



Journal of Computational Innovations and Engineering Applications

Vol. 3 No. 1

July 2018



Editorial Board

Editor-in-Chief

Prof. Elmer Dadios

De La Salle University, Manila, Philippines
elmer.dadios@dlsu.edu.ph

Managing Editor

Robert Kerwin Billones

De La Salle University, Manila, Philippines
robert.billones@dlsu.edu.ph

Editors

Prof. Abdoullah A. Afjeh

University of Toledo, USA

Prof. Marcelo Ang

National University of Singapore, Singapore

Prof. Kathleen Aviso

De La Salle University, Manila, Philippines

Prof. Argel Bandala

De La Salle University, Manila, Philippines

Prof. John-John Cabibihan

Qatar University, Qatar

Prof. Anthony SF Chiu

De La Salle University, Manila, Philippines

Prof. Kukjin Chun

Seoul National University, South Korea

Prof. Joel Cuello

University of Arizona, USA

Prof. Alvin Culaba

De La Salle University, Manila, Philippines

Prof. Eryk Dutkiewicz

University of Technology Sydney, Australia

Prof. Alexis Fillone

De La Salle University, Manila, Philippines

Prof. Laurence Gan Lim

De La Salle University, Manila, Philippines

Prof. Noel Gunay

Mindanao State University, General Santos City

Prof. Kaoru Hirota

Tokyo Institute of Technology, Japan
Japan Society for Promotions of Science, China

Prof. Rodrigo Jamisola, Jr.

*Botswana International University of Science
and Technology*

Prof. Oussama Khatib

Stanford University, USA

Prof. Ioan Marinescu

University of Toledo, USA

Prof. Janina Mazierska

James Cook University, Australia

Prof. Raouf Naguib

BIOCORE, International U.K.
Liverpool Hope University, U.K.

Prof. Yong-Jin Park

Waseda University, Japan

Dr. Nguyen Thi Quynh

*Royal Melbourne Institute of Technology (RMIT) University,
Vietnam*

Prof. Raymond Sison

De La Salle University, Manila, Philippines

Prof. Raymond Girard Tan

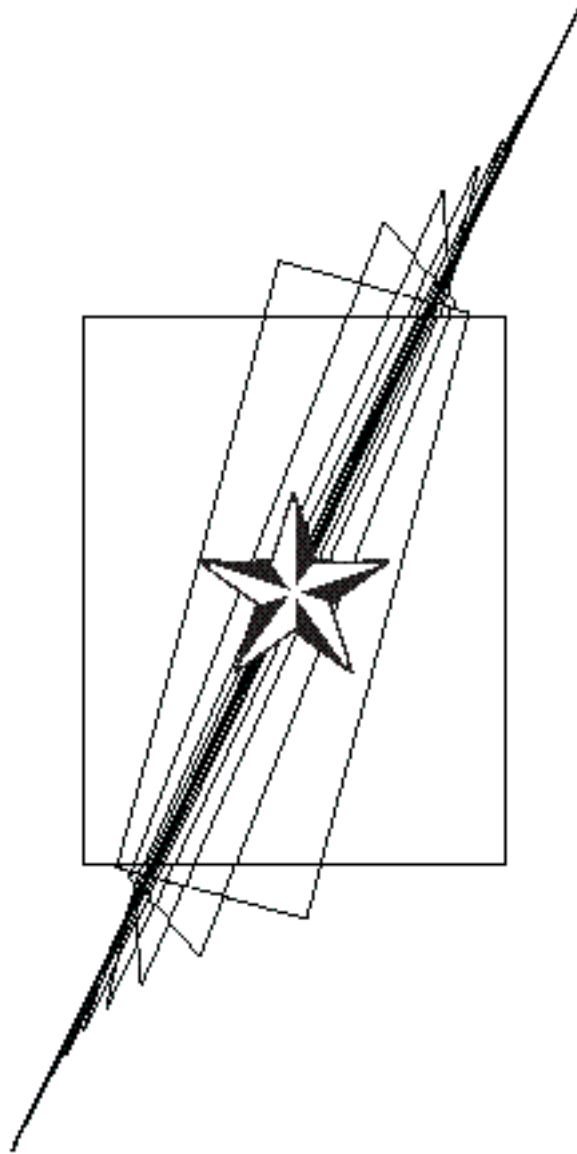
De La Salle University, Manila, Philippines

Dr. Ryan Vicerra

De La Salle University, Manila, Philippines

Prof. Lawrence Wong

National University of Singapore, Singapore



JOURNAL OF COMPUTATIONAL
INNOVATIONS AND
ENGINEERING APPLICATIONS

Volume 3 Number 1
JULY 2018

The **Journal of Computational Innovations and Engineering Applications (JCIEA)** is a peer-reviewed, open access journal of De La Salle University, Manila. The JCIEA aims to promote the development of new and creative ideas on the use of technology in solving different problems in different fields of our daily lives. The JCIEA solicits high quality papers containing original contributions in all areas of theory and applications of Engineering and Computing including but not limited to: Computational Applications, Computational Intelligence, Electronics and Information and Communications Technology (ICT), Manufacturing Engineering, Energy and Environment, Robotics, Control and Automation, and all their related fields. The JCIEA editorial board is comprised of experts from around the world who are proactively pushing for the development of research in these fields.

Annual Subscription Rates: Foreign libraries and institutions: US\$60 (airmail). Individuals: US\$50 (airmail). Philippine domestic subscription rates for libraries and institutions: Php1,800, individuals: Php1,300. Please contact Ms. Joanne Castañares for subscription details: telefax: (632) 523-4281, e-mail: dlsupublishinghouse@dlsu.edu.ph

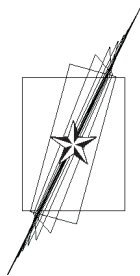
Copyright © 2017 by De La Salle University

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without written permission from the copyright owner.

ISSN 2507-9174

Published by De La Salle University Publishing House
2401 Taft Avenue, Manila 0922 Philippines
Telephone: (63 2) 523-4281 / 524-2611 loc 271
Fax: (63 2) 523-4281
Email: dlsupublishinghouse@dlsu.edu.ph
Website: <http://www.dlsu.edu.ph/offices/publishing-house/journals.asp>

The De La Salle University Publishing House is the publications office of De La Salle University, Manila, Philippines.



JOURNAL OF COMPUTATIONAL INNOVATIONS AND ENGINEERING APPLICATIONS

Table of Contents

From the Editor

Elmer P. Dadios
Editor-in-Chief

Research Articles

- Jackfruit Phytophthora Palmivora (Butler) Disease Recognizer Using Naïve Bayes Classifier 1
Jonah Flor V. Oraño, Jomari Joseph A. Barrera, and Elmer A. Maravillas
- ReTeSoil: A Temperature, Relative Humidity, and Soil Moisture Monitoring System 8
Using GSM with Blynk
Sittie Alyzza M. Garcia, Ariel James T. Abarquez, and Cristina P. Dadula
- A LabVIEW-Based Target Optimization Genetic Algorithm for Biological Predators 15
Ryann A. Alimuin, Elmer P. Dadios, and Argel A. Bandala
- Hybrid Sensor Based Fuzzy Clustering Neural Network Classification for 21
Human Activity Recognition
Ramtin Aminpour and Elmer P. Dadios
- Implementation of k-Nearest Neighbor (KNN) for PC-Based Character Recognition of 27
Currently Used Philippine Vehicle Standard Licensed Plate
*Klariz Donna Mae B. Bundal, Arlyn P. Lauron, Jan Jeffrey R. Camiña,
and Cristina P. Dadula*
- Real-Time Vehicle Classification Using MobileNet 38
Reagan L. Galvez, Melvin K. Cabatuan, and Argel A. Bandala

The Contributors

Guidelines for Contributors

From the Editor

The Journal of Computational Innovations and Engineering Applications (JCIEA) is a peer-reviewed and abstracted journal published twice a year by De La Salle University, Manila, Philippines. JCIEA aims to promote and facilitate the dissemination of quality research outputs that can push for the growth of the nation's research productivity.

In its third volume first issue, six articles were selected to provide valuable references for researchers and practitioners in the field of agricultural crop and livestock monitoring and control, crop disease detection, environmental condition assessment, wireless sensor networks, vision systems, intelligent transport systems, fuzzy logic control, and human activity recognition.

The first article is "Jackfruit Phytophthora Palmivora (Butler) Disease Recognizer Using Naïve Bayes Classifier". This paper presents a technique to detect the presence of *P. palmivora* disease in jackfruit trunk using Naïve Bayes classifier. Based on the result, the classifier achieved 94% accuracy in detecting the disease incidence.

The second article is "ReTeSoil: A Temperature, Relative Humidity, and Soil Moisture Monitoring System Using GSM with Blynk". This monitoring system was developed based on the temperature, relative humidity, and soil moisture content of a plant to minimize the occurrence of excessive and irregular irrigation that leads to certain problems like shortage in soil nutritive elements and decrease in productivity.

The third article is "A LabVIEW-Based Target Optimization Genetic Algorithm for Biological Predators". In the livestock industry, one of the major causes of destructive loss is reptilian predators; these are hunters that feed on fowls and their by-product. The aim of this research paper is to provide a solution for determining the locus (gene position) of a target such as predators and take consideration of some parameters such as speed of movement and location status.

The fourth article is "Hybrid Sensor Based Fuzzy Clustering Neural Network Classification for Human Activity Recognition". In this study, the fuzzy c-means has been considered by the ANFIS model to produce the fuzzy inference system (FIS) to make the classification with the neural network algorithm to detect the six major human activities. The results of the experiments show that the 97.2% accuracy could be acceptable in the field of study and the clustering structure could make the simulation more robust and faster.

The fifth article is "Implementation of k-Nearest Neighbor (KNN) for PC-Based Character Recognition of Currently Used Philippine Vehicle Standard Licensed Plate". This study focused on the development of a PC-based license plate recognition system using visual basic language. A system that can recognize the Philippines' currently standard vehicle plate numbers using EmguCV methods in image processing and KNN machine learning algorithm in recognizing characters.

The sixth article is "Real-time Vehicle Classification Using MobileNet". Image classification is an important part of vision systems and has several applications like autonomous cars and surveillance. This is a challenging task because computer sees images differently from humans. This paper used MobileNet model for training the data and tested it on android device. This model is lightweight and efficient compared with previous developed models.

The JCIEA editorial board expresses their warmest thanks and deepest gratitude to the distinguished authors for their outstanding contribution to JCIEA second volume first issue. They likewise express profound appreciation to the peer reviewers for their assistance and cooperation.

Original research outputs are most welcome to JCIEA. There is no publication fee in this journal, and the research papers are assured of fair and fast peer review process. For further information, please visit www.dlsu.edu.ph/offices/publishinghouse/journals.asp.

Prof. Elmer P. Dadios, PhD
Editor-in-Chief, JCIEA

Jackfruit *Phytophthora Palmivora* (Butler) Disease Recognizer Using Naïve Bayes Classifier

Jonah Flor V. Oraño,^{1,*} Jomari Joseph A. Barrera,² and Elmer A. Maravillas,³

Abstract

Abstract— This paper presents a technique to detect the presence of *P. palmivora* disease in jackfruit trunk using Naïve Bayes classifier. In this study, 200 sample images of jackfruit trunk were used, which were divided into two sets: for training and for testing. Each set contains 50 images for healthy and 50 images for disease infected. The input images were subjected to image pre-processing such as cropping, scaling, and brightness and contrast adjustment. Then, the images were segmented into two regions using color masking. Texture features such as angular second moment (uniformity) and sum of squares (variance) were also extracted from the images. Next, Naïve Bayes classifier was used to classify whether the jackfruit is infected with the disease or not. Finally, the performance of the classifier was evaluated by computing the overall accuracy of the system. Based on the result, the classifier achieved 94% accuracy in detecting the disease incidence. Moreover, this rate can be further improved by adding texture features and by applying other classification algorithms.

Keywords: *Phytophthora palmivora*, Naïve Bayes classification, texture features, image processing

I. INTRODUCTION

Jackfruit, which is scientifically known as *Artocarpus heterophyllus* Lam and locally known as “nangka” or “langka,” is one of the most widely grown fruit crops in the Philippines, which produces the largest edible fruit that may weigh as much as 50 kg [1]. It is an emerging industry in the country especially in Eastern Visayas, where it is marketed as the “flagship fruit.” It is a multipurpose

tropical fruit tree which can be the source of food, timber, fodder, dyes, latex, and medicinal and other value-added products [2]. In 2013, jackfruit was planted in a total area of 14,526 hectares (ha) with a total production of 46,080 metric tons (mt) [3]. A follow-up report in [4] shows that production of jackfruit amounted to about 44,605 mt in 2014, 43,666 mt in 2015, and 42,021 mt in 2016. It indicates that the country’s production continued to decline.

