19. Kakani V, Nguyen VH, Kumar BP, Kim H, Pasupuleti VR. A critical review on computer vision and artificial intelligence in food industry. *J Agric Food Res*. 2020;2:100033. doi:10.1016/j.jafr.2020.100033

20. Vedaldi A, Fulkerson B. Vlfeat - an open and portable library of computer vision algorithms. In: *Proceedings of the International Conference on Multimedia - MM '10*. ACM Press; 2010:1469-1472. doi:10.1145/1873951.1874249

21. Krizhevsky A, Sutskever I, Hinton GE. *ImageNet Classification with Deep Convolutional Neural Networks*.; 2012.

22. Accessed August 11, 2020. https://www.stemmerimaging.com/media/uploads/general/sis/20/2018-03-Market-Brochure-Food-en-WEB.pdf

23. Min W, Jiang S, Liu L, Rui Y, Jain R. A survey on food computing. *ACM Comput Surv*. 2020;52(5):1-36. doi:10.1145/3329168

24. Lo FPW, Sun Y, Qiu J, Lo B. Image-Based Food Classification and Volume Estimation for Dietary Assessment: A Review. *IEEE J Biomed Health Inform*. 2020;24(7):1926-1939. doi:10.1109/jbhi.2020.2987943

25. Qin J, Pan W, Xiang X, Tan Y, Hou G. A biological image classification method based on improved CNN. *Ecol Inform*. 2020;58:101093. doi:10.1016/j.ecoinf.2020.101093

26. Matsuda Y, Hoashi H, Yanai K. Recognition of multiple-food images by detecting candidate regions. In: *Proceedings - 2012 IEEE International Conference on Multimedia and Expo*. IEEE; 2012:25-30. doi:10.1109/icme.2012.157

27. Kawano Y, Yanai K. Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag; 2015:3-17. https://doi.org/10.1007/978-3-319-16199-0_1

28. Bossard L, Guillaumin M, Van Gool L. Food-101 – Mining Discriminative Components with Random Forests. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag; 2014:446-461. https://doi.org/10.1007/978-3-319-10599-4_29

29. Kaur P, Sikka K, Wang W, Belongie S, Divakaran A. FoodX-251: A Dataset for Fine-grained Food Classification. Published online 2019.

30. Zhang W, Wu J, Yang Y. Wi-HSNN: A subnetwork-based encoding structure for dimension reduction and food classification via harnessing multi-CNN model high-level features. *Neurocomputing*. 2020;414:57-66. doi:10.1016/j.neucom.2020.07.018

31. Sengur A, Akbulut Y, Budak U. Food Image Classification with Deep Features. In: *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE; 2019. doi:10.1109/idap.2019.8875946

32. Kogias K, Andreadis I, Dalakleidi K, Nikita KS. A Two-Level Food Classification System For People With Diabetes Mellitus Using Convolutional Neural Networks. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE; 2018:2603-2606. doi:10.1109/embc.2018.8512839

33. Ciocca G, Napoletano P, Schettini R. CNN-based features for retrieval and classification of food images. *Comput Vis Image Underst*. 2018;176-177(https://doi.org/10.1016/j.cviu.2018.09.001):70-77. doi:10.1016/j.cviu.2018.09.001

34. Yang H, Kang S, Park C, Lee J, Yu K. A Hierarchical deep model for food classification from photographs. *KSII Trans internet Inf Syst*. 2020;14:1704-1719.

35. Aliady H, Utari DT. GPU based image classification using convolutional neural network chicken dishes classification. *Int J Adv Soft Comput Appl*. 2018;10:1-13.

36. Rajayogi JR, Manjunath G, Shobha G. Indian Food Image Classification with Transfer Learning. In: *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*. IEEE; 2019:17-20. doi:10.1109/csitss47250.2019.9031051

37. Hnoohom N, Yuenyong S. Thai fast food image classification using deep learning. In: *2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON)*. IEEE; 2018:116-119. doi:10.1109/ecti-ncon.2018.8378293

38. Sukvichai K, Maolanon P, Sawanyawat K, Muknumporn W. Food categories classification and Ingredients estimation using CNNs on Raspberry Pi 3. In: *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*. IEEE; 2019. doi:10.1109/ictemsys.2019.8695967

