Article

A Novel Deep Learning Model For Detection Of Severity Level of the Disease In Citrus Fruits

Poonam Dhiman[1], Vinay Kukreja[2], Poongodi M[3*], Amandeep Kaur[2], Kamruzzaman M M [4*], Imed Ben Dhaou [5,6,7*] and Celestine Iwendi [8]

[1] Research Scholar, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab. e-mail: poonamdhiman19@gmail.com
[2] Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India, onlyvinaykukreja@gmail.com, amandeep@chitkara.edu.in,
[3] Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar, dr.m.poongodi@gmail.com
[4] Department of Computer Science, College of Computers and Information Science, Jouf University, Sakaka, Al-Jouf, KSA. mmkamruzzaman@ju.edu.sa
[5] Department of Computer Science, Hekma School of Engineering, Computing, and Informatics, Dar Al-Hekma University, Jeddah 22246-4872, Saudi Arabia.
[6] Department of Computing. University of Turku, Finland. imed.bendhaou@utu.fi
[7] Higher Institute of Computer Sciences and Mathematics, Department of Technology, University of Monastir, Monastir 5000, Tunisia
[8] School of Creative Technologies, University of Bolton, United Kingdom. celestine.iwendi@ieee.org
* Correspondence: imed.bendhaou@utu.fi ; dr.m.poongodi@gmail.com and mmkamruzzaman@ju.edu.sa

Abstract: Citrus fruit diseases have an egregious impact on both the quality and quantity of the citrus fruit production and market. Automatic detection of severity is essential for quality productions of fruits. In current work, citrus fruits dataset is preprocessed by rescaling and establishing bounding boxes with labeled image software. Then Selective search, which combines the capabilities of both an extensive search and graph based segmentation, is applied. The proposed DNN (deep neural network) model is trained to detect targeted area of the disease with its severity level using citrus fruits that have been labeled by taking help of a domain expert with four severity level(high, medium ,low and healthy) as ground truth. Transfer learning using VGGNet is applied to implement multi-classification framework for each class of severity. The model predicts the low severity level with 99% accuracy, and the high severity level with 98% accuracy. Model produces 96% accuracy in detecting healthy conditions and 97% accuracy in detecting medium severity levels. The result of the work shows that the proposed approach is valid, and it is efficient for detecting citrus fruit disease at four levels of severity.

Citation: P. Dhiman;V. Kukreja; M Poongodi;A. Kaur ; MM. K ; Dhaou I M; Iwendi C . *Journal Not Specified* **2022**, *0*, 0. https://dx.doi.org/

Received:
Accepted:
Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

According to the FAO (FAOSTAT 2019) [1], world citrus fruit production is estimated to be at 157.98 million of tons, with oranges accounting for more than half of total. Producers seek to produce superior fruits at a cheaper cost that are free of any disease insects and pathogens, this task can be accomplished through the use of appropriate mechanized standards and predictive maintenance techniques [2]. Fruit diseases create a substantial danger to modern farming production of the citrus. The citrus sector needs early automatic identification of diseases in post-harvesting since a few contaminated fruits might disseminate the disease to the entire sequence during processing or shipment. The severity of the disease is a crucial parameter for determining the extent of the disease that affects yield production. The ability to diagnose disease severity quickly and accurately would help to prevent production deficits usually; disease severity has been determined by trained professionals by visually inspecting plant tissues. The high cost and limited efficiency of human disease assessment stymies modernized agriculture's rapid progress [3]. This paper presents deep learning models for image-based automatic diagnosis of citrus fruit disease severity levels. We address the issues of determining the severity of disease in citrus fruits in a multi-classification framework using the deep learning model in this paper. The section 1 describes about the introduction and contribution of the paper. The rest of the paper is organized as follows: Section 2 provides the literature review. Section 3 presents the proposed algorithm of the disease and severity detection of the citrus fruits. Section 4 comprised of detailed description of the material and methodology used for severity detection of the disease. Further, Result evaluation has been presented in section 5. Finally, the paper is concluded in section 6.

1.1. Contribution of the Paper:

The objective of this paper is to develop a deep learning model that classifies the disease according to their severity level and to identify the diseased affected area of the citrus fruits. The proposed model has the ability to recognize and classify the infected areas of citrus fruits. It is a powerful approach for automatically identifying the citrus fruit disease severity and further extending to reinforce a unified citrus disease identification system for real word applications. The current study helps to mitigate and prevent the fruits disease at initial stages and can able to control the cost of pestilent during safeguarding the surroundings globally.

2. Literature Review:

Effective surveillance and diagnosis of resistant cultivars is critical for disease control and prevention for healthy yields. Using watershed segmentation, a novel machine vision system for automatic identification of diseases is proposed. Two kinds of diseases i.e Yellow rust and Septoria are accurately detected using the proposed approach [4]. The severity of the leaf rust disease can result in reduction of sugar production. As a result, illness signs must be discovered as soon as possible, and appropriate actions should be implemented to prevent the disease from spreading or progressing. A faster Region-based Convolutional Neural Network framework is constructed by altering the parameters of the model and a faster R-CNN framework is developed for the detection of leaf spot infestation in sugar. The technique provided for severity detection of disease with image-based systems is trained on 155 images, and classification accuracy of 95.48 % is obtained [5]. The citrus industry is still working on developing technologies for automatically identifying deterioration in citrus fruit all through quality control. Using three distinct manifold learning approaches, the viability of reflectance spectroscopy in the visible and near infrared regions are tested for early identification of root cause of rot by the penicillium digitatum in citrus fruit [6]. Controlling the spread of disease requires its diagnose and then destroying the cause of citrus huanglongbing (HLB)-infected trees. Ground investigation is arduous and time-taking task. It is rare to find a large-area analysis tool for citrus orchards with excellent efficiency. The possibility of large-area monitoring of citrus HLB using low-altitude remote sensing is explored [7]. Nowadays, Citrus fruits exporting to international markets are significantly hampered by fruit disorders like citrus canker, black spot, and scab. As a result, thorough procedures must be performed prior to transportation of fruits to mitigate the presence of citrus damaged by them. A model based on a feature selection method with a classifier trained on quarantine disease for disease detection is being deployed [8]. Among the most significant components used for enhancing agricultural product, scalability and waste reduction is considered to be criterion for evaluating quality. An optimized Convolutional Neural Network system is developed to identify visible flaws in sour lemon, evaluate them, and give a better way for doing it. To detect and characterize abnormalities, lemon images were taken and divided into two category i.e. healthy and impaired. Following pre-processing, the images are classified by using an improved CNN model. To improve the outcomes, stochastic pooling mechanism with augmentation techniques is implemented [9]. A machine vision system to detect irregularities in citrus peel and evaluating the nature of defect is designed. The image is segmented into defective zones using the Sobel gradient. Following that color and texture features are retrieved, some of which are associated with high order statistics [10]. Disease detection is now done manually by domain experts using harmful ultraviolet rays on fruits. The utilization of hyperspectral imaging technologies allows for the advancement of systems for automatic detection of disease. A methodology is proposed to develop a multi-classification system using receiver operating characteristics curve to detect fungal infections in citrus fruits. The developed system helps in reducing the set of features and achieved the accuracy rate of 89%. [11].