This decline can be attributed to a variety of reasons such as natural calamities, pests, and boring insects; however, damage caused by pathogens plays a significant role in crop reduction in both quality and quantity. To name one, *Phytophthora palmivora* (Butler), which is known as one of the most destructive genera of plant pathogens in temperate and tropical regions [5], was identified as the major cause of jackfruit production decline in southern Philippines, which affects 85% of jackfruit orchards in Leyte and Samar [6]. The occurrence of this disease hampered its production and threatened the livelihood of local farmers.

Jackfruit infected with *P. palmivora* shows symptoms including trunk cankers (Fig. 1), chlorosis and wilting of the foliage, root lesions, and tree death. As described by the authors in [7], stem cankers appeared firstly as wet lesions on the bark surfaces, often close to the insertion of large branches, but more frequently at trunk bases. A reddish-brown resin oozed from cracks in the bark. The wood tissues under the lesions showed cream to reddish brown discoloration. The infected areas enlarged, girdling the stems and causing severe decline of the trees.

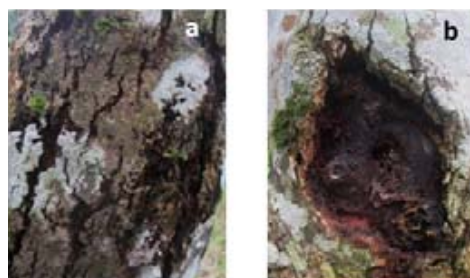


Fig. 1. Jackfruit infected with *P. palmivora* disease. a. Exterior tree trunk showing canker lesions. b. Exterior surface removed to show reddish color disease

¹Jonah Flor V. Oraño, Visayas State University, Visca Baybay City, Leyte, Philippines (e-mail: jforano@gmail.com)

²Jomari Joseph A. Barrera, Visayas State University, Visca Baybay City, Leyte, Philippines

³Elmer A. Maravillas CIT-University, N. Bacalso Ave., Cebu City, Philippines

If timely handling and proper management are not taken into consideration, this disease poses a serious threat to jackfruit yield and long-term viability of plantation. In this context, early and advanced disease detection is of utmost importance. Currently, jackfruit growers and domain experts identify the occurrence of the disease through naked-eye observation and laboratory tests. On the other hand, these could be time consuming and laborious especially in monitoring big jackfruit orchards. With the advent of technology, this process can be automated through image processing and machine learning techniques. These techniques have been applied to various agricultural applications such as to detect the incidence and severity of plant diseases, to determine plant varieties, and to identify the quality grading level of fruits and vegetables. For instance, in the study of [8], features such as color, morphology, and color coherent vector (CCV) were extracted and support vector machine (SVM) classification was used to determine incidence of pomegranate disease, in which the authors obtained an accuracy rate of 82%. Aside from SVM, other classification algorithms were proven to be effective, notably the Naïve Bayes [9] classifier wherein it outperforms the conventional classifiers as indicated in the studies of [10] and [11] in terms of classification accuracy.

There are other numerous studies pertaining to automated crop diseases identification [12] [13]; however, detection of *P. palmivora* disease occurrence in jackfruit does not exist yet. This could be beneficial in monitoring big farms and in the absence of domain experts, hence the conduct of this study.

II. METHODOLOGY

Figure 2 illustrates that the system architecture applied in this study consists of two main components: image processing and Naïve Bayes classification. In the following, details about each component were presented.

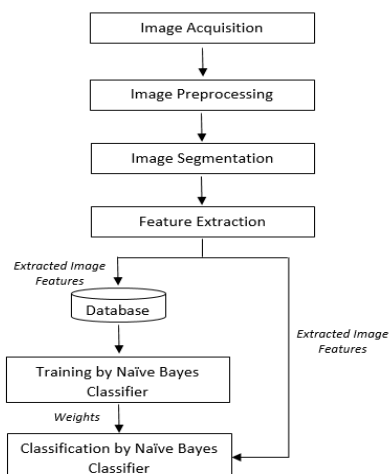


Fig. 2. System architecture of jackfruit *P. palmivora* disease recognizer.

A. Image Acquisition and Image Pre-Processing

The images of jackfruit trunk (healthy and infected) were captured using a 12-megapixel digital camera last July 24, 2017, and September 5, 2017, in two jackfruit farms located at Mahaplag, Leyte. Cropping was applied to images to remove unwanted details/objects and to emphasize the region of interest. Further pre-processing such as scaling into 600×600 and brightness and contrast adjustments were also performed.

B. Image Segmentation and Feature Extraction

Features such as color and texture were extracted from the images. For the color features, images were segmented into two regions using color masking based on the formula shown in Eq. (1). Threshold values that indicate the color characteristics of the infected region are reflected in Figure 3.

$$f(C_i) = \begin{cases} R_1, & \text{if } \angle_i(\text{hue}(C_i)) < 10^\circ \mid \text{brightness}(C_i) < 0.0625 \\ R_2, & \text{otherwise} \end{cases} \quad (1)$$

where

- C_i is color
- R_1 is infected region
- R_2 is not infected region

Hue is an angle between 0° and 360° . Equations (2) and (3) show the computations for hue angle and brightness [14].

$$\text{hue}(C_i) = \begin{cases} 0 & \text{if max} = \text{min} \\ (60^\circ \times \frac{g-b}{\text{max}-\text{min}} + 0^\circ) \bmod 360^\circ, & \text{if max} = r \\ 60^\circ \times \frac{b-r}{\text{max}-\text{min}} + 120^\circ, & \text{if max} = g \\ 60^\circ \times \frac{r-g}{\text{max}-\text{min}} + 240^\circ, & \text{if max} = b \end{cases} \quad (2)$$

$$\text{brightness}(C_i) = \frac{1}{2} (\text{max} + \text{min}) \quad (3)$$

The angle of incidence (\angle) is used to compute the distance between hue of red and the hue of C_i .

$$\angle_i(\theta) = \min(360^\circ - \theta, \theta) \quad (4)$$

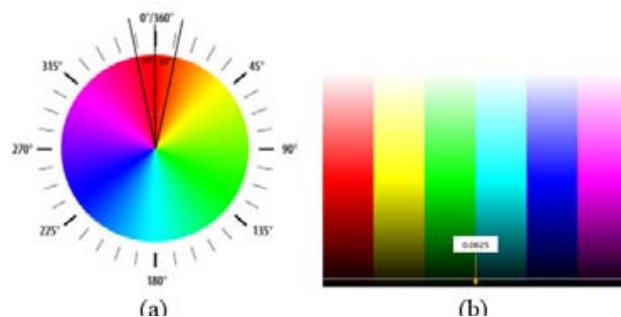


Fig. 3. Color space for (a) hue and (b) brightness.

Furthermore, for the texture features, two of the 14 descriptors defined by Haralick [15] from GLCM were extracted. These include angular second moment (Eq. 5) and sum of squares: variance (Eq. 6). GLCM produces features which describe well the relationship of adjacency among pixels in a texture image.

$$\text{Angular Second Moment (ASM)} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} p(i,j)^2 \tag{5}$$

$$\text{Sum of Squares: Variance} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - \mu)^2 p(i,j) \tag{6}$$

The extracted features values were then categorized based on the defined range of values for a specific category level shown in Table 1 to Table 3. These features from all images in the training set were stored in the database to be used in the classification process.

TABLE 1
CATEGORY OF VALUES FOR THE MASKED REGIONS

Category	Values
Very low	>0 to <=10
Low	>10 to <=20
Moderate	>20 to <= 35
High	>35 to <= 55
Very high	>55

TABLE 2
CATEGORY OF VALUES FOR ASM

Category	Values
Very low	<= 0.00105200
Low	>0.00105200 to <=0.00203048
Moderate	>0.00203048 to <=0.00300897
High	>0.00300897 to <=0.00398745
Very high	>0.00398745

TABLE 3
CATEGORY OF VALUES FOR VARIANCE

Category	Values
Very low	<=58.05314194
Low	>58.05314194 to <=92.45157227
Moderate	>92.45157227 to <=126.85000259
High	>126.85000259 to <=161.24843291
Very high	>161.24843291

C. Training and Classification Phase

During the training phase, a data set of 100 images was used, which is composed of 50 healthy trunks and 50 infected trunks.

Class values for Naïve Bayes classifier were defined, which include: not infected and infected. The probabilities for each attribute/feature conditional on the class value were computed. Then, product rule was applied to obtain a joint conditional probability for the attributes while Bayes' rule in Eq. (7) was used to derive the conditional probabilities for each class, wherein the class with the highest probability was considered as the outcome of the prediction [16].

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{7}$$

where

- $P(c|x)$ —posterior probability
- $P(x|c)$ —likelihood
- $P(c)$ —class prior probability
- $P(x)$ —predictor prior probability

III. EXPERIMENTAL RESULTS

The graphical interface for the training phase is shown in Figure 4. It illustrates that once the image is loaded into the system, the extracted feature values and its equivalent category level are displayed. The user then needs to indicate the expert's classification on the image before clicking the save button. These values will be saved into the database and will be used for the training process.



Fig. 4. Features extraction result of the system.

The difference between healthy jackfruit trunks and those with *P. palmivora* disease infection is indicated in Table 4.

TABLE 4
SAMPLE IMAGES OF INFECTED AND NOT INFECTED
JACKFRUIT TRUNKS

INFECTED	NOT INFECTED
	
	
	

Table 5 illustrates the training set for the system after extracting features from all 100 images. It contains columns for extracted numerical values, its equivalent category, and its expected classification values.

Apart from 100 jackfruit trunk images used during the training phase, a different set of 100 images was used for the testing phase. These images were loaded into the system and were then converted into features form. For each image, the predicted class label obtained by Naïve Bayes classifier was saved into the database table alongside with its values (Table 6).

The system can now be used to determine whether the image of the jackfruit trunk shows occurrence of the disease or not. Figure 5 illustrates the decision of the classifier based on the features extracted from the given image.

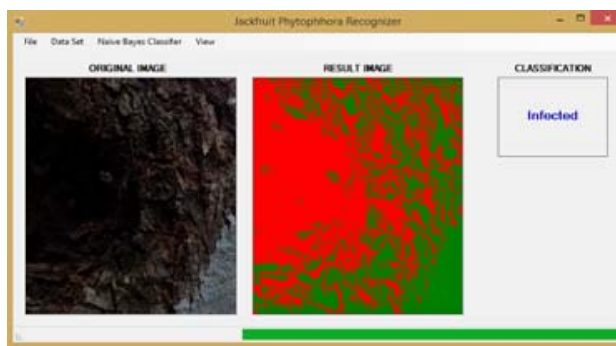


Fig.5. Recognition result of the system.

