Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 1 (2023) 168 - 178 ISSN Media Electronic: 2580-0760

Image Classification of Vegetable Quality using Support Vector Machine based on Convolutional Neural Network

Hanny Nurrani[1], Andi Kurniawan Nugroho[2*], Sri Heranurweni[3]
[1,2,3] Electrical Department-Faculty of Engineering, Universitas Semarang, Semarang, Indonesia
[1]hannynurrani6qq@gmail.com, [2]andikn@usm.ac.id, [3]heranur@usm.ac.id

Abstract

As part of an effort to develop intelligent agriculture, new methods for enhancing the quality of vegetables are being continually developed. In recent years, the Convolutional Neural Network (CNN) has shown to be the most successful and extensively used approach for identifying the quality of pre-trained vegetables. However, this method is time-consuming due to the scarcity of truly large, significant datasets. Using a pre-trained CNN model as a feature extractor is a straightforward method for utilizing CNNs' capabilities without investing time in training. While, Support Vector Machine (SVM excels at processing data with tiny dimensions and significantly larger instances. SVM more accurately classifies the flatten/vector feature supplied by the CNN fully connected layer with small dimensions. In addition, implementing Data Augmentation (DA) and Weighted Class (WC) for data variety and class imbalance reduction can improve CNN-SVM performance. The research results show highest accuracy during training always achieves 100% across all experimental options. With an average accuracy of 69.66% in the testing process and 92.51% in the prediction process for all data, the experimental findings demonstrate that CNN-SVM outperforms CNN in terms of accuracy performance in all possible experiments, with or without WC and or DA approach.

Keywords: *vegetable quality; image classification; convolutional neural network; support vector machine; feature extration.*

1. Introduction

This world has a variety of plants and wildlife. There are around 374000 plant species, 308312 of which are vascular plants and 295383 vegetable plants (angiosperms; monocots: 74273; eudicot species: 210008) [1]. With 416 families, 13164 genera, and 295,383 species, vegetable plants are the most varied group of land plants in the world [1]. As part of an effort to develop intelligent agriculture, new methods for enhancing the quality of vegetables are being continually developed. For example, the use of machine learning techniques in computer vision applications to identify images of vegetable quality.

Machine learning is able to accurately detect visual patterns, making it easier to manage quality issues of food goods such as vegetables and fruit and preventing food contamination [2], and time efficiency. However, classifying vegetable quality using machine learning is a complicated topic that requires additional investigation [3]-[5] due to similarities across classes and irregular intra-class characteristics [3]. In addition, because of the vast range of disciplines, the selection of suitable data collecting and feature representation approaches is particularly crucial [3], as well as the foreground and background colors which are very diverse and sometimes very similar. Consequently, this presents a challenge for the development of machine learning methods that can identify or forecast the quality of vegetable items with greater accuracy.

Based on these problems, this research focuses on improving the accuracy performance of the machine learning method for classifying the quality of vegetables. In recent years, several machine learning techniques have been used to identify and forecast the quality of vegetables, such as the categorization of plant leaves which demonstrated a 90% accurate Probabilistic Neural Network (PNN) [6], while using Artificial Neural Network (ANN) demonstrated 80% accuracy [7]. In the meantime, the application of ANN to the categorization of tomato quality demonstrated an accuracy of 98.50% [8], the identification of bruising Apples demonstrated a 94.94% accuracy rate [9], and the introduction of interest demonstrated an accuracy of 81.19% [10]. Decision Tree (DT) for vegetable classification demonstrated 95% accuracy [11], while

Accepted: 25-11-2022 | Received in revised: 02-02-2023 | Published: 05-02-2023

the Random Forest (RF) for predicting papaya ripeness demonstrated an accuracy of 94.7% [12]. Furthermore, K-Nearest Neighbor (KNN) for interest recognition demonstrated 90% accuracy [13], Egg quality assessment demonstrated an accuracy of 88% [14], and 1-NN for vegetable classification demonstrated 80% accuracy [15]. Support Vector Machine (SVM) for Mango scoring demonstrated 100% accuracy [16], while Grapevine detection demonstrated 97.70% accuracy [17]. Convolutional Neural Network (CNN) for vegetable recognition demonstrated 97.58% accuracy [18], the classification of fruits and vegetables demonstrated an accuracy of 95.6% [19] and 92,23% [20], the diagnosis of plant diseases demonstrated an accuracy of 99.53% [4], the classification of the type of rice demonstrated an accuracy of 99.31%, for the classification of the variety of Barley demonstrated an accuracy of 93% [21], identification of diseases on Cucumber leaves demonstrated an accuracy of 94.65% [5], vegetable classification demonstrated an accuracy of 96.5% [22], 99% [23] and 98,58% [24], for fruit classification demonstrated 98% accuracy [25], and for banana ripeness classification demonstrated 96.18% accuracy [26]. Multilayer Deep CNN (MDCNN) for fruit detection demonstrated 97.4% accuracy [27], Deep CNN (DCNN) for Cucumber disease recognition demonstrated 93.4% accuracy [28], and CNN + SVM for fruit detection demonstrated 97.50% accuracy [29].

These experimental results demonstrated that machine learning method is suitable for classifying and predicting the quality of vegetables. CNN has been demonstrated to be the best and most popular machine learning approach for classifying the quality of vegetables, as illustrated in Figure 1 and described in Table 1. This is consistent with the viewpoint expressed by Hamed et al. in his assessment of studies concerning the categorization of the quality of vegetables and fruit [3].

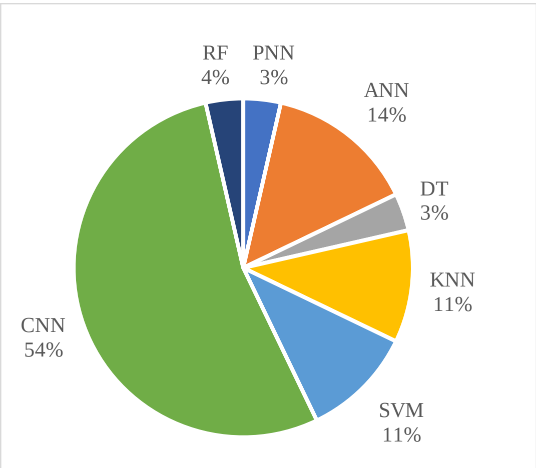


Figure 1. Constribution (in percent) of Machine Learning Methods for Classifying the Quality of Vegetables and Fruits

