2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)
Pune, India. Oct 29-30, 2021

Fungus Detection and Identification using Computer Vision Techniques and Convolution Neural Networks

Ajish Nair, Nija K Manomohan, Hrishikesh Reddy, Sankul Pandey, Prof. Priyanka Kadam
SIES GRADUATE SCHOOL OF TECHNOLOGY
ajish.nair17@siesgst.ac.in, nija.manomohan17@siesgst.ac.in, balram.hrishikesh17@siesgst.ac.in, sankul.pandey17@siesgst.ac.in, priyanka.kadam@siesgst.ac.in

***Abstract* – In this paper we have presented a fungus detection system. As we all know that Fungus is a very big risk for the human Life as it can come in contact to humans through food, water, Air etc. Exposure to high concentrations of fungal spores from air can cause many health problems, such as bronchial asthma and other deadly diseases. Fungus is the main risk for food logistics. We aim to build a model which can correctly detect and identify fungus. Datasets plays a critical role throughout the procedure. For detection purposes, an accurate dataset is required. The dataset we are collecting was made using an optical acquisition system. The main task of our project is the image analysis. Image analysis can be done using computer vision techniques or with the help of machine learning. We have explained both the methods of analysis of CVT using Histogram oriented gradients (HOG) and Support Vector Machine and Convolution neural networks (CNN) in detail with various processes and layers involved.**

Keywords - Fungus, fungus detection, CVT, CNN, Segmentation, Thresholding, SVM, Feature Extraction, HOG, CNN layer.

I. INTRODUCTION

In this research, we develop a fungus detection system and algorithms to automatically detect fungus. Three Computer-vision based techniques were considered by us for the detection of fungus spores. The technique which used Histogram of Oriented Gradients was implemented by us. There were other techniques which consisted of fusion of Fourier transform and SIFT features and a third method based on super-pixel and handcrafted features. The results of all the HOG technique encourage for the possibility of early detection of fungus spores from dirt particles with the help of data set which was collected by us online. The dataset thus obtained went through 4 steps which are explained in detail in methodology before passing it to SVM for training and testing purposes. Computer Vision Technique makes use of various image processing techniques. The various steps involved here are as follows:

1. Pre-processing – This step involves of image smoothening, image sharpening and image enhancement which improves the quality of images and suppresses unwanted distortions. Various filters like mean filter, Gaussian filters, etc cab be used for image smoothening.

2. Segmentation – After pre-processing, the second step is to focus on areas of interest, called as 'Foreground'. This process of separating foreground from background is called Segmentation. There are two types: Threshold and Water-Shed. Threshold Segmentation was preferred by us due to its simplicity.

3. Feature Extraction – Histogram of Oriented Gradients (HOG) was implemented as it is invariant to illumination changes because local histograms are normalized with contrast.

4. Identification - SVM classifier was utilized for differentiation between fungus and non-fungus spores. The main function of SVM is to detect and utilize complex patterns from testing data and utilize it for classifying the fungus data in high dimensional feature space.

The other main objective of this research was classification of fungus into its different classes. We used Convolution Neural Networks for this. CNN is a machine learning technique which is used for image, video or data classification. It involves various layers such as:

1. Convolution Layer – This is the main layer that holds the responsibility of automatic feature extraction. Feature extraction is the main part of identification. The feature extraction is done by convolving a filter on the entire image. A new map comes out which contains the specific feature mentioned in the applied filter. So, an image is convolved with multiple filters to extract multiple features out of the same image.

2. Pooling layer – The main purpose of the pooling layer is to reduce the image size and number of parameters. This layer can be the input of the classifier directly but the answer will be still big and huge computational power will be utilized. Therefore, it is better to apply the pooling layer first to further reduce the dimensions. The pooling layer is also normally utilized to insert in between two convolution layers. We used average pooling layer in our method i.e., the output of the depth slice of input is average of all the values of the depth slice.