TABLE 5
TRAINING DATASET

Sample No.	Mask 1	Mask 2	ASM	Variance	Mask 1 Category	Mask 2 Category	ASM Category	Variance Category	Expected Result
1	34.76	65.24	0.00051	55.11715	Moderate	Very High	Very Low	Very Low	Infected
2	48.62	51.38	0.00252	23.65471	High	High	Moderate	Very Low	Infected
3	36.17	63.83	0.00033	46.04551	High	Very High	Very Low	Very Low	Infected
4	22.99	70.01	0.00108	30.03209	Moderate	Very High	Low	Very Low	Infected
5	59.31	40.69	0.00497	23.97449	Very High	High	Very High	Very Low	Infected
6	18.80	81.20	0.00027	54.99160	Low	Very High	Very Low	Very Low	Infected
7	4.22	95.78	0.00011	140.87128	Very Low	Very High	Very Low	High	Not Infected
8	3.15	96.85	0.00018	124.68015	Very Low	Very High	Very Low	Moderate	Not Infected
9	10.31	89.69	0.00012	134.12725	Low	Very High	Very Low	High	Not Infected
10	7.22	92.78	0.00016	169.53530	Very Low	Very High	Very Low	Very High	Not Infected
11	12.50	87.50	0.0009	178.21163	Low	Very High	Very Low	Very High	Not Infected
:	:	:	:	:	:	:	:	:	:
100	0.16	99.84	0.00025	78.19798	Very Low	Very High	Very Low	Low	Not Infected

TABLE 6
TESTING DATASET

Sample No.	Mask 1	Mask 2	ASM	Variance	Mask 1 Category	Mask 2 Category	ASM Category	Variance Category	Expected Result	Actual Result
1	43.57	56.43	0.00031	36.31061	High	Very High	Very Low	Very Low	Infected	Infected
2	33.09	66.91	0.00008	31.70688	Moderate	Very High	Very Low	Very Low	Infected	Infected
3	30.21	69.79	0.00043	71.98274	Moderate	Very High	Very Low	Low	Infected	Infected
4	19.57	80.43	0.00012	103.2699	Low	Very High	Very Low	Moderate	Infected	Not Infected
5	16.54	83.46	0.00007	105.0948	Low	Very High	Very Low	Moderate	Infected	Not Infected
6	16.04	83.96	0.00021	67.90312	Low	Very High	Very Low	Low	Infected	Infected
7	13.71	86.29	0.00030	84.07898	Low	Very High	Very Low	Low	Infected	Infected
8	12.3	87.70	0.00018	160.4463	Low	Very High	Very Low	High	Not Infected	Not Infected
9	10.15	89.85	0.00023	96.82886	Low	Very High	Very Low	Moderate	Infected	Not Infected
10	9.82	90.18	0.00012	166.3141	Very Low	Very High	Very Low	Very High	Not Infected	Not Infected
11	7.28	92.72	0.00016	118.4576	Very Low	Very High	Very Low	Moderate	Not Infected	Not Infected
:	:	:	:	:	:	:	:	:	:	:
100	7.02	92.98	0.00015	107.681	Very Low	Very High	Very Low	Moderate	Not Infected	Not Infected

A. Discussions

Table 7 shows the classifier created from the training set where NI means not infected and I means infected. The probabilities are computed as the number of samples of a class having the value (category) for a feature divided by the frequency of a class. In this case, the probability of not infected p(NI) is 0.50 and the probability of infected p(I) is 0.50 as well.

TABLE 7
PROBABILITY OF FEATURE CONDITIONAL ON THE CLASS

Category	Features							
	Mask 1		Mask 2		ASM		Variance	
	NI	I	NI	I	NI	I	NI	I
Very low	.47	.00	.00	.00	.50	.43	.00	.44
Low	.03	.20	.00	.00	.00	.05	.01	.06
Moderate	.00	.22	.00	.00	.00	.01	.11	.00
High	.00	.07	.00	.02	.00	.00	.34	.00
Very high	.00	.01	.50	.48	.00	.01	.04	.00

Table 8 below contains the values for all the extracted features from a sample image to be classified as not infected or infected.

TABLE 8
SAMPLE EXTRACTED VALUES TO BE CLASSIFIED AS INFECTED OR NOT INFECTED

Feature	Value
Mask 1	Low
Mask 2	Very high
ASM	Very low
Variance	Low

For the classification as not infected, the posterior is given by Eq. (8).

$$\begin{aligned}
 & \text{posterior(NotInfected)} \\
 &= \frac{p(\text{Mask 1}|\text{NotInfected}) p(\text{Mask 2}|\text{NotInfected}) p(\text{ASM}|\text{NotInfected}) p(\text{Variance}|\text{NotInfected}) p(\text{NotInfected})}{\text{predictor}} \quad (8)
 \end{aligned}$$

While for the classification as infected, the posterior is given by Eq. (9).

$$\begin{aligned}
 & \text{posterior(Infected)} \\
 &= \frac{p(\text{Mask 1}|\text{Infected}) p(\text{Mask 2}|\text{Infected}) p(\text{ASM}|\text{Infected}) p(\text{Variance}|\text{Infected}) p(\text{Infected})}{\text{predictor}} \quad (9)
 \end{aligned}$$

where

$$\begin{aligned}
 \text{predictor} = & p(\text{Mask 1}|\text{NotInfected}) p(\text{Mask 2}|\text{NotInfected}) p(\text{ASM}|\text{NotInfected}) \\
 & p(\text{Variance}|\text{NotInfected}) p(\text{NotInfected}) + \\
 & p(\text{Mask 1}|\text{Infected}) p(\text{Mask 2}|\text{Infected}) p(\text{ASM}|\text{Infected}) \\
 & p(\text{Variance}|\text{Infected}) p(\text{Infected})
 \end{aligned}$$

However, given the sample, the predictor is a constant and thus scales both posteriors equally. In this case, it does not affect the classification and can be ignored.

Therefore, considering the posterior numerator of Not Infected, which was calculated as

$$\begin{aligned} \text{posterior}(\text{NotInfected}) \\ = .03 \times .50 \times .50 \times .01 \times .50 = .00004 \end{aligned}$$

and the posterior numerator of Infected, which was calculated as

$$\begin{aligned} \text{posterior}(\text{Infected}) \\ = .20 \times .48 \times .43 \times .06 \times .50 = .00012 \end{aligned}$$

it can be predicted that the sample is infected.

For each image, the predicted class label obtained by Naïve Bayes classifier was then compared to the actual class label specified by the domain expert. The overall results are displayed in the confusion matrix (Table 9) to emphasize how many images from the total of each class are accurately predicted.

Then detection accuracy for each class was compute as shown in Eq. (10).

$$\frac{\text{sum of correctly predicted class label}}{\text{total number of predictions}} \quad (10)$$

While the overall accuracy of the system was calculated using Eq. (11).

$$\frac{\text{sum of correct classification}}{\text{total number of classification}} \quad (10)$$

TABLE 9
CONFUSION MATRIX

Class	Not Infected	Infected	Detection Accuracy
Not Infected	50	0	100.00%
Infected	6	44	88.00%
Overall Accuracy			94.00%

The result shows that the detection accuracy of infected class is lower (88%) compared to the not infected class (100%). However, it is notable that the system was able to achieve a high overall accuracy rate of 94%.

IV. CONCLUSION AND RECOMMENDATION

The overall accuracy rate of the system, which is 94%, indicates that the application of Naïve Bayes classifier on the extracted color and texture features can significantly support an accurate detection of the *P. palmivora* disease.

The recognition accuracy rate can be further improved by using more high-resolution images, applying other image processing techniques, adding more features, and applying other classification algorithms such as artificial neural network, C4.5 classifier, SVM, etc. It would also be better that the system be able to classify the severity of the disease infection for a basis of appropriate disease management strategies.

REFERENCES

- [1] "Jakfruit in the Philippines, Part 1," March 1985. [Online]. Available: <http://rfcarchives.org.au>.
- [2] C. R. Elevitch and H. I. Manner, "Artocarpus heterophyllum (jackfruit)," April 2006. [Online]. Available: www.traditionaltree.org.
- [3] "Philippines: Jackfruit planted area and production 2008-2013," [Online]. Available: <https://www.statista.com/>.
- [4] "CountrySTAT Philippines," 10 November 2017. [Online]. Available: <http://countrystat.psa.gov.ph>. [Accessed 18 April 2018].
- [5] A. Drenth and D. Guest, "Diversity and Management of Phytophthora," *ACIAR Monograph series no. 114. Australian Centre for International Agricultural Research, Canberra, 2004*.
- [6] L. Borines, V. Palermo, G. Guadalquiver, C. Dwyer, A. Drenth, R. Daniel and D. Guest, "Jackfruit decline caused by Phytophthora palmivora (Butler)," *Australasian Plant Pathology*, vol. 42, pp. 123-129, 2013.
- [7] M. Van Tri, N. Van Hoa, N. Minh Chau, A. Pane, R. Faedda, A. De Patrizio, L. Schena, C. H. Olsson, S. A. Wright, M. Ramstedt and S. O. Cacciola, "Decline of jackfruit (*Artocarpus heterophyllum*) incited by Phytophthora," *Phytopathologia Mediterranea*, pp. 9-14, 2015.
- [8] M. Bhange and H. Hingoliwala, "Smart Farming: Pomegranate Disease Detection Using Image Processing," *Procedia Computer Science*, pp. 280-288, 2015.
- [9] M. M. Roomi and S. Saranya, "Bayesian Classification of Fabrics Using Binary Cooccurrence Matrix," *International Journal of Information Sciences and Techniques (IJIST)*, vol. 2, no. 1, pp. 1-9, 2012.
- [10] D.-C. Park, "Image Classification Using Naive Bayes Classifier," *International Journal of Computer Science and Electronics Engineering (IJCSEE)*, vol. 4, no. 3, pp. 135-139, 2016.
- [11] M. Mursalin, M. M. Hossain, M. K. Noman and M. S. Azam, "Performance Analysis among Different Classifier Including Naive Bayes, Support Vector Machine and C4.5 for Automatic Weeds Classification," *Global Journal of Computer Science and Technology Graphics & Vision*, vol. 13, no. 3, 2013.
- [12] A. Fahrrozi, S. Madenda, Ernastuti and D. Kerami, "Wood Classification Based on Edge Detections and Texture Features Selection," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 5, p. 2016, 2167-2175.

- [13] S. Arivazhagan, R. N. Shebiah, S. Ananthi and S. V. Varthini, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features," *Agric Eng Int: CIGR Journal*, vol. 15, no. 1, pp. 211-217, 2013.
- [14] M. K. Agoston, *Computer Graphics and Geometric Modeling: Implementation and Algorithms*, Springer, 2005.
- [15] R. Haralick, K. Shanmugam and L. Dinstein, "Textural Features for Image Classification," *IEEE Transaction on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610-621, 1973.
- [16] M. K. Leung, "Naive Bayesian Classifier," 28 November 2007. [Online]. Available: cis.poly.edu/~mleung/FRE7851/f07/naiveBayesianClassifier.pdf. [Accessed 2018].

ReTeSoil: A Temperature, Relative Humidity, and Soil Moisture Monitoring System Using GSM With Blynk

Sittie Alyzza M. Garcia, Ariel James T. Abarquez , and Cristina P. Dadula

Abstract—This monitoring system was developed based on the temperature, relative humidity, and soil moisture content of a plant to minimize the occurrence of excessive and irregular irrigation that leads to certain problems like shortage in soil nutritive elements and decrease in productivity. The system is composed of two nodes: the sensor node and the base node, and BLYNK mobile application program. The sensor node consists of sensors and a transmitter for the sending of data to base node. The base node is responsible for accepting data from the sensor node and it consists of a receiver and a Global System for Mobile communication (GSM) module to relay the received data to the BLYNK cloud. The system used RF communication for sensor to base node data transmission and Global Packet Radio Service (GPRS) for base node to BLYNK cloud data transmission. The BLYNK application program is used to monitor and display the data from the sensors, and sends email notification when the measured value of at least one of the sensors is below or above the set limit. All functionalities of the application program were working. The maximum operating distance for the RF module was 35 m when both receiver and transmitter modules have antenna installed. In addition, it was observed that GPRS communication was not stable, a noticeable time delay was experienced in displaying sensor data to the application. Hence, the use of more stable communication like Wi-Fi is highly recommended to avoid some communication issues.

Keywords: monitoring temperature, relative humidity, and soil moisture; GSM, GPRS, RF communications, Wi-Fi, wireless sensor network.

Sittie Alyzza M. Garcia, Ariel James T. Abarquez , and Cristina P. Dadula, Mindanao State University, General Santos City, Philippines (e-mail: garciasittie21@gmail.com, ajs.abarquez97@gmail.com, cris_dadula@yahoo.com)

I. INTRODUCTION

Excessive and irregular irrigation is not good for both plants and soil. This will result to some serious problems like shortage in soil nutritive elements, decreased productivity, or increase in salinity. Thus, causing the soil nutrients to be submerged and remain in the subsoil while the groundwater comes up to the surface, the groundwater undergoes evaporation and forms salts on the soil [1]. This leads to decrease in quality and productivity of the soil. To avoid these problems, some systems were designed to remotely monitor the humidity of the soil. Whenever the humidity reached a certain level, a motor pump is turned on to deliver the required water [2]. There is also a monitoring system designed for precision agriculture implemented using wireless sensor nodes. These sensors are spread through the field to periodically collect and relay soil data to the processing centers [3]. Gerard Rudolph Mendez et al. [4] have performed temperature and soil humidity monitoring by using Wi-Fi communication; for sensor node, they used the WSN802G module. Sensor measurements are collected and stored in the server for further analysis. Kishore Babu and Saggam Divyasri [5] have made a similar system which used GPRS for transmitting data instead of WiFi. Nurul Fahmi et al. [6] also implemented a wireless sensor network for the monitoring of a precision agriculture system. They made a simple prototype that used temperature, humidity, pressure, and soil moisture sensors. This prototype can monitor environment status through a website and even on smartphone. In 2013, H.A. Mansour et al. [7] have studied the effect of automatic control on closed circuit drip irrigation system as a modified irrigation system on yellow corn crop vegetative and yield parameters.