Table 1. Review on Classifying the Quality of Vegetables and Fruits

Year	Ref.	Accuracy	Data, Proposed Method
2007	[6]	90	32 species and 1800 leaf, PNN (classifier), PCA (feature extraction).
2008	[7]	80	1039 x 1392 pxl, ANN (classifier), RBF.
2012	[11]	95	296 images, DT (classifier), textured features.
2014	[13]	80	10 species and 150 images, KNN (classifier), Hu's Seven Moment.
2016	[18]	97.58	160 images, Explored CNN.
2016	[16]	100	200 Mango images, SVM (classifier), Multi Attribute Decision Making, Fuzzy (maturity and quality).
2017	[8]	98.50	Tomato images, ANN (classifier), RBF.
2017	[9]	94.94	Apel fruits, ANN (classifier), Grey Level Co-occurrence Matrix (GLCM), K-means (clustering segmentation).
2017	[14]	88%	50 Eggplant images, KNN (classifier), Otsu
2017	[17]	97.70	760 Grapevine images, SVM (classifier), features transform and bag of features.
2017	[19]	95.6	26 categories, CNN (classifier), VGG (pre-trained).
2018	[29]	97.50	1778 fruit images, CNN-SVM (classifier), Image region selection and improved object proposals, deformable parts model, cascade detection framework and faster RCNN.
2018	[12]	94.7	114 fruit images, RF (classifier), muturation stage.
2018	[4]	99.53	25 species and 87,848 images, CNN (classifier), VGG (pre-trained).
2018	[28]	93.4	14,208 images, DCNN (classifier), AlexNet (pre-trained).
2018	[30]	99.31	1600-4000 images, CNN (classifier), VGG and ImageNet (Pre-Trained).
2018	[10]	81.19	102 species and 8189 images, ANN (classifier), HSV (color descriptor), GLCM (texture descriptor), IM (shape descriptor).
2019	[21]	93	6 species and 60,000 images, CNN (classifier), image segmentation, extraction of individual kernels, correction of anteroposterial orientation, background removal and cropping, and resampling.
2019	[20]	92.23	13 species and 2700 images, CNN (classifier), GoogleNet (pre-trained).
2019	[5]	94.65	6 diseases and 600 images, CNN (classifier), Combining dilated convolution with global pooling using GPD, AlexNet (pre-trained).
2019	[15]	80	SAR time series data, 1NN (classifier), Dynamic Time Wrapping.
2020	[22]	96.5	CNN (classifier), VGG-M-BN (pre-trained).
2020	[25]	98	1400 images, CNN (classifier), pre-trained.
2021	[23]	99	21 images, CNN (classifier), InceptionV3 (pre-trained).
2021	[27]	97.4	6783 images, Modified CNN (classifier).
2021	[26]	96.18	436 images, CNN (classifier), MobileNet V2 (pre-trained).
2022	[24]	98.58	2300 images, CNN (classifier), DenseNet 201 (pre-trained).

DOI: https://doi.org/10.29207/resti.v7i1.4715
Creative Commons Attribution 4.0 International License (CC BY 4.0)

CNN can automatically adapt to data and prediction tasks in certain fields without the need for feature extraction using other methods [31]. CNN work for the categorization of vegetable picture quality is currently centered on the pre-trained technique. Because CNN is more adept at analyzing varied and vast data sets, it can develop a robust pattern recognition model. However, this method is dependent on the scarcity of really large, meaningful datasets [3]. Implementing Data Augmentation (DA) is another approach to resolving this issue. On both training data and test data, transformation techniques such as picture rotation, scale, translation, blur, and noise can be utilized as DA. In addition to these issues, data typically suffers from unbalanced class issues. A straightforward method such as Weighted Class (WC) can effectively address this issue.

However, CNN's training approach is highly sophisticated and hence time-consuming. Utilizing a pre-trained CNN model as a feature extractor is a simple approach to take use of the capability of CNNs without investing time in training [32]-[35]. In the meantime, it is well-known that Support Vector Machine (SVM) excels in processing small-dimensional data on a massive scale. Thus, flatten/vector characteristics generated by a fully connected layer CNN with modest dimensions are more effectively classified by SVM.

In order to classify the picture quality of Pumpkin, Eggplant, Tomato, and Carrot Vegetables using three class labels—Fresh, Wilted, and Rotten—this study will employ the WC approach for unbalanced class reduction, DA for data diversity, CNN for feature extractors, and SVM for classifier.

2. Research Method

This study employes experimental research methodologies because it examines the effect of applying the SVM method as a classification, CNN as a feature extraction, DA for data variety, and WC for reducing class imbalances on increasing the accuracy performance of machine learning methods, especially CNN in classifying the quality of vegetables. Primary data are images (jpg file) of vegetable species (Pumpkin, Eggplant, Tomato, and Carrot) collected by observation using a camera with a resolution of 416*312 pixels. 72 dpi, and 24 bit depth. While the experimentation equipment is Matlab tools.

2.1. Dataset

The vegetable quality dataset collected in supermarkets consisted of four types of vegetables, including 260 Pumpkin images, 247 Eggplant images, 259 Tomato images, and 255 Carrot images measuring 416*312 pixels, with each type of vegetable containing the three class labels Fresh, Withered, and Rotten, as depicted in Figure 2.

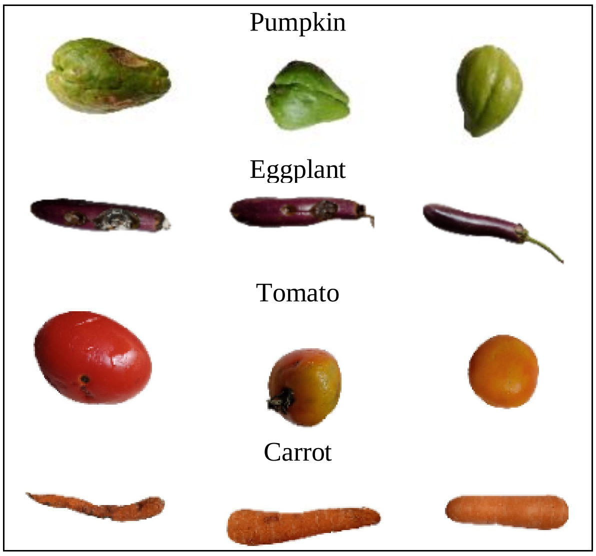


Figure 2. Frequency (in percent) of Each Class on Dataset (From Left to Right: Rotten, Withered, Fresh)

While the composition/frequency of each class of each type of vegetable can be shown in Figure 3.