3. ReLU Layer – ReLU stands for Rectified Linear Units. This is an activation function which is utilized in the proposed approach. The shape of this function similar to the ramp. It has the advantage of less computation cost over other activation functions like sigmoid, tangent hyperbolic. In result of that, it greatly reduces the training time.

4. Fully Connected Layer – After passing through all of the mentioned layers, input fed into the fully connected layer. This layer is responsible for the high-level feature extraction.

5.Softmax Log Loss layer – Right after the application of the fully connected layer, softmax log loss layer is introduced. This includes both softmax and logistic loss. Softmax layer is also a general type of fully connected layer.

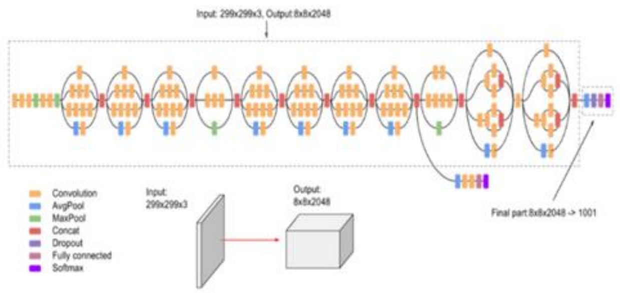

Fig. 1. Layers in InceptionV3 model

We used InceptionV3 architecture for our base model. We read in the pretrained weights and added our own layers on top. We defined the parameters of the model and setup the base model. Pre-processing was done on the images using Image Data Generator. It is used to configure random transformation and normalization operations on the images during training. It creates batches with real time data augmentation. We add a function to ensure that the input training images are preprocessed the same way as original InceptionV3 images were trained. An appropriate compiler is used as we have more than two categories. Figure 1 shows different layers in an InceptionV3 architecture.

II. RELATED WORKS

A. *Computer Vision Technique*

Researchers have been done in the recent years about fungus and other microbials. One of the most prominent is the detection of fungus in air. It started out with an acquisition system with the help of which dataset could be collected for further processes. A feasible approach was proposed using various hardware for same. Initially glass slide was used for sample collection. Air pump was used for sucking in a desired amount of air whose images were clicked using industrial camera. Further analysis was done using MATLAB and OpenCV. Image analysis was done using Computer Vision Technique. The process then included thresholding, segmentation and then support vector machine for classification. Some papers were published which wrote about the comparison of different methods for the analysis of spores. Data acquisition system was developed which consisted of six subsystems namely, Sampling unit, Handling unit, Light sources, Camera unit, Computer and Atmel SAM4S. Fungus spores were collected from air onto an adhesive tape. Handling unit had three motors for different movement of the air sample and place it correctly under the camera. Different light sources were used to click three types of images – Bright field images, dark field images and autofluoroscent images. These images were then analysed in software and the results were compared. Then we saw an approach which involved slight change in the hardware area. To make a cost-effective project, adhesive tape was used instead of a glass slide which turned to yield better results in terms of clarity of the images clicked as well. The industrial camera was used with a microscopic lens. Latest technologies are used to analyse the images.

Two different methods of computer vision were introduced in this. First and more effective one was the Histogram Oriented Gradients method followed by Support vector machine for classification. HOG features included orientation gradient, binning, block normalization. Second method was handcrafted features involving colour, size, saturation etc. Annotation was a key feature in the whole method which also plays an important role in Convolution Neural Networks.

B. *Convolution Neural Network*

It was later when machine learning came into picture and changed the approach for image analysis. Dataset required for analysis was obtained in a similar way as before but the annotation was done by trained engineering students. They manually annotated the images which was a tedious job but results were great. They made a dataset of 40,000 images out which 36,500 were for training set and the rest 3,500 were testing dataset. Convolution neural network is a machine learning technique which involves various layers for extraction and classification. Unlike computer vision technique, CNN has a layer which helps in classification of data. Though the CNN technique gives high accuracy of more than 90%, annotating so many images manually is a tedious task which requires time and concentration.