This research aims to develop a microcontroller-based system for monitoring of essential plant growth factors that can be remotely accessed by the user wirelessly through an application program. The objectives of the study are: 1) to develop a system prototype that uses two main sensors (DHT11 humidity and temperature sensor, and soil moisture sensor), microcontroller, RF module and

GSM module, 2) to use BLYNK to build an application for displaying data, 3) to transmit data from the sensor node to the base node using RF communication, 4) to transmit data from the base node to the BLYNK cloud using GPRS, and lastly 5) to test the functionality of the RF communication range, stability of connection of the GSM module, and BLYNK cloud data transmission.

II. AGRICULTURAL MONITORING AND CONTROL SYSTEMS USING WIRELESS SENSOR NETWORK (WSN)

The study discussed in [3] demonstrate the design and development of a Wi-Fi-based WSN capable of intelligent monitoring of agricultural environment conditions using a pre-programmed control and management system that can be updated as required. The WSN has three components which are the sensor nodes, access point, and central station. The sensor node is capable of data collection of parameters such as relative humidity, air pressure, temperature, soil moisture, presence of light, and nutrient levels. On the other hand, the study conducted in [4] used WSN to aid farmers for real-time monitoring of different agricultural areas. A simple prototype for precision agriculture was developed that monitors humidity, soil moisture, temperature, and atmospheric pressure. The study discussed in [5] shows the effect of automatic control closed circuits drip irrigation system on yellow corn crop. This was conducted in the Al-Hasa region of Saudi Arabia. A microcontroller-based irrigation system was discussed in [6]. The researcher wants to develop a system that can work constantly in a remote location, even under abnormal conditions. The system monitors the soil humidity and delivers water whenever the humidity reaches a certain level. The study discussed in [7], developed a distributed wireless sensor network which is powered by photovoltaic panels. Soil moisture and temperature sensors are placed in the plant's roots to gather information. Sensor measurements are sent into a gateway to display values in a web application. These measurements are also used to trigger actuators for control of some system variables, such as water quantity. The system also used a communication link that used cellular-Internet interface for data transmission that allows for remote inspection and irrigation scheduling.

III. METHODOLOGY

This section presents the design and development of both the software and hardware components of the system. For the software application, BLYNK was used to develop a user interface and mobile application to display the data

coming from the sensor node. It is an Internet of Things (IoT) platform suitable only for smartphones [8]. Meanwhile, Arduino IDE was used for the development of algorithm needed by the hardware component of the system. Arduino IDE is an open-source software that is compatible with the application and board used in the system [9]. This allows for the hardware component to interact with the user application.

The system has two circuits: the base node and sensor node, as shown in Figure 1. The sensor node is composed of an Arduino Uno board, sensors (namely, DHT11, the temperature and humidity sensor, and the soil moisture sensor), and RF transmitter module to send the data to the base node. This base node is composed of an Arduino Uno board, a GSM module, and an RF receiver module. Sensor node will send data to the base node through RF communication using a TX/RX RF module. Moreover, the data received by the base node will also be sent to the BLYNK cloud by the GSM module so that the developed application can access the data anytime and anywhere.

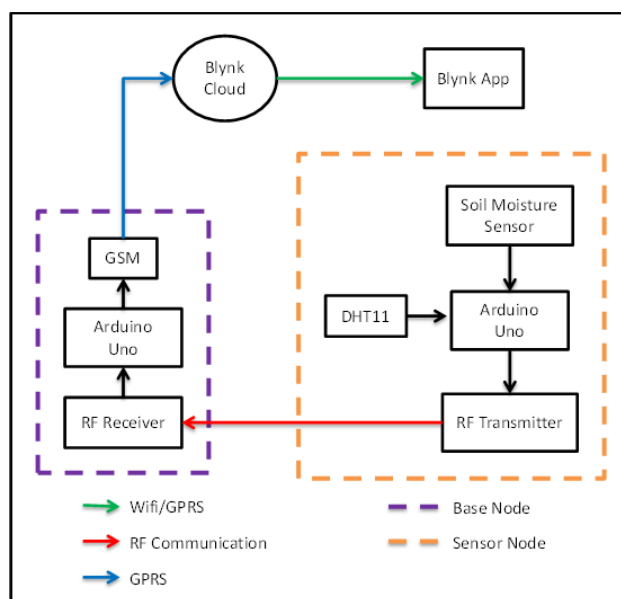


Fig. 1. Conceptual system framework.

A. Hardware Development

The project's hardware development consists of two circuitries: the sensor node and the base node construction.

Sensor Node

Sensor node is responsible for remote measurements and transmission of data [10]. It consists of Arduino Uno, RF transmitter module, DHT11, soil moisture sensor, and a 9V battery. The DHT11 sensor will gather the data for the temperature and relative humidity. It can also determine the soil's moisture using soil moisture sensor. Once all the data

are gathered, the RF transmitter will transmit all the data to the base node. Figure 2 shows the schematic diagram of the sensor node.

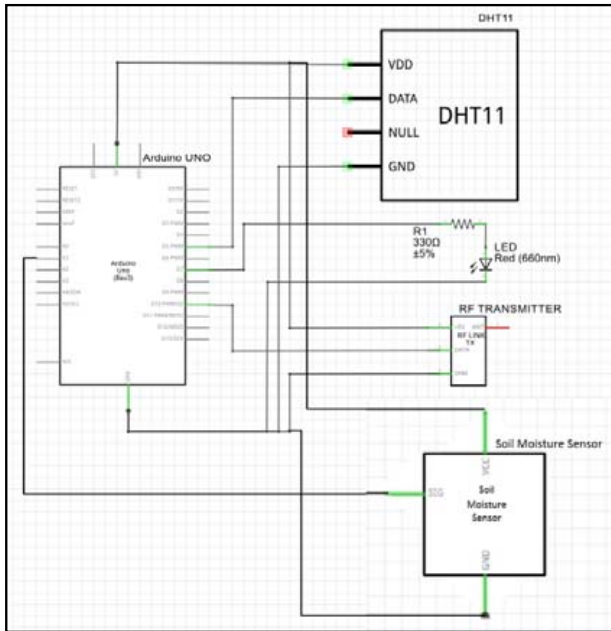


Fig. 2. Sensor node schematic diagram.

Base Node

The transmitted data coming from the sensor node will be received by the base node. It consists of Arduino Uno, GSM module, RF receiver module, and a 9V battery. When the data are transmitted from sensor node to the RF receiver module, the data will be forwarded to the BLYNK cloud by the GSM module so that the developed application can access the data anytime and anywhere.

The RF module that was used in the system operates in 433 MHz frequency band. This means that the RF transmitter and receiver will operate at the same frequency [11]. RF modules are commonly used in wireless communication because of their stability and wide coverage but is very prone to multipath fading and other error sources [12]. Another module that was used is the GSM sim900a module [13]. This module is capable of data communications between the base node and the BLYNK server. Figure 3 shows the schematic diagram of the base node.

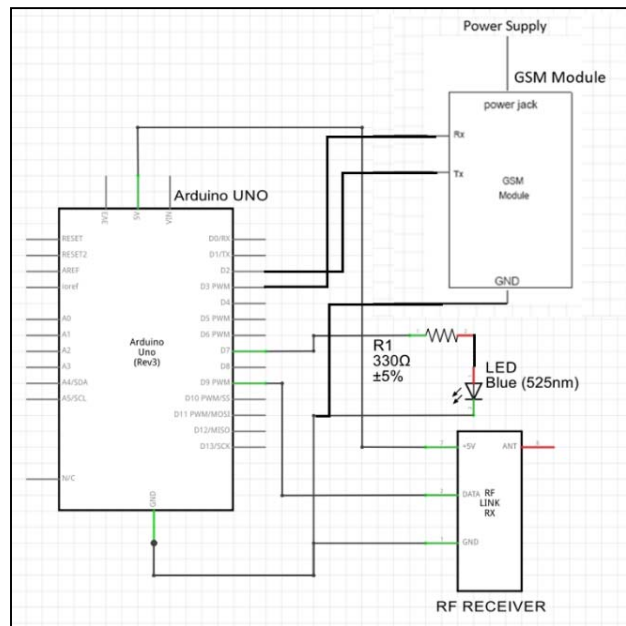


Fig. 3. Base node schematic diagram.

B. Software Application Development using BLYNK

BLYNK application was used in the software development. This application has the necessary functions in developing the system. Figure 4 shows how Blynk will be setup. It used a drag-and-drop feature in building the application and can communicate with Arduino IDE just by installing the needed libraries. The functions include the monitoring of the following sensors: (1) soil moisture, (2) temperature, (3) humidity, and (4) the *Timeline* for the real-time data of the sensors. With these in mind, the main layout of the BLYNK application should contain three gauges for every data of the sensors, SuperChart for the timeline of real-time data, and a notification and email widget. The notification function is used if sensor readings are above the set limit. However, to be able to monitor the sensors with these gauges and the SuperChart, it is then necessary for the BLYNK application to be able to communicate with the Arduino UNO board via the GSM Sim900a module. To do so, the GSM Sim900a module must have a data connection to access the BLYNK server, to monitor the data from the sensors which will be displayed by the BLYNK application. The codes are modified to make sure that the gauge and SuperChart can get the data from the sensors to monitor the soil moisture temperature, and relative humidity. The codes include the communication between the application and the Arduino. Figure 5 shows the layout of the BLYNK application.



Fig. 4. BLYNK setup [14].

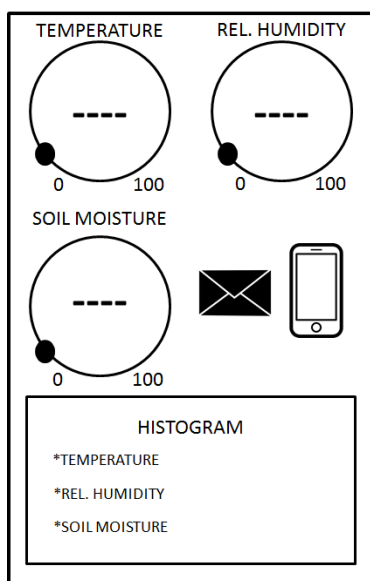


Figure 5. Application layout.

C. Hardware and Software Integration

After the necessary codes are finalized, testing whether the current codes are suitable to complete the anticipated functionalities of the hardware takes place. The researchers used the Arduino IDE Serial Monitor to check the initialization and connectivity of the GSM module and the BLYNK cloud, as shown in Figure 6. This initialization of the GSM module takes time depending on signal strength and where the node is being set up. Figure 7 shows that the BLYNK application is already connected to the Arduino IDE. It also displays the data from the sensors.

D. Distance Test

The transmission of data from the sensor node to base node was also tested for different distances (from 0 to 45 m) with the presence or absence of the antenna of the transmitter module and receiver module.

```

Initializing modem...
[15335]

v0.5.2 on Arduino Uno

[15342] Modem init...
[16247] Connecting to network...
[16559] Network: Globe Telecom-PH
[16559] Connecting to internet.globe.com.ph ...
[23806] Connected to GPRS
[24086] Connecting to blynk-cloud.com:80
[27984] Ready (ping: 836ms).
[60425] Connecting to blynk-cloud.com:80
[62874] Ready (ping: 852ms).
[96156] Connecting to blynk-cloud.com:80
    
```

Fig. 6. The initialization of the GSM module, network connection, Access Point Name (APN), Global Packet Radio System (GPRS) status, and BLYNK cloud connection.

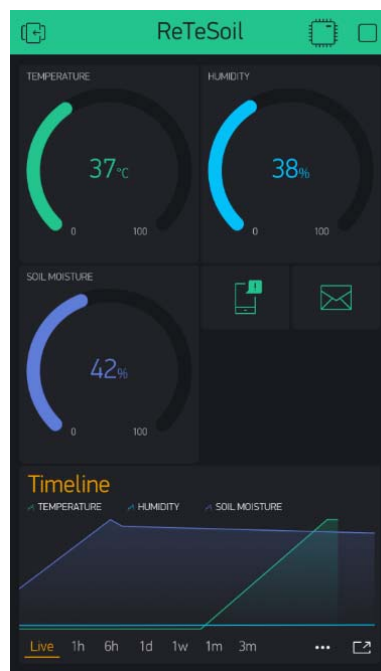


Fig. 7. Displaying of data using BLYNK application.

IV. RESULTS AND DISCUSSION

Figure 8 shows the final version of the BLYNK application, which has an interface that allows the user to monitor sensor data, through the GSM module connection.