3. Materials and Methods

The proposed model for detecting affected area and the severity levels of the citrus fruits disease comprised of five modules as shown in figure 1. The first module targets the collection of citrus fruit images.The second module is used to label the healthy and infected images by using expert knowledge.For labeling the images, an open source tool is used [12]. Labeling is the process of providing annotation to the graphical images and label the bounding box for object detection. Annotations of the images are stored as XML files in Pascal VOC form and the process of annotating the image is further explained in section 3.2. The third module is the combination of graph based segmentation and object detection process to produce region of proposal that is independent of the class. The most similar regions are grouped together and similarity will be calculated between the regions which is further explained in section 3.4. A CNN network using transfer learning extracts a fixed-length feature map for each region in the fourth module. The last module represents the implementation of multi-class sequential CNN models that determine the severity level of citrus fruit diseases using softmax function explained in section 3.6.

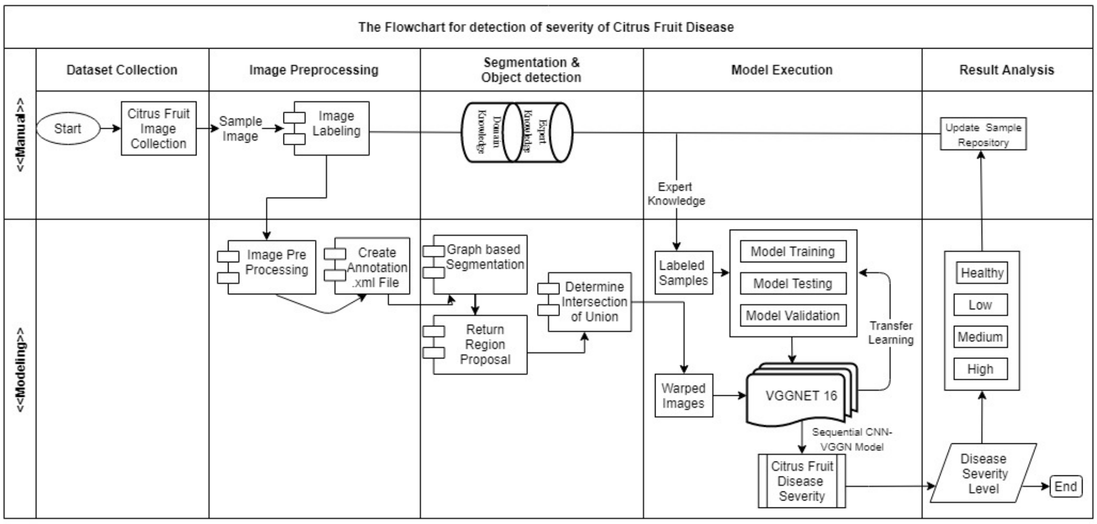

Figure 1. The overall process of severity detection of the citrus fruits severity levels.

3.1. Dataset

Fruit diseases badly affect the product quality, market segment and revenue. Citrus is an important source of vitamins A and C. Citrus illnesses, on the other hand, have a negative impact on citrus fruit output and quality [11]. Citrus plants like lemons, oranges, grapefruit, and limes are susceptible to a variety of citrus diseases such as anthracnose, HLB, scab, black spot, and other fungal infections [13]. Adequate datasets are necessary for object detection and the classification process using deep learning. All the images collected for the dataset were downloaded from the online like dataset collected from the sources i.e. plant village, and kaggle [14,15]. After taking the images from publically available source,the images are prepared for getting the severity of the disease on the infected images with the help of domain expert.

3.2. Annotation

Before training a model, image annotation is an essential image pre-processing step During the training phase, a model can learn the labeled features. As a result, the quality of the training model is strongly influenced by the precision of the feature labeling. As several types of disease appear to be relatively similar, knowledge of the different types of fruit diseases, which could aid the machine in learning traits important to different fruit diseases. Scientist of horticulture helped in data annotation. The expert considered the diameter, color features, shape and the surface area of the affected portion of the disease present in the image for deciding extent of damage in the fruit. The labeling only includes the exterior feature of the image while interior damage was not considered. The outcome of the annotated image is coordinates and bounding boxes and the practice of image annotation requires the labeling of disease locations in the image. Labeling is a free graphical image annotation tool that locates and categorizes the disease severity in an image and stores it as an.xml file with the matching xmin, xmax, ymin, and ymax data for each bounding box [16], [17]. There is an.xml file in the Annotation folder for single jpg file in the JPEG Images folder. Each object's bounding box is saved in an xml file. It's a little difficult to work with annotation data for each image in a separate file. So using Panda modules combine each of these xml files into an one csv file. Annotations are first made in a panda data frame called "df anno," which is then saved as a csv file. Then after, csv file is segregated, which contains annotated data of citrus fruits, into four disease severity categories: healthy, medium, high and low, and build an object for each class of severity. Then iterate each row of an object to extract the image name and url from the object file and read it. Then, on each category's object, the accuracy of object detection is measured. Table 1 represents the total number of citrus samples taken for training and testing.

