Mushroom Recognition using Neural Network

Johaira U. Lidasan[1] and Martina P. Tagacay[2]

**[1] College of Computer Studies, Notre Dame University
Cotabato City, 9600, Philippines**

**[2] College of Computer Studies, Notre Dame University
Cotabato City, 9600, Philippines**

Abstract

An application would be beneficial if it is real time and could give its users enough information. This would be of greater advantage for mobile applications. Mushroom Recognition using Neural Network is a mobile-based application that combined the power of neural network with image processing to recognize mushroom image based on its order and family and if it is edible or inedible/poisonous. It is a multi-class classification program that recognizes mushroom image from 3 orders and 8 families defined in this research. The application used the GrabCut algorithm for image segmentation and Probabilistic Neural Network (PNN) as its classifier that trains and classifies the mushroom image. This application used 133 mushroom images as its training data and obtained an accuracy rate of 92%. This could be used as an educational tool both for Biology students and people in IT fields. It could also help mycologists identify wild mushrooms.

Keywords: *Image Segmentation, Probabilistic Neural Network, Machine Learning.*

1. Introduction

Machine learning is one of the field in artificial intelligence that is very useful in different subjects. Shwartz and David discussed the definition of machine learning as it pertains to the automated detection of meaningful patterns in data similar to pattern recognition. [1] This means that computers must be able to learn after several training by means of some patterns. One of its example is the Probabilistic Neural Network which is an example of feed forward network that has four layers: input layer, pattern layer, summation layer and output layer. It is predominantly a classifier since it could map any input pattern to a number of classification as stated by Mishra et al in their study about classification of vehicles. [2]

This classifier has been widely used because of the following advantages: easy and instantaneous training and real time application as what Specht argued about this network. [3] As soon as one pattern representing each category has been observed, the network can begin to generalize new patterns by choosing the appropriate value of the smoothing parameter. Because of these advantages, it is commonly used as the classifier in applications running in Android devices. One of this is the study conducted by Prasvita and Herdiyeni. [4] Here PNN has been used as a classifier for classifying medicinal plants based on leaf image. Its accuracy result is only 56.33% and it is said that this is due to the quality of the leaf image captured by the mobile device.

Because mushroom is a popular food, this has been a good subject for study that would be beneficial for mushroom pickers. One of this is the study of Damien Matti which used Android phone for capturing mushroom image and classified as edible or poisonous mushroom. [5] However, this application needed an internet connection before it could classify the image because it has to send the image to a central server where the actual processing is done. In other words, this application is not real-time and time consuming. An application would be more beneficial to its users if it is real-time or could process data immediately. Furthermore, it would be good to give additional information of the subject. Mushrooms has family and order and it is much more beneficial for its user to include this information.

The study is conducted in order to improve the features of previous applications development for mushroom industry. The following are the objectives of the study:

1. To develop an application which uses Android to process images of mushroom
2. To use PNN as a classifier for recognizing the order, family and status of mushrooms (with cap and stalk) through image processing

IJCSI International Journal of Computer Science Issues, Volume 15, Issue 5, September 2018
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.5281/zenodo.1467659

3. To test the accuracy and efficiency of the application

The result of the study is useful to a) biology students who conduct experiments in the laboratory b) information technology students who develop applications that involve neural network used image processing; c) Mycologists and other researchers of allied areas to explore the use of applications and image processing in carrying out further research and d) interested groups in the nearby settlers in the forests for them to detect safe and unsafe mushrooms.

2. Methodology

This research uses 133 mushroom images for each class as a training sample set for the Probabilistic Neural Network. Each mushroom image is resized to 215 x 280 pixels before extracting its features. This is done to minimize the percentage of errors and lessen the processing time for image segmentation.

After resizing, image segmentation is done for each image through the Grabcut algorithm. It is an iterative GraphCut algorithm that starts with a user-specified bounding box around the object to be segmented, the algorithm estimates the color distribution of the target object and that of the background using a Gaussian mixture model. However, in this application, it does not need to ask user to specify the background and foreground. Grabcut algorithm is used to mask and remove the background of the image. The sample segmented image is display in figure 1 and figure 2.

