

# A Multilayer Perceptron Approach for *Ficus carica* (fig) Ripening Classification

Mohd Ikmal Fitri Maruzuki<sup>1</sup>, Aisyah Sakina Shahrin<sup>2</sup>, Samsul Setumin<sup>2</sup>, Rafidah Aida Ramli<sup>4</sup>, Syahrul Fithry Senin<sup>5</sup>, Mohd Shukri Mohamed Talib<sup>6</sup>, Mohamed Syazwan Osman<sup>1,3,7\*</sup>

<sup>1</sup>Faculty of Electrical Engineering, Universiti Teknologi MARA, Cawangan Pulau Pinang, 13500 Permatang Pauh, Penang, Malaysia.

<sup>2</sup>Faculty of Chemical Engineering, Universiti Teknologi MARA, Cawangan Pulau Pinang, 13500 Permatang Pauh, Penang, Malaysia.

<sup>3</sup>Center of Excellence Geopolymer & Green Technology (CEGeoGTech), Universiti Malaysia Perlis (UniMAP), 01000 Kangar, Perlis, Malaysia.

<sup>4</sup>Faculty of Hotel & Tourism Management, Universiti Teknologi MARA, Cawangan Pulau Pinang, 13500 Permatang Pauh, Penang, Malaysia

<sup>5</sup>Faculty of Civil Engineering, Universiti Teknologi MARA, Cawangan Pulau Pinang, 13500 Permatang Pauh, Penang, Malaysia.

<sup>6</sup>EES Biotech Sdn Bhd, No 25, Jalan Guar Perahu, Taman Guar Perahu, 14400 Bukit Mertajam, Penang, Malaysia

<sup>7</sup>EMZI-UiTM Nanoparticles Colloids & Interface Industrial Research Laboratory (NANO-CORE), Universiti Teknologi MARA, Cawangan Pulau Pinang, 13500 Permatang Pauh, Penang, Malaysia

\*corresponding author: [syazwan.osman@uitm.edu.my](mailto:syazwan.osman@uitm.edu.my)

## ARTICLE HISTORY

## ABSTRACT

Received  
5 February 2021

Accepted  
4 March 2021

Available online  
30 March 2021

The ripening stage is a stage where the fruit is ready to be harvested. During ripening, pectin activity is observed to trigger parenchyma cell wall middle lamella dissolution of a fruit. Additionally, the ripening stage also affects the changing appearance of the fruit. Thus, this research aims to develop a classification model based on ANN that can predict or classify the ripening stage based on either pectin activity or fruit appearance. The study will focus specifically on *Ficus carica* (fig). To achieve the objective, the researchers developed two Multilayer Perceptron (MLP) models: figNN and pectinNN. We trained figNN using features extracted from images of figs with different ripening stages, and pectinNN with a set of the statistical value of pectin activity such as weight (W), brix of sugar (BS), extraction yield (EY), and degree of esterification (ED) from 30 figs with varying degree of ripening. From the result of this research, figNN and pectinNN can distinguish the ripening stage based on either the chemical properties or the images. Furthermore, we can also show that the image-based classification is more accurate than the pectin-based classification. For future work, the study of the correlation between pectin and image features is highly encouraged.

**Keywords:** Multilayer Perceptron (MLP); figs; fruit ripening; pectin; image analysis.

## 1. INTRODUCTION

Pectin is a heteropolysaccharide acidic structure found in the primary and middle lamellas and cell walls of land-dwelling plants [1]. Galacturonic acid is the main component of the pectin. This acid is a sugar acid that is derived from galactose. In commercial production, it is produced as a white to light brown powder extracted from fruits. This powder is usually used as a gelling agent in food like jams and jellies. It is also used as a stabilizer for fruit juices and milk

beverages, a source of dietary fibre, dessert toppings, medicines, and candies [2]. The quantity, structure, and chemical composition of pectin varies across plants, over time within the plant itself, or at different fruit's ripening stages.

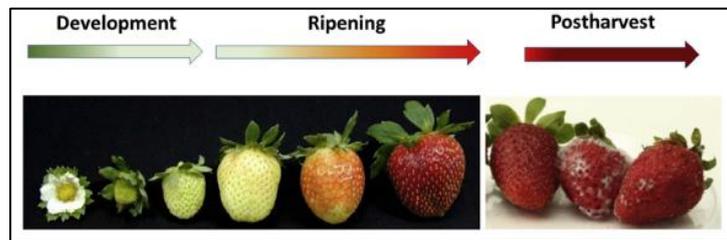


Figure 1: The ripening processes in fruits development [3]

Due to ripening enzymes such as pectin, the cells' mechanical properties lead to different levels of fruits' ripening as shown in Figure 1. There are three stages in the fruiting process and this activity would be identical in all types of fruits. The first process is known as *development*, followed by *ripening*, and lastly *postharvest*. The most desired stage for harvesting is ripening [3].