Table 1. Citrus samples count in training and testing.

Classes	Sample count for Training	Sample count for Testing
Healthy	1173	293
Low Severity	737	184
Middle Severity	774	194
High Severity	625	156

3.3. Proposed Algorithm for Detecting Severity Regions of the Citrus Diseases

Input the colored image(Img)

1) Perform BoundingBox(Img) and annotate image i.e. Annotate(Img) where BoundingBox(Img) is used to create boundary coordinates on affected areas of the image and Annotate(Img) function is used to create and extract the annotated image as .xml file for each image.
2) Create object for each category(i.e. healthy, low, medium and high)
3) Repeat step 5 for each object
4) Repeat step 6 for each row of single object
5) Extract Img_name and Img_url from object and perform preprocessing
6) Extract region using Graph based segmentation for finding out the region proposal
7) Repeat step 9-11for each extracted segment region
8) Compute texture gradient of the image (using LBP)
9) Extract HSV for entire image using color histogram having COLOUR_CHANNELS (3)* bins with a total of 25 bins
10) Augment regions with histogram parameters and return region proposal
11) Repeat step 13, 14 for neighboring pair of region $(r_\alpha,\ r_\beta)$
12) (Compute similarity $\text{Sim}_{(r_\alpha,r_\beta)}$ = colour similarity $\text{Sim}_{\text{color }(r_\alpha,r_\beta)}$ + texture similarity $\text{Sim}_{\text{ texture }(r_\alpha,r_\beta)} +$ size similarity $\text{Sim}_{\text{ size}}\,(r_\alpha,r_\beta) +$ fill similarity $\text{Sim}_{fi\ ll}(r_\alpha,r_\beta))$
13) Merge Regions, in order $(Sim_{(r_\alpha,\ r_\beta)}$, R)

14) Calculate IOU for Regions

Precision of object detection highly affects disease and severity recognition accuracy so a robust automatic detection system is proposed using image processing techniques. This algorithm performed the pre-processing and objects identification task for different disease location and severity of the disease present in citrus fruits. Graph based segmentation is implemented to get the region of proposal of each image. The above steps of the algorithm are implemented to get the region of proposal and object detection is given.

3.4. Steps of Selective Search to get the Region Proposal:

Initial regions are generated using Felzenszwalb's graph-based segmentation approach. After implementation; results are represented in figure 2

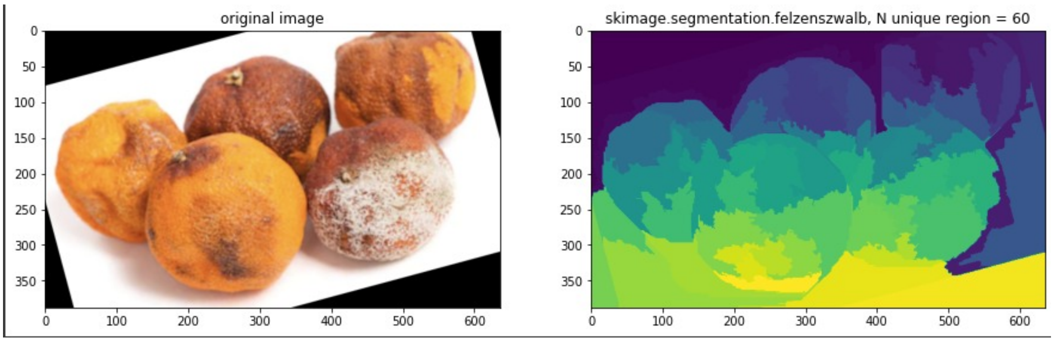

Figure 2. original image and segmented image sample of the citrus fruit

Next step is to add labels to segmented regions of image [18]. Visualization of labels output after Felzenszwalb segmentation is shown in figure 3.

After segmentation a lot of useless labels or labels are generated belonging to one object. Next step is to group labels that belong to one object based on the most similar regions. For this grouping LBP (Local Binary Pattern) will be implemented [19]. To capture the texture similarities of the initial regions, for each initial region, LBP features is calculated. Calculated texture gradient for entire image is computed and the results shown in figure 4.

Now, collect the RGB values on a scale of 0 to 1, and also the biggest and lowest RGB values, as well as the point of difference by following the equations from 1 to 7..

$$R = \frac{r}{255}\ ,\ G = \frac{g}{255}\ ,\ B = \frac{b}{255}. \tag{1}$$

$$\mathrm{V_{max}} = \mathrm{MAX(R, G, B)}. \tag{2}$$

$$\mathrm{V_{min}} = \mathrm{MIN(R, G, B)}. \tag{3}$$

$$\delta = Vmax - Vmin. \tag{4}$$

$$\mathrm{H_{hue}} = \begin{cases} 60^\circ * \left(\dfrac{G-B}{\delta} \bmod 6\right), \mathrm{Vmax = R} \\ 60^\circ * \left(\dfrac{B-R}{\delta} + 2\right), \mathrm{Vmax = G} \\ 60^\circ * \left(\dfrac{R-G}{\delta} + 4\right), \mathrm{Vmax = B} \end{cases} \tag{5}$$

$$S_{\mathrm{saturation}} = \begin{cases} 0, & \delta = 0 \\ \dfrac{\delta}{V\max}, & \delta \neq 0 \end{cases} \tag{6}$$