Fig. 1 Original Image

Fig. 2 Segmented Image

After the segmentation, the first feature extracted is the color of the cap. [6] To do this, the intersection of stalk and cap must be identified; hence the user would be asked to mark it. After separating the cap and the stalk of the mushroom, each pixel is calculated to obtain its Red, Green and Blue values.

After extracting the color of the cap, the ratio of the image is extracted from the original segmented image. The edge of the image is detected through Canny Image. Figure 3 shows the resulting image after Canny Image detection.

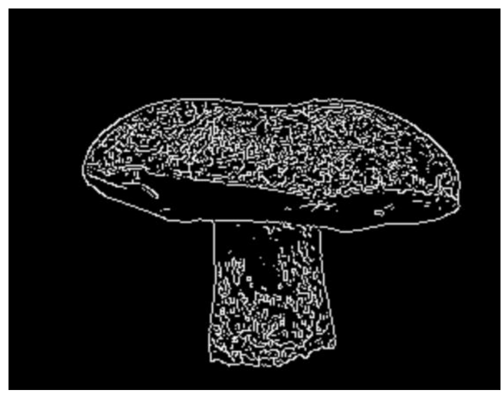

Fig 3 Illustration for Canny Edge Detection

This process converts first the image into gray scale before determining the edges. To get the width of the mushroom, the leftmost and rightmost white pixel of the image and the height of the mushroom are determined as shown in figure 4. [6]

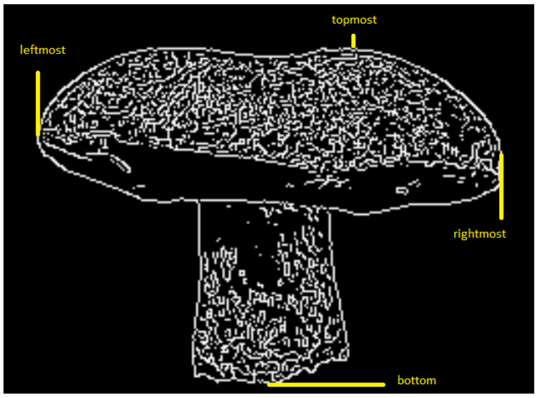

Fig. 4. Leftmost, Rightmost, topmost and bottom of the image

The segmented image is processed again for its white blob detection. [5] Using image thresholding, the images are segmented into 255 making them binary. The number of white pixels is counted and serves as one of the features of the image. All these features are normalized before putting the normalized features to the vector to be fed to the network. [6] Normalization refers to the conversion of the values between 0 and 1 using the Min-max normalization algorithm.

The application uses Probabilistic Neural Network in recognizing the mushroom image. Prasvita and Herdiyeni reported on her research on medicinal plant that training in PNN is faster and instantaneous, hence it is a good classifier. The same reason is applied in this study. [4] The testing of the application is done both for the actual mushroom and image of a mushroom.

3. Experimental Results

The application flowchart for its process is displayed in figure 5.

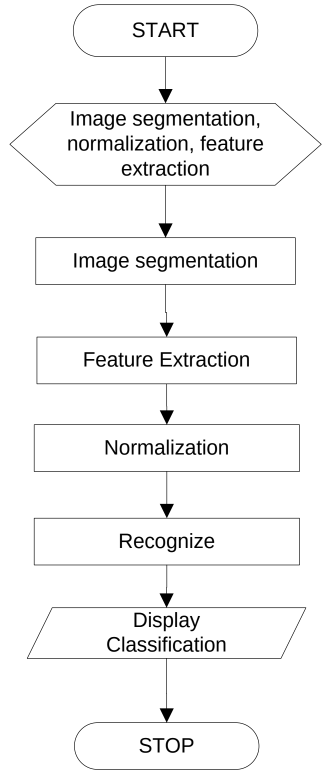

Fig. 5. MRNN Process Flowchart

For the Recognize process, it uses Probabilistic Neural Network which has 4 layers that serves as the classifier of the study:

