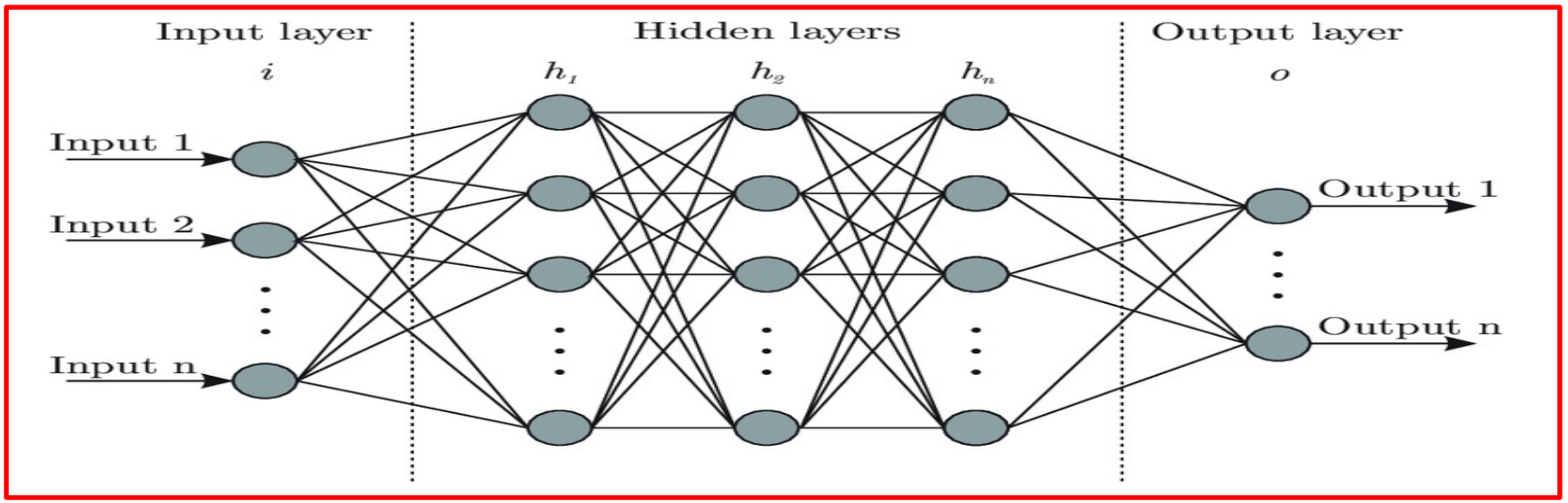
Analysis of Badminton Matches and Classification of Grape Leaf Diseases Using Convolutional Neural Network

P. Raveendran
Department of Electrical Engineering
University of Malaya

N. Ramesh Babu
Department of Mathematics
The Gandhigram Rural Institute, India

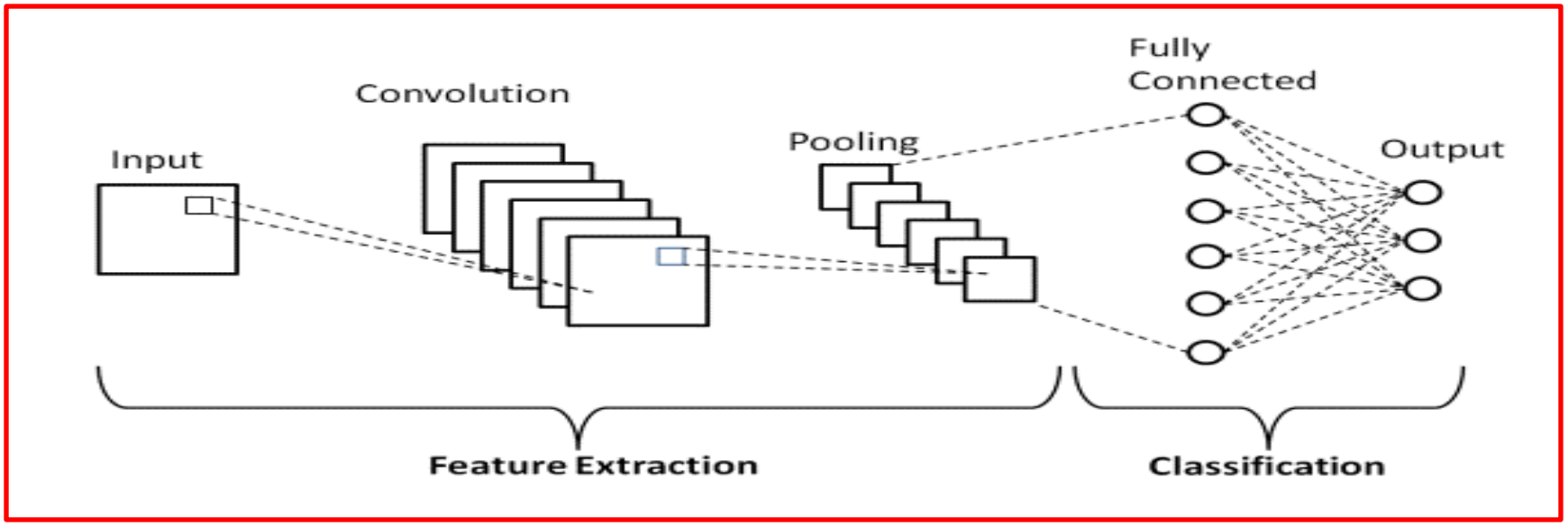
See Shin Yue
UCSI University

Background



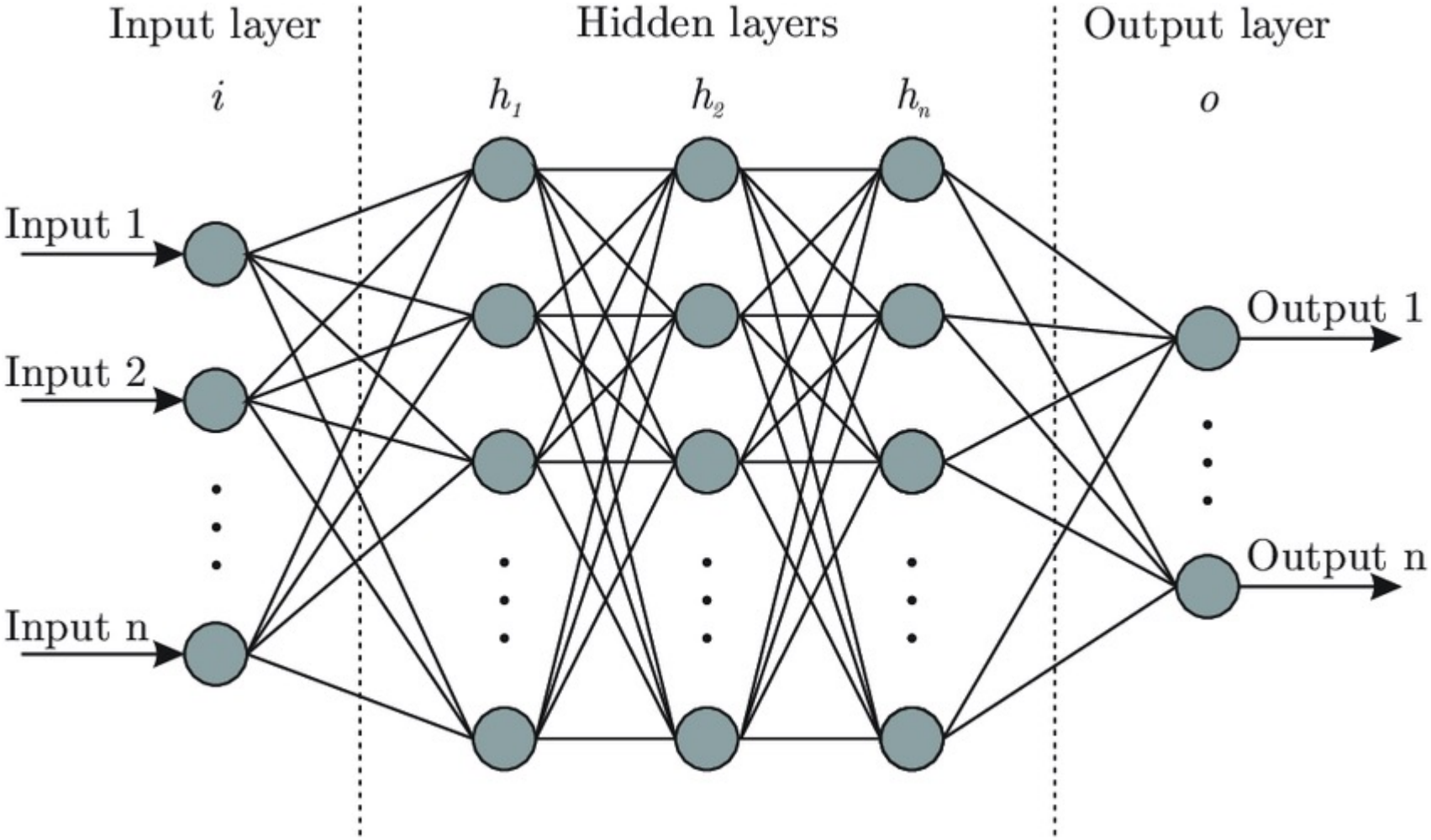
NNs

Vs



CNNs

Neural Network Architecture



Grape Leaf Disease Classification using Neural Networks

Healthy

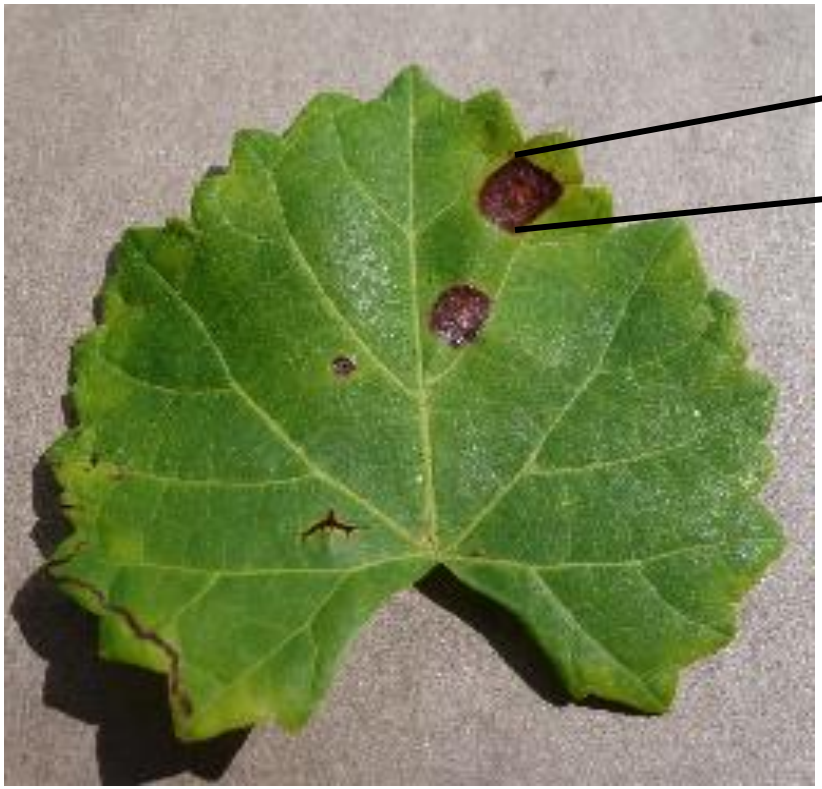
Grapes are the most widely produced commercial fruit crop in the world. They are often eaten fresh but are also commonly used to produce wine.

❖The Common or European grapevine (*Vitis vinifera*) is a long stemmed, woody vine (liana) which produces high value berries, or grapes.

❖The vines can reach lengths in excess of 30 m and can live for many years with proper management.

❖The leaves of the grape vine are alternately arranged on the stem, typically reaching sizes of 5–20 cm (2.0–7.9 in).

❖Flowers are produced in clusters and fruit. The fruit is a berry known as a grape and grows in clusters from the vine.



Black Rot

Black rot (*Guignardia bidwellii (Ellis))* is a potentially devastating fungal disease that can infect the leaves, shoots, berries and cluster stems of grapes. It is originated in eastern North America but is now in portions of Europe, South America, and Asia.

Effects: It can cause complete crop loss in warm and rainy climates.

Feature: Black spheres inside the tan field. Various factors can cause tan lesions on grape leaves, only black rot lesions will feature black sphere inside the tan field.

Black Measles

One of the common fungal diseases is Esca (Black Measles) which is found in the Grape Plants and can be easily identified as brown streaking lesions on any part of the leaf. The affected leaves can dry off completely and fall off from the plant prematurely which eventually results in death of the plant.

Effects: It can cause complete crop loss.

Feature: Interveinal striping that looks like tiger stripes. White cultivars show chlorotic and nectrotic strips. Red cultivars show red areas and necrotic strips.

Leaf Blight

Leaf blight *(Isariopsis Leaf Spot)* is commonly visible in tropical and subtropical grapes. The disease appear late in the season. Cynthiana and Cabernet Sauvignon are susceptible to this pathogen. Fungicides sprayed for other diseases in the season may help to reduce this disease.