Fig is a source of nutritional substance for human populations. Fig can be freshly consumed, either unpeeled or not. Pectin activity is observed to trigger parenchyma cell wall middle lamella dissolution of a fruit [4]. Hence, many figs are commonly lost to spoilage every year due to pectin activity in fruit [5]. Therefore, it is imperative to determine the ripening stage to avoid further waste. However, to the best of our knowledge, no comprehensive work has been performed to utilise pectin activity and image analysis to predict the various fruit ripening stages using Artificial Neural Network (ANN). Thus, this research's main objective is to classify fruit ripeness based on pectin activity and image analysis. In this paper, we propose two ANN models, namely *figNN* and *pectinNN*. The key contributions of the proposed models include:

1. Our model *figNN* classify the ripening level based on image analysis of the fruit, i.e., figs skin colour. These figs skin colour can be obtained by extracting the colour from a static image captured by a camera.
2. Our model *pectinNN* classify the ripening level based on the pectin activity of the fruit, i.e., the weight, brix sugar level, extraction yield, and degree of esterification.

It should be noted here that, to the best of our knowledge, this work is among the first attempt to run an image and pectin into ANN to realize ripening classification for fruit, notably for fig. We will also discuss how the choice of the number of hidden nodes will affect classification performance. The paper is organized as follows: Section 2 covers common theoretical background for neural network algorithms. Section 3 present the methodology of the study. Section 4 presents the results and discussions on both models from various perspectives. The final Section 5 concludes the work and suggest possible future work.

## 2. THEORETICAL BACKGROUND

ANN is a form of artificial intelligence that possesses the ability to learn from features or data. The use of ANN is currently wide-spreading through the applications of pattern recognition, control, estimation, and classification [6, 7, 8, 9]. In this work, our choice of models is based

on Multilayer Perceptron (MLP) network, a feed-forward ANN class. The structure of an MLP network consists of three categories of layers: an input layer, hidden layer, output layer. The network structures often designed with a nonlinear activation function to provide nonlinear behaviour, which is crucial in performing the applications [10]. Figure 2 shows a typical model of an MLP network.

The ability to learn in MLP is partly attributed to the back-propagation (BP) method [10]. This training algorithm is based on error minimization at the output layer. One factor that can improve the learning ability of MLP is the number of hidden nodes in the hidden layer.

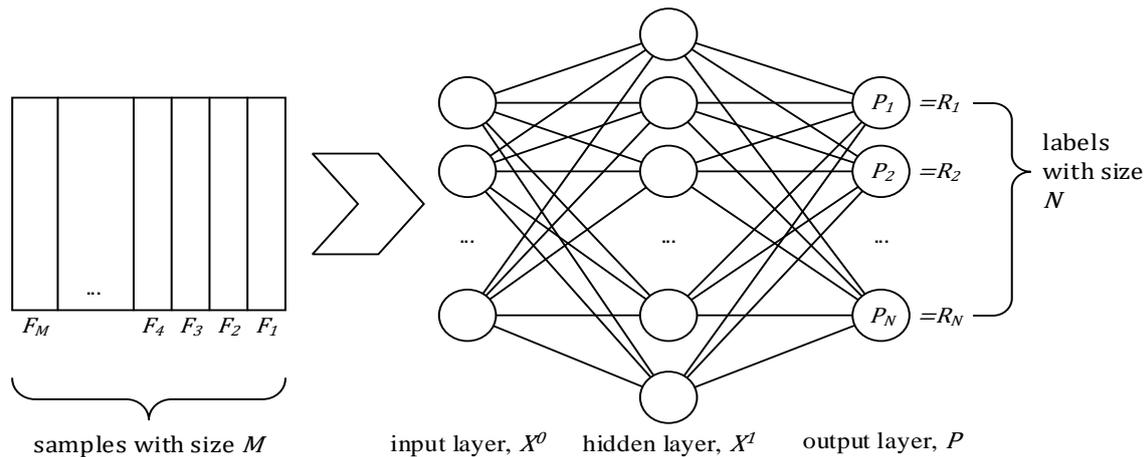


Figure 2: A Hypothetical Example of a Multilayer Perceptron Network

### 3. METHODOLOGY

The *figNN* and *pectinNN* proposed in this work are evaluated based on the two case studies that include:

1. classification of fig using skin colour of the fig; and
2. classification of fig using pectin activity of the fruit.