Functionality of PNN Structure [6]

1. Input layer is an input *x* consisting of *k* value to be classified in one class of n classes. In this study, each neuron in this layer refers to the normalized extracted features of the mushroom image that is tested. It must be normalized to avoid getting an infinity result when calculating the sigmoid value of the data, hence before feeding the vector to the classifier, the data must be normalized first with the following formula

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (1)$$

2. Pattern layer performs dot product between input x and weight x_{ij}, or $Z_i = x \cdot x_{ij}$, *Zi* then divided by a certain bias σ then inserted into the radial basis functions, that is *radbas(n) = exp(-n)*. This is shown in Eq (1). Thus, the equation used in pattern layer is computed as:

$$f(x) = \exp\left(-\frac{(x-x_{ij})^T(x-x_{ij})}{2\sigma^2}\right) \quad (2)$$

where x_{ij} express training vector class i order j.

The study uses this layer for the extracted features of the training image. There are 133 training images and 5 features are extracted from each image; hence, there are 665 neurons in this layer.

3. It is in Summation layer that each pattern in each class was summed up to produce a population density function for each class. Eq (2) shows the formula used at this layer:

$$p(x) = \frac{1}{(2\pi)^{\frac{k}{2}}\sigma^k t}\sum_{i=1}^{t}\exp\left(-\frac{(x-x_{ij})^T(x-x_{ij})}{2\sigma^2}\right) \quad (3)$$

But eliminate common factors and absorb the "2" into the σ as shown in Eq (3). [7]

$$p(x) = \frac{1}{k}\sum_{i=1}^{t}\exp\left(-\frac{(x-x_{ij})^T(x-x_{ij})}{\sigma^2}\right) \quad (4)$$

In this study, there are 13 classes used. The result of the pattern layer are summed up to this layer.

4. Output layer is the decision layer input x will be classified into class I if the value $p_1(x)$ is larger than any other class. The study uses an algorithm to determine the maximum value among the 13 classes to determine the class I.

Each mushroom image is processed and extracted and grouped according to its order and family. This would serve as the class for the pattern layer of the network.

During testing, the application recognizes mushroom image according to 3 mushroom orders and 8 mushroom families and identifies whether it is edible or inedible/poisonous. As for the neural network, it uses 665 neurons for the pattern layer and 13 neurons for the summation layer. After a series of testing, the application obtained an accuracy result of 92% as shown in fig. 6. The 8% error rate is due to the quality of the image and incorrect result of the image segmentation. The incorrect result of the GrabCut algorithm could be accounted for the quality of the image, angle of the mushroom or the brightness of the image.

The calculation of the Accuracy rate is done through Confusion Matrix.

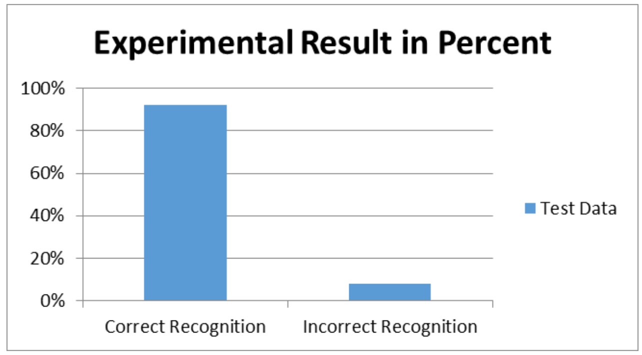

Fig. 6 Experimental Result

The output of the application is shown in fig. 7, fig. 8 and fig. 9.