39. Teng J, Zhang D, Lee DJ, Chou Y. Recognition of Chinese food using convolutional neural network. *Multimed Tools Appl*. 2019;78(9):11155-11172. doi:10.1007/s11042-018-6695-9

40. Mezgec S, Seljak BK. Using Deep Learning for Food and Beverage Image Recognition. In: *Proceedings - 2019 IEEE International Conference on Big Data (Big Data)*. IEEE; 2019:5149-5151. doi:10.1109/bigdata47090.2019.9006181

41. Ciocca G, Micali G, Napoletano P. State Recognition of Food Images Using Deep Features. *IEEE Access*. 2020;8:32003-32017. doi:10.1109/access.2020.2973704

42. Pawara P, Okafor E, Surinta O, Schomaker L, Wiering M. Comparing local descriptors and bags of visualwords to deep convolutional neural networks for plant recognition. In: *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*. SCITEPRESS - Science and Technology Publications; 2017:479-486. doi:10.5220/0006196204790486

43. Zhou L, Zhang C, Liu F, Qiu Z, He Y. Application of Deep Learning in Food: A Review. *Compr Rev Food Sci Food Saf*. 2019;18(6):1793-1811. doi:10.1111/1541-4337.12492

44. Chen H, Xu J, Xiao G, Wu Q, Zhang S. Fast auto-clean CNN model for online prediction of food materials. *J Parallel Distrib Comput*. 2018;117:218-227. doi:10.1016/j.jpdc.2017.07.004

45. Hoi S. FoodAI: Food Image Recognition by Deep Learning. Published online 2016.

46. Foodai - State-of-the-art food image recognition technologies. https://foodai.org/

47. Knez S, Šajn L. Food object recognition using a mobile device: evaluation of currently implemented systems. *Trends Food Sci Technol*. 2020;99:460-471. doi:10.1016/j.tifs.2020.03.017

48. Furtado P, Caldeira M, Martins P. Human Visual System vs Convolution Neural Networks in food recognition task: an empirical comparison. *Comput Vis Image Underst*. 2020;191:102878. doi:10.1016/j.cviu.2019.102878

49. Nakamoto H, Nishikubo D, Okada S, Kobayashi F, Kojima F. Food Texture Classification Using Magnetic Sensor and Principal Component Analysis. In: *Proceedings - 2016 Third International Conference on Computing Measurement Control and Sensor Network (CMCSN)*. Vol C. IEEE; 2016:114-117. doi:10.1109/cmcsn.2016.39

50. He Y, Xu C, Khanna N, Boushey CJ, Delp EJ. *Analysis of Food Images : Features and Classification School of Electrical and Computer Engineering*. Purdue University Department of Electronics and Communication Engineering, Graphic Era University Cancer Epidemiology Program, University of Hawaii Can; 2014.

51. Momeny M, Jahanbakhshi A, Jafarnezhad K, Zhang YD. Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach. *Postharvest Biol Technol*. 2020;166:111204. doi:10.1016/j.postharvbio.2020.111204

52. Jahanbakhshi A, Momeny M, Mahmoudi M, Zhang YD. Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks. *Sci Hortic (Amsterdam)*. 2020;263:109133. doi:10.1016/j.scienta.2019.109133

53. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR; 2015.

54. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society; 2015:1-9. doi:10.1109/cvpr.2015.7298594

55. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2016:770-778. doi:10.1109/cvpr.2016.90

56. ImageNet. Accessed January 12, 2021. http://www.image-net.org

57. Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis*. 2015;115(3):211-252. doi:10.1007/s11263-015-0816-y

58. Marcelino P. Transfer learning from pre-trained models. Towards Data Science. https://towardsdatascience.com/transfer-learning-from-pre-trained-models

59. AlexNet convolutional neural network - MATLAB alexnet - MathWorks India. https://in.mathworks.com/help/deeplearning/ref/alexnet.html;jsessionid=defd62336b19cc09701bf6ba7c81

60. VGG-16 convolutional neural network - MATLAB vgg16 - MathWorks India. Accessed April 23, 2021. https://in.mathworks.com/help/deeplearning/ref/vgg16.html

61. Karri SPK, Chakraborty D, Chatterjee J. Transfer learning based classification of optical coherence tomography images with diabetic macular edema and dry age-related macular degeneration. *Biomed Opt Express*. 2017;8:579.