All experiments are carried out using both *figNN* and *pectinNN* to understand better how well our proposed models are compatible in predicting the figs ripening behaviours. The procedure to obtain those data are outlined in the following subsections.

#### 3.1 Experimental data preparation

The dataset is the first case study that comes from the skin colour of the figs. A camera captures 30 sets of figs images from a predetermined distance. There are seven variations in a single set, which include different angles (left, right, top, bottom, front, back) and cross-section of the fruit. Thus, a total of  $30 \times 7$  images are taken for colour feature extraction, with each image has a size of  $4000 \times 3000$ . Then, to extract the colour features from the fig images, *HSV* (Hue, Saturation, Value) model is employed. This model is said to be robust to the influence of light intensity from the environment. The features are extracted by computing the colour histogram of each channel into a dedicated number of bins. The numbers of bins are 15, 5, and 5 for the *H*, *S*, and *V*, respectively. These features are then reshaped to become a 1D feature vector  $F_m$  with a size of 375. In the case where the total number of variations are seven, the feature vectors

are concatenated to build a single feature vector with 2625 dimensions as illustrated in Figure 3.

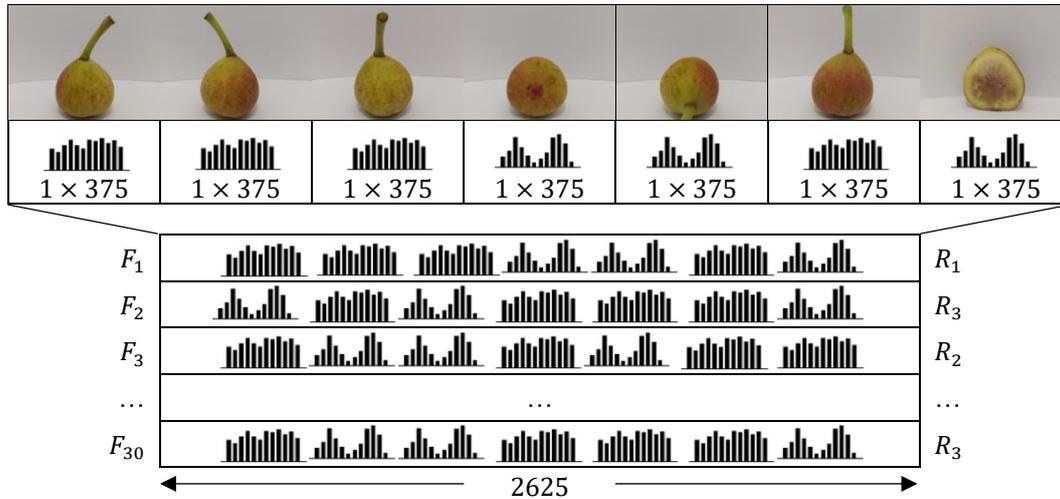


Figure 3: Example of extracted feature vectors from 30 figs colour images

There is a total of  $30 \times 2625$  feature vectors in the dataset. Every 2625 feature vector is labelled to ripening level  $R_1$ ,  $R_2$ , and  $R_3$  accordingly, where  $R_1$  is *development*,  $R_2$  is *ripening* and  $R_3$  is *postharvest*. The only pre-processing technique to be applied to the dataset is scalarisation to the range of 0~1, which requires the computation of the Equation (1):

$$\mathbf{X}_{norm} = \frac{\mathbf{X}_{HSV}}{H \times W} \quad (1)$$

where  $\mathbf{X}_{HSV}$  is the value of a single feature vector,  $H$  and  $W$  are the height and width of the image, respectively. After the scalarized feature vectors  $\mathbf{X}_{norm}$  are obtained from the fig's images, the vectors are then fed into *figNN* for the classification task.

In the second case study, the dataset consists of a set of weight ( $W$ ), brix sugar level ( $BS$ ), extraction yield ( $Y$ ), and degree of esterification ( $DE$ ) from 30 figs with varying ripening stages. There is a total of 30 sets. Table 1 shows an example of data collected from 30 figs. Similar to the first dataset, each set is labelled to ripening level  $R_1$ ,  $R_2$ , and  $R_3$  accordingly. The procedures to acquire the dataset are as follows:

Table 1: Sample data of weight ( $W$ ), brix sugar level ( $BS$ ), extraction yield ( $Y$ ), degree of esterification ( $DE$ ), and ripening label ( $R$ ) from multiple figs

W (g)	BS (%)	Y (%)	DE (%)	R
48	5.10	2.58	55.50	$R_1$
33	6.50	2.19	39.21	$R_1$
71	6.70	4.52	10.30	$R_2$
69	6.30	8.43	13.72	$R_2$
78	6.90	26.19	23.43	$R_3$
82	8.60	27.77	31.93	$R_3$
...	...	...	...	...

