NATIONAL INSTITUTE OF TECHNOLOGY, HAMIRPUR

Image Processing Lab

Topic: ***Pest Detection from Tomato Leaves***

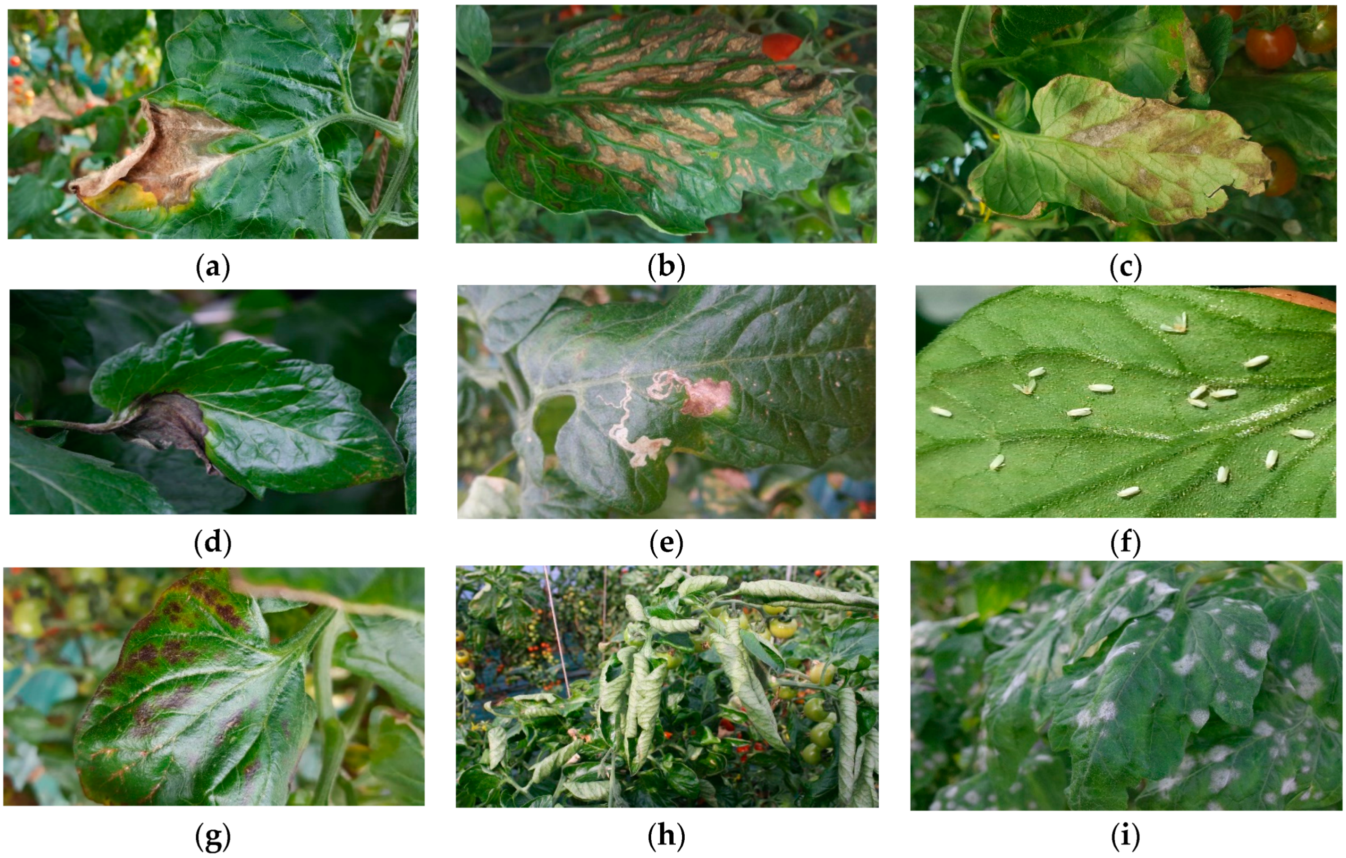
Submitted to: **Dr. Nagendra Pratap Singh**

by:

- Ritwik Duggal , 185528
- Himanshi , 185539
- Abhishek Kumar , 185540
- Akshay Kumar , 185541

COURSE CODE: CSD_328

Department of Computer Science and Engineering



Pest on Tomato Leaves

Aim:

To Classify Pest on Tomato Leaves using Second-Order Statistical Feature Extraction method on Gray Level Co-occurrence Matrix on PCA Compressed Images

Abstract:

A Group of various Pests on Tomatoes Leaves images are converted to gray-level and then the acquired 2D matrix is compressed to 100 px *100 px size and further to 1 D Array. After this, PCA analysis is applied to the 1D Array to extract Pixel columns with top 50 eigenvalues which affecting the overall result. The acquired Pixels column is further converted to the 2D matrix[10,5]. In the end, part of image segmentation Feature Extraction like Contrast, Correlation, Energy, Homogeneity is applied using GLCM matrix. In the image classification, part ML model and Convolution Neural Network is applied to get the High Accuracy.