Other related works include detection of mould in wheat and maize. It made use of electrodes in a reaction chamber. A fungus detection system in greenhouse gases using wireless visual sensor network was also done. It made use of machine learning algorithm. Cameras were inserted in a green house and images of leaves and plants are clicked and analysed on computer using machine learning and image processing. If and when fungus is detected, the sensor sends message to check the humidity and temperature.

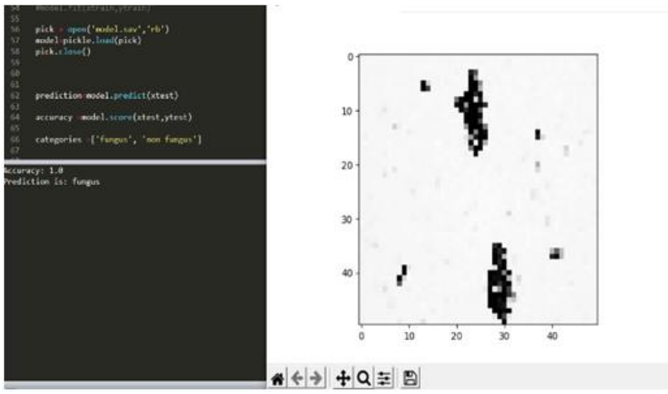

Fig. 2. Detection of fungus and non-fungus spores using CVT.

III. METHODOLOGY

A. *Dataset Acquisition*

One of the most important aspect of the new methods is the dataset. Dataset is basically a collection of images of different types of fungi. We acquired our dataset from various online dataset websites. We collected 9 different type of fungi mainly from the class Candida. We split the dataset into training and testing set and trained our model.

Fig. 3. Accuracy of CNN model.

We collected a dataset of almost 2000 images. But, larger the dataset, better will be the accuracy.

B. CNN Architecture

Convolution Neural Network is a machine learning technique which makes use of various convolutional, pooling, fully connected layers. There are different types of architectures available for classification. They differ with respect to different number of layers which are present in each. We implemented our model using the InceptionV3 architecture. Inception V3 has an architecture which is 48 layers deep. We can even add a pretrained layer on top which has been trained with images from ImageNet.

```
Number of Fungi Classes:                                   9
------------------------------------------------------------
Input Image:                                     299 x 299 x 3
Batch Size:                                               32
------------------------------------------------------------
Total Number of Training Images:                        1428
Total Number of Validation Images:                       476
```

Fig. 4. Training and Testing dataset

We imported the dataset and split it into training and testing dataset.

We read in the pretrained weights and skipped the fully connected output layers. Instead, we added our own layers on top. It included global average pooling layer, batch normalization and activation layers.

```
IMAGE_SIZE = [299, 299]

inception = InceptionV3(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

```
for layer in inception.layers:
    layer.trainable = False

X = inception.output
X = GlobalAveragePooling2D()(X)
X = BatchNormalization()(X)

X = Dense(128, activation=None)(X)
X = BatchNormalization()(X)
X = Activation('relu')(X)

X = Dense(128, activation=None)(X)
X = BatchNormalization()(X)
X = Activation('relu')(X)
X = Dropout(0.3)(X)

X = Dense(num_classes, activation=None)(X)
X = BatchNormalization()(X)
Y = Activation('softmax')(X)

model = models.Model(inputs=inception.input, outputs=Y)
```

Fig. 5. Added layer in InceptionV3 architecture.

For compiling, since we have more than two categories, categorical cross entropy loss was used. We selected Adam optimizer that is based on adaptive estimation and a default learning rate of 0.001. Image Data Generator was used to configure random transformation and normalization operations on the images during the training process.

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

idg = preprocessing.image.ImageDataGenerator(preprocessing_function=inceptionv3_preprocess_input, validation_split=train_val_split)


train_generator = ImageDataGenerator(rescale = 1./255,
                                     shear_range = 0.2,
                                     zoom_range = 0.2,
                                     horizontal_flip = True)

val_generator = ImageDataGenerator(rescale = 1./255)

train_generator = idg.flow_from_directory(
        image_dir,
        target_size=(image_size, image_size),
        batch_size=batch_size,
        class_mode='categorical',
        subset='training',
        shuffle=True)

val_generator = idg.flow_from_directory(
        image_dir,
        target_size=(image_size, image_size),
        batch_size=batch_size,
        class_mode='categorical',
        subset='validation',
        shuffle=True)
```

Fig. 6. CNN model compiling.