### 3.1.1 Brix Sugar Level of figs Method

First, a fig is measured by weighing scale at the beginning of the experiment ( $W, g$ ). The fruit is cut into small pieces and crushed by either a lab blender or mortar and pestle. Then, the pulp or seeds are removed by filtering the solution through a filter paper to a beaker.

Next, we clean the Brix refractometer (Kern, Germany) carefully with a soft cloth. We use a pipette (Eppendorf, Germany) to place a few drops of the filtered solution onto the prism. We gently close the cover to avoid producing bubbles. Then, we observe the scale through the eyepiece while pointing directly to a light source ( $BS, \%$ ).

### 3.1.2 Extraction yield (Extraction method)

First, a fig is peeled by a knife. We dried off the fig's skin in an oven at  $50\text{ }^\circ\text{C}$  for 36 hours and grounded the dried skins in a coffee blender to produce skin powder ( $W_{dried}, g$ ). We keep the powder figs in the dark air-tight containers at ambient temperature ( $24 \pm 1\text{ }^\circ\text{C}$ ).

For the extraction,  $3\text{ g}$  of powdered figs ( $W_{powder}, g$ ) is mixed with  $50\text{ mL}$  of distilled water acidified with citric acid ( $pH = 2.0$ ) into the  $100\text{ mL}$  beakers. Then, the mixture is heated at  $90\text{ }^\circ\text{C}$  for 60 minutes in a water bath. We immediately filter the heated solution through a muslin cloth before performing centrifuge at  $5350\text{ } \times g$  for 15 min to obtain a clear solution. Later, we cool the solution to  $24 \pm 1\text{ }^\circ\text{C}$  for 1.5 h, and two volumes of ethanol are added into the solution. We repeat the centrifuge process at  $5350\text{ } \times g$  for 60 min. Then, we dry off the solution in an oven at  $50 \pm 1\text{ }^\circ\text{C}$  for 16 h and measure the dried solution weight ( $W_{pectin}, g$ ). Lastly, we calculate the extraction yield ( $Y, \%$ ) using the Equation (2):

$$Y = \frac{W_{dried} - W_{pectin}}{W_{dried} - W_{powder}} \times 100 \quad (2)$$

### 3.1.4 Degree of esterification (Titration method)

First, we add  $3\text{ mL}$  of ethanol (96%) to dried figs pectin and dissolve the mixture with  $20\text{ mL}$  of deionized water. We stir the solution at under  $150\text{ rpm}$ . Then, we add a few drops of phenolphthalein reagent to the solution before titrating the solution with the volume ( $V_1, \text{ mL}$ ) of  $0.1\text{ N}$  NaOH. Next, we add  $10\text{ mL}$  of  $0.1\text{ N}$  NaOH to the solution and stir for 15 minutes. We repeat the stirring process while adding  $0.1\text{ N}$  HCl is added into the solution until the pink colour dissipates. Lastly, we titrate the solution with volume ( $V_2, \text{ mL}$ ) of  $0.1\text{ N}$  NaOH until pink colour appears. We measure the degree of esterification ( $DE, \%$ ) by using the Equation (3):

$$DE = \frac{V_2}{V_1 + V_2} \times 100 \quad (3)$$

## 3.2 MLP Configurations

ANN models developed in our experimental work are the two MLP, namely *figNN* and *pectinNN*. Table 2 provides the details of both models. Exponential linear unit (*elu*) activation function from the Equation (4) is applied to the outputs of all hidden layers due to their superior performance, as suggested by many previous works [11]:

$$\mathbf{X}^L = \begin{cases} \mathbf{X}^{L-1} & \text{if } \mathbf{X}^{L-1} > 0 \\ a \times \exp(\mathbf{X}^{L-1} - 1) & \text{else} \end{cases} \quad (4)$$

where  $L$  is a current layer,  $L - 1$  is a previous layer and  $\mathbf{X}^{L-1}$  is the input from the previous layer and  $a$  is a trainable hyperparameter.

Finally, a softmax layer at the output layer is responsible for classifying the input patterns into several predefined labels  $P_n$  using the Equation (5):

$$P_n = \exp(X_n^{L-1}) \left\{ \sum \exp(X_n^{L-1}) \right\}^{-1} \quad (5)$$

Where  $N$  is the number of outputs at  $P$  [12].