Features: On leaf surface we will see lesions which are irregularly shaped (2 to 25 mm in diameter). Initially lesions are dull red to brown in color turn black later. If disease is severe this lesions may coalesce. On berries we can see symptom similar to black rot but the entire clusters will collapse.

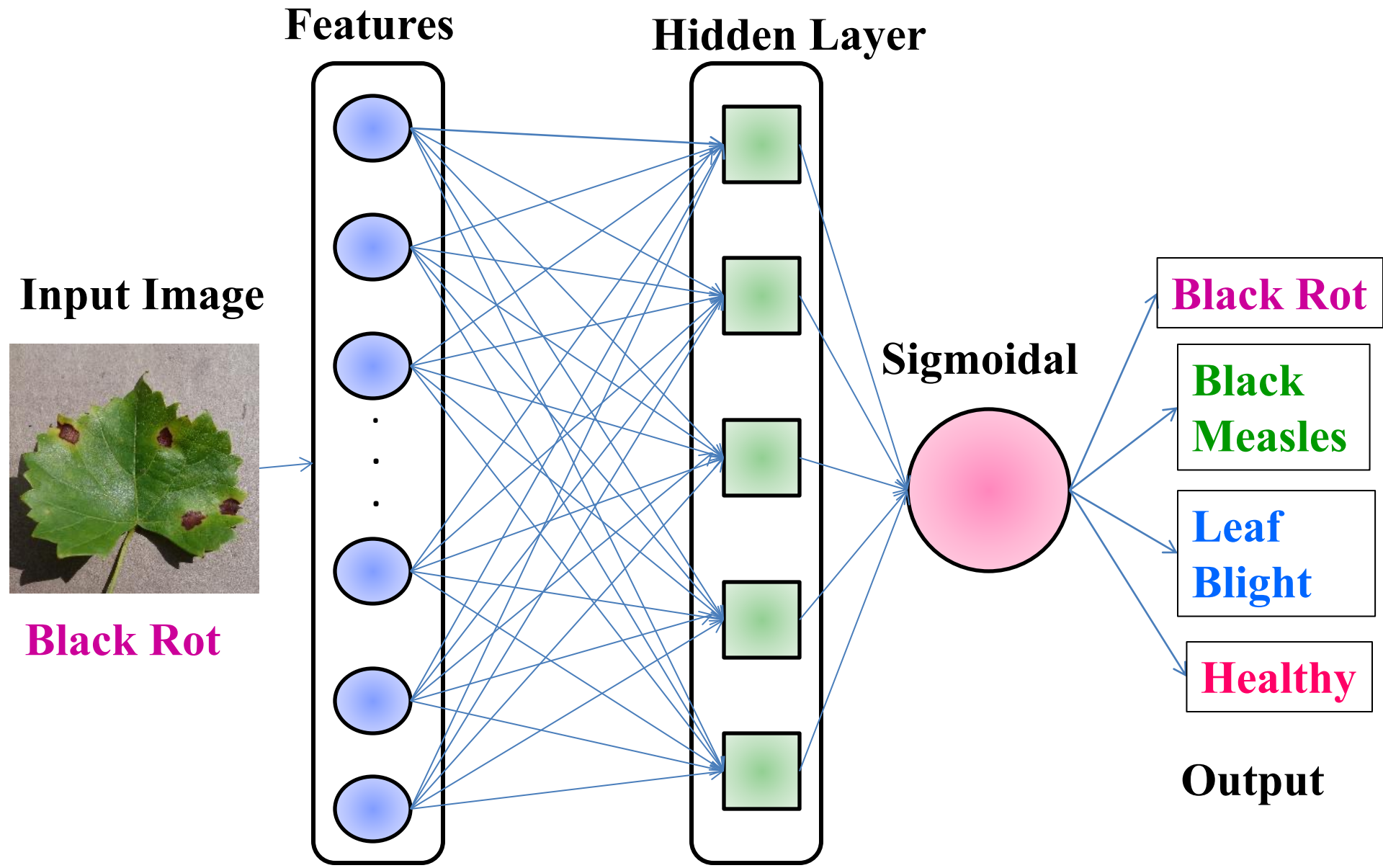
Grape Leaf Data Set - 1

Leaf Disease	Total No. of Images	No. of Training Images	No. of Testing Images
Black Rot	100	80	20
Black Measles	100	80	20
Leaf Blight	100	80	20
Healthy	100	80	20
Total	**400**	**320**	**80**

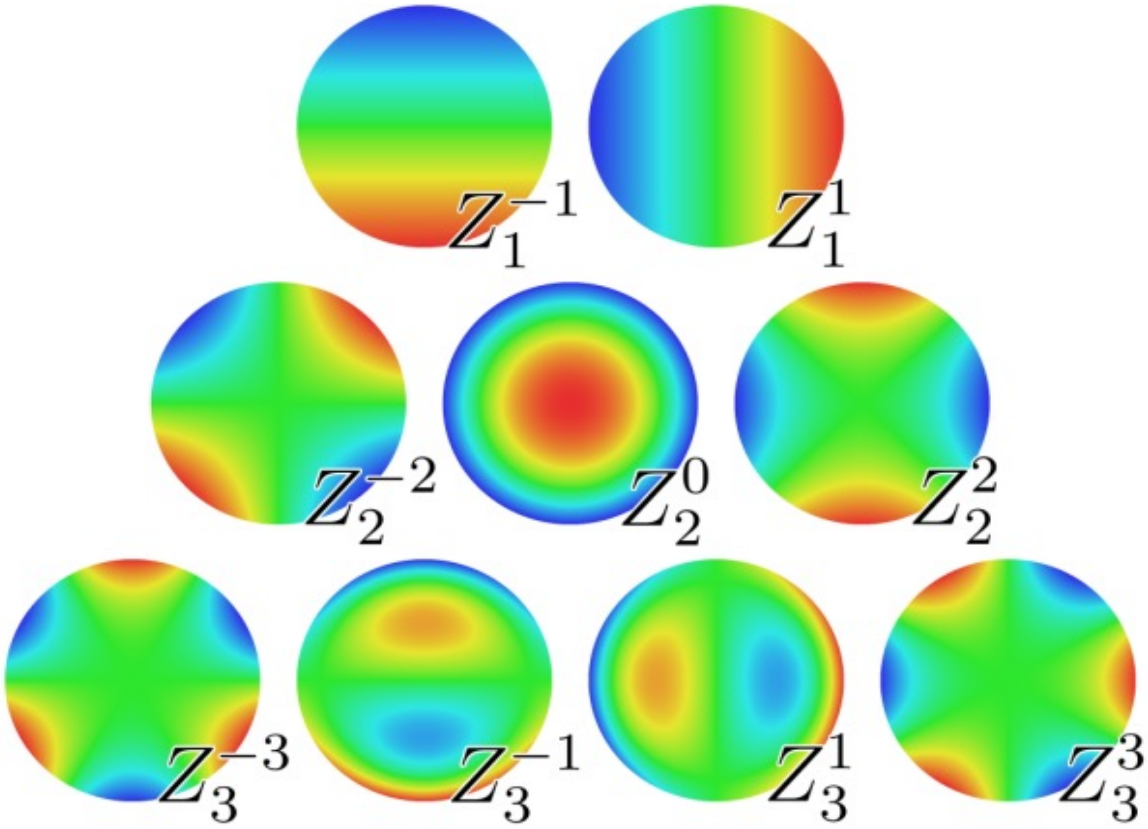
https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset

Grape Leaf Data Set - 2

Leaf Disease	Total No. of Images	No. of Training Images	No. of Testing Images
Black Rot	250	200	50
Black Measles	250	200	50
Leaf Blight	250	200	50
Healthy	250	200	50
Total	**1000**	**800**	**200**


Features
Hidden Layer
Input Image
Black Rot
Sigmoidal
Black Rot
Black Measles
Leaf Blight
Healthy
Output

Zernike Moments



Zernike moments were originally introduced in the 1930s by physicist and Noble prize winner ***Fritz Zernike*** *to describe optical aberrations.*

Orthogonal Functions

$$\int_a^b y_m(x)\,y_n(x)\,dx = 0; \qquad m \neq n$$

Zernike Polynomial

A Set of orthogonal polynomial defined on the disk

$$V_{n,m}(\rho,\theta) = R_{n,m}(\rho)e^{jm\theta}$$

$$R_{n,m}(\rho) = \begin{cases} \sum_{k=0}^{(n-|m|)/2} \dfrac{(-1)^k (n-k)!}{k!\left[\left(\dfrac{n+|m|}{2}\right)-k\right]!\left[\left(\dfrac{n-|m|}{2}\right)-k\right]!}\rho^{n-2k}, & \text{for } n-m \text{ even} \\ 0, & \text{for } n-m \text{ odd} \end{cases}$$

Zernike Moments

$$Z_{\mathrm{n,m}} = \frac{n+1}{\pi} \int_x \int_y f(x,y)[V_{n,m}(x,y)]^* dxdy$$

Where $x^2 + y^2 \leq 1$

For the digital image, let us denote the intensity of the image pixel as f(x,y), Hence the Zenike moment is defined as

$$Z_{n,m} = \frac{n+1}{\pi} \sum_x \sum_y f(x,y)\left[V_{n,m}(x,y)\right]^*$$

Each image features can be represented with a sequence of Zernike moments

$$\left\{Z_{1,1}, Z_{2,0}, Z_{2,2}, \cdots, Z_{n_{\max},m}\right\}$$

Each moment relates a different piece of information pertaining to the image.

Order	Moments	# of Moments
1	$Z_{1,1}$	1
2	$Z_{2,0}$ $Z_{2,2}$	2
3	$Z_{3,1}$ $Z_{3,3}$	2
4	$Z_{4,0}$ $Z_{4,2}$ $Z_{4,4}$	3
5	$Z_{5,1}$ $Z_{5,3}$ $Z_{5,5}$	3
6	$Z_{6,0}$ $Z_{6,2}$ $Z_{6,4}$ $Z_{6,6}$	4
7	$Z_{7,1}$ $Z_{7,3}$ $Z_{7,5}$ $Z_{7,7}$	4
8	$Z_{8,0}$ $Z_{8,2}$ $Z_{8,4}$ $Z_{8,6}$ $Z_{8,8}$	5
9	$Z_{9,1}$ $Z_{9,3}$ $Z_{9,5}$ $Z_{9,7}$ $Z_{9,9}$	5
10	$Z_{10,0}$ $Z_{10,2}$ $Z_{10,4}$ $Z_{10,6}$ $Z_{10,8}$ $Z_{10,10}$	6
11	$Z_{11,1}$ $Z_{11,3}$ $Z_{11,5}$ $Z_{11,7}$ $Z_{11,9}$ $Z_{11,11}$	6
12	$Z_{12,0}$ $Z_{12,2}$ $Z_{12,4}$ $Z_{12,6}$ $Z_{12,8}$ $Z_{12,10}$ $Z_{12,12}$	7

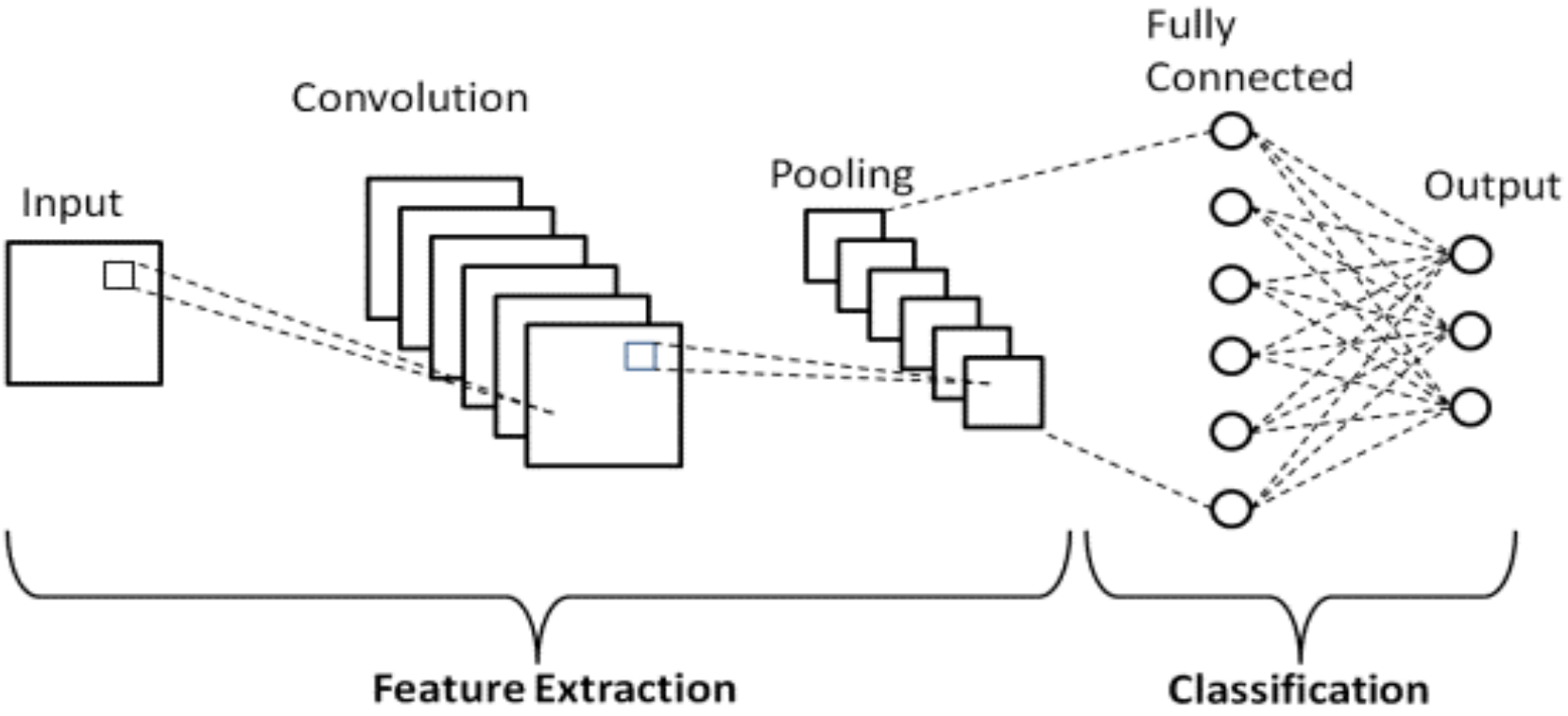
Experimental Results (Dataset 1)