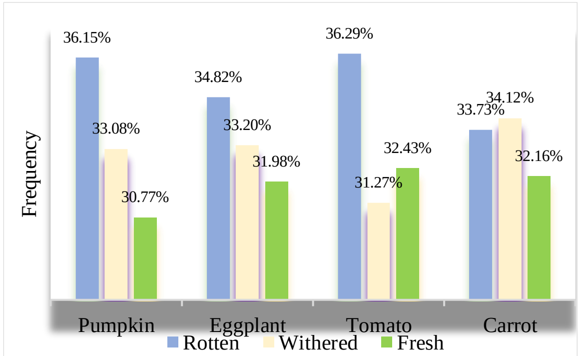


Figure 3. Frequency (in percent) of Each Class on Dataset

2.2. Proposed Method

The experiment stages are collect vegetable image data using a camera; application of the WC approach for imbalanced class reduction, namely by assigning a weight to each class based on its respective data frequency to be used in the output layer process or CNN classification; application of the HoldOut technique for random validation or data partitioning with a composition of 75% training data and 25% test data; application of DA for data diversity in training data and test data using rotation, scale, translation, blur, and noisy techniques; CNN modeling/training using input (x) and target (t) on the training data; feature extraction for RGB images on training, test, and prediction data (all data) by utilizing the full connected layer CNN model that has been trained by activating it to produce flatten/vector feature data from each training, test, and

DOI: https://doi.org/10.29207/resti.v7i1.4715
Creative Commons Attribution 4.0 International License (CC BY 4.0)

data data. those predictions; modeling/training of input (x) and target (t) SVM on training data feature vector data. The eighth stage is predicting the output (y) of the input feature vector (x) of test data and prediction data using the SVM model that has been trained; The final step is to evaluate the output (y) predicted by the SVM model with the target (t) or actual output using the confusion matrix approach to obtain accurate performance from the model.

Thus, the proposed method can be described in the form of a framework shown in Figure 4.

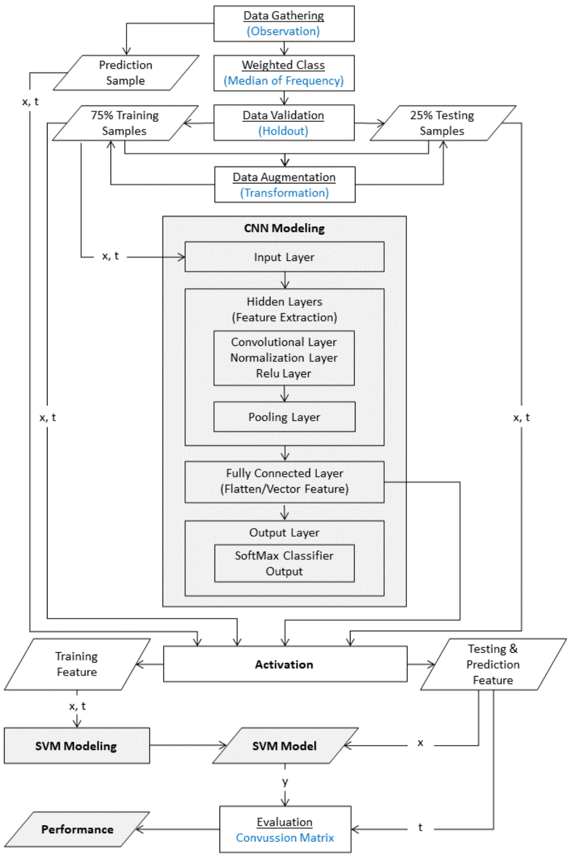


Figure 4. The Framework of Proposed Method

Based on these stages, the proposed method with various experiment options can be shown in Table 2.

Table 2. Experiment Options

Method	Modeling	WC	DA	FE	Classifier
CNN	CNN	No	No	No	CNN
CNN-WC	CNN	Yes	No	No	CNN
CNN-WC-DA1	CNN	Yes	Yes	No	CNN
CNN-WC-DA2	CNN	Yes	Yes	No	CNN
CNN-WC-DA1-SVM	CNN, SVM	Yes	Yes	CNN	SVM
CNN-WC-DA2-SVM	CNN, SVM	Yes	Yes	CNN	SVM
CNN-WC-SVM	CNN, SVM	Yes	No	CNN	SVM

2.3. Convolutional Neural Network

The Convolutional Neural Network (CNN) approach is a potent Deep Learning method and a development of the ANN method for image processing in order to eliminate redundancy, overfitting, and convergence difficulties [31]. CNN is better suited for large-scale image processing because the design of the input and hidden layers consists of layers of neurons organized in three dimensions, namely width, height, and depth (see Figure 5) [36]. With the convolution method, the size of neurons in layers may be decreased, lowering computational complexity and redundancy [36]. Through CNN's convolution filter, feature extraction can be carried out and the results, which is an additional advantage of CNN [36]. In addition, the CNN filter parameters have the benefit over traditional filters in that the convolution and pooling filters experience a learning process throughout the training stage [36]. Figure 5 shows the CNN network architecture for image classification.

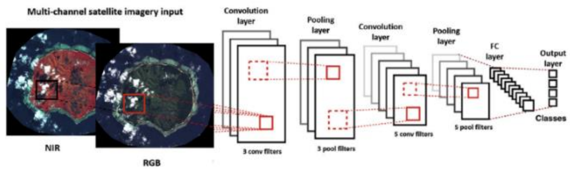


Figure 5. CNN Network Architecture for Image Classification

The input layer is a network for input (x) and target (t) of the data that will be used to train CNN. In this layer, the height, width, and channel size of the image is determined. Three channels for RGB images and 1 channel for gray images.

The hidden layer or also known as the feature extraction process can have many layers, generally consists of [36]: (1) Convolutional layer with filter size parameter which is the height and width of the filter used by the training function in scanning the image and number of filter parameter which is the number of neurons which will determine the number of feature maps; (2) Normalization layer to normalize the activation and gradient propagating through the network, making the training process an easier optimization problem; (3) ReLu layer as non-linear activation function in the convolutional layer; (4) The pooling layer may not be used, but the Convolutional layer (with its activation function) is sometimes followed by a down-sampling process to reduce the spatial size of the feature map and reduce redundant spatial information. The parameters are the column sizes of the matrix and the stride which determines the step size in the training function.

The fully connected layer is the layer where the neurons are connected to all the neurons in the previous layer [36]. This layer combines all the features learned by the previous layer (Hidden layer) throughout the image to identify the larger pattern [36]. This layer combines the results of feature extraction into flattens or vectors so that the classification or regression process for images can now be carried out [36]. Therefore, the Output size

DOI: https://doi.org/10.29207/resti.v7i1.4715
Creative Commons Attribution 4.0 International License (CC BY 4.0)

parameter in this layer must be equal to the number of classes. Thus, this layer can also be used as a feature extractor which allows other classifier methods, such as SVM, ANN, etc. to take advantage of it.

The output layer is the process of determining the classification and regression/estimation outputs [36]. This layer generally consists of SoftMax and output/classification layer [36]. SoftMax classifier as an activation function to normalize the output of the Fully connected layer [36]. The output from SoftMax can then be used as a classification probability at the output/classification layer to determine the predicted output or assign the input to one of the classes.

The parameter settings for CNN are shown in Table 3 .