Table 2: *figNN* and *pectinNN* model for the fig ripening classification

Model	Layer	Type	Total Neurons	Activation function
<i>figNN</i>	X0	Input	2625	<i>elu</i>
	X1	Fully Connected	<i>varied</i> (Table 3)	<i>elu</i>
	P	Output	3	<i>softmax</i>
<i>pectinNN</i>	X0	Input	4	<i>elu</i>
	X1	Fully Connected	<i>varied</i> (Table 3)	<i>elu</i>
	P	Output	3	<i>softmax</i>

### 3.3 Training Methodology

Table 3 depicts the default hyperparameter for our training. We divided both datasets were divided to 50% training, 10% validation and 40% in the test. In this work, all training procedures and MLP models were developed from scratch. We trained our models using the  $k$ -fold cross-validation method with  $k = 3$ .  $k$ -fold cross-validation method is useful in avoiding classification overfitting problem [13]. The models are trained for 1000 epochs within each  $k$ . At the beginning of every training epoch, the training samples are shuffled to ensure that they can learn robustly and produce better learning performance.

Table 3: Training configuration

CPU	Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz
GPU	GeForce GTX 1050 4GB
RAM	16GB
Framework	TensorFlow, Keras
OS	Windows 10
Dataset & model	skin colour ( <i>figNN</i> ), pectin ( <i>pectinNN</i> )
Training epoch	1000
$k$ -fold	3
Optimizer	Adam: $\beta_1 = 0.9, \beta_2 = 0.99$
Initial learning rate	0.001
Hidden nodes setting	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Glorot normalisation is performed in the weight initialisation process at the start of every new training repetition [14]. As for the learning algorithm, all models use Adam as an optimiser [15]. We will also discuss how the choice of the number of hidden nodes will affect the classification performance.

We apply accuracy and Random Mean Square Error (RMSE) to evaluate MLP models' fitness. Accuracy is computed using the Equation (6):

$$accuracy = \left( \frac{1}{M} \sum^M D_m \right) \times 100 \quad (6)$$

$$D_m = \begin{cases} 1 & \text{if } \mathbf{P}_m = \mathbf{R}_m \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Where  $M$  is the total samples in the training dataset. At  $m^{th}$  sample,  $\mathbf{P}_m$  is a set of predicted ripening levels and  $\mathbf{R}_m$  is the set of actual ripening levels. Higher accuracy means a higher ability to determine the label correctly. RMSE is computed using the Equation (8):

$$RMSE = \sqrt{\frac{1}{M \times N} \sum^M \sum^N (P_{nm} - R_{nm})^2} \quad (8)$$

Where  $N$  is the number of labels at the output layer. In this case,  $N=3$  corresponds to the three ripening levels. At  $m^{th}$  sample and  $n^{th}$  label,  $P_{nm}$  is predicted ripening levels while  $R_{nm}$  is actual ripening levels.

#### 4. RESULT AND DISCUSSION

We examined the proposed models' learning performance by studying their classification accuracy and RMSE on validation and test dataset. Table 4a and Table 4b show the results of *figNN*. Table 5a and Table 5b are results for *pectinNN*, with various hidden nodes setting within 1000 training epochs. All average values are calculated from  $k$ -fold cross-validation results. Figure 4 shows the overall performance of both models.

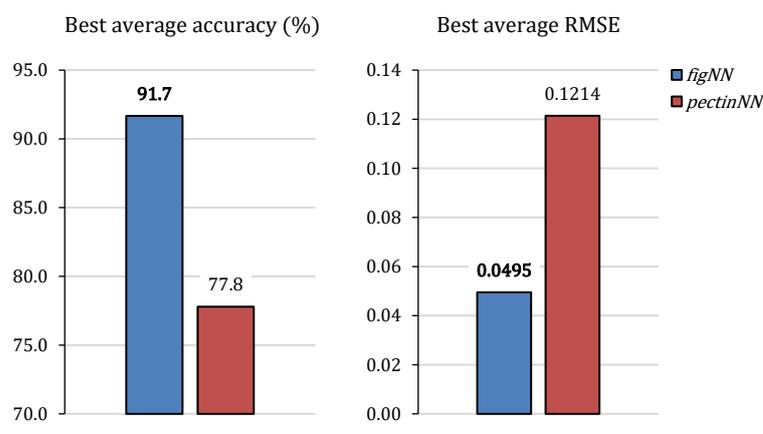


Figure 4: Overall performance of *figNN* vs *pectinNN*

Table 4a: *figNN* Validation and Test Accuracy after 1000 epoch on skin colour dataset