$$\mathrm{V} = V_{\max}$$

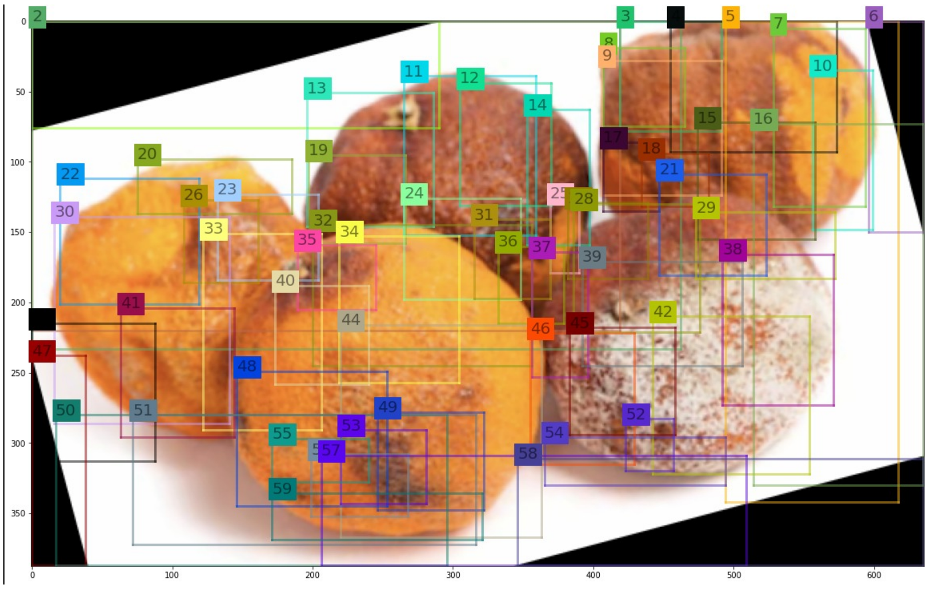

Figure 3a. Labels on Original image and on Felzenszwalb Segmentated image

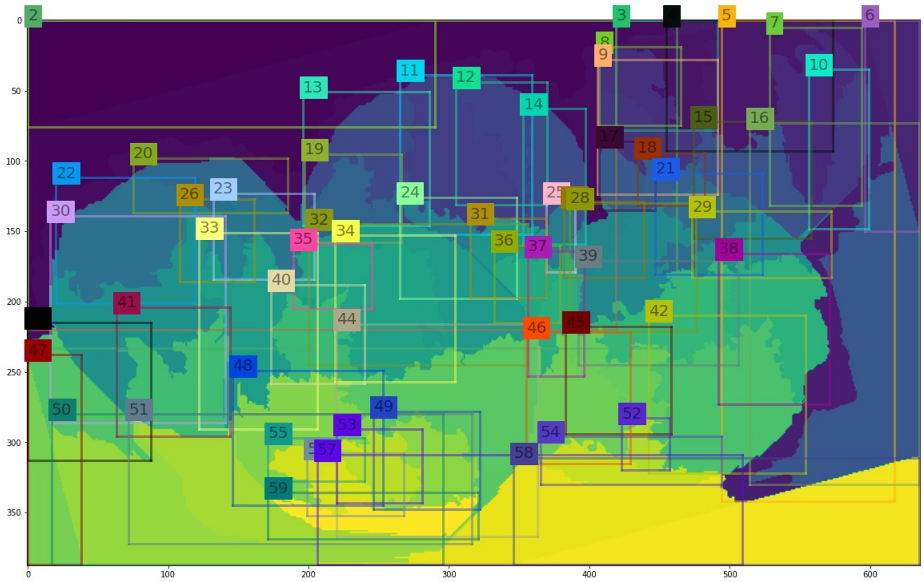

Figure 3b. Felzenszwalb Segmented image

The Hue Saturation Value (HSV) format symbolizes how paints of multiple colors blend altogether, also with saturation component representing different intensities of vibrantly colored paints and the value component representing the combination of each of these paints with different ratios of black or white paints [20]. Figure 5 represent the HSV image with calculated min max values.

Sum of histogram intersection of color $Sim_{color(r_\alpha, r_\beta)}$ is calculated to measures color similarity. One-dimensional color histograms are derived for individual color channel for each region using 25 bins, which is found to be effective. Three rgb colour channels results into a color histogram with dimensions d = 75 for each region. The $L1$ norm is used to normalize the color histograms. The histogram intersection is used to determine similarity using equation 8.

$$Sim_{color(r_\alpha, r_\beta)} = \sum_{l=1}^{d=75} min(c_{hist}{}_{\alpha}^{l}, c_{hist}{}_{\beta}^{l}). \tag{7}$$