Table 3. Parameters Setting of CNN

Parameter	Setting
Training function	adam
Validation data	Testing Data
Gradient decay factor	0.9
Squared gradient decay factor	0.999
Epsilion	1.00E-08
Initial learning rate	1.00E-03
Learning rate schedule	piecewise
Learning rate drop factor	0.3
Learning rate prop period	10
L2 Regularization	0.005
Gradient treshold method	l2norm
Gradient treshold	inf
Max epoch	10
Mini batch size	8
Verbose	1
Verbose frequency	10
Validation patience	inf
Shuffle	every-epoch
Check point path	none
Execution environment	auto
Worker load	[]
Output function	[]
Plots	training-progress
Sequence length	longest
Sequence padding value	0
Sequence padding direction	right
Dispatch in background	0
Reset input normalization	1
Batch normalization statistics	population

2.4. Support Vector Machine

The Support Vector Machine (SVM) excels in processing large data with small dimensions. In SVM, each training data is expressed by notation (x_i, y_i), where $i = 1, 2, 3, \ldots, n$, so n is the amount of data. Available data is denoted as: $\vec{x}_i \in R^d$, where $x_i = \{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iq}\}$ is a feature for data i. While each class label is denoted as: $y_i \in \{-1, +1\}$ where $i = 1, 2, 3, \ldots, n$. It is assumed that the two classes -1 and +1 can be perfectly separated by a d-dimensional *hyperplane*, which is defined in Equation (1).

$$w.x_i + b = 0 \quad (1)$$

w and b are the weight and bias parameters whose values you want to find. $w.x_i$ is the inner inner product between w and x_i. Data/pattern x_i that enters class –1 can be defined as a pattern that satisfies Inequality (2), while data/pattern x_i that enters class +1 can be defined as a pattern that satisfies Inequality (3).

$$w.x_i + b \leq -1 \quad (2)$$

$$w.x_i + b \geq +1 \quad (3)$$

If there is data in class -1 that is located in the *hyperplane*, then Inequality (2) will be satisfied. Likewise, if there is data in class +1 which is located in the *hyperplane*, then Inequality (3) will be fulfilled. Thus, the margin can be calculated by subtracting the two inequalities as shown in Equation (4).

$$w.(x_b - x_a) = 2 \quad (4)$$

The *hyperplane margin* is given by the distance between the two *hyperplanes* of the two classes, so Equation (4) can be summarized into Equation (5) as follows.

$$\|w\| * d = 2 \; atau \; d = \frac{2}{\|w\|} \quad (5)$$

So the objective function is to minimize the following Equation (6).

$$\min_{\vec{w}} \tau(w) = \frac{1}{2}\|\vec{w}\|^2 \quad (6)$$

Condition:

$$y(w.x_i + b) - 1 \geq 0 \quad (7)$$

$$y(w.x_i + b) \geq 1, i = 1, 2, \ldots, n \quad (8)$$

By labeling -1 for the first class and labeling +1 for the second class, data prediction can use Equation (9).

$$y_i = sign(o_i) \quad (9)$$

$$y = \begin{matrix} +1, jika \; w.z + b > 0 \\ -1, jika \; w.z + b < 0 \end{matrix}$$

y is the class label resulting from the prediction of the data i, which if the output resulting from the prediction of o-i data > 0 then y = +1 and vice versa if <0 then y = -1.

If in ANN there are Perceptron and Multi Layer Perceptron (MLP), in SVM there are linear SVM and non-linear SVM (Kernel Trick). Like Perceptron, SVM is actually a linear *hyperplane* that only works on data that can be separated linearly. For data whose class distribution is not linear, the Kernel approach is used on the initial data features of the dataset to make it possible to solve non-linear problems.

To solve non-linear problems, SVM is modified by including Kernel functions. First of all the data $\vec{x}$ in the input space is mapped by the function $\phi(\vec{x})$ to a higher dimensional vector space (feature space). In this new vector space, a *hyperplane* that separates the two

DOI: https://doi.org/10.29207/resti.v7i1.4715
Creative Commons Attribution 4.0 International License (CC BY 4.0)

classes can be constructed. This approach is different from classification methods in general which actually reduce the initial dimensions to simplify the computational process to provide better prediction accuracy. The Kernel functions that are usually used in SVM are Linear (10), Polynomial (11), Radial Basis Function (12), Tangent Hyperbolic or Sigmoid (13), and Inverse Multiquadratic (14).

$$K(x_i, x_j) = x_i . x_j \tag{10}$$

$$K(x_i, x_j) = (x_i . x_j + 1)^d \tag{11}$$

$$K(x_i, x_j) = exp\left(\frac{\|x_i . x_j\|^2}{2\sigma^2}\right) \tag{12}$$

$$K(x_i, x_j) = tanh\,(\sigma(x_i . x_j) + c) \tag{13}$$

$$K(x_i, x_j) = \frac{1}{\sqrt{\|x_i - x_j\|^2 + c^2}} \tag{14}$$

$\boldsymbol{\sigma}$, c, $d > 0$, is a constant.

The parameter settings in SVM are shown in Table 4.

Table 4. Parameters Setting of SVM

Parameter	Setting
Multi class approach	One vs All
Kernel function	Linear
Kernel scale	1
Prior	[0.34, 0.34,0.33]
Cost	[0,1,1;1,0,1;1,1,0]
Binary loss	hinge
Coding matrix	[1,-1,-1;-1,1,-1;-1,-1,1]
Scale transform	none

2.5. Weighted Class

As the name implies, the Weighted Class (WC) approach is an approach that gives a certain weight value to each class. Its function is to class reduction imblanced. Actually, the ensemble approach can also be used for imbalanced class reduction. However, a simpler way is to simply take the median value of the class frequency distribution divided by the class frequency distribution, more details are shown in Equation (15).

$$w_c = \frac{median\left(\frac{\sum_{i=1}^{n} x_{i=c}}{\sum_{i=1}^{n} x_i}\right)}{\frac{\sum_{i=1}^{n} x_{i=c}}{\sum_{i=1}^{n} x_i}} \tag{15}$$

The w weight value of the c-class can then be used in the CNN output/classification layer process as a weighted class to balance the predicted output class.

2.6. Data Augmentation

The Data Augmentation (DA) approach is part of the Deep Learning process to overcome the weaknesses of Deep Learning in processing little data or data with patterns that are not too diverse. This is done to enrich information in a data so that it can produce a deep learning model that is strong in recognizing data patterns. In addition, this approach can also reduce overfitting symptoms.

In simple terms, the approaches that can be used for DA are general transformations such as rotation, scale, translation, cropping, coloring, reflection, shear, synthetic blue, and synthetic noise. However, this research will only apply rotation, scale, translation, synthetic blur, and synthetic noise transformation.

Rotation transformation creates a random rotation transformation that rotates the image at a randomly selected angle within a certain range of degrees. Translation transformation shifts the image horizontally and vertically with randomly selected distances within a certain pixel range. Scale transformation changes the size of the image using a scale factor that is randomly selected from a certain range, but with the same horizontal and vertical directions. Synthetic noise applies artificial noise to images with several noise models that can be used, such as Gaussian, Poisson, salt and pepper, and multiplicative noise. Synthetic blur applies Gaussian blur randomly to the image by specifying a certain amount of smoothing.