We used different parameters such as rescale, shear range, zoom and horizontal flip. Model is trained for 15 epochs and then finally evaluated on a test image.

```
checkpoint = ModelCheckpoint('weights.best.hdf5', monitor='val_acc', verbose=1, mode='max')

history = model.fit_generator(train_generator,\
                              steps_per_epoch=math.ceil(n_train/batch_size),\
                              epochs=15,\
                              validation_data=val_generator,\
                              validation_steps=math.ceil(n_val/batch_size))
```

Fig. 7. Training model.

C. CVT STEPS

As mentioned in the introduction, CVT consists of 4 steps and its implementation is shown below step-wise:

1) Pre-processing

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread("Resources/CA17.png")

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

kernel = np.ones((5, 5), np.float32)/25
dst = cv2.filter2D(img, -1, kernel)
blur=cv2.blur(img, (5, 5))
gblur=cv2.GaussianBlur(img, (5, 5), 0)
median=cv2.medianBlur(img, 5)

titles = ['image', '2D Convolution','blur','GaussianBlur', 'Median Blur']
images= [img,dst, blur,gblur,median]

for i in range(5):
    plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])
    plt.show()
```

Fig. 8. Pre-processing stage.

As seen above, OPENCV methods which is ‘blur’, ‘gaussian blur’, ‘median blur’ where used of which gaussian blur produced best results with no distortion.

2) Segmentation

As seen in the following image, threshold method was used for segmentation which takes source image, threshold value(T), minimum value and threshold type as an input.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread("Resources/CA17.png")

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
kernel = np.ones((5, 5), np.float32)/25
dst = cv2.filter2D(img, -1, kernel)
blur=cv2.blur(img, (5, 5))


gblur=cv2.GaussianBlur(img, (5, 5), 0)
median=cv2.medianBlur(img, 5)

print(gblur.shape)
width ,height = 400,400
gblurResize = cv2.resize(gblur,(width,height))
print(gblurResize.shape)

titles = ['image', '2D Convolution','blur','GaussianBlur', 'Median Blur']
images=  [img,dst, blur,gblur,median]

for i in range(5):
    plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])
    plt.show()
    _, th1 = cv2.threshold(gblurResize, 100, 255, cv2.THRESH_TRUNC)
    cv2.imshow("th1", th1)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

Fig. 9. Segmentation stage.

3) *Feature Extraction*

```
from skimage.io import imread
from skimage.transform import resize
from skimage.feature import hog
import matplotlib.pyplot as plt

#reading the image
img = imread('Resources/CA17.png')
plt.axis("off")
plt.imshow(img)
print(img.shape)

#resizing image
resized_img = resize(img, (128*4, 64*4))
plt.axis("off")
plt.imshow(resized_img)
plt.show()
print(resized_img.shape)

#creating hog features
fd, hog_image = hog(resized_img, orientations=9, pixels_per_cell=(8, 8),
                    cells_per_block=(2, 2), visualize=True,
multichannel=True)
print(fd.shape)
print(hog_image.shape)
plt.axis("off")
plt.imshow(hog_image, cmap="gray")
plt.show()

# save the images
plt.imsave("resized_img.jpg", resized_img)
plt.imsave("hog_image.jpg", hog_image, cmap="gray")
```

Fig. 10. Histogram of Oriented Gradiensts

Histogram of oriented gradients was used for feature extraction and HOG images were saved for training and testing of SVM.

4) *Identification*

C:\Users\HRISHIKESH REDDY\Desktop\hrishi\SVM\classifier.py • - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

classifier.py Package Control Messages