# Neurons in the Hidden Layer	Training 320	Testing 80
20	96.87 %	71 %
30	97.81 %	70 %
40	97.81 %	70.5 %
50	97.87 %	71 %
60	97.875 %	70 %

Experimental Results (Dataset 2)

# Neurons in the Hidden Layer	Training 800	Testing 200
20	91.875 %	74 %
30	91.5 %	76 %
40	91 %	77 %
50	92. 1 %	75%
60	92.87 %	72 %

Convolutional Neural Network (CNN)



Historical View

LeNet-5 (1989–1998)

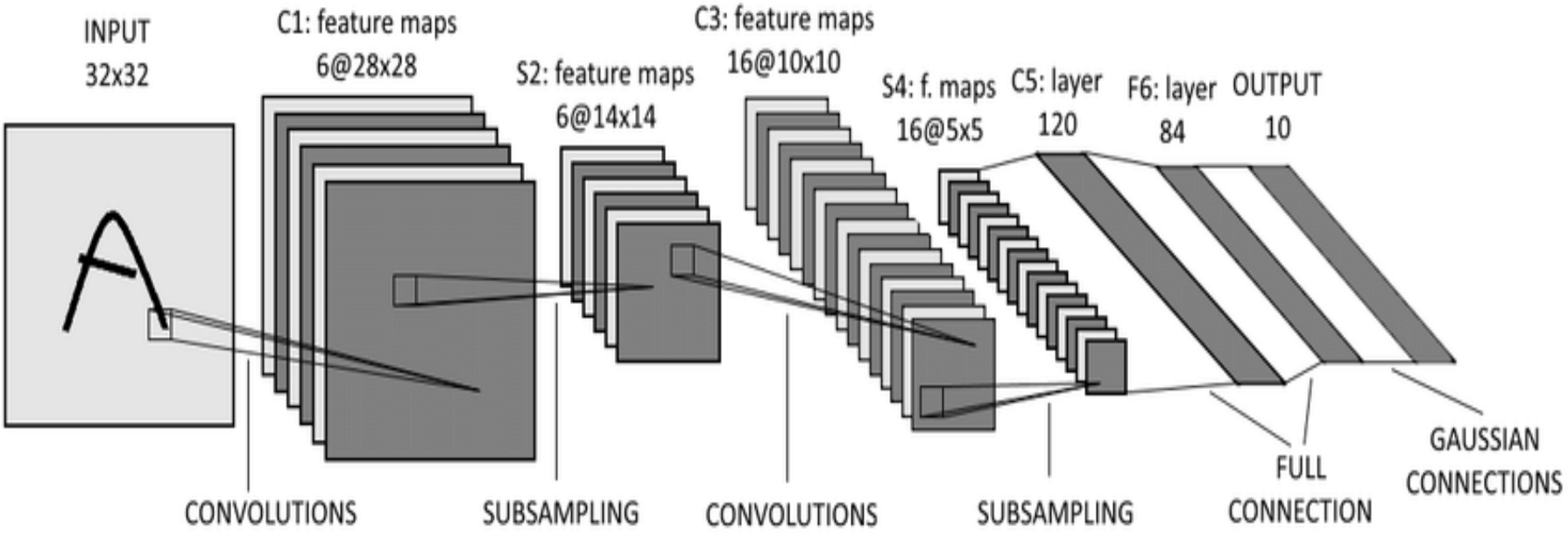
The first successful applications of convolutional networks Were developed by Yann LeCun. In 1989, LeCun's team published the first study to successfully train CNNs via backpropagation.

At the time LeNet achieved outstanding results matching the performance of support vector machines, then a dominant Approach in supervised learning, achieving an error rate of less than 1% per digit.

LeNet was eventually adapted to recognize digits for processing deposits in ATM machines. To this day, some ATMs still run the code that Yann LeCun and his colleague Leon Bottou wrote in the 1990s!

http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

LeNet Architecture



The credit for newer architectures of CNNs goes to ImageNet (a dataset) classification challenge named 'ImageNet large scale visual recognition challenge (ILSVRC)'.

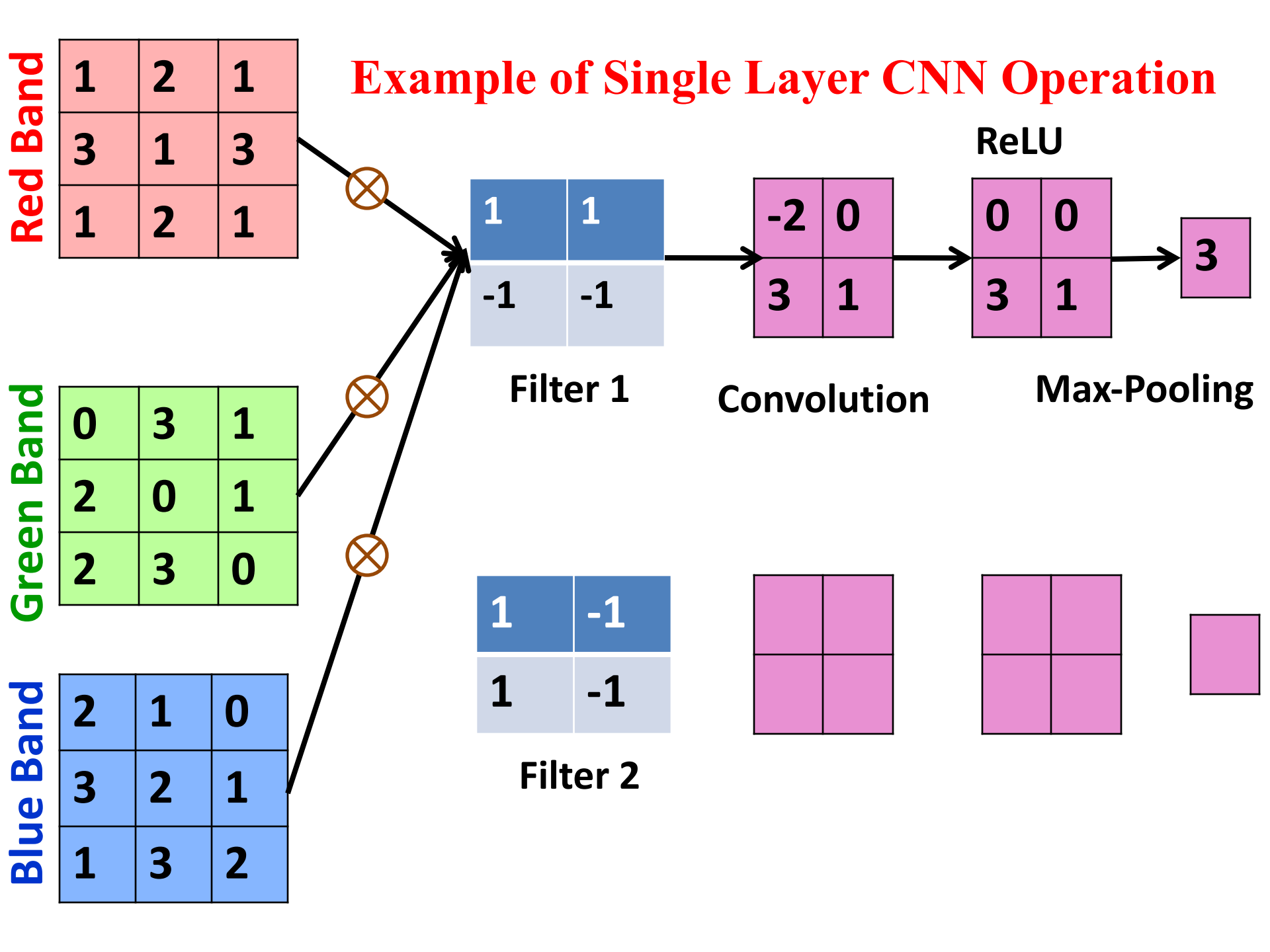
https://image-net.org/index.php

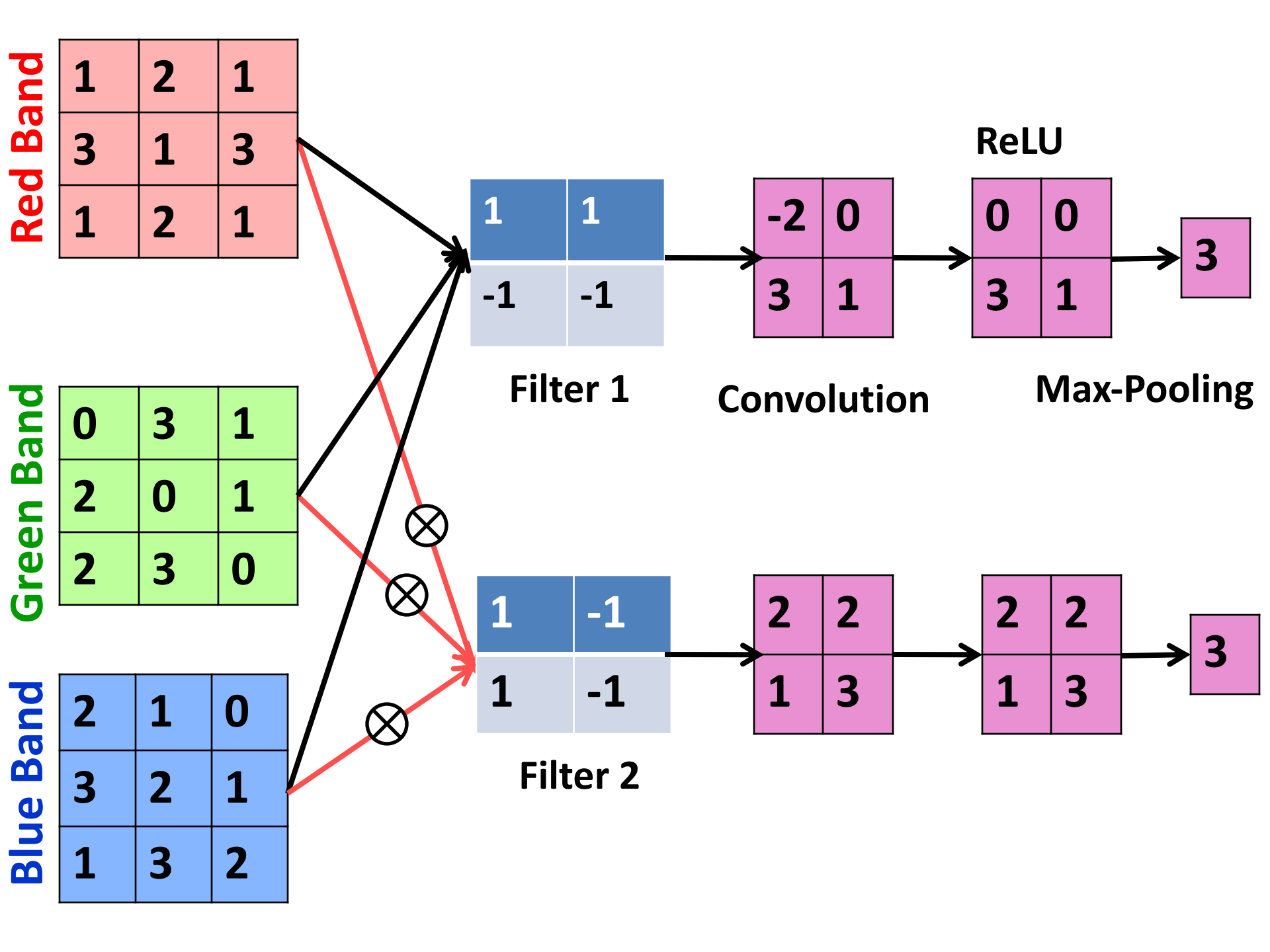
It was started in 2010 which led to a significant effort across researchers to benchmark their machine learning and computer vision models, in particular for image classification, on a common dataset.