A. Final Version of the BLYNK Application

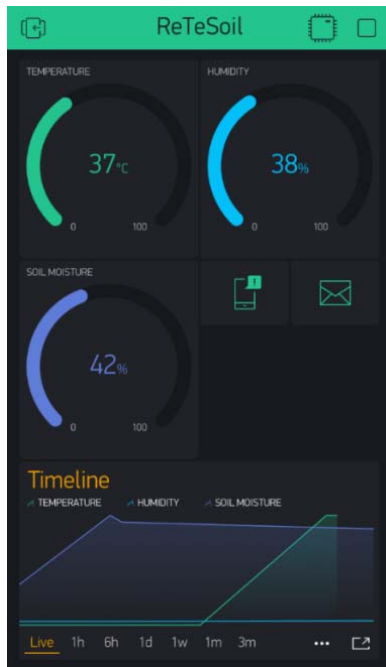


Fig. 8. BLYNK application main interface.

To monitor the data from the sensor node, the three gauges display the numeric data of the sensors, which are the DHT11 sensor (temperature and relative humidity sensor) and soil moisture sensor. The SuperChart shows the live and historical data of the sensors. The email widget allows the user to send email from the hardware to any email address if the sensor readings are above the set limit. The push notification widget allows the user to send push notification from the hardware to the user's device in case the hardware went offline and when the data from the sensors are above the set limit [15].

B. Initializing the GSM Modem

This section shows the results in the Arduino IDE serial monitor when the GSM module is being initialized, as shown in Figure 9. The initialization of the GSM module to connect to the data services of the network takes a bit longer depending upon the signal strength or the area where the base node is being set up. After the initialization, establishment of connectivity between the application and the module was initiated. Some problems encountered about the initialization of the GSM module include: 1) there were times that the module did not initialize properly and proceeded on restarting to re-initialize the module, and 2) there were also times that the researchers need to unplug the power supply and then plug it again.

```

Initializing modem...
[15335]

  _ _  ) / /  _ _ _ _  / / /
 / _ / / / / / / _ \ \ ' \
 / _ _ / \ \ / / / / \ \ \
 / _ / v0.5.2 on Arduino Uno

[15342] Modem init...
[16247] Connecting to network...
[16559] Network: Globe Telecom-PH
[16559] Connecting to internet.globe.com.ph ...
[23806] Connected to GPRS
[24086] Connecting to blynk-cloud.com:80
[27984] Ready (ping: 836ms).
[60425] Connecting to blynk-cloud.com:80
[62874] Ready (ping: 852ms).
[96156] Connecting to blynk-cloud.com:80

```

Fig. 9. Arduino IDE Serial Monitor—GSM modem initializing.

C. Establishment of Connection between the BLYNK Application and the GSM Module

This section shows the application when the GSM module is connected to the BLYNK application. Figure 10 shows the Arduino's response to establish the connection.

```

Initializing modem...
[15335]

  _ _  ) / /  _ _ _ _  / / /
 / _ / / / / / / _ \ \ ' \
 / _ _ / \ \ / / / / \ \ \
 / _ / v0.5.2 on Arduino Uno

[15342] Modem init...
[16247] Connecting to network...
[16559] Network: Globe Telecom-PH
[16559] Connecting to internet.globe.com.ph ...
[23806] Connected to GPRS
[24086] Connecting to blynk-cloud.com:80
[27984] Ready (ping: 836ms).
[60425] Connecting to blynk-cloud.com:80
[62874] Ready (ping: 852ms).
[96156] Connecting to blynk-cloud.com:80

```

Fig. 10. BLYNK is connected to the GSM module.

The researchers encountered some stability problems with the establishment of connection between the application and the GSM module. Figure 10 shows how the connection was established for a couple of seconds, and then it will be disconnected, and connected again afterwards.

D. Distance Test for RF Communication

Table 1 shows the summary of all the processes that can be performed by the RF module as mentioned above. Note that this evaluation was done while performing a distance test. As such, the table below shows the results for a distance test of up to 45 m.

TABLE 1
RF COMMUNICATION RANGE TESTING SUMMARY

Distance	TX With Antenna	RX With Antenna	Remarks
0 m	NO	NO	Transmit data
5 m	NO	NO	Failed
5 m	YES	NO	Transmit data
10 m	YES	NO	Transmit data
15 m	YES	NO	Failed
15 m	YES	YES	Transmit data
20 m	YES	YES	Transmit data
25 m	YES	YES	Transmit data
30 m	YES	YES	Transmit data
35 m	YES	YES	Transmit data
40 m	YES	YES	Failed
45 m	YES	YES	Failed

During testing, the transmission stability was poor when the distance was about on its limit (40 m). The base node, most of the time, can receive data as long as the receiver antenna was installed. Otherwise, the connection was not stable, causing it to poorly receive the arriving data. With the antenna installed, the transmission was successful up to 35 m.

E. Displaying of Data using BLYNK Application

The application successfully established the necessary hardware-to-hardware communication, hardware-to-software communication, hence it successfully displayed the data coming from the sensor node as shown in Figure 11. The researchers set the threshold value of each gauges, as shown in Table 2. So if the reading value of the temperature, for example, is above the threshold value, notification will appear, as shown in Figure 12.

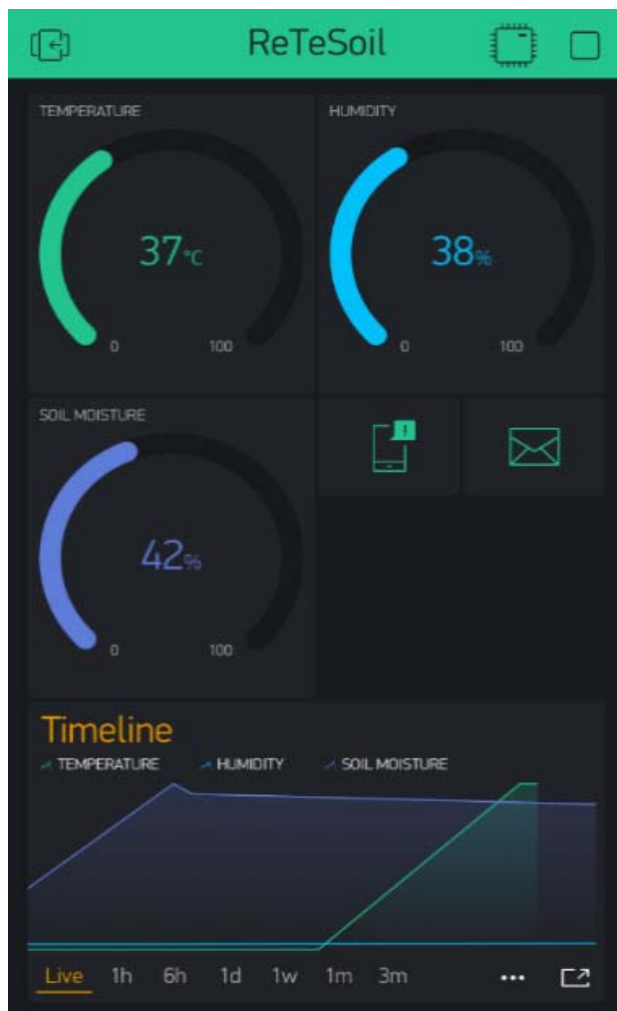


Fig. 11. Gauges reading.

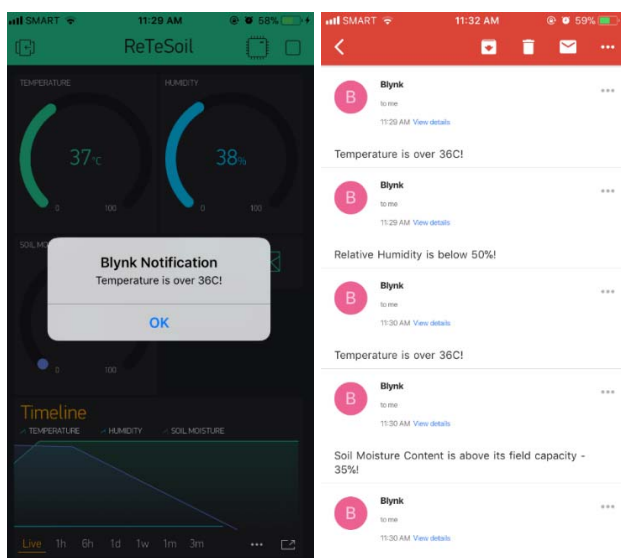


Fig. 12. Push and email notification.

TABLE 2
THRESHOLD SUMMARY

	Threshold Value
Temperature	36°C
Relative Humidity	50%
Soil Moisture	30% and 70%

The threshold value can be altered by changing the values in the source code. Once the values are set and the prototype is ready, it cannot be changed since the microcontroller was enclosed in a box.

V. CONCLUSION AND RECOMMENDATIONS

The researchers implemented a wireless monitoring system using BLYNK that can monitor temperature, relative humidity, and soil moisture content of the plant. With this, the sensors' data coming from the sensor node are displayed in the application with push notification and push email that allow users to remotely monitor the environment conditions in an area, if the threshold values are above or below the set limit. The researchers observed that the RF communication used in the base node connection has a maximum operating distance of 35 meters when both modules have antenna. Within this operating distance, data can be transmitted to the base node without any issues except when the distance is about its maximum limit (35 m). The initialization of GSM modem and connection to GPRS for base node and the BLYNK application communication was not stable. Delay was observed, it took a couple of seconds or minutes depending upon the signal strength of the network.

Future works may consider the improvement of the stability of data transmission from base node to BLYNK cloud by using Wi-Fi and/or other more stable communication. The use of a higher frequency operating RF modules can also be considered for wider range compare to 433-MHz RF modules that covers only 35 m, as well as, improving the hardware for both base node and sensor node. The use of 5-12 volts of power source with at least 1 Ampere rating to power up the Arduino board and GSM modem for the base node is highly recommended. This is also applicable for the sensor node since RF transmitting module consumes a lot of power when transmitting data remotely. Lastly, the use of a high-quality module and sensors can also be considered to avoid too much calibration.

REFERENCES

- [1] "Excessive and Irregular Irrigation Has the Following Effects," [Online]. Available: <http://www.biologydiscussion.com/articles/excessive-and-irregular-irrigation-has-the-following-effects/2403>. [Accessed 4 January 2018].
- [2] S. R. Kumbhar and A. P. Ghatule, "Microcontroller based Controlled Irrigation System for Plantation," p. 4, 2013.
- [3] H. Sahota, R. Kumar and A. Kamal, "A Wireless Sensor Network for Precision Agriculture and its Performance," p. 18, 2011.
- [4] G. R. Mendez, "A WiFi based Smart Wireless Sensor Network for an Agriculture Environment," Palmerston North, 2012.
- [5] S. Divyasri, and K. Babu, "Automated Irrigation System using a Wireless Sensor Network and GPRS Module," *International Journal of Innovative Technologies*, pp. 1307-1317, 2015.
- [6] N. Fhami, S. Huda, E. Prayitno, M. Rasyid, M. C. Roziqin and M. U. Pamenang, "A Prototype of Monitoring Precision Agriculture System Based on WSN," p. 6, 2017.
- [7] H. Mansour and Y. El-Melhem, "Impact the Automatic Control of Closed Circuits Drip Irrigation Systems on Yellow Corn Growth and Yield," p. 10, 2013.
- [8] M. S. Shah, U. A. Shaikh, and H. S. Doshi, "Internet of Things (IoT): Integration of Blynk for Domestic Usability," *Vishwakarma Journal of Engineering Research*, pp. 149-157, 2017.
- [9] Arduino, [Online]. Available: <http://arduino.cc>. [Accessed 6 May 2018].
- [10] A. Terzis, A. Szalay, and Q. Wang, "A Novel Soil Measuring Wireless Sensor Network," in *IEEE Instrumentation and Measurement Technology Conference Proceedings*, Austin, 2010.
- [11] S. M. A. Alim, M. S. Islam, K. B. R. Kawshik, F. Amid and S. Islam, "433 MHz (Wireless RF) Communication between Two Arduino UNO," *American Journal of Engineering Research*, pp. 358-362, 2016.
- [12] M. Maroti and A. Ledeczi, "Wireless Sensor Node Localization," 2011.
- [13] M. N. Jivani, "GSM Based Home Automation System Using App Inventor," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, pp. 12121-12128, 2014.
- [14] "Kickstarter," [Online]. Available: <http://create.arduino.cc/projecthub/vijayvikram11844/first-project-arduino-blynk-a5a297>.
- [15] Blynk, [Online]. Available: <http://docs.blynk.cc/>. [Accessed 6 May 2018].