The parameter settings in SVM are shown in Table 5.

Table 5. Parameters Setting of Data Augmentation

Transformation	Setting
Rotation	[-30, 30]
Scale	[0.95, 1.95]
X Translation	[-3, 3]
Y Translation	[-3, 3]
Blur	Randomized Gaussian Blur
Noise	Salt and Paper

The experiment options in DA are divided into two parts. The first part, DA1, uses the Rotation, Scale, X Translation, and Y Translation transformations. While the second part, namely DA2, uses the transformation DA1 + synthetic blur and noise.

2.7. Confussion Matrix

The performance of a classification model can be measured based on its accuracy performance using Equation (16) based on the Confusion Matrix (Table 6).

$$Acc. = (TP + TN)/(TP + TN + FN + FP) \tag{16}$$

Table 6. Confussion Matrix

Label/Class		PREDICTED (Output)	
		Positive (P)	**Negative (N)**
ACTUAL (Target)	**Positive (P)**	True Positive (TP)	False Negative (FN)
	Negative (N)	False Positive (FP)	True Negative (TN)

DOI: https://doi.org/10.29207/resti.v7i1.4715
Creative Commons Attribution 4.0 International License (CC BY 4.0)

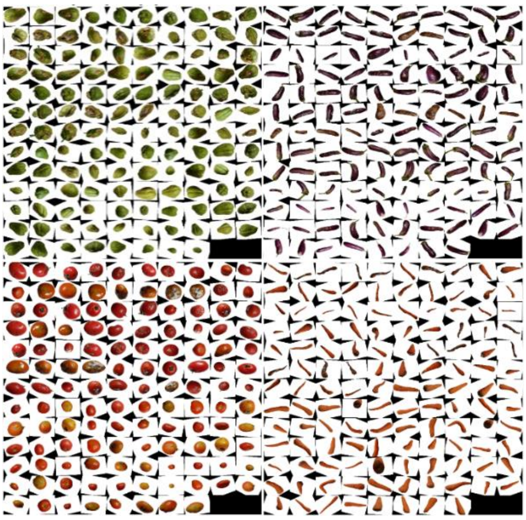
3. Results and Discussion

This section consists of modeling, evaluation, comparison, and discussion.

3.1. Modeling

Modeling is done on 2 methods, namely CNN and SVM. However, CNN is only used as a feature extractor for SVM so that it is enough to only make one CNN model, then there is no need to do CNN modeling/training which is generally quite long. By activating the fully connected layer of the CNN model that has been trained, the training, test and prediction data can be extracted into small dimension feature vectors so that it is very appropriate for SVM modeling and also for classification/prediction with SVM. This of course will increase time efficiency and also improve model accuracy performance due to SVM's superiority in processing data with small dimensions. However, only for comparison purposes, a classifier process was also carried out with CNN.

Meanwhile, the application of WC is only used in the CNN output/classification layer in the CNN modeling process. Meanwhile, DA is only used for training data and test data for modeling and classification/prediction of CNN and/or SVM. Image samples after applying DA can be shown in Figure 6.



from DA1

from DA2

Figure 6. Sample of Data Augmentation

CNN network consists of several layers, namely: (1) image input layer; (2) feature extraction layer consisting of convolutional 1, normalization 1, relu 1, pooling 1, convolutional 2, normalization 2, relu 2, pooling 2, and convolutional 3, normalization 3, relu 3 layers; (3) fully connected layers; and (4) output/classification layer which consists of softmax and output layer. In detail, it can be shown in Table 7.

Table 7. CNN Networks Architecture

Name	Description	Activations
imageinput	Image input	[312, 416, 3]
conv1	Feature extraction 1	[3, 4] filterSize, 4 numFilter
batchnorm1	Normalization 1	[312, 416, 4]
relu1	Activation conv. 1	[312, 416, 4]
maxpool1	Down-sampling 1	[156, 208, 4]
conv2	Feature extraction 2	[3, 4] filterSize, 8 numFilter
batchnorm2	Normalization 2	[156, 208, 8]
relu2	Activation conv. 2	[156, 208, 8]
maxpool2	Down-sampling 2	[78, 104, 8]
conv3	Feature extraction 3	[3, 4] filterSize, 16 numFilter
batchnorm3	Normalization 3	[78, 104, 16]
relu3	Activation 3	[78, 104, 16]
fc	Flatten feature	3 numClass = [1, 1, 3]
softmax	Output activation	[1, 1, 3]
label	Output prediction	[1, 1, 3]

3.2. Performance Evaluation and Comparison

Maximum accuracy performance in the training process can always reach 100% for all trial options or all models/methods. While the accuracy performance in the prediction process (all data) generally shows better accuracy than the testing/evaluation process. That is, the training data is able to provide good results for the model training process so that it is able to predict different data well. Meanwhile, the application of the WC approach can slightly improve accuracy performance, this is indicated by the average accuracy of CNN-WC in the testing process of 66.88% which is 2.1245% better than CNN, as well as in the prediction process of all data of 91.44% which is better 1.3521% than CNN. In detail, shown in Figure 7.

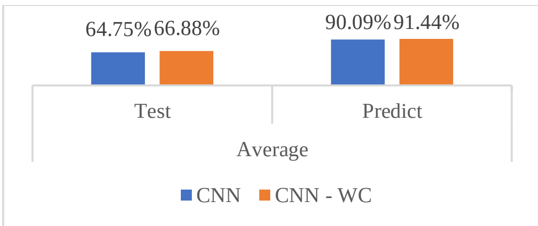


Figure 7. CNN vs CNN-WC Comparison

However, the application of DA cannot improve the accuracy performance of various models/methods. This happens because basically the combination of patterns in the data is not that diverse, so that by applying DA it actually adds information that is not important, can cause bias, makes the model training process more

DOI: https://doi.org/10.29207/resti.v7i1.4715
Creative Commons Attribution 4.0 International License (CC BY 4.0)

complex, and actually reduces model performance. Thus it can be concluded that the application of DA to the CNN model cannot always improve model performance, even though CNN is superior in processing larger data, but for data that is in principle not very diverse, CNN does not need additional data pattern combinations which could actually damage the process. pattern recognition. However, the application of DA can reduce the symptoms of overfitting, because even with lower accuracy performance, the difference in accuracy performance between the training and testing processes in each iteration is not significantly different, compared to without the application of DA. In detail, the results of measuring the accuracy performance of each experimental option (model/method) can be shown in Figure 8.