Dataset:

Tomato Leaves Pest DATASET - This DATASET is further classified into 8 groups of Pest species.

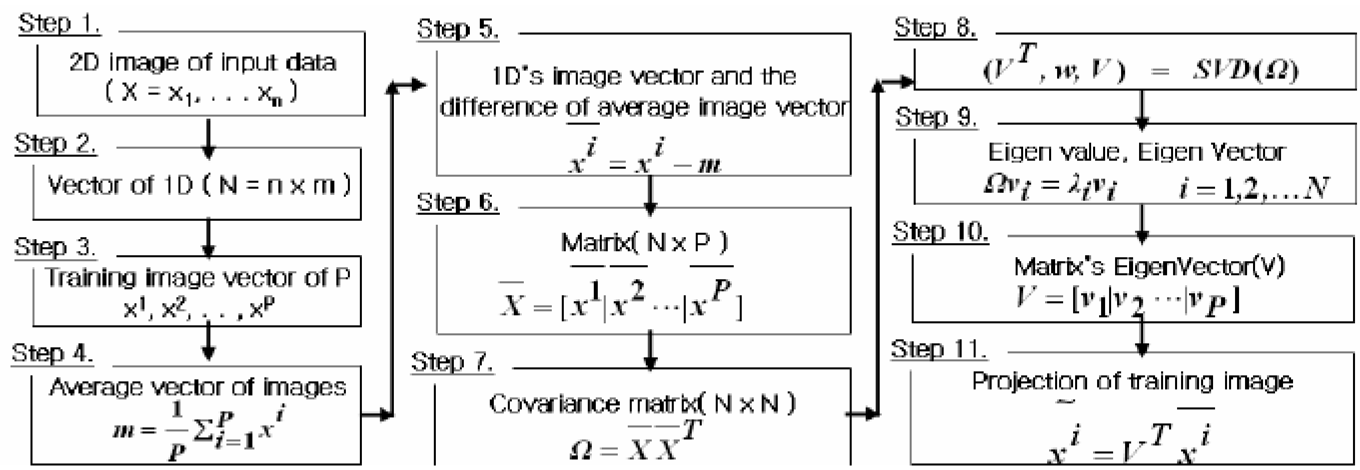
Language:

Matlab -> For Feature Extraction ***Python*** -> Deep Neural Network

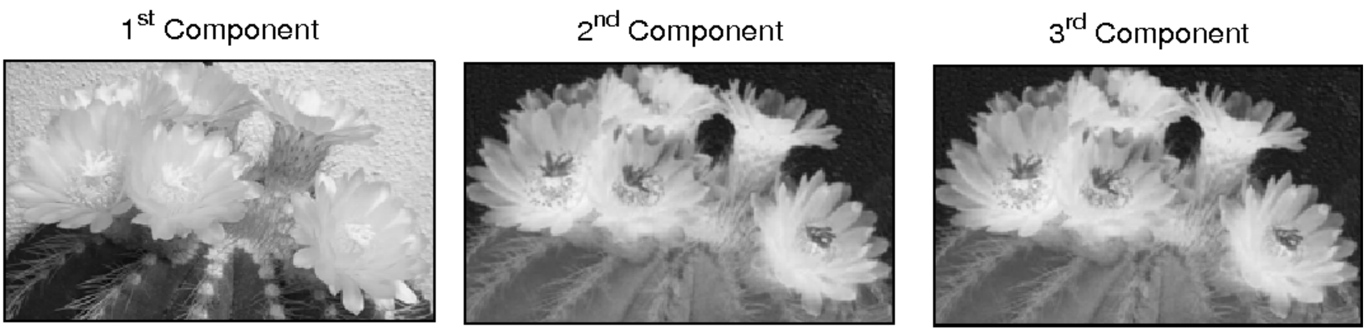
Desciption:

1. PCA Analysis: PCA is a statistical method that uses orthogonal transformations to turn a potentially correlated set of data into a linearly uncorrelated set of data that contain principal components. The number of principal components will be less than or equal to the total number of variables in the original dataset.

Furthermore, the principal components are sorted in a way so that the first component contains the largest possible variance in the data, and each succeeding component has the next highest variance. We can use the ideas presented in PCA to compress an image. Since pest pictures are not noise and randomly colored pixels, it is not unreasonable to think that there should be some underlying structure. In other words, if given a pure white pixel in an image, the chances that the surrounding pixels will be white or some variant of white is probably high. Most images do not contain sharp white to black transitions (except edges!)