In 2010, the winning error rate was 28.2%. In 2011 researchers improved the score from 28.2% to 25.8% error rate.

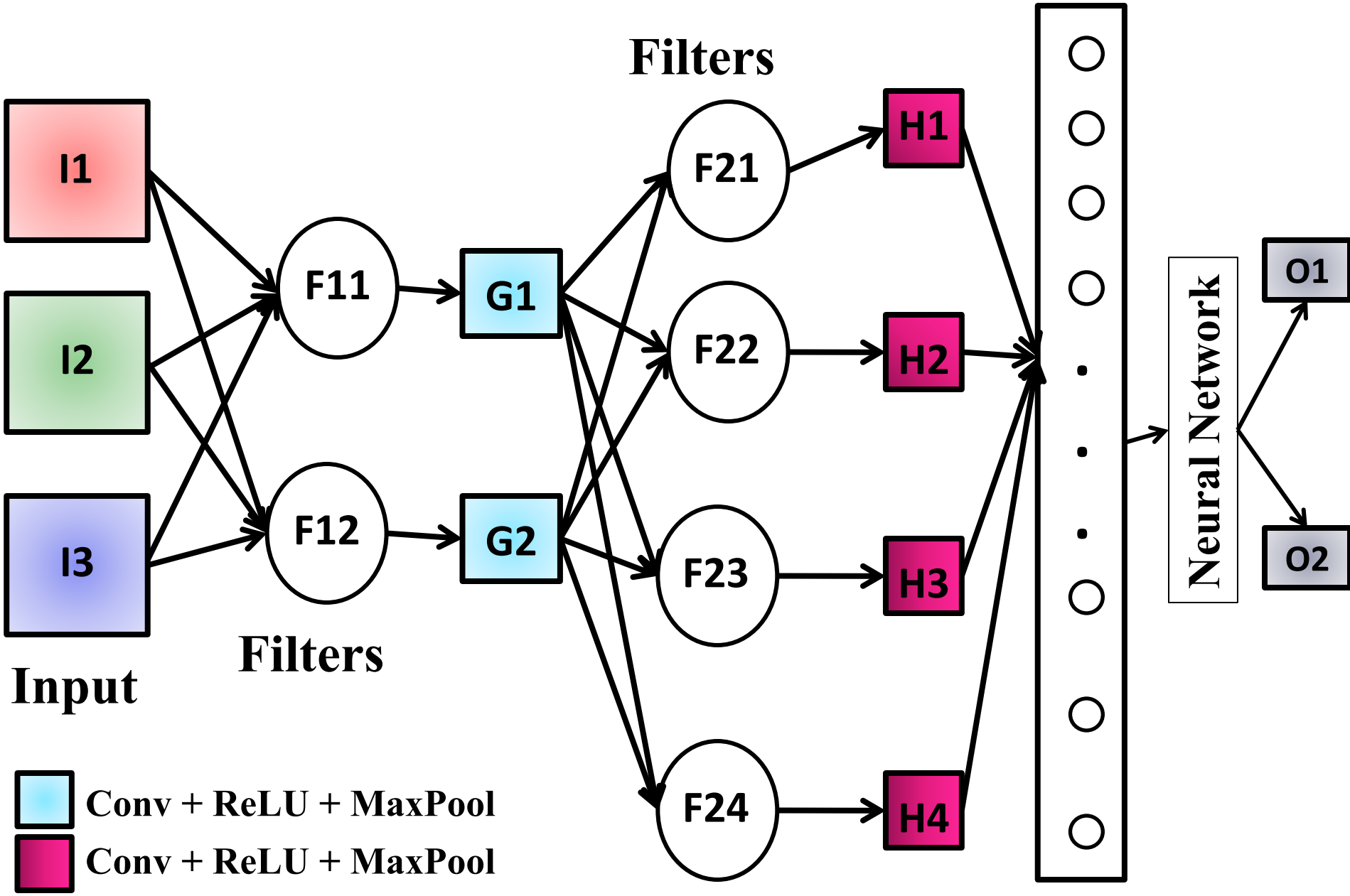
AlexNet (2012)

Finally in 2012, Alex Krizhevsky and Geoffrey Hinton came up with a CNN architecture popular to this day as AlexNet, which reduced the error from 25.8% to 16.4% which was a significant improvement at that time.

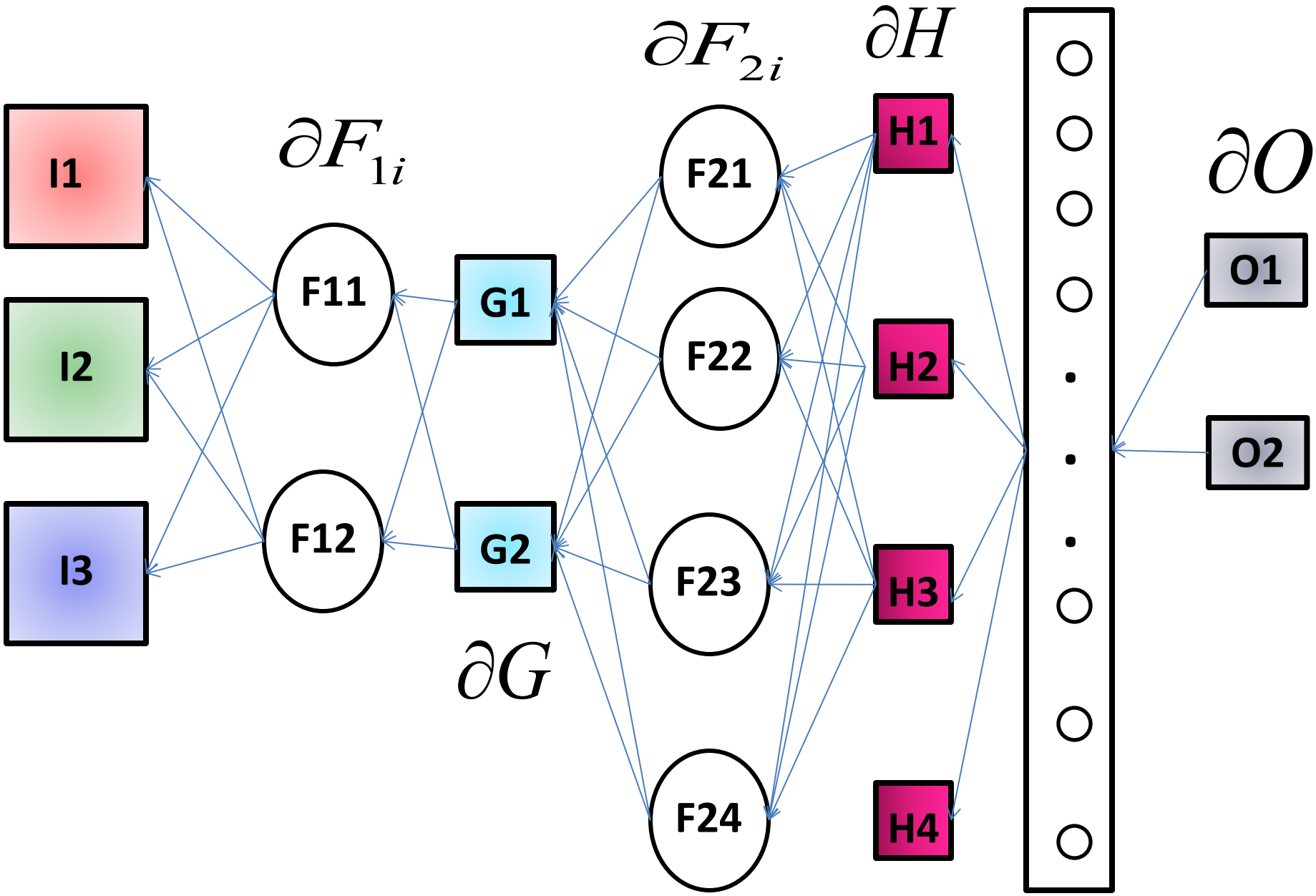

Example of Single Layer CNN Operation
Red Band
1 2 1
3 1 3
1 2 1
Green Band
0 3 1
2 0 1
2 3 0
Blue Band
2 1 0
3 2 1
1 3 2
1 1
-1 -1
Filter 1
1 -1
1 -1
Filter 2
Convolution
-2 0
3 1
ReLU
0 0
3 1
Max-Pooling
3


Red Band
1 2 1
3 1 3
1 2 1
Green Band
0 3 1
2 0 1
2 3 0
Blue Band
2 1 0
3 2 1
1 3 2
1 1
-1 -1
Filter 1
-2 0
3 1
Convolution
ReLU
0 0
3 1
Max-Pooling
3
1 -1
1 -1
Filter 2
2 2
1 3
2 2
1 3
3

Training Phase: Forward Propagation of CNN



Training Phase: Backpropagation of CNN



$$E_{Total} = \frac{1}{2}\left[(Actual_{O1} - Out_{O1})^2 + \left[(Actual_{O2} - Out_{O2})^2\right]\right]$$

$$\frac{\partial E_{Total}}{\partial out_{O1}} = Out_{O1} - Actual_{O1}$$

$$Out_{O1} = \frac{1}{1 + e^{-net_{O1}}}$$

$$\frac{\partial out_{O1}}{\partial net_{O1}} = Out_{O1}(1 - Out_{O1})$$

$$net_{O_1} = w_{11}H_1 + w_{12}H_2 + w_{13}H_3 + w_{14}H_4$$

$$\frac{\partial net_{O1}}{\partial w_{11}} = H_1$$

$$\frac{\partial E_{Total}}{\partial w_{11}} = \frac{\partial E_{Total}}{\partial out_{O1}} * \frac{\partial out_{O1}}{\partial net_{O1}} * \frac{\partial net_{O1}}{\partial w_{11}}$$

$$\partial F_{2i} = Convolution(G, \partial H)$$

$$\partial F_{1i} = Convolution(I, \partial G)$$

Grape Leaf Disease Classification using CNNs

Grape Leaf Data Set - 1

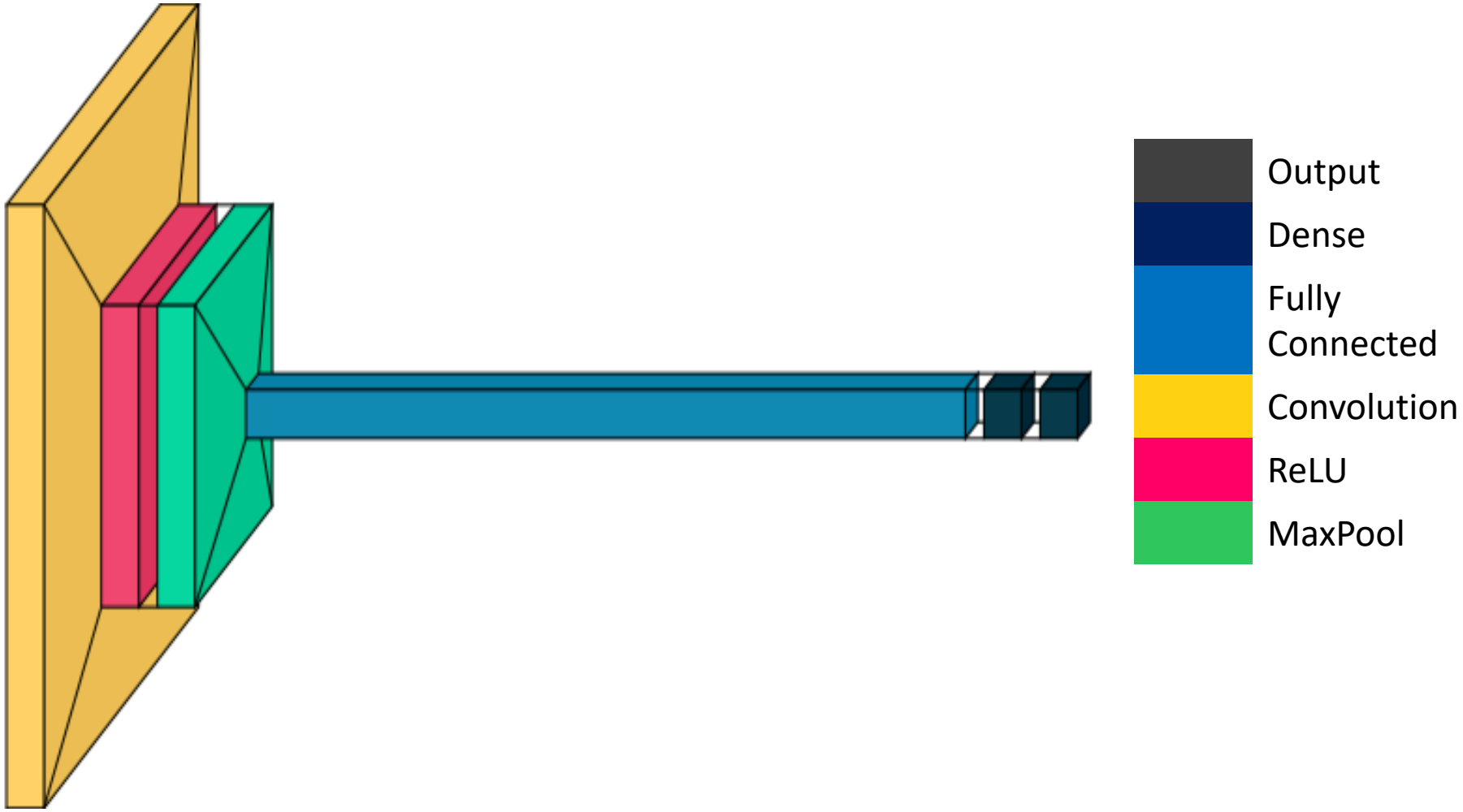
Leaf Disease	Total No. of Images	No. of Training Images	No. of Testing Images
Black Rot	100	80	20
Black Measles	100	80	20
Leaf Blight	100	80	20
Healthy	100	80	20
Total	**400**	**320**	**80**