A LabVIEW-Based Target Optimization Genetic Algorithm for Biological Predators

Ryann Alimuin,^{1,*} Elmer Dadios,² and Argel Bandala²

Abstract—Initial projection of a continuously repositioning target is a setback in genetic algorithm; a GA program needs constant input sampling to predict and declare a targets status. Another difficulty is the incorporation of GA to hardware and software which considered as the most important tool in sensor integration. Familiarity in programming is essential in utilizing the NI LabVIEW and NI myDAQ environment. The aim of this research paper is to provide a solution for determining the locus (gene position) of a target through distinctly employed multiple sensors which employs low-frequency (LF) ground-wave oscillations as its signal sources. The targets' position as well as the speed is continuously monitored through virtual instrument (VI) software; the user will be able to visually analyze the constant system mutation plots and the number of completed generations. Upon completion on the number of generations, the plot points can be imported to a spreadsheet for further analysis. The multiplatform software will be able to plot the response under real-time circumstances.

Keywords: LabVIEW, locus, mutation, low-frequency, multiplatform

I. INTRODUCTION

Today, multiple programming methods are applied to perform different tasks; one of them is the genetic algorithm (GA). GA in artificial intelligence is a type of evolutionary computer algorithm in which symbols (often called “genes” or “chromosomes”) representing diverse solutions undergo a process called crossover or sometimes addressed as breeding. This process involves recombination of possible genes and multiple mutations at a specific rate. GA mimics the way evolution acts and allows us to improve the performance of controllers or

adapt them to different systems [1]. GA is a probabilistic behavior, global searching, and optimization algorithm which is primarily intended to solve complex problems [2].

In comparison with common classical algorithms, GA has loads of advantages, such as a widely viable solution, searching in small and large groups, assistance without further information on the target, internal heuristic random search, parallel computing, etc. [3]. GA is a process for solving both forced and unrestrained optimization problems that is based on natural selection; the mutation process drives biological evolution [4]. It is mainly an overall arbitrary search and optimization method which aims to emulate the natural biological evolution [5].

In the livestock industry, one of the major causes of destructive loss is reptilian predators; these are hunters that feed on fowls and their by-product. This research aims to employ this method under a different target such as predators and take consideration of some parameters such as speed of movement and location status. Through LabVIEW, a programming platform which is widely utilized due to its simple graphical environment yet able to adopt to diverse interface on various hardware and software [6], the user is enabled to analyze the result through point-plotting approach. Unlike other object-oriented programming, LabVIEW allows wiring graphical objects in block diagrams which function as various utilities and modules [7]. The NI DAQ (Data Acquisition) hardware is an external module interface of LabVIEW which consists of several modules that can imitate analog-to-digital converters (ADC), signal conditioners, isolators, filters, and interfacing circuits [8].

II. PRIOR RELATED WORK

LabVIEW GA is a virtual instrument that integrates fitness evaluation, mutations, crossovers and selection [9] through prediction modelling, determination of different signals coming from multiple sources where attenuation and interferences can occur can be used [10].

Animal behavioral studies have been recently being studied especially on the use of sound pressure [11], [12] especially on common reptiles such as snakes through

¹Ryann Alimuin, Technological Institute of the Philippines, Quezon City, Philippines (e-mail: ryann.alimuin@tip.edu.ph)

²Elmer Dadios and Argel Bandala, De La Salle University–Manila, Philippines

vibration stimulation [13]. At 150–450 Hz in the range of 65–75 dBre at 20 mPa, Young and Aguiar observed a substantial decrease in the rattlesnake's *C. atrox* bodily movement and tongue flicking whilst showing a significant rise in numbers of head twitches and tail rattles [11]. Young and Morain added that olfactory-denervated and temporarily blinded Saharan sand vipers' (*Cerastes cerastes*) striking distance, angle, and accuracy were significantly reduced in capturing a free-running prey while employing only target vibrations [13]. Using GA, animal modeling rationalizes non-random mating and complex data organizations which utilizes parental phenotypes and offspring [14].

III. COMPONENTS AND SYSTEM ARCHITECTURE REQUIREMENTS

The system composition is a LabVIEW Virtual Instrument (VI) that employs a block diagram, front panel, and a DAQmx assistant which serves as a medium for acquiring data. The intent of the experiment is to record and plot multiple generations of data coming from the motion sensors. The sets of data that are sent to the computer are the position as well as the speed of the target. The program will be able to predict the succeeding point location where the target would move through a series of consecutive mutations and generations.

A. The following are the hardware components of the GA system:

1. Motion sensors/transducers
2. Signal transceivers/transducers
3. NI myDAQ
4. Personal computer/laptop
5. NI LabVIEW
6. Dual polarity power supply

B. The following are the computer hardware specifications necessary to operate the GA structure:

1. Pentium 4M processor or higher
2. 1 GB of RAM or higher
3. 32- or 64-bit Windows 7/8/8.1/10
4. Free 20-GB hard drive storage for system software requirements
5. Screen resolution of at least 1024×768 pixels

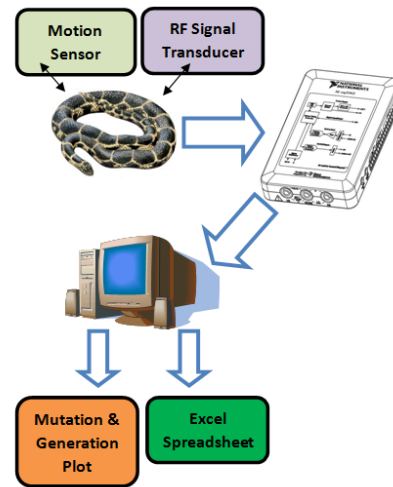


Fig. 1. System block diagram.

The system as shown in Figure 1 consists of two pre-positioned transducers, a motion sensor for detecting the rate of the specimen movement, and an RF signal transducer comprised of a ground-wave oscillator for quantifying the position of the target.

The transducers continuously monitor the movement of the target such as change in speed and position and send it to the data acquisition hardware. The GA virtual instrument takes the samples (parents) and generates consecutive mutations within the user assigned number of iterations. The VI will display the optimum value of mutation and plots the activity. The accumulated results can be analyzed through an integrated spreadsheet.

The LabVIEW VI is composed of the following:

1. Waveform chart for signal analysis
2. Generation progress monitor
3. Value indicator
4. Generation control
5. Mutation probability control
6. Interrupter button
7. Time estimator
8. Time generation monitor
9. "Save on spreadsheet" button

The hardware and software system requirements stated are based upon the minimum requirements needed to run LabVIEW 2012 to 2015 and NI myDAQ. Higher specification computers can increase the boot speed of the program as well as the necessary modules; this will also affect the effectiveness of system simulation.

IV. HARDWARE AND SOFTWARE INTEGRATION

The hardware and software integration is implemented through the following procedures:

- A motion sensor is installed in a defined location with minimal irregular geographic terrain.
- An RF signal transducer is utilized to monitor the location of the specimen.
- The transducers are connected to an NI MyDAQ signal acquisition hardware to integrate LabVIEW.
- The LabVIEW program acquires the samples and applies genetic algorithm to perform mutations.
- The user defines the number of generations that will be iterated by the program.
- LabVIEW GA plots each performed mutation and restarts after completing the defined generations.

The circuit in Figure 2 is the RF oscillator that serves as an input to the signal transmitter. It is comprised of multiple square wave generators and sinusoid converters. The waveforms produced are generally combined by a masking stage and are used to generate ground wave oscillations.

Using roulette selection [15], let the population size be p , the population members be c_i for $i \leq p$, and the fitness of chromosome be written $f(c)$ and the wheel size W by

$$W = \sum_i^p f(c_i)$$

Selecting a chromosome,

$$j < \sum_i^k f(c_i)$$

where each $f(c)$ slots corresponds to each chromosome. Assigning slot counts according to

$$f'(c) = F - f(c) + s$$

where

$$F = \max f(c_i)$$

The fitness function is given by with if and only if x is a solution.

The loose fitness function satisfies with if x is a solution where the sum of any member of fitness functions with non-negative weights is also a fitness function.

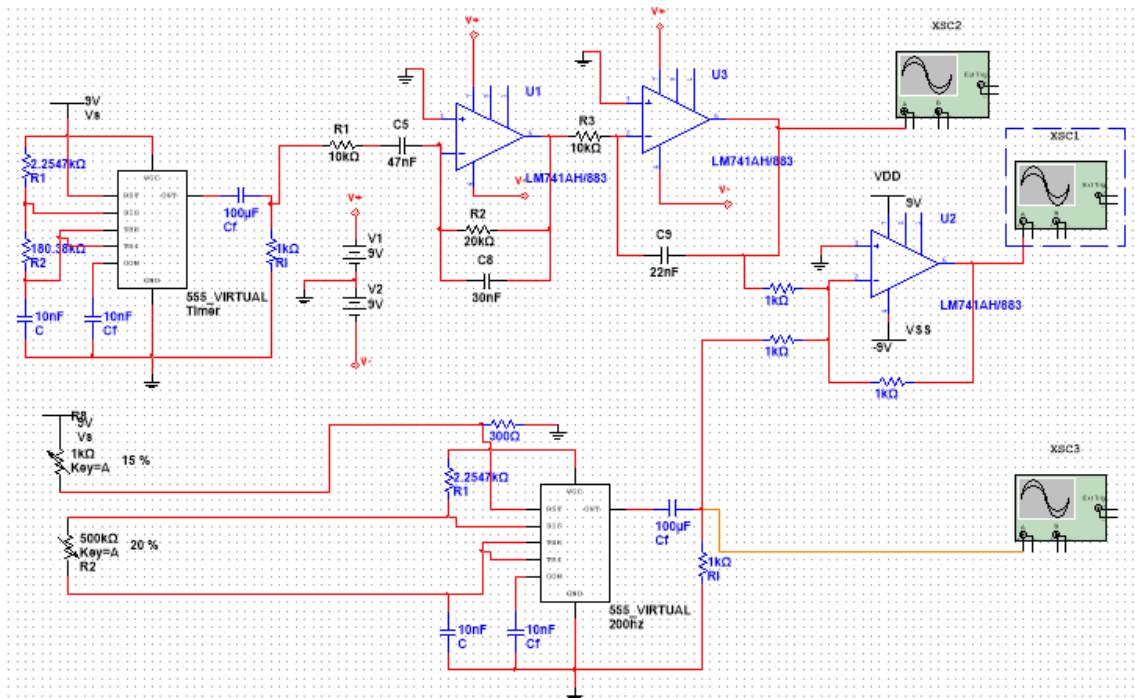


Fig. 2. RF oscillator schematic diagram.

V. SYSTEM FUNCTIONALITY DEMONSTRATION

The block diagram, as shown in Figure 3, illustrates the main VI program of the GA system. It is also composed of multiple sub-VIs which perform the mutation of each input sample. The DAQ Assistant serves as the interface between the external experiment proper and the software. The speed of iterations varies depending upon the capacity of the simulating CPU.

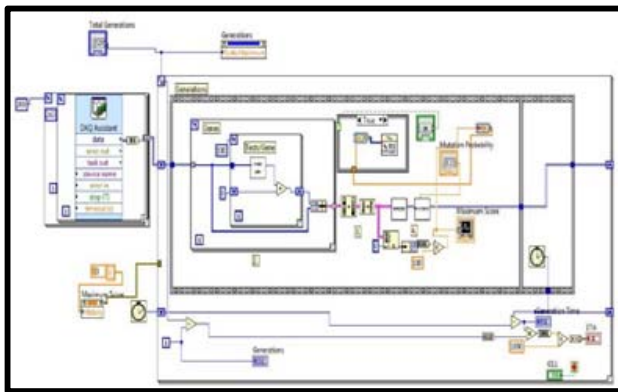


Fig. 3. GA system block diagram.

The virtual instrument is simulated and displays the plot on the number of mutations and completed generations. The optimum mutation probability is 0.006. The most effective number of generations should be at least 100 before it reaches the best fitness.