```
import os
import numpy as np
import cv2
import matplotlib.pyplot as plt
import pickle
import random
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

pick_in = open('data1.pickle','rb')
data=pickle.load(pick_in)
pick_in.close()
random.shuffle(data)
features = []
labels = []
for feature ,label in data:
    features.append(feature)
    labels.append(label)
xtrain, xtest, ytrain,ytest = train_test_split(features, labels, test_size=0.50)
#model = SVC(C=1,kernel='poly',gamma= 'auto')
#model.fit(xtrain,ytrain)
pick = open('model.sav','rb')
model=pickle.load(pick)
pick.close()
prediction=model.predict(xtest)
accuracy =model.score(xtest,ytest)
categories =['fungus', 'non fungus']
print('Accuracy:', accuracy)
print('Prediction is:',categories[prediction[0]])
rel=xtest[0].reshape(50,50)
plt.imshow(rel,cmap='gray')
plt.show()
```

Fig. 11. Support Vector Machine.

Here, dataset was divided into positive and negative dataset where positive dataset consisted of HOG images of fungus and negative dataset consisted of HOG images of non-Fungus.

IV. RESULTS

A. *Computer Vision Technique.*

1) *Pre-processing*

The following two figures represent the output of the first step i.e., Image smoothening and Image Enhancement.

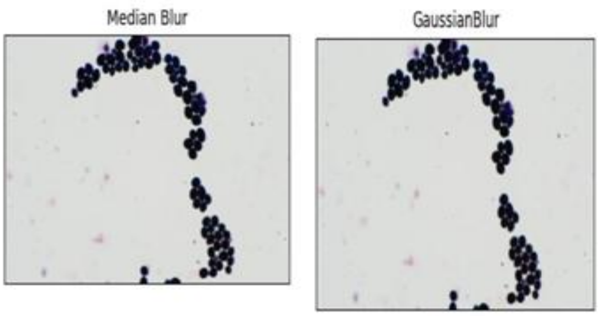

Fig. 12. Output of pre-processing step using Median Blur and Gaussian Blur

2) *Segmentation*

Fig. 13. Output od segmentation step.

Here, threshold segmentation was used.

3) *HOG Feature Extraction*

The HOG Feature image is used to train and test SVM classifier by creating positive and negative Training dataset. Figure 14 is a part of positive training data.

Fig. 14. HOG Feature image.

4) *Support Vector Machine*

Figure 7. shows the output of SVM classifier that we used for the final classification between fungus and non-fungus spores. The trained model successfully predicts a fungus patch. It was observed that accuracy remained above 90% throughout the identification process. However, limited dataset proved to be one of the drawbacks of this model.

B. *Convolution Neural Network*

Our CNN model was saved and trained with many parameters and different layers. The performance of the model was analysed on 2000 images. As shown in Figure 3, after running the model for 15 epochs, we obtained an accuracy of 83%. A CNN model is said to have an accuracy of more than 90% but in our project the comparatively less accuracy is because of the reduced number of datasets. By preparing a much vast dataset, the results can be improved. Some false identification can be observed in some cases due to the lack of adequate images in our dataset. We were able to classify 9 different classes of fungi.

TABLE I. COMPARISON OF THE TWO METHODS USED.

	CVT	CNN
Type of identification	Fungus or Non fungus	Classification of fungi
Segmentation	Thresholding	CNN Layer
Feature extraction	HOG	CNN Layer
Classification	SVM	CNN Layer
Accuracy	More than 90%	83%

```
from keras.preprocessing import image as image_utils
test_image = image_utils.load_img(r"C:\Users\Nija\OneDrive\Desktop\SB10.png", target_size=(299, 299))
test_image = image_utils.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis=0)

result = model.predict_on_batch(test_image)
result

array([[1.9924269e-10, 1.8653415e-08, 1.2575943e-02, 5.0618861e-02,
        4.3675173e-03, 4.6161770e-07, 4.7184363e-09, 8.7211472e-01,
        6.0322516e-02]], dtype=float32)

print(result.argmax())
for category, value in val_generator.class_indices.items():
        if value == result.argmax():
            print(category)