https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset

Training Dataset

Testing Dataset

Single Layer CNN Architecture

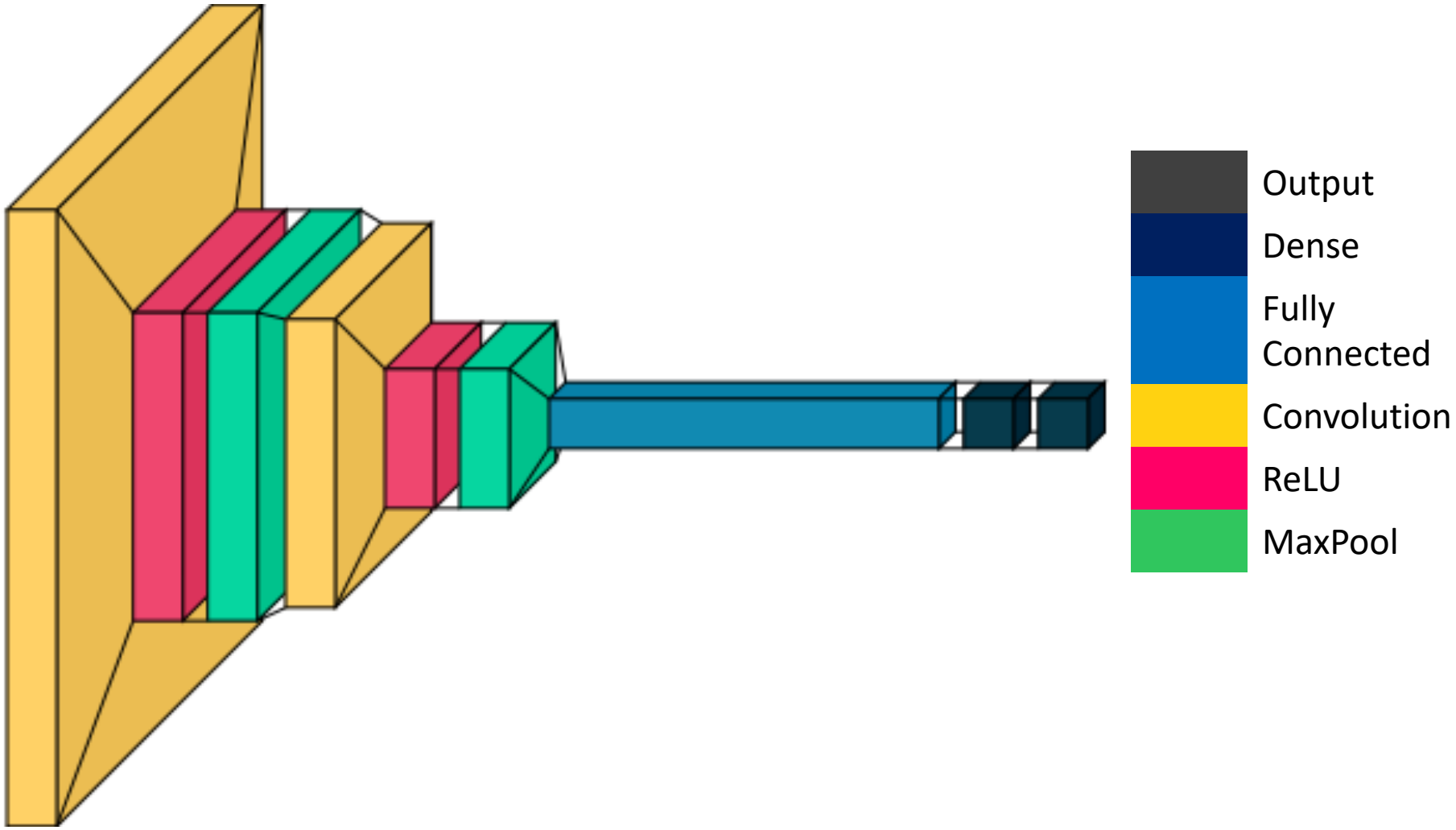


Layer	Output shape	# Parameters K*(m*n*l)+b
Convolution + ReLU	(62, 62, 4)	4*(3*3*3)+4 = 112
Max Pooling	(31, 31, 4)	0
Flattern	3844	0
Dense (Hidden)	16	61520
Output	4	68
	Total # Parameters	61,700

Filter Size = m * n* l, # filters = k, and # bias = b

Input Size (Training / Testing)	# Filters in First Layer	# Filters in Second Layer	# Filters in Third Layer	Training Accuracy	Testing Accuracy
400 (320 / 80)	4	-	-	100 %	75 %
1000 (800 / 200)	4	-	-	100 %	79 %
400 (320 / 80)	8	-	-	100 %	86.25 %
1000 (800 / 200)	8	-	-	98.2 %	84.5 %
400 (320 / 80)	16	-	-	100 %	87.5 %
1000 (800 / 200)	16	-	-	100 %	83.5 5

Two Layer CNN Architecture

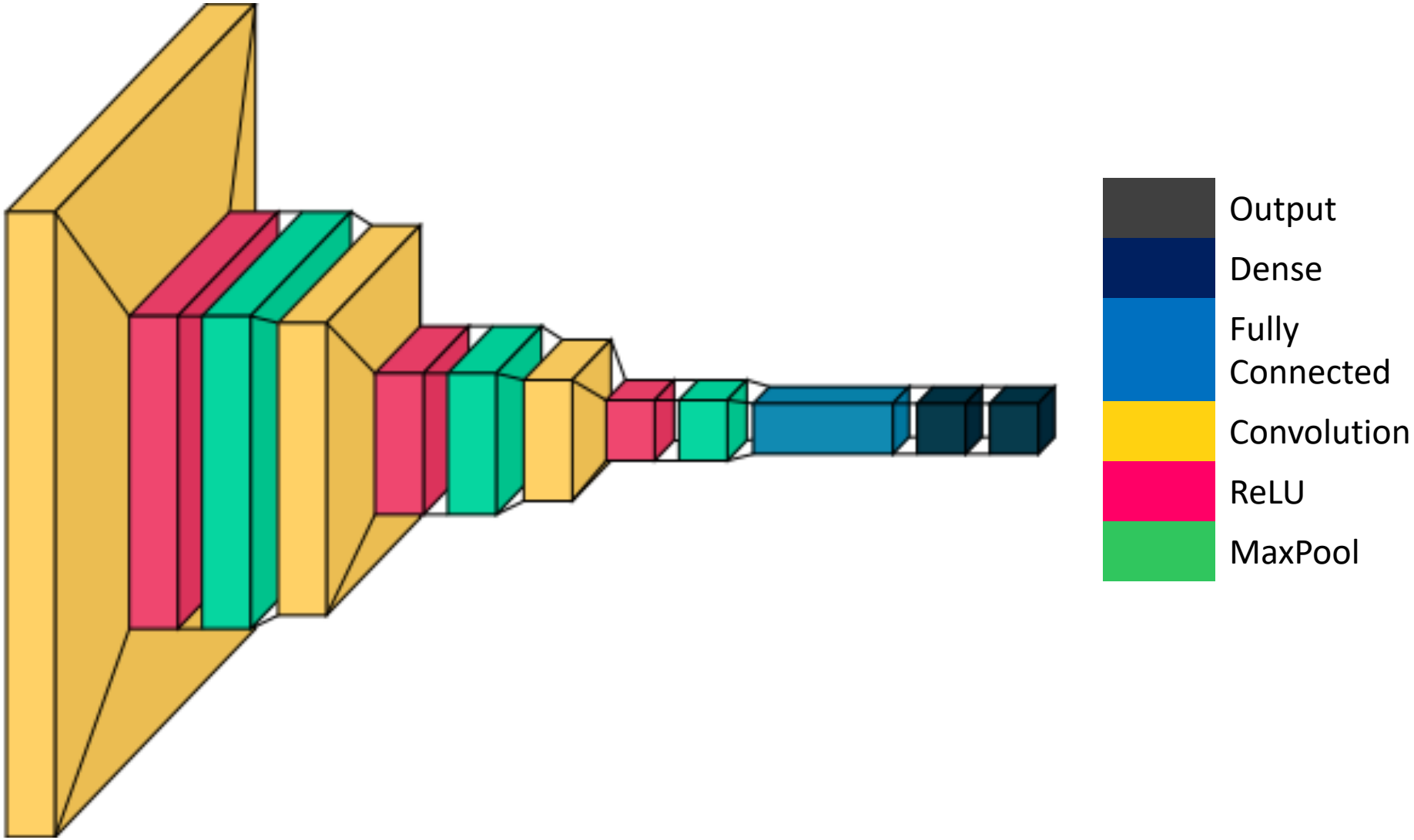


Layer	Output shape	# Parameters K*(m*n*l)+b
Convolution + ReLU	(62, 62, 4)	4*(3*3*3)+4 = 112
Max Pooling	(31, 31, 4)	0
Convolution + ReLU	(29, 29, 8)	296
Max Pooling	(14, 14, 8)	0
Flattern	1568	0
Dense (Hidden)	16	25104
Output	4	68
	Total # Parameters	25,580

Filter Size = m * n* l, # filters = k, and # bias = b

Input Size (Training / Testing)	# Filters in First Layer	# Filters in Second Layer	# Filters in Third Layer	Training Accuracy	Testing Accuracy
400 (320 / 80)	4	8	-	99 %	83.75 %
1000 (800 / 200)	4	8	-	99 %	84.5 %
400 (320 / 80)	4	16	-	99.6 %	80 %
1000 (800 / 200)	4	16	-	100 %	84.5 %
400 (320 / 80)	8	16	-	99 %	87.5 %
1000 (800 / 200)	8	16	-	97 %	86.5 %