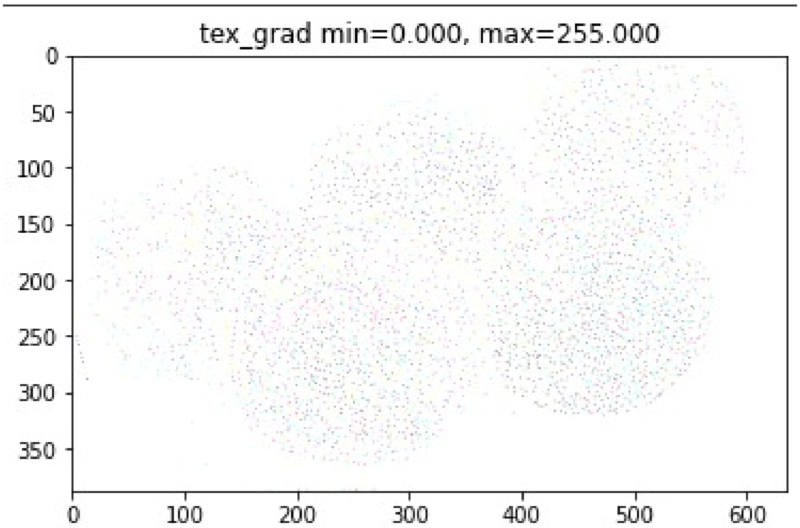

Figure 4. Texture gradient for LBP feature

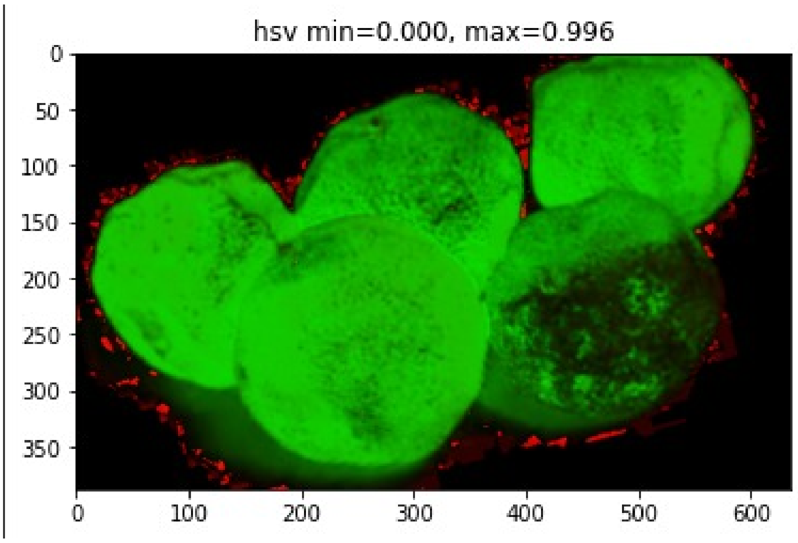

Figure 5. HSV image with min-max value

The color histograms can be efficiently propagated through the hierarchy by using the following equation 9. 188 189

$$c_{hist} = \frac{size(r_\alpha) * c_\alpha + size(r_\beta) * c_\beta}{size(r_\alpha) + size(r_\beta)}. \tag{8}$$

Sum of histogram intersection of texture $Sim_{texture(r_\alpha, r_\beta)}$ is calculated to measures texture similarity. $L1$ norm is to normalize the Texture histograms. In equation 9 histogram intersection is used to determine similarity: 190 191 192

$$Sim_{texture(r_\alpha, r_\beta)} = \sum_{l=1}^{d} min({t_{hist}}_\alpha^l, {t_{hist}}_\beta^l). \tag{9}$$

Now, Calculate the image's size similarity $Sim_{size(r_\alpha, r_\beta)}$, which promotes rapid fusion of tiny regions. This constrains the size of regions in S, i.e. regions that have not yet been merged, throughout the procedure. That's also advantageous since this enables the generation of object locations at all scales throughout the image. For instance, it inhibits an individual region from devouring most other regions one after the other, giving all scales exclusively at the location of this developing region. $Sim_{size(r_\alpha, r_\beta)}$ is defined as the 193 194 195 196 197 198 percentage of the image that r_α and r_β collectively inhabit, whereas $size(img)$ specifies the image's pixel size in equation 11:

$$Sim_{size(r_\alpha, r_\beta)} = \frac{size(r_\alpha) + size(r_\beta)}{size(img)}. \tag{10}$$

Following that, compute the fill similarity throughout the image $Sim_{fill(r_\alpha, r_\beta)}$determines how effectively the region r_α and $r_\beta$fit together. The goal is to fill up the gaps: if r_α is included in r_β,it stands to reason for merging them first to prevent any gaps. If r_α and $r_\beta$are barely contacting one another they would most certainly form an odd region and should not be combined. Only the sizes of the regions and the enclosed boxes are incorporated in order to keep the quick evaluation. Particularly, defined $BBox_{\alpha\beta}$ as the compact bounding box encompassing r_α andr_β . $Sim_{fill(r_\alpha, r_\beta)}$ Now represents the proportion of the image in $BBox_{\alpha\beta}$ that is not covered by the regions of r_α andr_β in equation 12.

$$Sim_{fill(r_\alpha, r_\beta)} = \frac{size(BBox_{\alpha\beta}) - size(r_\alpha) - size(r_\beta)}{size(img)}. \tag{11}$$

Then Retrieve a list of regions that intersect. Calculate the similarities between each pair of neighboring regions and then produce the sum of the regions' similarities using equation 13. Obtain the total of two regions' similarity, which is a composite of the four types of similarity mentioned previously.

$$\begin{aligned} Sim_{(r_\alpha, r_\beta)} = Sim_{color(r_\alpha, r_\beta)} + \ Sim_{texture(r_\alpha, r_\beta)} + \\ Sim_{size(r_\alpha, r_\beta)} + Sim_{fill(r_\alpha, r_\beta)}. \end{aligned} \tag{12}$$

Calculate similarity of all regions using equation 9

$$Sim_{overall} = \textstyle\sum_{ij=n}^{N} Sim_{(r_\alpha, r_\beta)}. \tag{13}$$

Nextly, merge regions and then delete already merged regions and calculate new similarity. The following steps to be followed in order to merge the regions.

Merge regions in order s (ri, r j, R)

1) Retrieve the pair of regions with the highest degree of similarity from the similarity dictionary.

2) Merge the region pairs and add them to the dictionary of regions.

3) Eliminate all pairs of regions from the similarity dictionary in which one of the regions is defined in Step 1.