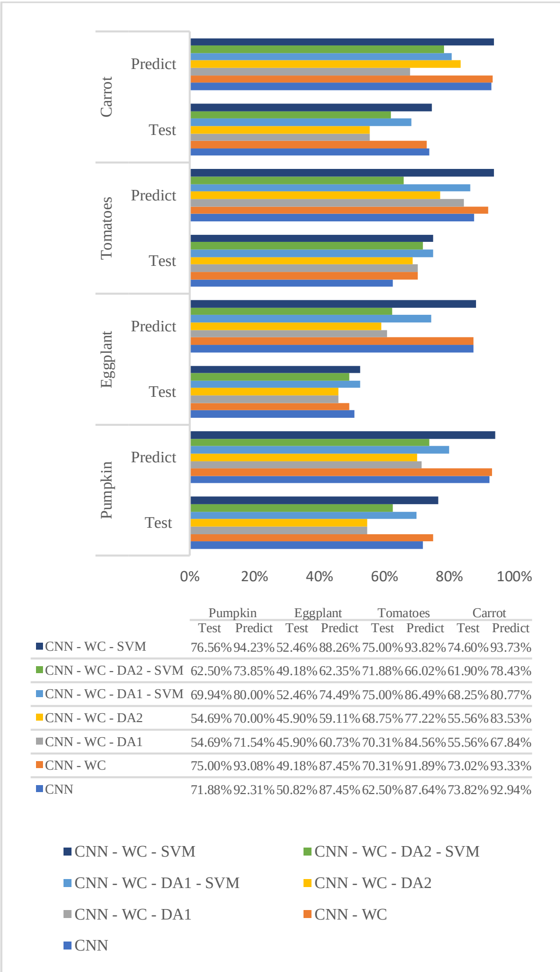


Figure 8. Performances Comparison

Thus, the application of SVM as a classifier is able to improve accuracy performance in each of the various experimental options, both in the application of WC and/or DA or without the application of the two approaches, except for part of the prediction process for all data. The increase in model accuracy performance by SVM can occur because in terms of the size of the amount, the data that is processed is not large. In addition, the use of flatten feature extraction results from the fully connected CNN layer which only produces 3 features/attributes is indeed very appropriate for processing by SVM because of SVM's superiority in processing fewer input features/attributes with a much larger number of instances. Meanwhile, data on Eggplant Vegetables is the most difficult to identify, while Pumpkin Vegetables is the easiest to identify. The average accuracy performance measurement of each experimental option (model/method) can be shown in Figure 9.

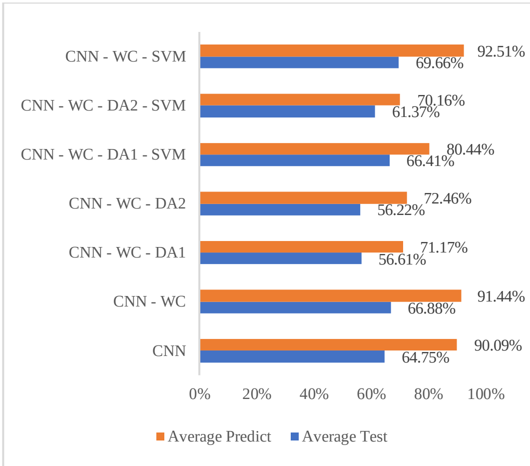


Figure 9. Average of Performances Comparison

3.3. The Best Model

Based on the results of the evaluation and comparison of the various models tested, the best model/method is CNN-WC-SVM with an average accuracy of 69.66% at the testing stage and 92.51% at the prediction stage for all data as shown in Figure 7 previously. While the highest accuracy was obtained in the Pumpkin Vegetable test and prediction with an accuracy of 76.56% and 94.23% respectively. The CNN-WC-SVM model is a model that implements WC in the CNN output/classification layer, does not implement DA, trains CNN to act as a feature extractor, and trains SVM to act as a classifier by utilizing the feature extraction results (activation) from the results of the fully connected CNN layer.

For more details, the CNN-WC-SVM training process for Pumpkin Vegetables can be shown in Figure 10. While The results of the CNN-WC-SVM training performance on Pumpkin Vegetables can be shown in Figure 11. Both of them can shown the accuracy and loss performance of CNN in more details for each multiple of 10 training iterations. These results also show that the performance of the CNN model accuracy in training and testing is not significantly different, so there are no overfitting indication.

DOI: https://doi.org/10.29207/resti.v7i1.4715
Creative Commons Attribution 4.0 International License (CC BY 4.0)

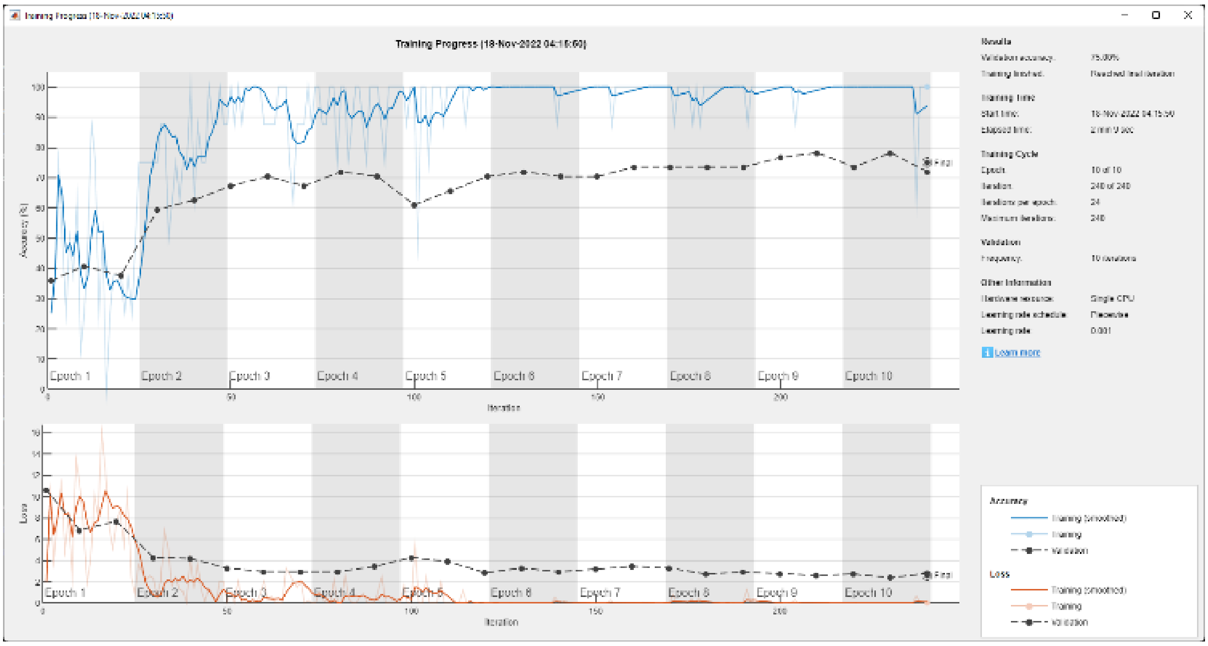


Figure 10. Training-Progress of CNN-WC-SVM for Image Classification of Pumpkin Quality