Three Layer CNN Architecture

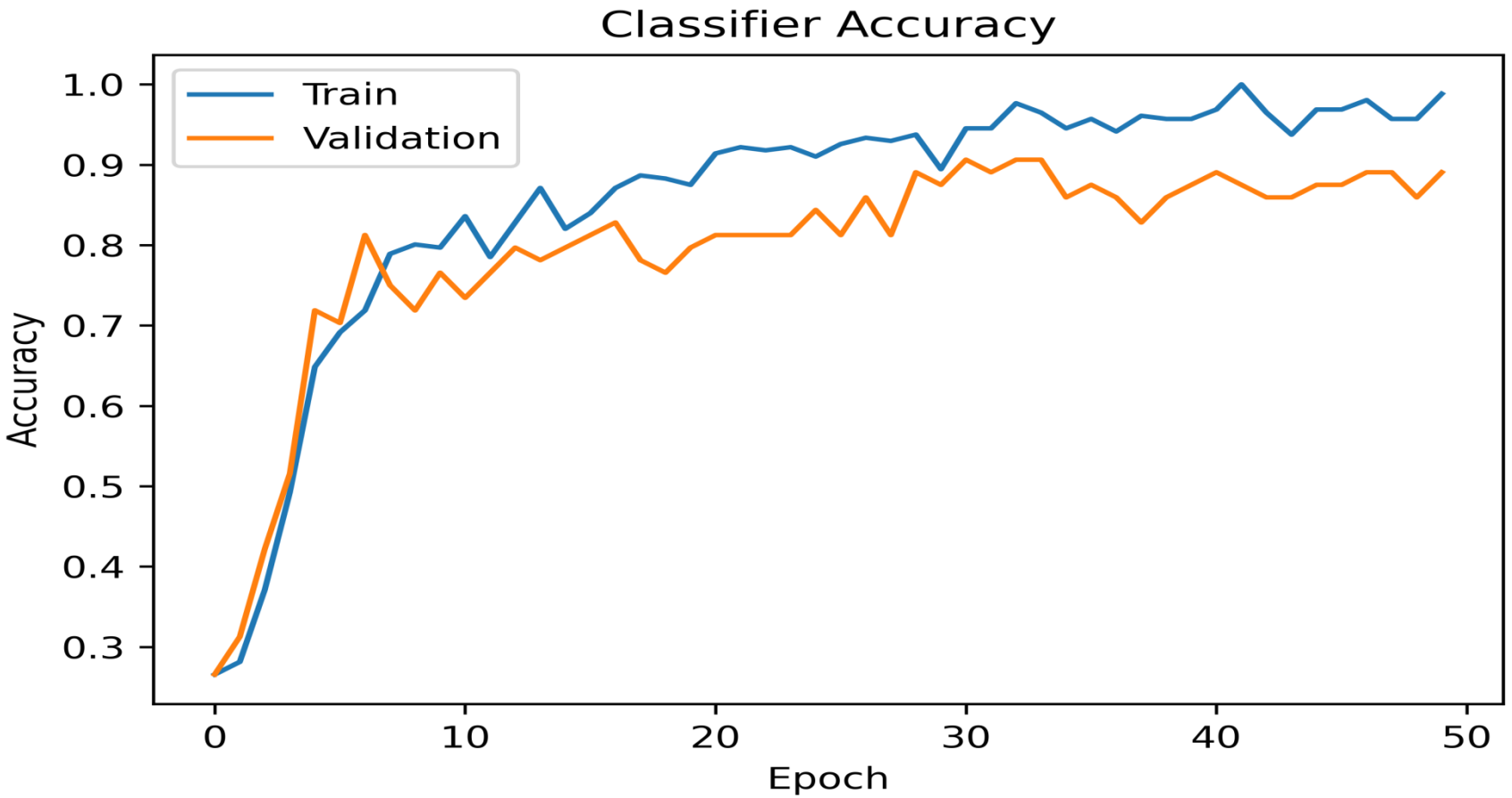


Layer	Output shape	# Parameters K*(m*n*l)+b
Convolution + ReLU	(62, 62, 4)	4*(3*3*3)+4 = 112
Max Pooling	(31, 31, 4)	0
Convolution + ReLU	(29, 29, 8)	296
Max Pooling	(14, 14, 8)	0
Convolution + ReLU	(12, 12, 16)	1168
Max Pooling	(6, 6, 16)	0
Flattern	576	0
Dense (Hidden)	16	9232
Output	4	68
	Total # Parameters	10,876

Filter Size = m * n* l, # filters = k, and # bias = b

Input Size (Training / Testing)	# Filters in First Layer	# Filters in Second Layer	# Filters in Third Layer	Training Accuracy	Testing Accuracy
400 (320 / 80)	4	8	16	95 %	86 %
1000 (800 / 200)	4	8	16	99 %	90.5 %
400 (320 / 80)	4	16	4	94 %	86.25 %
1000 (800 / 200)	4	16	4	94.5 %	87.5 %
400 (320 / 80)	16	8	16	98 %	90 %
1000 (800 / 200)	16	8	16	98.5 %	91 %
400 (320 / 80)	8	8	8	97 %	90 %
1000 (800 / 200)	8	8	8	95 %	91 %

Experimental Results 1



Training : 95 %

Testing : 86%

Receiver Operating Characteristic (ROC) Curve

An **ROC curve** is a graph showing the performance of a classification model at all classification thresholds.

This curve plots two parameters:

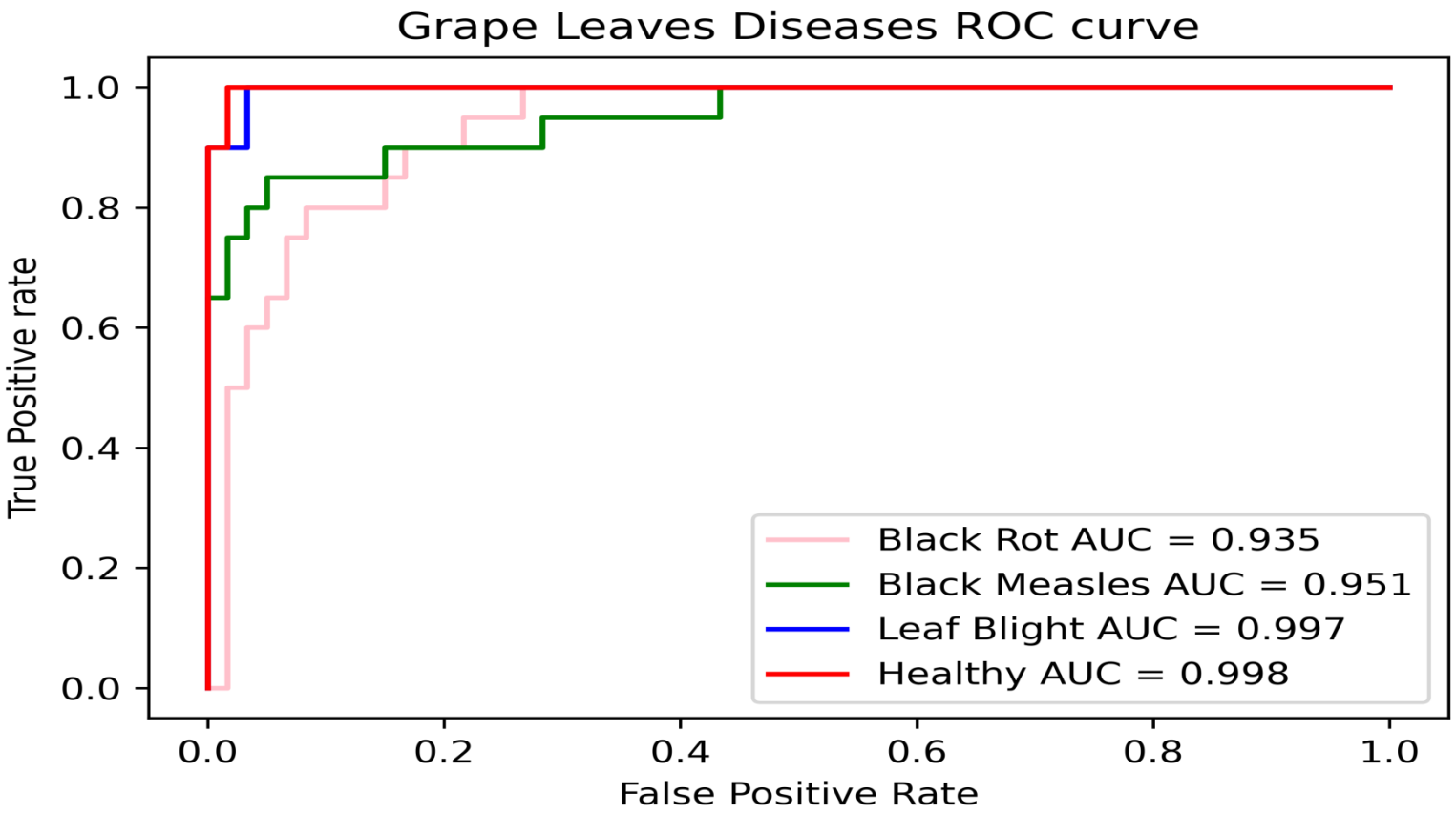
- True Positive Rate
- False Positive Rate

True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

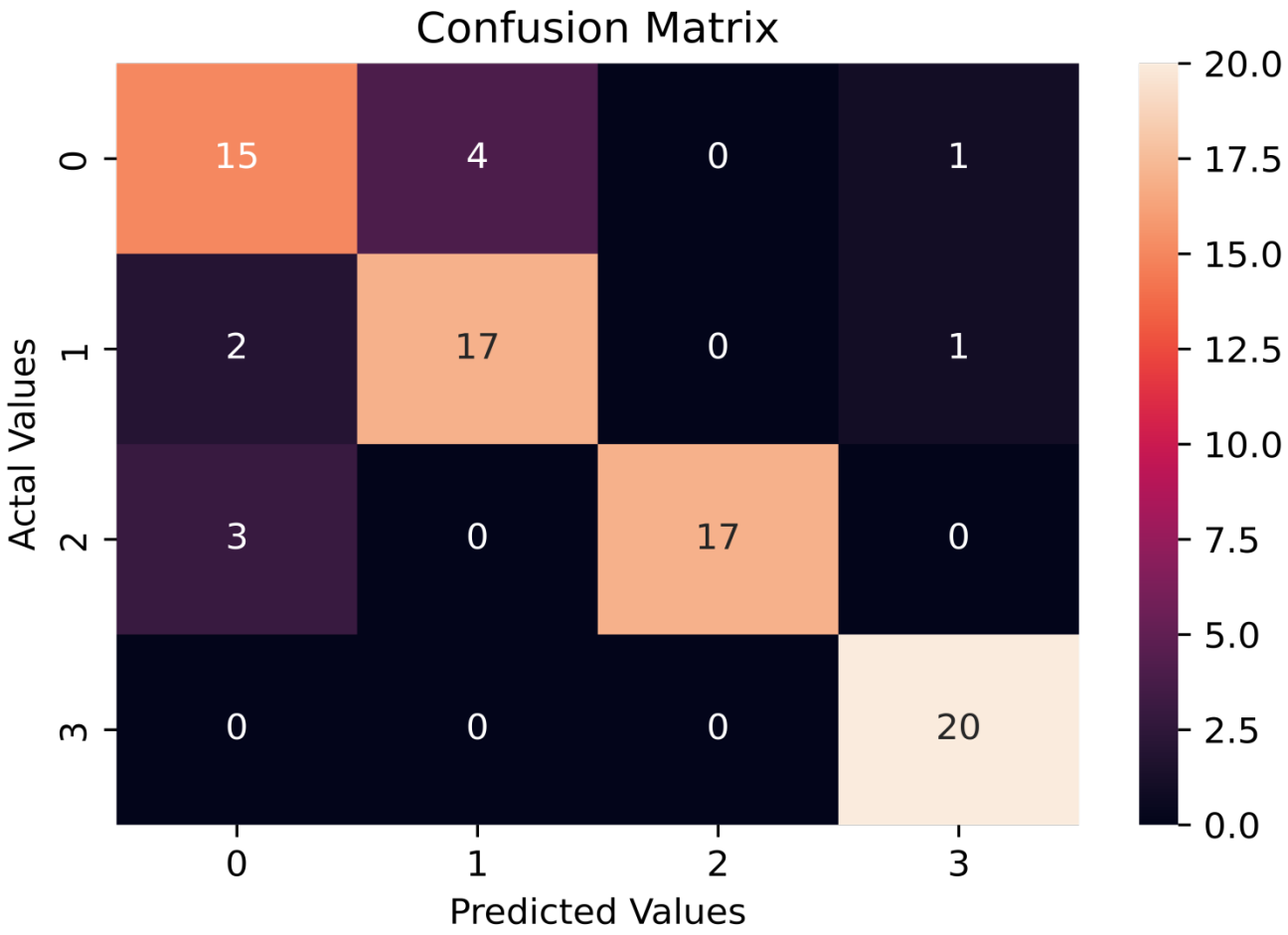
$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (FPR) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$



Confusion Matrix



Filter Size	# Filters in First Layer	# Filters in Second Layer	# Filters in Third Layer	Training Accuracy	Testing Accuracy
3×3	4	8	16	95 %	86 %
3×3	4	16	8	93.7 %	75 %
3×3	8	4	16	92 %	81 %
3×3	16	4	8	95 %	86 %
3×3	8	16	4	96 %	82 %
3×3	16	8	4	94 %	81 %
3×3	4	4	8	90 %	85 %
3×3	8	4	4	89 %	80 %
3×3	4	8	4	88.6 %	77 %

Filter Size	# Filters in First Layer	# Filters in Second Layer	# Filters in Third Layer	Training Accuracy	Testing Accuracy
3×3	4	4	16	96 %	90 %
3×3	16	4	4	93 %	85 %
3×3	4	16	4	94 %	86.25 %
3×3	8	8	4	86.7 %	81 %
3×3	4	8	8	96 %	80 %
3×3	8	4	8	92.5 %	80 %
3×3	8	8	16	99.6 %	86.25 %
3×3	16	8	8	98.8 %	82.5 %
3×3	8	16	8	96.5 %	83.7 %

Filter Size	# Filters in First Layer	# Filters in Second Layer	# Filters in Third Layer	Training Accuracy	Testing Accuracy
3×3	16	16	4	26 %	25 %
3×3	4	16	16	93 %	73.75 %
3×3	16	4	16	96 %	77.5 %
3×3	16	16	8	93.75 %	86 %
3×3	8	16	16	95 %	81 %
3×3	16	8	16	98 %	90 %
3×3	4	4	4	92 %	85 %
3×3	8	8	8	97 %	90 %
3×3	16	16	16	95 %	86 %