Hidden nodes	validation accuracy (%)				test accuracy (%)			
	<i>k</i> =1	<i>k</i> =2	<i>k</i> =3	Average	<i>k</i> =1	<i>k</i> =2	<i>k</i> =3	Average
1	100.00	100.00	83.33	94.44	83.33	66.67	75.00	75.00
2	100.00	100.00	83.33	94.44	91.67	83.33	91.67	88.89
3	100.00	100.00	83.33	94.44	91.67	91.67	91.67	91.67
4	100.00	100.00	83.33	94.44	91.67	91.67	91.67	91.67
5	100.00	100.00	83.33	94.44	91.67	91.67	91.67	91.67
6	100.00	100.00	83.33	94.44	91.67	91.67	91.67	91.67
7	100.00	100.00	83.33	94.44	91.67	91.67	91.67	91.67
8	100.00	100.00	83.33	94.44	91.67	91.67	91.67	91.67
9	100.00	100.00	83.33	94.44	91.67	91.67	91.67	91.67
10	100.00	100.00	83.33	94.44	91.67	91.67	91.67	91.67

Table 4b: *figNN* Validation and Test RMSE after 1000 epoch on skin colour dataset

Hidden nodes	validation RMSE				test RMSE			
	<i>k</i> =1	<i>k</i> =2	<i>k</i> =3	Average	<i>k</i> =1	<i>k</i> =2	<i>k</i> =3	Average
1	0.0026	0.0328	0.0957	0.0437	0.0909	0.1486	0.1056	0.1150
2	0.0000	0.0027	0.0557	0.0195	0.0554	0.0697	0.0481	0.0577
3	0.0000	0.0000	0.0611	0.0204	0.0556	0.0543	0.0385	0.0495
4	0.0000	0.0000	0.0604	0.0201	0.0600	0.0560	0.0410	0.0524
5	0.0000	0.0000	0.0550	0.0183	0.0557	0.0572	0.0461	0.0530
6	0.0000	0.0000	0.0531	0.0177	0.0562	0.0553	0.0503	0.0540
7	0.0000	0.0000	0.0604	0.0201	0.0561	0.0578	0.0427	0.0522
8	0.0000	0.0000	0.0615	0.0205	0.0556	0.0606	0.0480	0.0547
9	0.0000	0.0000	0.0604	0.0201	0.0559	0.0577	0.0421	0.0519
10	0.0000	0.0000	0.0614	0.0205	0.0558	0.0595	0.0439	0.0530

The proposed *figNN* and *pectinNN* both clearly show high average accuracy in classifying the ripening level, at 91.67% and 77.78% respectively. This demonstrates the effectiveness of the training. Moreover, both models show very low average RMSE, with *figNN* at 0.0495 and *pectinNN* at 0.1214. Figure 4 also shows a huge performance gap between *figNN* and *pectinNN*. This is owing to *figNN* is developed and trained with a high number of *HSV* features, whereas *pectinNN* are trained on limited features of pectin chemical properties.

Table 5a: *pectinNN* Validation and Test Accuracy after 1000 epoch

Hidden nodes	validation accuracy (%)				test accuracy (%)			
	<i>k</i> =1	<i>k</i> =2	<i>k</i> =3	Average	<i>k</i> =1	<i>k</i> =2	<i>k</i> =3	Average
1	50.00	33.33	66.67	50.00	50.00	25.00	16.67	30.56
2	100.00	66.67	33.33	66.67	41.67	66.67	33.33	47.22
3	83.33	50.00	33.33	55.56	41.67	41.67	25.00	36.11
4	83.33	100.00	33.33	72.22	83.33	58.33	75.00	72.22
5	83.33	66.67	100.00	83.33	58.33	41.67	83.33	61.11
6	66.67	66.67	66.67	66.67	75.00	66.67	50.00	63.89
7	83.33	66.67	50.00	66.67	83.33	33.33	33.33	50.00
8	66.67	50.00	33.33	50.00	50.00	66.67	66.67	61.11
9	66.67	66.67	33.33	55.56	66.67	66.67	66.67	66.67
10	83.33	66.67	50.00	66.67	83.33	83.33	66.67	77.78