4) determine the degree of similarity between the newly combined region and the regions and their intersecting regions (intersecting region is the region that are to be deleted)

return (regions)

3.5. Intersection of Union on overlapped region

To train a classifier using CNN features as input, we require ground truth labels for each candidate region. However, there is a quandary over how to identify a region that partially overlaps when a portion of the fruit is included. To address this issue, overlap threshold value will be used below which regions will be regarded as negatives. Intersect over Union (IoU) is a frequently used metric for determining the similarity of the projected bounding box to the ground truth bounding box using equation 15-17. The aim is to examine the area of overlap between two boxes to the cumulative area of the two boxes [21] [22]. Figure 6 shows the region of intersection over union.

$$(\alpha_1, \beta_1) = (\max(\mathrm{a}_1), \max(\mathrm{x}_1)). \tag{14}$$

Figure 6. Intersection of Union on overlapped region

$$(\alpha_2, \beta 2) = (\min(b_2), \min(x_2)). \tag{15}$$

$$\begin{aligned}&\text{Overlapping region} = \text{width} * \text{height}\\ &\text{Else}\\ &\text{Overlapping region} = 0\end{aligned}$$

$$\begin{aligned}&\text{Combined region=Area of (Box1)} + \text{ Area of (Box2)}-\\ &\text{Overlapping Region .}\end{aligned} \tag{16}$$

Training features is created and ground truth into a 4 pickled objects that contains candidate regions that contain having IOU > 0.75.Same object can have large number of small candidate regions that hardly provide new information so for each object only candidate region will be chosen. Other pickled object corresponding to the particular object captured in first object. Rest two picked object contains all the candidate regions that does not contain a citrus fruit object i.e., IOU < 0.4 and information regarding the particular object that was not captured in first object.

3.6. Warp the regions proposed by the selective search:

To calculate features for a region proposal, transformation of image samples in the region to a form which is compatible to the CNN is required [23]. All pixels in a tight bounding box around the candidate region is wraped to the desired size irrespective of its size or aspect ratio. We elongate the tight bounding box beforehand to warping so that there are exactly p pixels of warped image across the original box (we use p = 16). VGG16 specifies that the image must have the dimensions (height, width, Nchannel) = (224, 224, 3). The region proposal given by the selective search often does not correspond to the image with the dimensions 224 in height and width. So all pixel in the region proposal need to warped to the CNN's input size.

3.6.1. Feature extraction

Using the VGGNet16, 4096 features map is extracted from each region proposal. VGGNet is the current state of the art, with advanced and efficient identification capabilities, and it is frequently used for transfer learning due to its portability. Only 3x3 convolutions are used by VGGNet. VGGNet, on the other hand, contains many extra filters [24]. It has 16 layers, each with its own set of trainable weights. It is now the most popular method for obtaining features from images. The VGGNet's weight composition is open to the public.VGGNet is just used for feature extraction not for the classification purpose. For classification last three layers were removed from the network. Forward propagation of a mean-subtracted RGB 227×227 image through 5 convolution layers and 2 fully connected dense layers is used to compute features.

3.6.2. Transfer learning

Transfer learning is a powerful approach of machine learning that makes CNNs to learn for one goal are repurposed as the foundation for a model on a different task. In-spite of initiating the training from scratch by arbitrarily instantiating the weights, a pre-trained network can be used to initialize the weights on large labeled datasets like as public datasets [25]. The ImageNet project is a massive visual database developed for use in the development of visual object recognition [26]. In this article, leveraging pretrained model is investigated pretrained from the enormous large dataset ImageNet, and then them them to a get the severity trained on the citrus fruit dataset. The following are the key processes of the transfer learning technique. The proposed model using transfer learning is shown in figure 7.

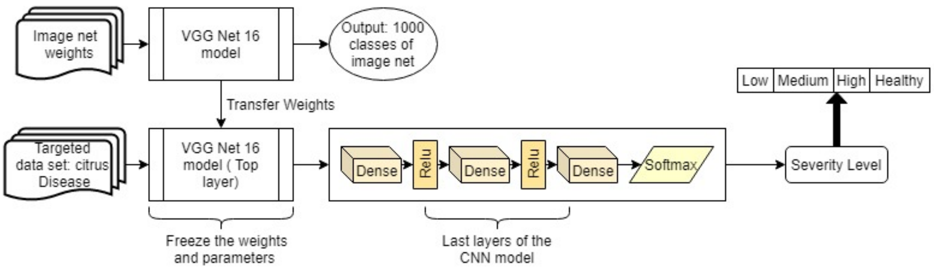

Figure 7. Proposed CNN model with Transfer Learning

The first step is to the find out the base networks of the transfer learning and assigns the network's weights by using the pre-trained CNN model. These weights are available for download from an online source. Then reconstruct the network structure by manipulating bottom layers of the network. A new modified network structure can be obtained using this approach. The newly constructed networks can then be fine-tuned in order to minimize the loss function using dataset and associated labels. Specifically, Adaptive Moment Estimation (Adam) algorithm is used to determine the optimized weights with controlling of the loss function using sparse categorical cross entropy as a loss function. So, for transfer learning, a VGGNet pre-trained model was used on ImageNet, and a sequential CNN model was used to train the newly updated neural networks using citrus fruit datasets. The method offers the features of VGGNet with sequential CNN. From the initial layers i.e. $block1\_cov1$ to FC1(Dence) are from the VGGNet.Dense, $Dense\_1$, $Dense\_2$ is substituted with the sequential CNN model.Latly, softmax classifier is used for multi-classification of the severity classes of the citrus disease.Thus , new model bring about is generally consist of two sections in which first section is the pre-trained model and other section is the perpetuated layers employed on multi-scale feature vector for multi-classification and Table 2 listed the parameters of the implemented Deep learning model.

4. Result Analysis

The training accuracy is the percentage of the correctly defined data samples in the training set. Similarly, the validation accuracy refers to the percentage of the correctly elucidated data samples from some of the other samples. Dataset is divided into two sets, one set comprising of images for training and other is for validation. The 80-20 cross-validation process is used to train and validate the model. For validation, multiple investigations are carried out with shuffled images [27]. New randomly selected images are used to test the efficiency of the model. Sparse categorical cross entropy for loss function was used to determine the classification model's performance.The overall training accuracy achieved by the model is 95%. Adam optimizer is selected for model to optimize the cross-entropy function [28]. The result of the implemented convolution neural network model on randomly selected test images was analyzed and represented as the Confusion matrix in Table 3. Figure 8 depicts the classification accuracy and loss gained after the training and validation process of the model.