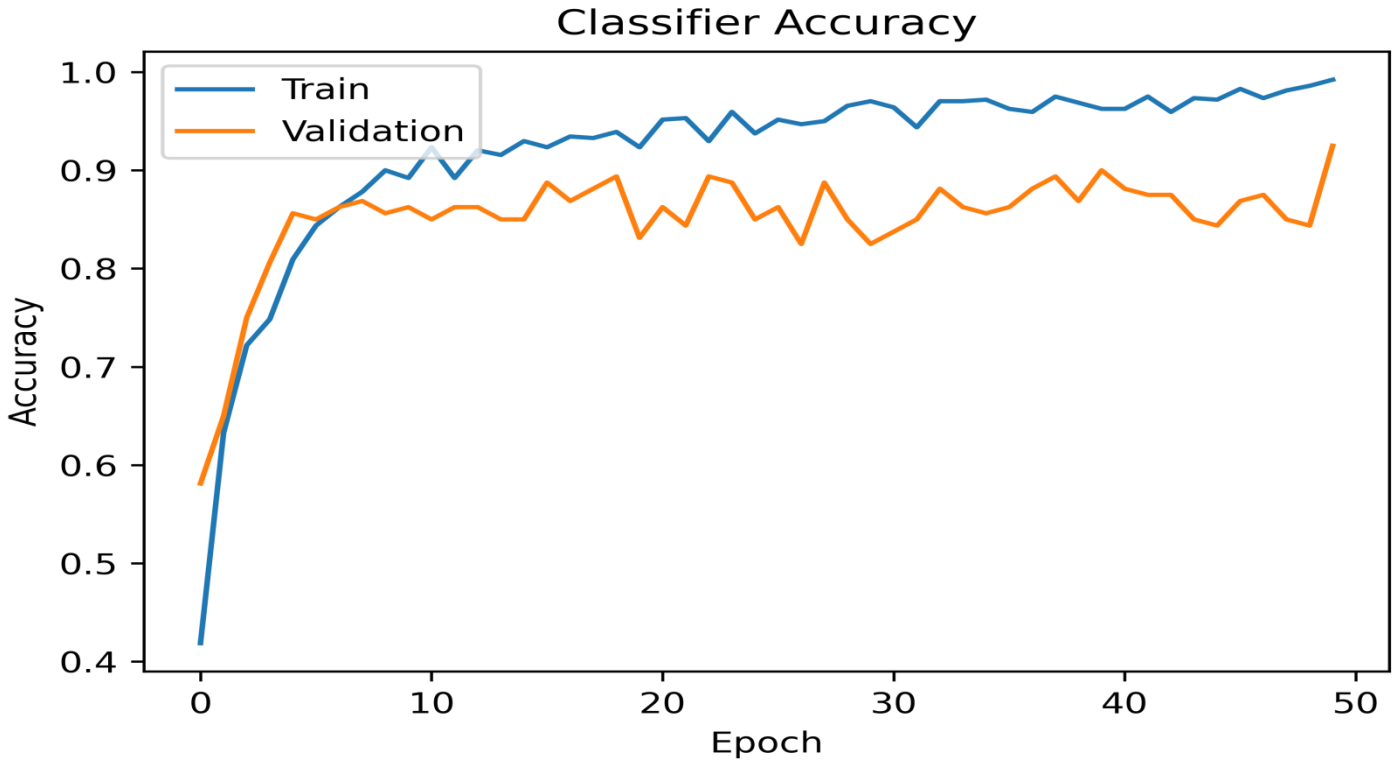
Grape Leaf Data Set - 2

Leaf Disease	Total No. of Images	No. of Training Images	No. of Testing Images
Black Rot	250	200	50
Black Measles	250	200	50
Leaf Blight	250	200	50
Healthy	250	200	50
Total	**1000**	**800**	**200**

Training Dataset

Testing Dataset

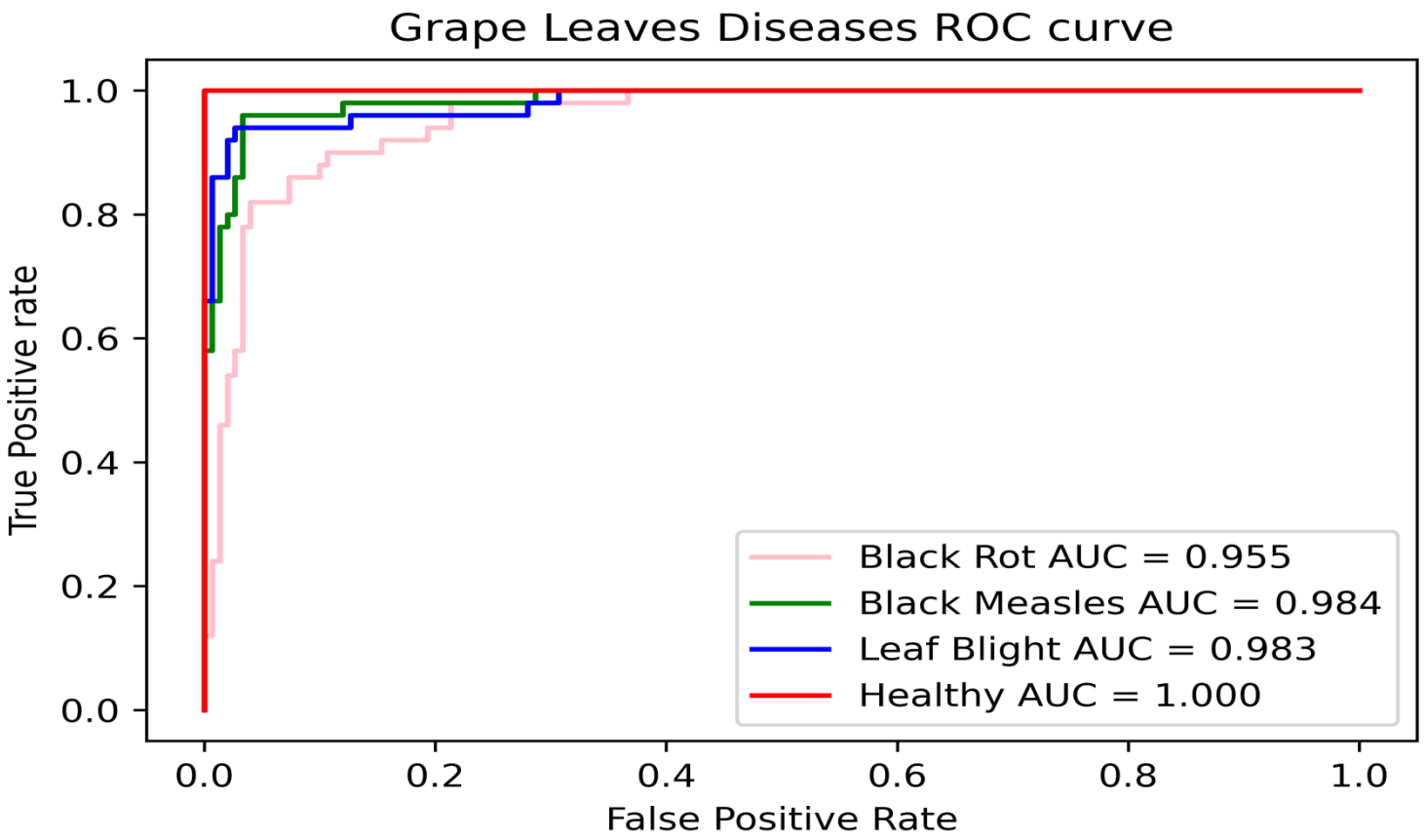
Experimental Results 2



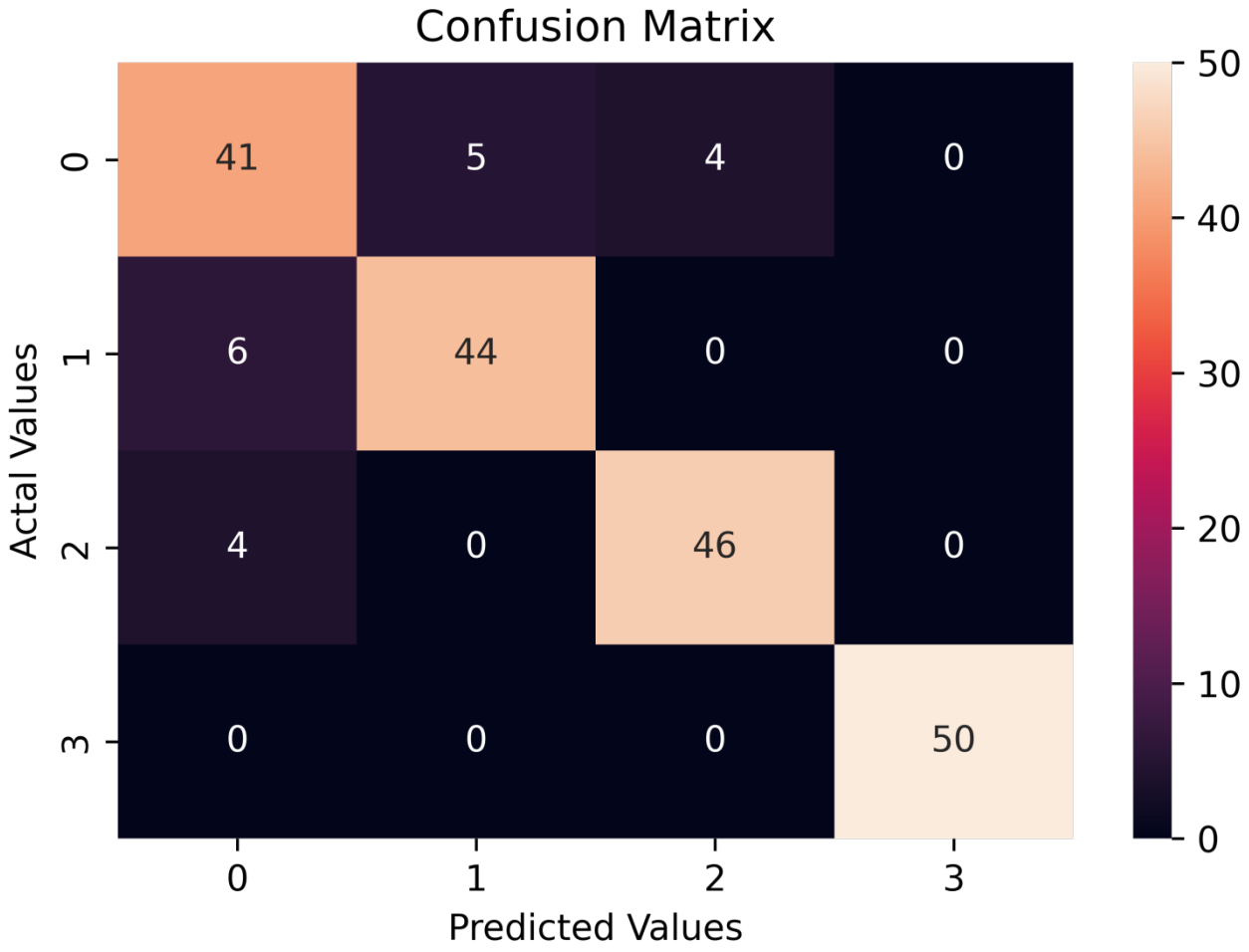
Training : 99 %

Testing : 90.5%

ROC Curve

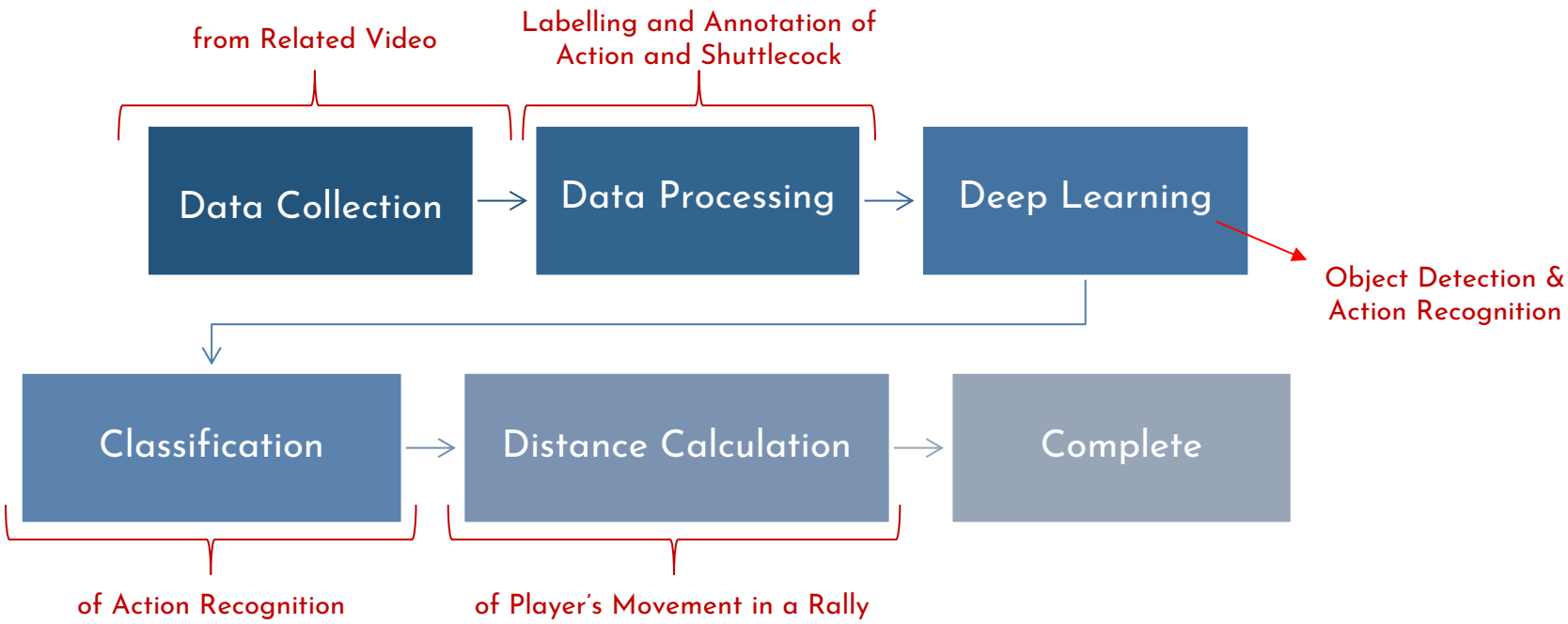

Grape Leaves Diseases ROC curve
True Positive rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Black Rot AUC = 0.955
Black Measles AUC = 0.984
Leaf Blight AUC = 0.983
Healthy AUC = 1.000

Confusion Matrix



Analysis of Badminton Match Between Lin Dan and Lee Chong Wei Using Convolutional Neural Network

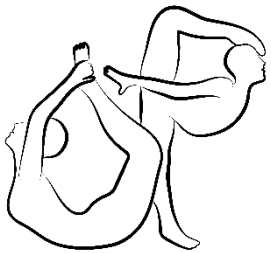
Flow Chart


from Related Video
Labelling and Annotation of Action and Shuttlecock
Data Collection
Data Processing
Deep Learning
Object Detection & Action Recognition
Classification
Distance Calculation
Complete
of Action Recognition
of Player's Movement in a Rally

Supported Library

YOLOv5

Family of **Object Detection** architectures and models pretrained on the COCO dataset.