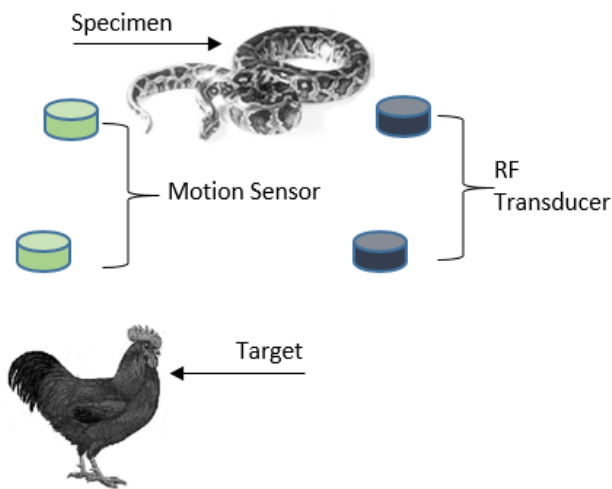


Fig. 4. Specimen and target setup.

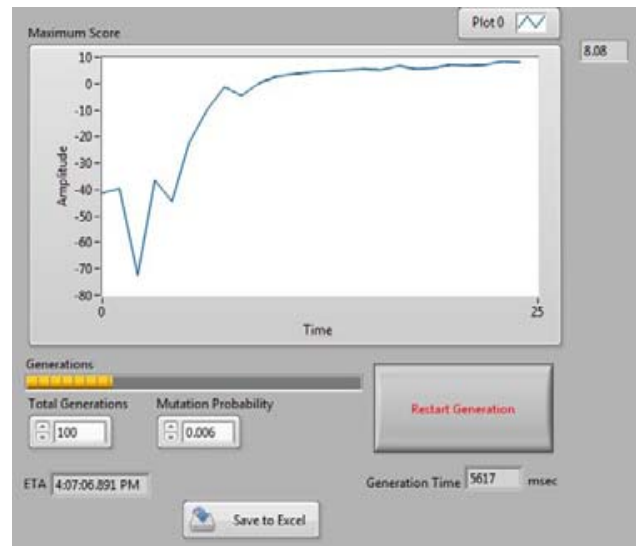


Fig. 5. First-generation plot.

Figure 5 shows the plot of the first generation. First mutations are low, yet the best fit value generated by the VI reaches an almost constant score.

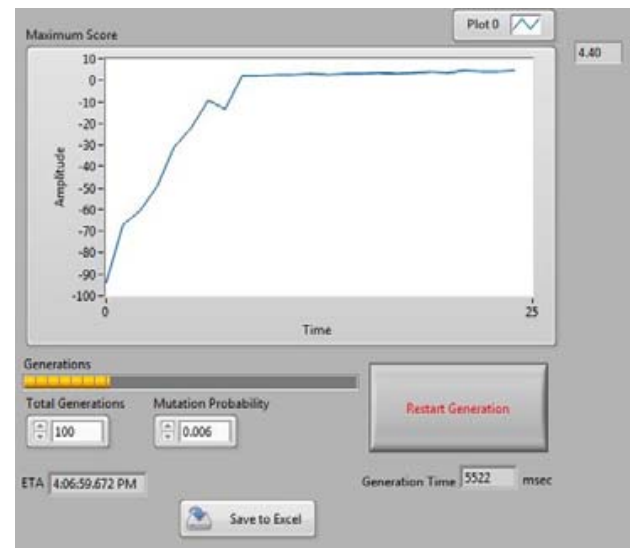


Fig. 6. Second-generation plot.

Figure 6 shows the plot of the second generation. First mutations are relatively lower than the first, yet the optimally fit value generated still reaches an almost constant score.

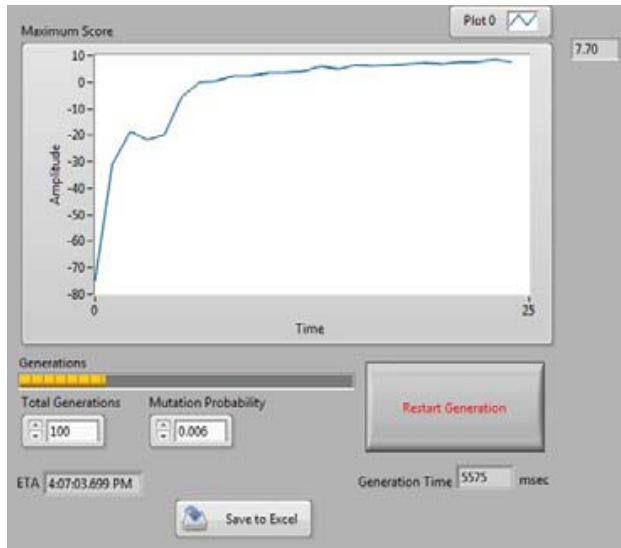


Fig. 7. Third-generation plot.

In Figure 7, the first mutations are low, but while finding the best fitness, the program still arrived at the optimum score which is nearly constant.

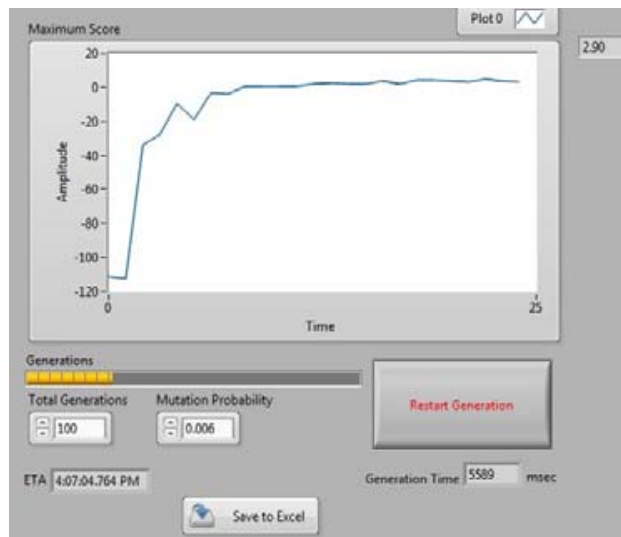


Fig. 8. Fourth-generation plot.

In the fourth generation as shown in Figure 8, the first samples started extremely low but still managed to recover and attained a maximum score throughout the generation.

Figure 9 illustrates the four generations of mutations done by the system. As shown, multiple iterations are repeated for mutation of the acquired samples. At the start of the experiment, mutations are low; however, the system continuous to attempt a prediction on how the specimen would succeed in reaching its target. The VI requires at least 11 ms before it settles for the best fitness outcome.

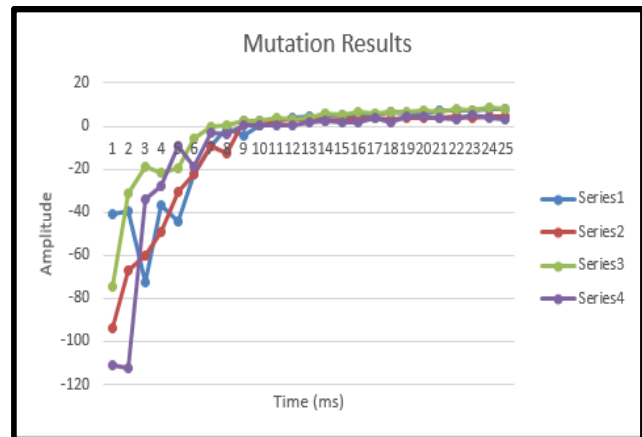


Fig. 9. Mutation results.

VI. CONCLUSIONS AND RECOMMENDATIONS

The point plotting GA system was able to execute locus plots through the reception of sensor data. Mutations of several samples on the target specimen's velocity and position were obtained and monitored through NI LabVIEW as well as the importation on a spreadsheet summary for further analysis and assessment. The VI software can be modified to perform other tasks which allows gathering of other parameters of moving samples.

The system structure can be enhanced through several software and hardware modifications. These enhancements will improve genetic mutations on movement predictions, namely,

- Increase in CPU processing speed by utilizing greater hardware specifications,
- Replacement of a higher sensitivity reception transducers,
- Modifications on the LabVIEW program for generation mutations, and
- Placement of transmitter and receiver transducers on terrains with less inclinations as well as geographical obstructions.

ACKNOWLEDGEMENTS

The research proponents express their deep gratitude to the Technological Institute of the Philippines–Quezon City and De La Salle University for the collaborative research and experimentation. We would also acknowledge their permission on the use of relevant equipment and apparatuses for the design.

REFERENCES

- [1] P. Ponce, F. Ramirez and V. Medina, "A Novel Neuro-Fuzzy Controller Genetically Enhanced Using LabVIEW," *IEEE*, pp. 1559-1565, 2008.
- [2] X. Li, Y. Zhang, S. Wang and G. Zhai, "Analog Circuits Fault Diagnosis by GA-RBF Neural Network and Virtual Instruments," *IEEE*, pp. 236-239, 2012.
- [3] J. Li and M. Zhu, "An Implementation of Genetic Algorithm in Matlab," *IEEE*, pp. 638-641, 2010.
- [4] S. Jaganathan, A. Sekar and W. Gao, "Formulation of Loss minimization Problem Using Genetic Algorithm and Line-Flow-based Equations," *IEEE*, pp. 1-6, 2008.
- [5] C. Zhang and J. Chen, "Methods of Optimization of Milling Parameters Based on Genetic Algorithm," in *The Ninth International Conference on Electronic Measurement & Instruments*, Nanjing, China, 2009.
- [6] P. Mishra, V. Kumar, K. P. S. Rana, S. S. Nair and J. Kumar, "Cuckoo Search Implementation in LabVIEW," *IEEE*, pp. 1-6, 2016.
- [7] R. Turley and M. Wright, "Developing Engine Test Software in LabVIEW," *IEEE*, pp. 575-579, 1997.
- [8] M. Murugappan, R. Thirumani, M. I. Omar and S. Murugappan, "Development of Cost Effective ECG Data Acquisition System for Clinical Applications using LabVIEW," in *IEEE*, Kuala Lumpur, Malaysia, 2014.
- [9] V. Kumar, K. P. S. Rana, A. Kumar, R. Sharma, P. Mishra and S. Nair, "Development of a Genetic Algorithm Toolkit in LabVIEW," in *Third International Conference on Soft Computing for Problem Solving*, India, 2014.
- [10] S. K. Bhargav, B. V. Ramana, K. C. B. Rao and P. S. Kumar, "Prediction and Analysis of Electromagnetic Interference in Radio Frequency Propagation for Transmitter - Receiver pair using LabVIEW," in *International Conference on Electromagnetic Interference Compatibility and Workshop*, India, 2016.
- [11] R. Shine, "Reticulated pythons in Sumatra: biology, harvesting and sustainability," *Elsevier*, 1999.
- [12] B. A. Young and A. Aguiar, "Response of western diamondback rattlesnakes *Crotalus atrox* to airborne sounds," *Journal of Experimental Biology*, pp. 205, 3087-3092, 2002.
- [13] B. A. Young and Harris, "Auditory sensitivity of the northern pacific rattlesnake, *Crotalus viridis oregonus* : do behavioral responses conform to physiological performance?," *Northwest Science*, pp. 80, 218-223, 2006.
- [14] J. Odegard, T. H. Meuwissen, B. Heringstad and P. Madsen, "A simple algorithm to estimate genetic variance in an animal threshold model using Bayesian inference," *BioMed Central*, pp. 1-7, 2010.
- [15] R. Booth, D. Bormotov and A. Borovik, "Genetic Algorithm and Equations in Free Groups and Semi Groups," *American Mathematical Society*, pp. 63-80, 2003.

Hybrid Sensor Based Fuzzy Clustering Neural Network Classification for Human Activity Recognition

Ramtin Aminpour¹ and Elmer Dadios²

Abstract—The smartphone is going to become an all-purpose gadget for the human life and all of them at least armed with accelerometer sensor. In this study, the fuzzy c-means has been considered in the ANFIS model to produce the fuzzy inference system (FIS) to make the classification with the neural network algorithm to detect the six major human activities. The data were taken in real life with the accelerometer sensor of a smartphone. The results of the experiments show that the 97.2% accuracy could be acceptable in the field of study and the clustering structure could make the simulation more robust and faster.

Keywords: Fuzzy clustering, Neural network, Human activity recognition

I. INTRODUCTION

Human activity recognition is widely used in human-computer interaction, fitness tracking, and maintenance of elderly people [1]–[4]. The data from activities could be gathered by camera sensors, audio sensors, and embedded sensors [5]. Nowadays, smartphones are personal importance in the societies. All of the smartphones have internal sensors to collect data with low power consumption and powerful processors. One of the most useful embedded sensors of the smartphones is the accelerometer which could be used to collect extensive data on human activities [6].

In this study, a Neuro-Fuzzy inference system combined with the fully-connected layer neural network has been used to classify the human activities. Integration of neural networks and fuzzy logic systems could be a

hybrid approach to the model of a system. The study of the architecture is shown in Figure 1.