Table 5b: *pectinNN* Validation and Test RMSE after 1000 epoch

Hidden nodes	validation RMSE				test RMSE			
	$k=1$	$k=2$	$k=3$	Average	$k=1$	$k=2$	$k=3$	Average
1	0.1895	0.2270	0.1146	0.1770	0.2903	0.2364	0.2966	0.2745
2	0.0168	0.1383	0.2720	0.1424	0.3842	0.1608	0.2622	0.2691
3	0.1061	0.2532	0.2268	0.1954	0.2089	0.2708	0.2361	0.2386
4	0.1156	0.0464	0.3594	0.1738	0.0941	0.1378	0.2450	0.1589
5	0.0930	0.2465	0.0447	0.1280	0.1443	0.3733	0.1007	0.2061
6	0.1262	0.1377	0.1452	0.1364	0.1526	0.1602	0.3084	0.2071
7	0.1117	0.1458	0.1796	0.1457	0.1029	0.3382	0.3548	0.2653
8	0.1572	0.2987	0.1717	0.2092	0.2828	0.2015	0.1340	0.2061
9	0.1663	0.1939	0.2951	0.2184	0.1278	0.1934	0.1661	0.1624
10	0.1009	0.1582	0.3350	0.1980	0.0659	0.1101	0.1884	0.1214

The number of hidden nodes in the hidden layer contributes to the overall accuracy and RMSE as well. However, the effects are different in both models. By referring to *figNN* results in Table 4a and Table 4b, the increasing number of hidden nodes do not necessarily mean that the accuracy and RMSE performance will be improved significantly. For instance, as the number of hidden nodes is getting higher, the *figNN* accuracy becomes stabilized. Still, RMSE becomes stagnant, suggesting that a high number of hidden neurons will not improve network performance. However, the opposite effect can be observed in *pectinNN* results. Based on *pectinNN* results in Table 5a and Table 5b, the impact of good performance is noticeable as we increased the number of hidden nodes were increased. This provides solid evidence that a higher number of nodes in *pectinNN* tend to result in better network performance.

We examined the impact of applying image features vs pectin chemical properties to MLP input. Figure 5 and Figure 6 illustrates the learning curves of *figNN* and *pectinNN* during training and validation. Overall, *figNN* converges better than *pectinNN*. Figure 5a vs Figure 6a gives a clear indication of how well the *figNN* performs on the training and validation set even within 100 epochs only. Referring to Figure 5b vs Figure 6b, the steep RMSE curve in *figNN* training and validation signifies an excellent generalisation performance of the proposed model. Given the same figs are used for both datasets, it is essential to note that the fig skin colour-based approaches are better than the pectin activity-based approach. This demonstrates the superiority of image-based solution in solving the ripening classification problem.

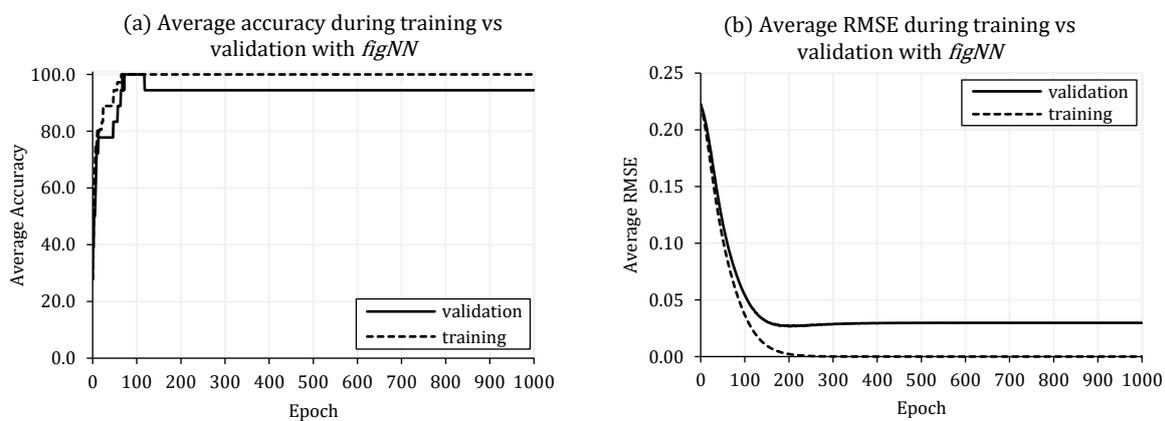


Figure 5: Average accuracy and average RMSE of *figNN* in training set and validation set during 1000 epoch