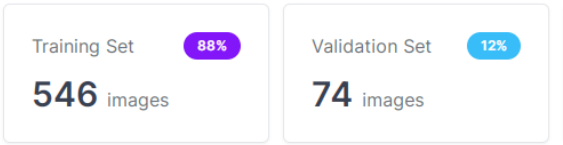
First real-time multi-person system to jointly **Detect Human Body**, hand, facial, and foot keypoints (in total 135 keypoints) on single image.

* Pre-trained model by OpenPose is used to detect human body.

Shuttlecock Detection Process

roboflow

1. Roboflow is used for Image Annotation and Labelling.

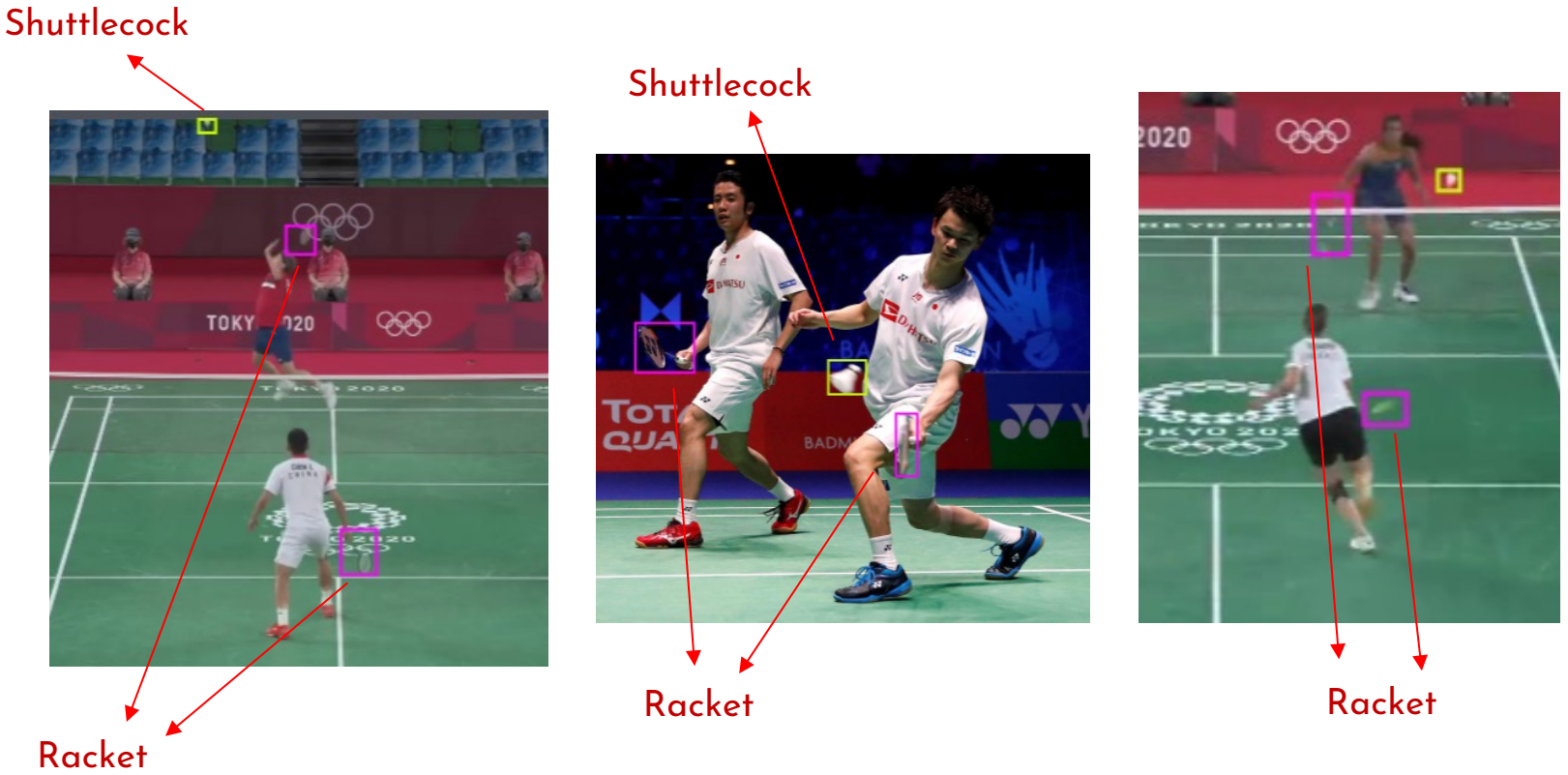


2. Total Annotate Images for Training and Validation

YOLOv5

3. Training Data using Yolov5

Shuttlecock and Racket Detection Training Sample



Shuttlecock Detection Result

Action Recognition Training Samples

Start Service

Defense

Attack

Jump Smash

Action Recognition Result

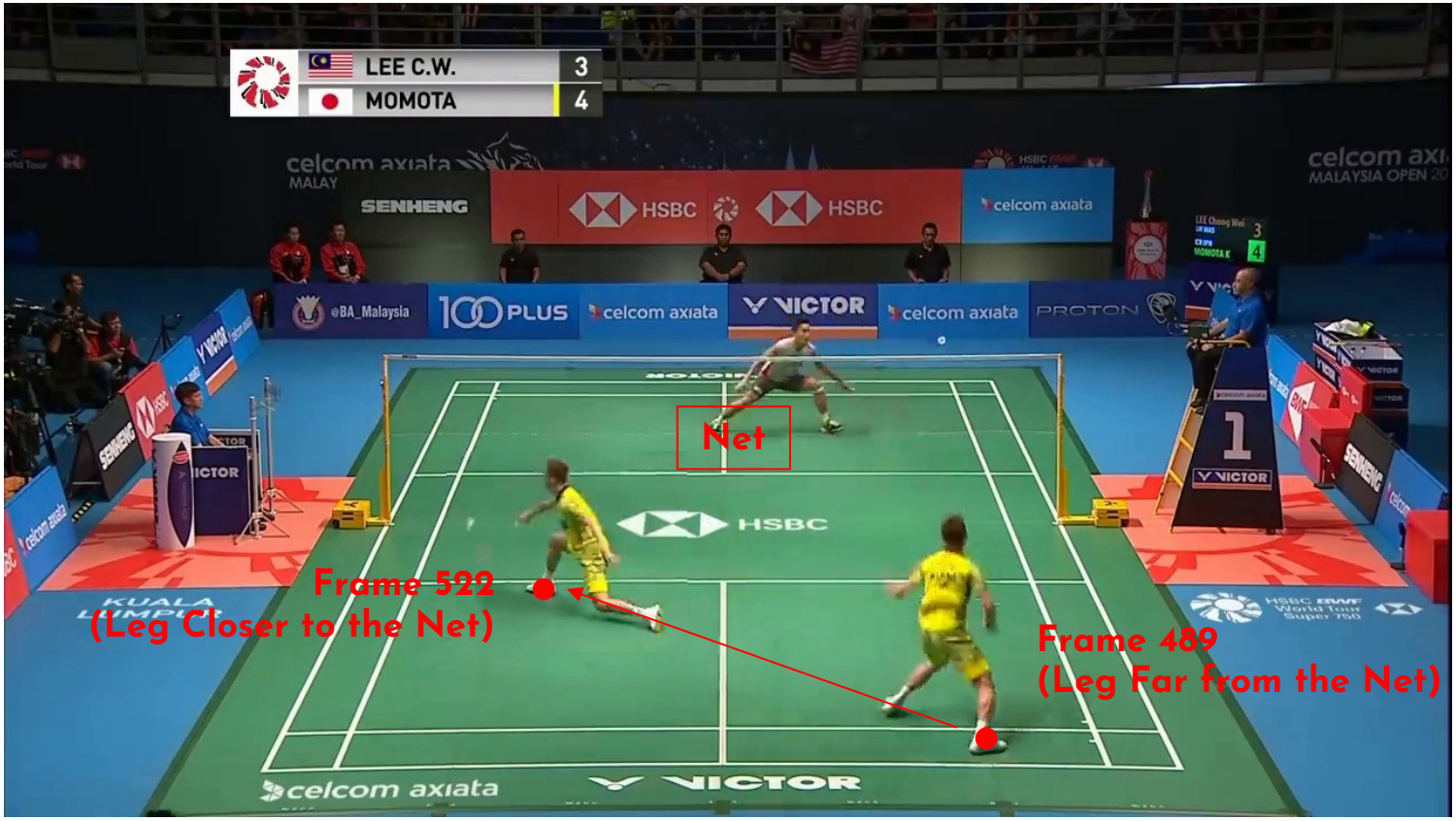
Original Image

Skeleton Detection

Classification

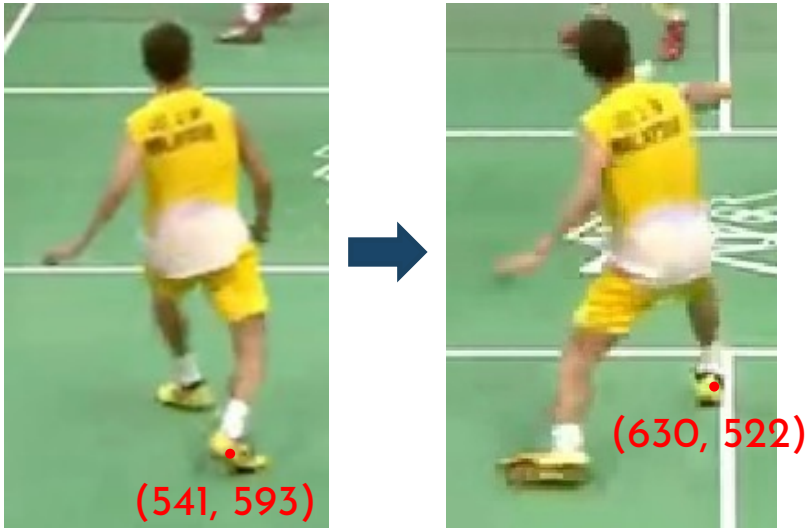
Action Recognize:
Smash

Distance Calculation

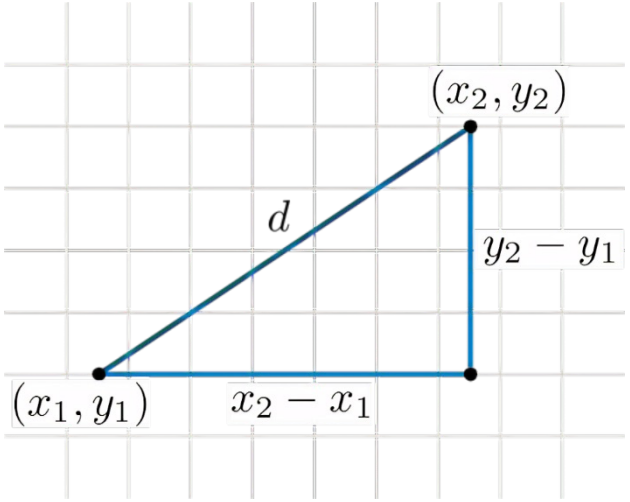


Method for Distance Calculation

Leg Movement



Euclidean Distance Formula



(First Set) Point 20-15

(Second Set) Point 10-20

Thank You

Should You Have Any Further Inquiries,

Please Do Not Hesitate To Contact Me At:

ravee58@gmail.com