Training on single CPU.
Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:03	25.00%	35.94%	1.9689	10.5867	0.0010
1	10	00:00:07	25.00%	40.62%	11.2137	6.8190	0.0010
1	20	00:00:11	37.50%	37.50%	7.6476	7.6721	0.0010
2	30	00:00:16	75.00%	59.38%	2.3021	4.2634	0.0010
2	40	00:00:22	62.50%	62.50%	4.2100	4.1600	0.0010
3	50	00:00:27	100.00%	67.19%	0.0415	3.2659	0.0010
3	60	00:00:32	87.50%	70.31%	1.9987	2.9394	0.0010
3	70	00:00:37	87.50%	67.19%	1.9806	2.9232	0.0010
4	80	00:00:43	100.00%	71.88%	1.3657e-06	2.9291	0.0010
4	90	00:00:48	100.00%	70.31%	2.0862e-07	3.4132	0.0010
5	100	00:00:53	100.00%	60.94%	0.0720	4.2564	0.0010
5	110	00:00:58	100.00%	65.62%	0.0024	3.8744	0.0010
5	120	00:01:03	100.00%	70.31%	-0.0000e+00	2.8270	0.0010
6	130	00:01:09	100.00%	71.88%	4.0899e-07	3.2604	0.0010
6	140	00:01:14	100.00%	70.31%	3.3639e-07	2.9202	0.0010
7	150	00:01:19	100.00%	70.31%	3.0920e-08	3.1662	0.0010
7	160	00:01:25	100.00%	73.44%	0.0004	3.4163	0.0010
8	170	00:01:30	100.00%	73.44%	-0.0000e+00	3.2707	0.0010
8	180	00:01:35	100.00%	73.44%	-0.0000e+00	2.7010	0.0010
8	190	00:01:41	100.00%	73.44%	1.1046e-06	2.9023	0.0010
9	200	00:01:46	100.00%	76.56%	-0.0000e+00	2.7368	0.0010
9	210	00:01:51	100.00%	78.12%	1.6632e-06	2.5828	0.0010
10	220	00:01:57	100.00%	73.44%	-0.0000e+00	2.6963	0.0010
10	230	00:02:02	100.00%	78.12%	-0.0000e+00	2.3676	0.0010
10	240	00:02:08	100.00%	71.88%	-0.0000e+00	2.7700	0.0010

Figure 11. Training-Performances of CNN-WC-SVM for Image Classification of Pumpkin Quality

Based on the evaluation results of the various models/methods tested, the application of WC can slightly increase the accuracy of CNN, but the application of DA can actually damage the performance of the model because the data is in principle homogeneous, while the application of CNN as a feature extractor and SVM as a classifier is superior in When processing data with small dimensions, such as feature vector data resulting from activation in the fully connected layer, CNN is able to show the best accuracy performance, with an average accuracy of 69.66% in the testing phase and 92.51% in the prediction stage for all data.

The highest accuracy was obtained in the pumpkin vegetable test and prediction with an accuracy of 76.56% and 94.23% respectively as shown in Figure 12.

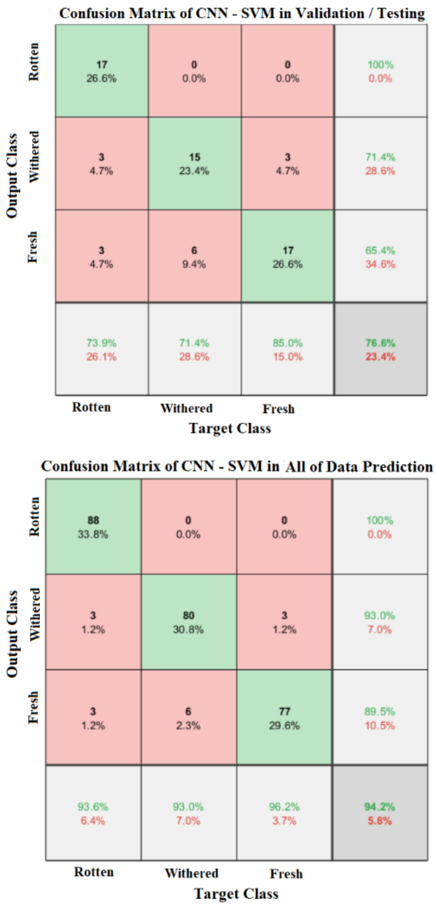


Figure 12. (Top) Testing and (Bottom) Prediction Performances of CNN-WC-SVM for Image Classification of Pumpkin Quality

4. Conclusion

This study proposes the application of Weighted Class (WC) in the CNN output/classification layer for imbalanced class reduction, data Augmentation (DA) in training data and test data for data diversity and overfitting reduction, CNN as a feature extractor, and SVM as a classifier by utilizing features in the activation result vector from the fully connected CNN layer. This proposed method can be termed a Convolutional Neural Network – Weighted Class – Support Vector Machine (CNN-WC-SVM).

Based on the experiment results, it can be concluded that the application of WC for imbalanced class reduction and SVM as a classifier can improve the performance of CNN accuracy for image quality classification of pumpkin, eggplant, tomato, and carrot vegetables. With this good performance (69.66% average accuracy in the testing phase and 92.51% average accuracy in the prediction stage for all data), this model is very feasible to implement to predict the quality image of vegetable products so that it can facilitate, improve time efficiency, and maintain the quality of vegetable products while increasing smart farming.

Nevertheless, this model is not without some drawbacks, such as the amount of data that is still not diverse due to limited data collection, which in fact results in the application of DA not being as expected. Therefore, future research can test the proposed method on public data or by collecting more and more diverse data.

References

[1] M. J. M. Christenhusz and J. W. Byng, "The number of known plants species in the world and its annual increase," *Phytotaxa*, vol. 261, no. 3, pp. 201–217, 2016.

[2] Q. Chen, C. Zhang, J. Zhao, and Q. Ouyang, "Recent advances in emerging imaging techniques for non-destructive detection of food quality and safety," *TrAC - Trends Anal. Chem.*, vol. 52, pp. 261–274, 2013.

[3] K. Hameed, D. Chai, and A. Rassau, "A comprehensive review of fruit and vegetable classification techniques," *Image Vis. Comput.*, vol. 80, pp. 24–44, 2018.

[4] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, no. January, pp. 311–318, 2018.

[5] S. Zhang, S. Zhang, C. Zhang, X. Wang, and Y. Shi, "Cucumber leaf disease identification with global pooling dilated convolutional neural network," *Comput. Electron. Agric.*, vol. 162, no. February, pp. 422–430, 2019.