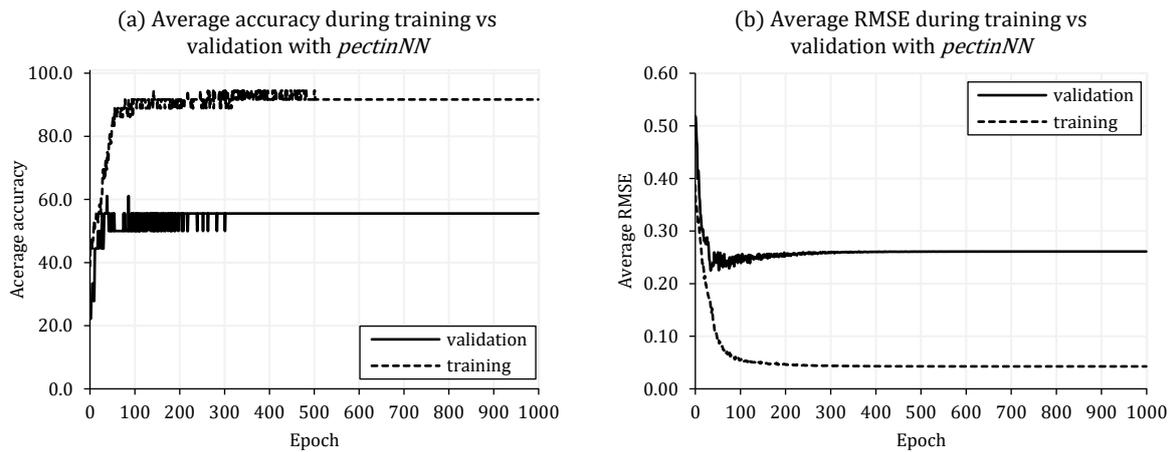


Figure 6: Average accuracy and average RMSE of *pectinNN* in training set and validation set during 1000 epoch

## 5. CONCLUSION

In conclusion, we have described the recent work on the classification of figs' ripening level. In this paper, we developed two MLP models that took the image and pectin properties of a fruit, namely fig, as its input and classify the fruit ripening level.

The first proposed MLP model, *figNN* can classifies effectively the ripening level based on figs skin colour. Our second MLP model, *pectinNN*, can classify successfully the ripening level based on the fruit's pectin activity. In the same experiment, we have also discussed the relationship between the number of hidden nodes and our models' learning capability. In summary, the proposed MLP learning ability can be affected by the number of hidden nodes.

For future work, we suggest using Machine Learning (ML) method such as Deep Neural Networks (DNN) to correlate pectin activity with image features of figs. Over the last decade, significant developments of the ML approaches have led to many advances in various science and engineering, but the application to predict pectin activity based on image features is scarce.

## REFERENCES

- [1] S. M. T. Gharibzahedi, B. Smith and Y. Guo, "Ultrasound-microwave assisted extraction of pectin from fig (*Ficus carica* L.) skin: Optimization, characterization and bioactivity," *Carbohydrate Polymers*, vol. 222, pp. 114992, 2019.
- [2] S. M. T. Gharibzahedi, B. Smith and Y. Guo, "Pectin extraction from common fig skin by different methods: The physicochemical, rheological, functional, and structural evaluations," *International Journal of Biological Macromolecules*, vol. 136, pp. 275–283, 2019.
- [3] P. M. A. Toivonen and D. A. Brummell, "Biochemical bases of appearance and texture changes in fresh-cut fruit and vegetables," *Postharvest Biology and Technology*, vol. 48, pp. 1–14, 2008.
- [4] C. Paniagua, S. Posé, V. J. Morris, A. R. Kirby, M. A. Quesada and J. A. Mercado, "Fruit softening and pectin disassembly: an overview of nanostructural pectin modifications assessed by atomic force microscopy," *Annals of Botany*, vol. 114, pp. 1375–1383, 2014.
- [5] S. Posé, C. Paniagua, A. J. Matas, A. P. Gunning, V. J. Morris, M. A. Quesada and J. A. Mercado, "A nanostructural view of the cell wall disassembly process during fruit ripening and postharvest

- storage by atomic force microscopy," *Trends in Food Science & Technology*, vol. 87, pp. 47–58, 2019.
- [6] H. Huang, R. He, Z. Sun and T. Tan, "Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution," *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [7] K. Zhang, W. Zuo, S. Gu and L. Zhang, "Learning deep CNN denoiser prior for image restoration," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [8] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, pp. 3142–3155, 2017.
- [9] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao and S. Yan, "HCP: A flexible CNN framework for multi-label image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 1901–1907, 2015.
- [10] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, pp. 2627–2636, 1998.
- [11] D.-A. Clevert, T. Unterthiner and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.
- [12] I. Goodfellow, Y. Bengio and A. Courville, 6.2. 2.3 *Softmax Units for Multinoulli Output Distributions*. *Deep Learning*, MIT Press, 2016.
- [13] K. Srinivasan, A. K. Cherukuri, D. R. Vincent, A. Garg and B.-Y. Chen, "An efficient implementation of artificial neural networks with K-fold cross-validation for process optimization," *Journal of Internet Technology*, vol. 20, pp. 1213–1225, 2019.
- [14] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.
- [15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.