Table 2. Shows the related parameters of the implemented model

Layer	Lyer Type	Kernel Size	Stride	Neuron Size	Maps	Param #
Block1_conv1	Convolutional layer	3×3	1	224×224	3	1792
Block1_conv2	Convolutional layer	3×3	1	224×224	64	36928
Block1_pool	Pooling layer P1	2×2	2	112×112	64	0
Block2_conv1	Convolutional layer	3×3	1	112×112	64	73856
Block2_conv2	Convolutional layer C4	3×3	1	112×112	128	147584
Block2_pool	Pooling layer P2	2×2	2	56×56	128	0
Block3_conv1	Convolutional layer	3×3	1	56×56	128	295168
Block3_conv2	Convolutional layer	3×3	1	56×56	256	590080
Block3_conv3	Convolutional layer	3×3	1	56×56	256	590080
Block3_pool	Pooling layer P3	2×2	2	28×28	256	0
Block4_conv1	Convolutional layer	3×3	1	28×28	256	1180160
Block4_conv2	Convolutional layer	3×3	1	28×28	512	23598038
Block4_conv3	Convolutional layer	3×3	1	28×28	512	23598038
Block4_pool	Pooling layer P4	2×2	2	14×14	512	0
Block5_conv1	Convolutional layer	3×3	1	14×14	512	23598038
Block5_conv2	Convolutional layer	3×3	1	14×14	512	23598038
Block5_conv3	Convolutional layer	3×3	1	14×14	512	23598038
Block5_pool	Pooling layer P5	2×2	2	7×7	512	0
Flatten	Flatten	—–	—–	——	25088	0
Fc1 (Dense)	———	—–	—–	——	4096	102764544
Dense (Dense)	Sequential CNN	—–	—–	——	32	131104
Dense_1 (Dense)	Sequential CNN	——	—–	——-	32	1056
Dense_2 (Dense)	Sequential CNN	——	—–	——	4	132
Output	Softmax	——-	——	Classifier	4	—–

Table 3. Confusion matrices for all level of severity of disease present in citrus fruits.

Class	Healthy	Low	Medium	High
Healthy	21	0	0	0
Low	0	25	0	1
Medium	3	0	25	0
High	1	0	0	24

Out of the four level diseases severity of the citrus fruits, the model can able to predict the low severity level with accuracy 0f 99%, precision of 100%, recall 84%, and F1Score 91%. For high severity level of the disease, our model recorded the accuracy of 98% when

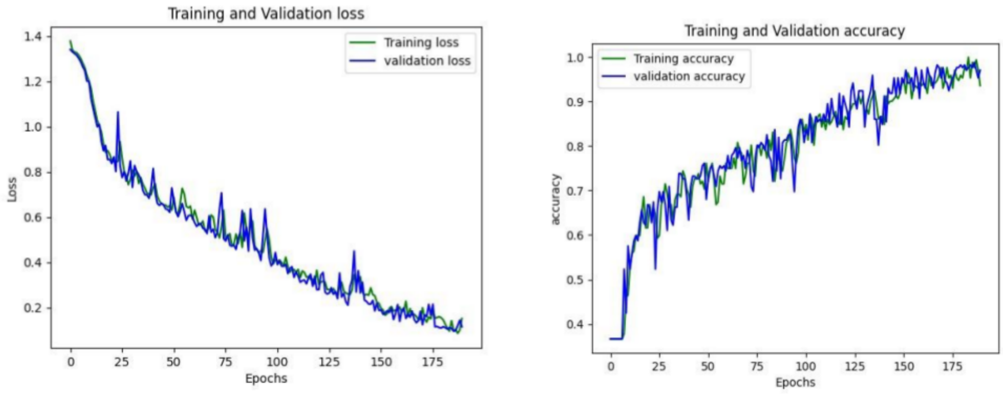

Figure 8. Loss and Accuracy curves of the implemented model.

compared to other classes. For the detection of healthy condition model produces the 96% of accuracy and 97% of accuracy in case of medium severity level. Accuracy, precision, recall and F1Score calculated for each severity level of the infected citrus fruit of the disease are encapsulated in Table 4.

Table 4. Accuracy, Precision, Recall and F1Score of the Model

Class	Accuracy	Precision	Recall	F1Score
Healthy	96%	100%	84%	91%
Low	99%	96%	100%	98%
Medium	97%	89%	100%	94%
High	98%	96%	96%	96%

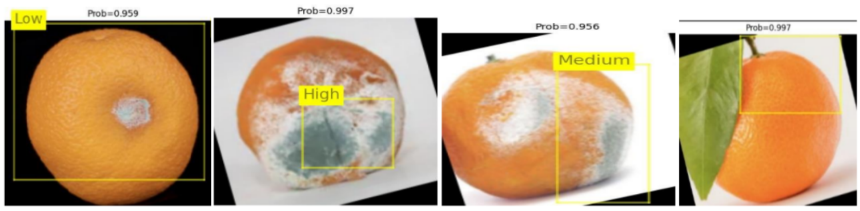

Figure 9. Result showing four level of severity in image samples

Figure 9 depicts some of the graphical outcomes of automatic disease recognition system proposed. The results demonstrate that the assessing accuracy of the disease severity level of citrus fruits as low severity (95.9%), high severity (99.7%), medium severity (95.6%) and healthy (99.7%). As demonstrated in figure 9, our system can efficiently diagnose the image dataset with four severity levels of disease, and has been compared to expert manual evaluation. The results reveal that disease severity identification is quite accurate and falls within the domain experts' acceptable range.