7
Saccharomyces boulardii
```

Fig. 15. Testing the model with input image.

As we achieved an accuracy of 83%, we tested the model with input images. Figure 8 represents one of the outputs of classification. Most of the images were classified correctly. However, some identifications were observed in the case where fungus patterns were quite similar.

Table 1 shows the comparison between CVT and CNN model.

V. FUTURE SCOPE

Fungus has a wide range of scope right from the construction industry to medical industry. There are many fungal diseases which affects a lot of people and a promising solution to it requires correctly identifying the fungus to make vaccines or to take other effective measures. Accuracy of the model with respect to both CVT and CNN can be increased by increasing the dataset images under microscopic conditions. Both these methods are useful right from food industry to many other where CVT can be used to determine whether fungus is present or not while CNN on the other side will give a deep insight into which type of fungus is been detected and as each type has its own characteristics, methods would differ to prevent each one of them.

VI. CONCLUSION

In this paper we have presented our fungus detection and identification model. Fungus detection is a topic of countless possibilities. Fungus poses a grave danger to human beings as well as our environment. It is not every day that we come across automated system to detect and identify fungus. All the research we have done and we might still end up being short of all the information we could collect. Fungus is a problem for the ages but the first step into solving the issue is detecting the fungus and this is what our project’s sole purpose is. We used various methods of computer vision technique and machine learning to correctly detect and identify 9 different classes of fungi. We used Inception V3 model of convolution neural networks to classify fungus into 9 different classes. Computer Vision was used to detect fungus and non-fungus spores using HOG features. Further SVM classifier was used. We achieved the desired results in both techniques of CNN and CVT obtaining an accuracy of 83% and more than 90% respectively.

REFERENCES

[1] Fungus detection (2016)- M. W. Tahir, N. A. Zaidi, R. Blank, P. P. Vinayaka and W. Lang, "Fungus Detection System," 2016 IEEE International Conference on Autonomic Computing (ICAC), Wurzburg, 2016, pp. 227-228, doi: 10.1109/ICAC.2016.50.

[2] R. Blank, P. Papireddy Vinayaka, M. W. Tahir, J. Yong, M. J. Vellekoop and W. Lang, "Comparison of Several Optical Methods for an Automated Fungal Spore Sensor System Concept," in IEEE Sensors Journal, vol. 16, no. 14, pp. 5596-5602, July15, 2016, doi: 10.1109/JSEN.2016.2567538.

[3] Simple and efficient fungus detection system (2017)-M. W. Tahir, N. A. Zaidi, R. Blank, P. P. Vinayaka, M. J. Vellekoop and W. Lang, "An efficient and simple embedded system of fungus detection system," 2017 International Multi-topic Conference (INMIC), Lahore, 2017, pp. 1-4, doi: 10.1109/INMIC.2017.8289477.

[4] Fungus detection using different feature vector (2017)- M. W. Tahir, N. A. Zaidi, R. Blank, P. P. Vinayaka, M. J. Vellekoop and W. Lang, "Fungus Detection Through Optical Sensor System Using Two Different Kinds of Feature Vectors for the Classification," in IEEE Sensors Journal, vol. 17, no. 16, pp. 5341-5349, 15 Aug.15, 2017, doi: 10.1109/JSEN.2017.2723052.

[5] Fungus detection using machine learning techniques (2018)- 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, doi: 10.1109/TNB.2018.2839585.

[6] R. Blank, P. P. Vinayaka, M. W. Tahir, M. J. Vellekoop and W. Lang, "Optical sensor system for the detection of mold: Concept for a fully automated sensor system for the detection of airborne fungal spores," 2015 IEEE SENSORS, Busan, 2015, pp. 1-4, doi: 10.1109/ICSENS.2015.7370346.

[7] A. Ali and H. S. Hassanein, "A Fungus Detection System for Greenhouses Using Wireless Visual Sensor Networks and Machine Learning," 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 2019, pp. 1-6, doi: 10.1109/GCWkshps45667.2019.9024412.