PCA Algorithm



PCA on Image

2.Gray Level Co-occurrence Matrix

The Gray Level Co-occurrence Matrix1 (GLCM) and associated texture feature calculations are image analysis techniques. Given an image composed of pixels each with an intensity (a specific gray level), the GLCM is a tabulation of how often different combinations of gray levels co-occur in an image or image section. Texture feature calculations use the contents of the GLCM to give a measure of the variation in intensity (a.k.a. image texture) at the pixel of interest.

0	1	0	2
0	2	1	1
3	1	0	0
0	0	2	3

4 x 4 image

2	1	3	0
2	1	0	0
0	1	0	1
0	1	0	0

GLCM

0.16	0.08	0.25	0
0.16	0.08	0	0
0	0.08	0	0.08
0	0.08	0	0

Nomalized GLCM

GLCM Process

3. Texture Analysis The texture is a feature used to partition images into regions of interest and to classify those regions.

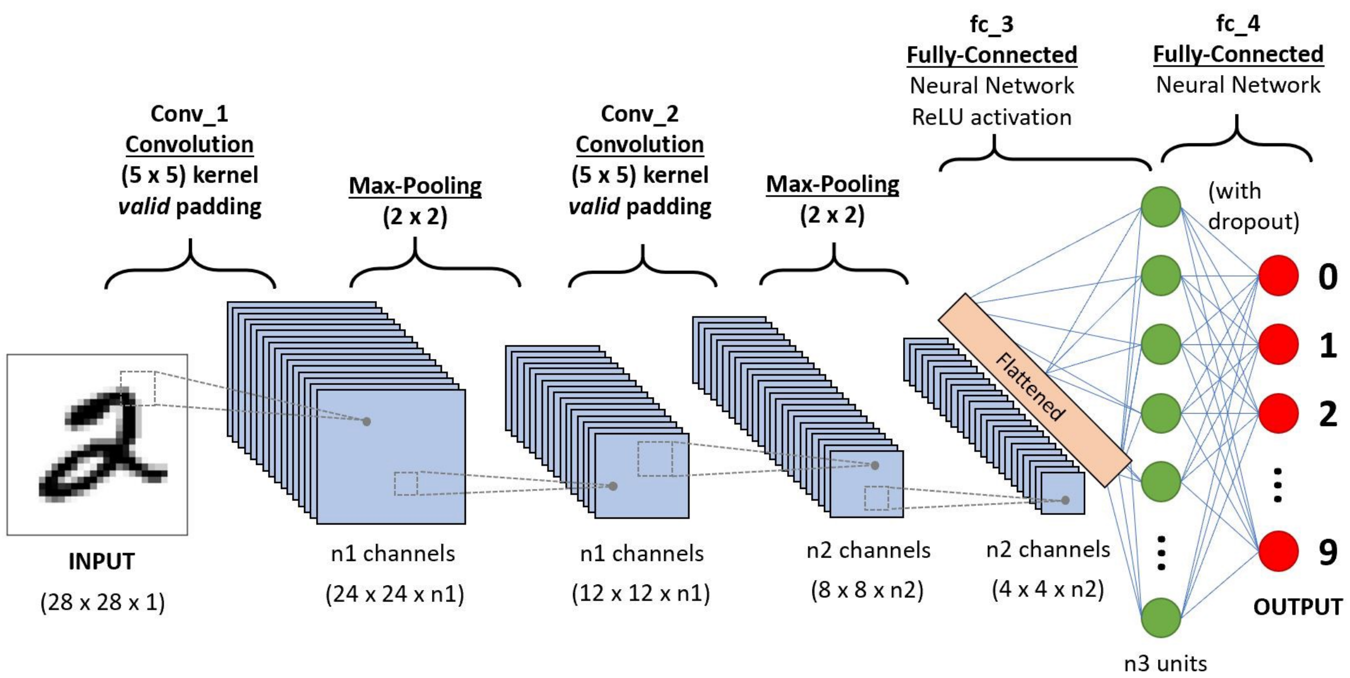
Contrast	Correlation	Energy	Homogeneity
$\sum_{i,j} \lvert i-j \rvert^2 p(i,j)$	$\sum_{i,j} \frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i \sigma_j}$	$\sum_{i,j} p(i,j)^2$	$\sum_{i,j} \frac{p(i,j)}{1+\lvert i-j \rvert}$
Density	**Variance**	**Std. Deviation**	**Max**
$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$	$var(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2$	$\sqrt{var(x)}$	Largest intensity value in window
Min	**Skewness**	**Kurtosis**	
Smallest intensity value in window	$kurtosis(x) = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^4}{(var(x))^2}$	$skewness(x) = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^3}{(\sqrt{var(x)})^3}$	