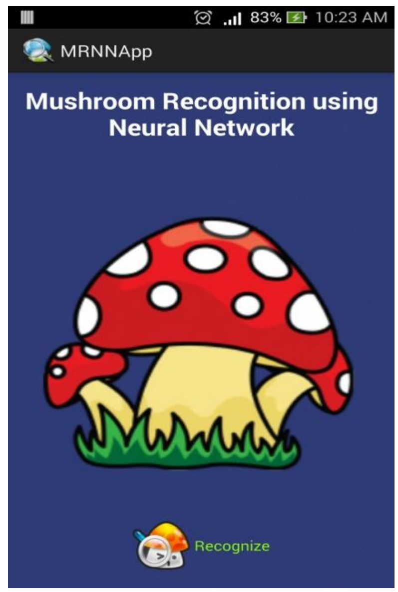

Fig. 7 MRNN GUI

Fig. 8 Segmented mushroom image

Fig. 9 Mushroom image with output

4. Conclusions

An application that recognizes mushroom with neural network has been successfully developed. It has 92% accuracy rate which produces a promising output. Moreover, the lacking 8% could be because of the following factors observed after a series of testing:

1. **Size of the neurons for each class in the pattern layer**.
   The size of the neurons for each class in the pattern layer matters in the accuracy of the application. Based on the series of testing, the larger the number of neurons used for each class, the lower the accuracy rate. Hence, determining the right number of neurons to be used for each layer was necessary.

2. **Value of the smoothing factor**
   The value of the smoothing factor must be between 0 and 1. To have a higher accuracy result, its value must be experimented. In this application, when the value of the smoothing factor was nearer to 0, its accuracy result was higher and vice versa.

3. **GrabCut algorithm output and quality of the image**
   The quality of the image affected the output of the image. Sometimes, the result of the GrabCut algorithm was dependent on the quality of the image. When the background and foreground image had the same color, the result of the image segmentation was not correct; hence the application recognized the image wrong.

Nevertheless, with the correct value of the smoothing factor and the number of neurons for each class, Probabilistic Neural Network was a good classifier for multi class problems. This is supported by the accuracy rate of the application and the result of the survey done to test the application’s efficiency and accuracy.

Acknowledgments

The authors would like to thank everyone who helped us in making this study possible.

References

[1] S. Shwartz and S. David (2014) Understanding Machine Learning: From Theory to Algorithms, *Cambridge University Press*. url: http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf

[2] M. Mishra, A.J. Jena and R. Das (2013). A Probabilistic Neural Network Approach for Classification of Vehicle. *International Journal of Application or Innovation in Engineering and Management 2*(7). url: https://pdfs.semanticscholar.org/8b35/74243ffd725ef597605f9e5a082a0f7f3999.pdf

[3] D. Specht (1990). Probabilistic Neural Networks. *Journal Neural Networks 3*(1). pages 110-117. Doi: 10.1016/0893-6080(90)90049-Q

[4] D.S. Prasvita and Y. Herdiyeni (2013). Medleaf: Mobile Application for Medicinal Plant Identification Based on Leaf Image. *International Journal on Advanced Science, Engineering and Information Technology 3*(2). Doi: http://dx.doi.org/10.18517/ijaseit.3.2.287

[5] D. Matti (2010). Mushroom recognition. (Unpublished masteral). Ecole polytechnique federale de Lausanne, pages 9-10

[6] J. Lidasan (2016). Mushroom Classification using Neural Network. (Unpublished masteral). Cebu Institute of Technology University, Cebu, Philippines

[7] P. Sangeetha (2014). Brain Tumor Classification using Probabilistic Neural Network and Clustering. *International Journal of Innovative Research in Science, Engineering and Technology 3*(3). url: https://www.rroij.com/open-access/brain-tumor-classification-using-pnn-andclustering.pdf

Johaira Lidasan is a faculty of the College of Computer Studies in Notre Dame University, Philippines and a member of Computing Society of the Philippines. She is a graduate of Master in Computer Science and a certified Microsoft Office Specialist and IBM DB2 Academic Associate. She is interested in image processing, fuzzy logic, expert system and neural networks.

Martina Tagacay is the college dean of the College of Computer Studies in Notre Dame University, Philippines and an office of the Philippine Society of Information Technology Educators. She is a graduate of Master in Business Administration and Master in Information Management. She is interested in data mining.