5. Conclusion

Fruit diseases are the most serious threats to global agricultural progress, and they do have a terrible influence on food safety. As a result, automatic diagnosis of citrus fruit diseases is increasingly desirable in analytic. Deep learning approaches, specifically CNNs, have demonstrated encouraging ability to resolve the majority of the difficult classification problems. Transfer learning for deep CNNs is investigated in this research with the goal of improving the learning ability of obtaining the severity level, and Sequential-VGGNet16 architecture is developed for the diagnosis of four severity level of the disease present in citrus fruits. The pre-trained VGGNet-16 is updated by substituting its last bottom layers with an extended convolutional layer that includes dense layer with ReLu activation and Sparse categorical cross entropy for loss function used to determine the classification model's performance. Adam optimizer is selected for model to optimize the cross-entropy function. Lastly, the fully-connected Softmax layer was inserted as the classification layer in order to get four severity level of the disease.Test accuracy achieved on randomly selected images for healthy,low level, high level and medium level is 96%,99%,98% and 97%.

6. Acknowledgement

The authors would like to thank Deanship of Scientific Research at Jouf University for partly supporting this Project (number DSR 2021 02 0339)

References

1. Food and agriculture organisation of the united states. http://www.fao.org/faostat/en/#home (accessed Jul. 15, 2021).
2. Cubero, S.; Lee, W.S.; Aleixos, N.; Albert, F.; Blasco, J. Automated Systems Based on Machine Vision for Inspecting Citrus Fruits from the Field to Postharvest-a Review. *Food Bioprocess Technol* **2016**, *9*, 1623–1639.
3. Taylor, P. Critical Reviews in Plant Sciences Plant Disease Severity Estimated Visually , by Digital Photography and Image Analysis , and by Hyperspectral Imaging Plant Disease Severity Estimated Visually , by Digital Photography and Image Analysis , and by Hyperspe. *CRC. Crit. Rev. Plant Sci* **2010**, *29*, 37–41.
4. Han, L.; Haleem, M.S.; Taylor, M. A Novel Computer Vision-based Approach to Automatic Detection and Severity Assessment of Crop Diseases, 2015.
5. Metin, M.; Adem, K. Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms. *Physica A* **2019**, *535*, 122537–122537.
6. Lorente, D.; Escandell-Montero, P.; Cubero, S.; Gómez-Sanchis, J.; Blasco, J. Visible-NIR reflectance spectroscopy and manifold learning methods applied to the detection of fungal infections on citrus fruit. *J. Food Eng* **2015**, *163*, 17–24.
7. Lan, Y. Comparison of machine learning methods for citrus greening detection on UAV multispectral images. *Comput. Electron. Agric* **2020**, *171*, 105234–105234.
8. Stegmayer, G.; Milone, D.H.; Garran, S.; Burdyn, L. Automatic recognition of quarantine citrus diseases. *Expert Syst. Appl* **2013**, *40*, 3512–3517.
9. Jahanbakhshi, A.; Momeny, M.; Mahmoudi, M.; Zhang, Y.D. Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks. *" Sci. Hortic. (Amsterdam)* **2020**, *263*, 109133–109133.
10. Lopez, J.J.; Aguilera, E.; Cobos, M. Defect detection and classification in citrus using computer vision. *proceeding International Conference on Neural Information Processing* **2009**, pp. 11–18.
11. Lorente, D.; Aleixos, N.; Gómez-Sanchis, J.; Cubero, S.; Blasco, J. Selection of Optimal Wavelength Features for Decay Detection in Citrus Fruit Using the ROC Curve and Neural Networks. *Food Bioprocess Technol* **2011**, *6*, 530–541.
12. https://github.com/tzutalin/labelImg/releases/tag/v1.8.1 .
13. Behera, S.K.; Jena, L.; Rath, A.K.; Sethy, P.K. Disease Classification and Grading of Orange Using Machine Learning and Fuzzy Logic. *Proceedings of IEEE International Conference on Communication and Signal Processing* **2018**, pp. 678–682.
14. https://data.mendeley.com/datasets/3f83gxmv57/2 .
15. https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification .
16. Yang, H.; Zhou, J.T.; Zhang, Y.; Gao, B.; Wu, J.; Cai, J. Exploit bounding box annotations for multi-label object recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2016**, pp. 280–288.
17. Garfinkel, S.L. Automating disk forensic processing with SleuthKit, XML and python. *4th International Workshop on Systematic Approaches to Digital Forensic Engineering* **2009**, pp. 73–84.
18. Uijlings, A.W.; Sande, J.R.V.D.; Gevers, K.E.; Smeulders, T. Selective Search for Object Recognition. *Int. J. Comput. Vis* **2012**, *104*, 154–171.

19. Timo Ojala, and Matti Pietikainen and Topi Maenpaa. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* **2002**, *24*, 971–987.
20. Saravanan, G.; Yamuna, G.; Nandhini, S. Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models. *International Conference on Communication and Signal Processing* **2016**, pp. 462–466.
21. Chen, J.W.; Lin, W.J.; Cheng, H.J.; Hung, C.L.; Lin, C.Y.; Chen, S.P. A smartphone-based application for scale pest detection using multiple-object detection methods. *Electron* **2021**, *10*, 1–14.
22. Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; Jiang, Y. Acquisition of localization confidence for accurate object detection. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **2018**, pp. 816–832.
23. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J.; Berkeley, U.C.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2014**, *1*, 5000–5000.
24. Wang, Q.; Qi, F.; Sun, M.; Qu, J.; Xue, J. Identification of Tomato Disease Types and Detection of Infected Areas Based on Deep Convolutional Neural Networks and Object Detection Techniques. *Comput. Intell. Neurosci* **2019**, *2019*.
25. Wimmer, G.; Vécsei, A.; Uhl, A. CNN transfer learning for the automated diagnosis of celiac disease, 2017. doi: 10.1109/IPTA.2016.7821020.
26. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric* **2019**, *173*, 105393–105393.
27. Chena, J.; Chena, J.; Defu.; Zhanga.; Sunb, Y.; Nanehkarana, Y. Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture* **2020**, *173*, 1–11.
28. Bock, S.; Weis, M. A Proof of Local Convergence for the Adam Optimizer. *Proc. Int. Jt. Conf. Neural Networks* **2019**.