4. Convolution Neural Network

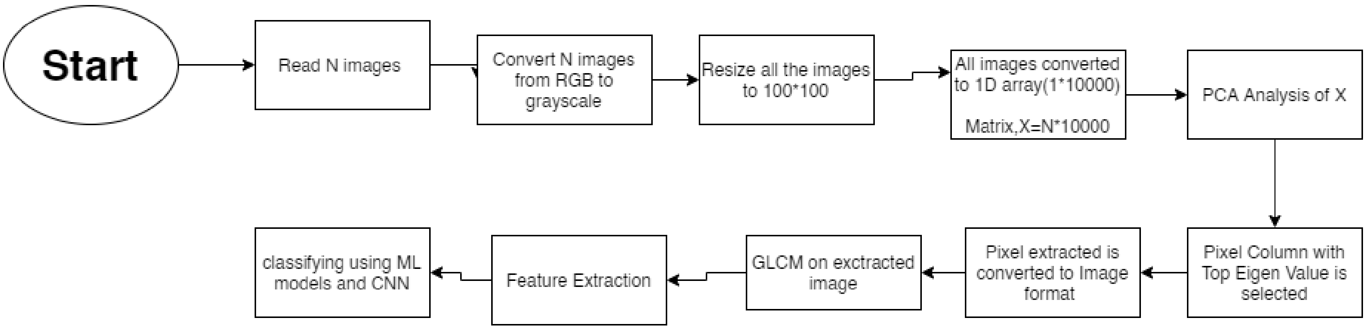
Convolutional layers are the major building blocks used in convolutional neural networks.

Convolution is the simple application of a filter to an input that results in activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in input, such as an image.

The innovation of convolutional neural networks is the ability to automatically learn a large number of filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification. The result is highly specific features that can be detected anywhere on input images.



Algorithm:



Algorithm

1. Read all the image
2. Convert the images from RGB scale to Grayscale
3. Resize all the images to 100*100
4. Convert the 2D matrix to a 1D matrix
5. PCA analysis of newly formed matrix
6. Pixel Column with Top 50 Eigen Value is selected.
7. Pixel extracted is converted to 2D Matrix [10,5]
8. GLCM on Extracted pixel
9. Feature Extraction like - Contrast - Correlation - Energy - Homogeneity
10. Classification using Convolution Neural Network
11. Accuracy is checked

Code:

1. **Image Preprocessing**

```
clc;
clear;
n=input('Enter No. of images for the training:      ');
l=input('Enter No. of Dominant Eigen Values to keep:    ');
M=72;N=72; %Required Image Dimensions
X=zeros(n,(M*N)); %Initialise Data set matrix [X]
T=zeros(n,l);%Initialize Transformed dataset [T] in PCA space
for count=1:n
    filename=sprintf("ZC/ZCimcrop (%d).JPG",count);
    I=imread(filename);%Reading image
    I=rgb2gray(I);
    I=imresize(I,[M,N]);
    X(count,:)=reshape(I,[1,M*N]);%reading images as 1D Vector
end
Xb=X;%Copy database for further use

m=mean(X);%mean of all Images
for i=1:n
    X(i,:)=X(i,:)-m;%Subtracting Mean from each 1Dimage
end
Q=(X'*X)/(n-1);%Finding Covariance Matrix
[Evecm,evalm]=eig(Q);%Getting Eigen values and Eigen Vectrs of COV matrix[Q]
```

```
Eval=diag(Evalm);%Extracting all eigen values
[Evalsorted,Index]=sort(Eval,'descend');%Sorting Eigne Values
Evecsorted=Evecm(:,Index);
Ppca=Evecsorted(:,1:l);%Reduced tranformation matrix [Ppca]
for i=1:n
    T(i,:)=(Xb(i,:)-m)*Ppca;%Projecting Each image to PCA space
end
features =zeros(n,5);%Matrix assignment for features
for i=1:n
        A=reshape(T(i,:),[10,5]);%Converting a 1d array afterPCA to image matrix
        glcm=graycomatrix(A);%Creating Gray correlation matrix
        stats=graycoprops(glcm,'Contrast Correlation Energy Homogenity');
        contrast=stats.Contrast;
        correlation=stats.Correlation;
        energy=stats.Energy;
        homogeneity=stats.Homogeneity;
        features(i,:)=[contrast,correlation,energy,homogeneity,"ZC"];
end
writematrix(features, 'MyFile_ZC.csv');
msgbox('Done','Done');
```

This part will provide us MYFile_ZC.csv which have 5 columns - Category - Contrast - Correlation - Energy - Homogeneity and N number of rows for each image.

2. **Convolution Neural Network**

This will classify the Image in various divisions and their probabilities will be used to detect accuracy scores. We have got 88% accuracy for our model.