[6] S. G. Wu, F. S. Bao, E. Y. Xu, Y. X. Wang, Y. F. Chang, and Q. L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *ISSPIT 2007 - 2007 IEEE International Symposium on Signal Processing and Information Technology*, 2007, pp. 11–16.

[7] P. Jiazhi and H. Yong, "Recognition of plants by leaves digital image and neural network," in *Proceedings - International Conference on Computer Science and Software Engineering, CSSE 2008*, 2008, vol. 4, pp. 906–910.

[8] M. Zaborowicz, P. Boniecki, K. Koszela, A. Przybylak, and J. Przybył, "Application of neural image analysis in evaluating the quality of greenhouse tomatoes," *Sci. Hortic. (Amsterdam).*, vol. 218, pp. 222–229, 2017.

[9] D. Jawale and M. Deshmukh, "Real time automatic bruise detection in (Apple) fruits using thermal camera," in *Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017*, 2017, vol. 2018-Janua, pp. 1080–1085.

[10] H. Almogdady, S. Manaseer, and H. Hiary, "A flower recognition system based on image processing and neural networks," *Int. J. Sci. Technol. Res.*, vol. 7, no. 11, pp. 166–173, 2018.

[11] M. Suresha, "Texture Features and Decision Trees based Vegetables Classification," *Int. J. Comput. Appl.*, no. April, pp. 21–26, 2012.

[12] L. F. Santos Pereira, S. Barbon, N. A. Valous, and D. F. Barbin, "Predicting the ripening of papaya fruit with digital imaging and random forests," *Comput. Electron. Agric.*, vol. 145, no. December 2017, pp. 76–82, 2018.

[13] T. Tiay, P. Benyaphaichit, and P. Riyamongkol, "Flower recognition system based on image processing," *Proc. 2014 3rd ICT Int. Sr. Proj. Conf. ICT-ISPC 2014*, no. July, pp. 99–102, 2014.

[14] Y. A. Akter and M. O. Rahman, "Development of a computer vision based Eggplant grading system," in *4th International Conference on Advances in Electrical Engineering, ICAEE 2017*, 2017, vol. 2018-Janua, pp. 285–290.

[15] M. Li and W. Bijker, "Vegetable classification in Indonesia using Dynamic Time Warping of Sentinel-1A dual polarization SAR time series," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 78, no.

DOI: https://doi.org/10.29207/resti.v7i1.4715
Creative Commons Attribution 4.0 International License (CC BY 4.0)

September 2018, pp. 268–280, 2019.
[16] C. S. Nandi, B. Tudu, and C. Koley, "A machine vision technique for grading of harvested mangoes based on maturity and quality," *IEEE Sens. J.*, vol. 16, no. 16, pp. 6387–6396, 2016.
[17] D. S. Pérez, F. Bromberg, and C. A. Diaz, "Image classification for detection of winter grapevine buds in natural conditions using scale-invariant features transform, bag of features and support vector machines," *Comput. Electron. Agric.*, vol. 135, pp. 81–95, 2017.
[18] Y. Sakai, T. Oda, M. Ikeda, and L. Barolli, "A vegetable category recognition system using deep neural network," in *Proceedings - 2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2016*, 2016, pp. 189–192.
[19] G. Zeng, "Fruit and vegetables classification system using image saliency and convolutional neural network," *Proc. 2017 IEEE 3rd Inf. Technol. Mechatronics Eng. Conf. ITOEC 2017*, vol. 2017-Janua, pp. 613–617, 2017.
[20] J. Steinbrener, K. Posch, and R. Leitner, "Hyperspectral fruit and vegetable classification using convolutional neural networks," *Comput. Electron. Agric.*, vol. 162, no. October 2018, pp. 364–372, 2019.
[21] M. Kozłowski, P. Gorecki, and P. M. Szczypinski, "2019, Kozłowski, CNN - Varietal Classification.pdf," *Biosyst. Eng.*, vol. 184, pp. 155–165, 2019.
[22] Z. Li, F. Li, L. Zhu, and J. Yue, "Vegetable recognition and classification based on improved VGG deep learning network model," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 559–564, 2020.
[23] M. I. Ahmed, S. Mahmud Mamun, and A. U. Zaman Asif, "DCNN-Based Vegetable Image Classification Using Transfer Learning: A Comparative Study," in *2021 5th International Conference on Computer, Communication, and Signal Processing, ICCCSP 2021*, 2021, no. May, pp. 235–243.
[24] H. Singh, R. Singh, P. Goel, A. Singh, and N. Sharma, "Automatic Framework for Vegetable Classification using Transfer-Learning," *Int. J. Electr. Electron. Res.*, vol. 10, no. 2, pp. 405–410, 2022.
[25] M. Khatun, M. Forhad Ali, P. Sarker, F. Ali Nakib Aman Turzo, and J. Nine Pritom Sarker, "Fruits Classification using Convolutional Neural Network," *GRD Journals- Glob. Res. Dev. J. Eng.*, vol. 5, no. 8, 2020.
[26] R. E. Saragih and A. W. R. Emanuel, "Banana Ripeness Classification Based on Deep Learning using Convolutional Neural Network," in *3rd 2021 East Indonesia Conference on Computer and Information Technology, EIConCIT 2021*, 2021, pp. 85–89.
[27] R. S. Latha *et al.*, "Automatic Fruit Detection System using Multilayer Deep Convolution Neural Network," in *2021 International Conference on Computer Communication and Informatics, ICCCI 2021*, 2021, no. January.
[28] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," *Comput. Electron. Agric.*, vol. 154, no. August, pp. 18–24, 2018.
[29] H. Kuang, C. Liu, L. L. H. Chan, and H. Yan, "Multi-class fruit detection based on image region selection and improved object proposals," *Neurocomputing*, vol. 283, pp. 241–255, 2018.
[30] V. A. Patel and M. V. Joshi, "Convolutional neural network with transfer learning for rice type classification," in *ICMV 2017 - The 10th International Conference on Machine Vision*, 2018, no. April, p. 21.
[31] A. Di Noia *et al.*, "Use of neural networks in ground-based aerosol retrievals from multi-angle spectropolarimetric observations," *Atmos. Meas. Tech.*, vol. 8, no. 1, pp. 281–299, 2014.
[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: Constructing a large-scale image database," in *Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
[33] A. Krizhevsky, I. Sutskever, and E. H. Geoffrey, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, vol. 60, no. 6, pp. 84–90, 2012.
[34] J. Donahue *et al.*, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," *arXiv*, vol. 32, 2014.
[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–14.
[36] M. Segal-Rozenhaimer, A. Li, K. Das, and V. Chirayath, "Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (CNN)," *Remote Sens. Environ.*, vol. 237, no. November 2018, p. 111446, 2020.

DOI: https://doi.org/10.29207/resti.v7i1.4715
Creative Commons Attribution 4.0 International License (CC BY 4.0)