II. RELATED WORK

Applications of computational intelligence have been used in many fields based on neural network, fuzzy logic, genetic algorithm, and hybrid approaches of these algorithms. NeuroFuzzy has been proposed by Jang [7] with the concept of integration with human and learning capability of the structure of the neural network. Neuro-Fuzzy systems could be powerful solutions in many applications [8], [9]. Neural network and fuzzy logic are dynamic with the ability of parallel processing to approximate the input and output functions.

One of the problems of fuzzy design is the difficulty to determine the number of rules and also the number of membership functions of every rule. Hybrid algorithms can optimize the systems to the trade of this problem [10], [11]. Hybrid algorithms have been used for fine-tuning and learning of parameters of the neural network and fuzzy logic. Lin [12] proposed a Takagi Sugeno (TS) type of fuzzy model with a hybrid learning algorithmic rule. The approach was used to modify the mean and the deviation of the membership functions. A combined Takagi Sugeno type Neuro-Fuzzy system has been done with the bee colony algorithm for parameter optimization [13].

A comparative study for classification of ECG signals with MultiLayer backPropagation learning (MLP) has been done by Ozbay [14]. They introduced a Fuzzy Clustering Method (FCM) based neural network, which shows faster and better accuracy to compare of ordinary MLP architecture.

Kim et al. [15] have developed the FCM algorithm for the color clustering problems. Mingoti [16] performed a clustering algorithm based on Self-Organizing Map (SOM) neural network and FCM and they found that the performance of the algorithm was improved in the presence of outliers. A hybrid study of Support Vector Machine (SVM) and FCM for gene dataset has been done

¹ Ramtin Aminpour, Department of Electronics and Communications Engineering, De La Salle University, Manila, Philippines (email: ramtin.aminpour@dlsu.edu.ph)

² Elmer Dadios, Department of Manufacturing Engineering and Management, De La Salle University, Manila, Philippines (email: elmer.dadios@dlsu.edu.ph)

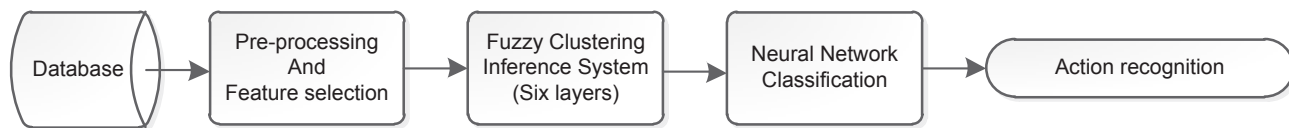


Fig. 1. Architecture of the FCM neural network model

by Mukhopadhyay [17]. Aydilek [18] proposed a hybrid approach of Genetic Algorithm (GA) with FCM and SVM to estimate missing values and optimization of the size of a cluster.

The hybrid algorithm could be an optimal way of a solution for system design. The learning algorithm of Neuro-Fuzzy is based on a steep descent optimization like the backpropagation algorithm. Similar to other gradient methods, steep descent optimization cannot prove to converge to the global solution. On the other hand, the parameters of the membership functions cannot be tuned in and modified. In these situations, optimization with hybrid algorithms could give an efficient result.

III. DATASET DESCRIPTION AND FEATURE EXTRACTION

An Android smartphone has been used to collect the data in this study. The accelerometer has been collected the data with a sample frequency rate of 50 Hz from seven adult persons to detect six activities of walking, jogging, running, jumping, using stairs, and standing. The smartphone was placed in the front pocket of the volunteers. Each activity was repeated five times during a period of 30 seconds. After collecting the data, pre-processing of the data by a low-pass filter has been done to clean the raw data from missing data, noise reduction, and outlier detection.

To detect the activities, proper feature set must be extracted from the raw data. A sliding window with the length of 2.5 seconds is used to separate the data. The acceleration sensor of the smartphone has three dimensions in which four features from time domain and two features of frequency domain have been extracted. In total 18 features have been selected to extract from X, Y, and Z axes of the accelerometer. These features are shown in Table I. In the Table x_i are the features, N is the total number of features, and FFT is the Fast Fourier Transform of the features.

The data might have a different scale because of the gait, the height and the weight of volunteers. For this reason, the

TABLE 1
FEATURES EXTRACTION FROM ACCELEROMETER OF SMARTPHONE

Features	Math explanation
Mean Absolute Value	$\frac{1}{N} \sum_{i=1}^N x_i$ (1)
Variance	$\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$ (2)
Root Mean Square	$\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i)^2}$ (3)
Skewness	$\frac{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^3}{\sigma^{3/2}}$ (4)
Dominant frequency	$Max(FFT)^2$ (5)
Energy	$\sum (FFT)^2$ (6)

data must be normalized to zero means before using into the classification algorithm.

$$f_{normalized(i)} = \frac{f_{raw(i)} - \mu}{\sigma} \quad (7)$$

Here μ is mean and σ is the standard deviation of the features data. All the raw data replaced with normalized data to build the features matrix.

IV. CLASSIFICATION MODEL ARCHITECTURE

The structure of the adaptive neuro-fuzzy framework model (ANFIS) consists of input variables and output variables with a Takagi-Sugeno type of rule-based. The type of this network is an adaptive, multi-layer, and feed-forward [7]. The rule sets are evaluated by linguistic variables, a linear combination between input values, and a constant parameter which is employed to obtain the result of the rules.

In the first layer of this network, input variables are mapped into fuzzy sets through the process of fuzzification and generate membership grades with a membership function such as triangular or sigmoid. The next layer is working as a fuzzy conjunction to combine the fuzzy sets on each input. Another layer calculates the ratio of the rule that fire to the sum of all firing strength and then multiplied by the function of TKS fuzzy rules and extract the sum of all outputs of each rule. Finally, the crisp fuzzy output is calculated during the process of defuzzification by the weighted average method.

The ANFIS has two types of parameters, linear parameters on the consequent part and nonlinear parameters in the commence part. To optimize these variables, ANFIS has several strategies such as gradient descent, steepest descent, and hybrid technique [7]. In the consequent part, the output of each layer is transferred toward the last layer and the parameters are adjusted by the least squares method. In the premise part, the error signals are transferred to the first layer and update the parameters by the gradient descent method. The processes are done with the limited number of iterations during the epochs. The error between the output and also the target goes to minimize by weight adjustment of the connections throughout the learning method [19].

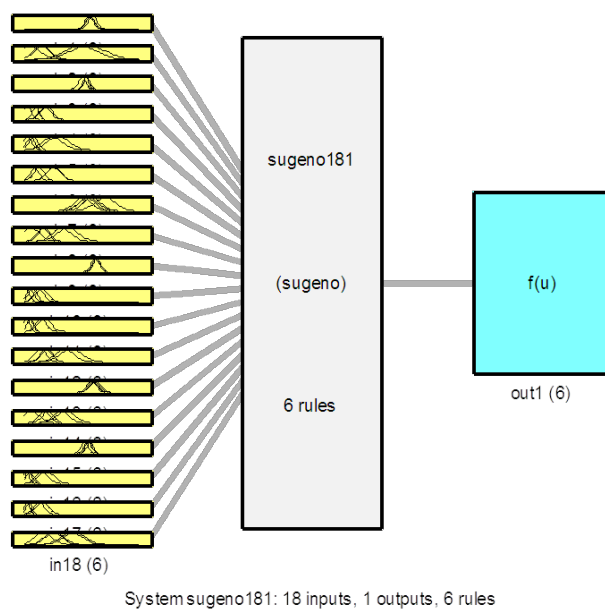


Fig. 2. Sugeno neuro fuzzy clustering model

The Sugeno Neuro-fuzzy clustering for this study is shown in Figure 2.

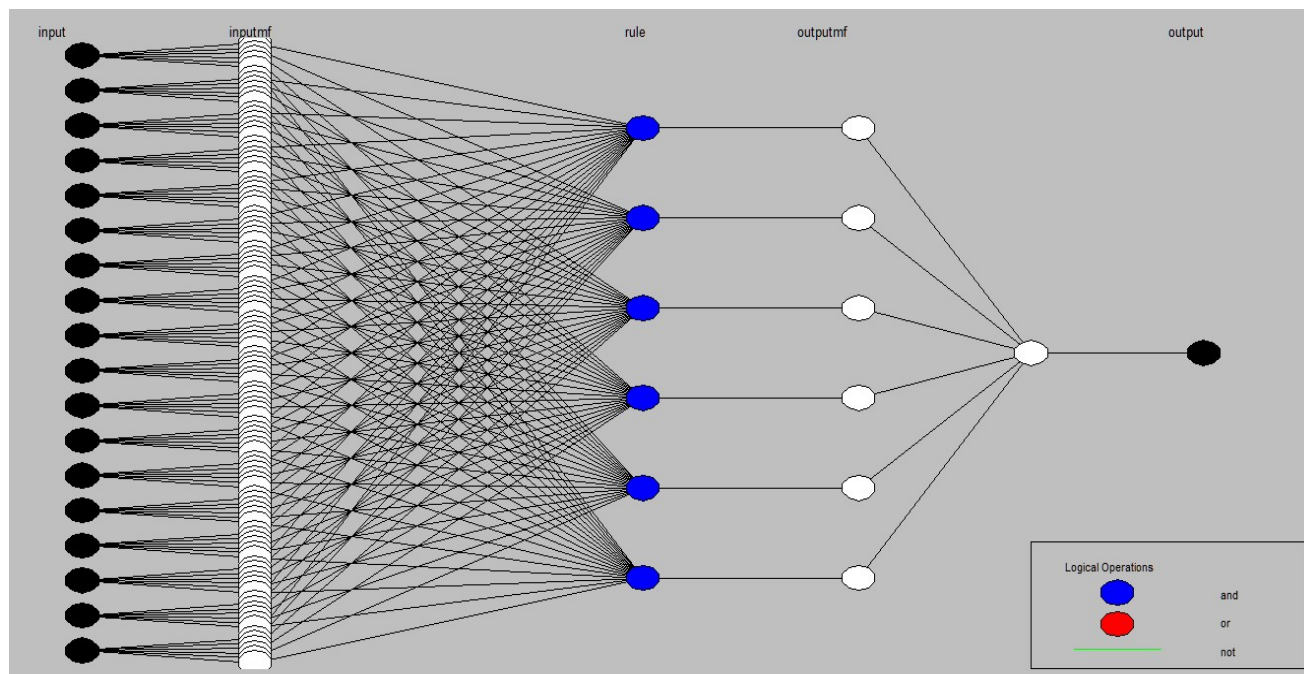


Fig. 3. ANFIS structure model

A. ANFIS Model Building

In this study, the data have been separated into three parts. Seventy percent have been used during the training of the algorithm, 15% as a checking data to prevent of the over-fitting of the model, and remaining 15% for testing of the model to check the predicted ability of the algorithm. During the training phase of FCM, the parameters are determined automatically in the specific epochs to minimize the checking error. Samples of the dataset for walking and running activities are given in Table II.

There are three methods to generate the fuzzy inference system (FIS) structure in ANFIS, grid partition, subtractive clustering, and fuzzy c-means clustering [20]. The fuzzy clustering method and the neural network which have been used as a classification are explained below. Because of six activities, the simulation has six different outputs. The ANFIS structure of one of the outputs of this study is shown in Figure 3.

B. Fuzzy Clustering Method

Fuzzy inference system (FIS) could be generated by fuzzy c-means (FCM) clustering. The FCM structure

can generate the rules from the behavior of the data and determine the number of rules and membership functions of the input and output variables of the algorithm. The number of clusters can be set in the FCM algorithm. The bigger cluster radius generates fewer clusters and fewer rules during the process of generating FIS. In the Takagi Sugeno structure, the membership functions for input and output are set with Gaussian and linear, respectively.

The steps of a fuzzy clustering algorithm are as follows [21], [22]:

- 1) Initialize the number of clusters.
- 2) The fuzzification step which is selecting a metric Euclidean norm and the weighting metric.
- 3) Initialize the cluster prototype and iterative counter.
- 4) Calculate the partition matrix.
- 5) Update the fuzzy cluster centers.
- 6) If the norm of cluster centers was smaller than epsilon then stop the algorithm otherwise, repeat step two up to four.

TABLE 2
SAMPLES OF DATASET FOR TWO ACTIVITIES

<i>Walking_x</i>	<i>Walking_y</i>	<i>Walking_z</i>	<i>Running_x</i>	<i>Running_y</i>	<i>Running_z</i>
-0.03	-1.2	-0.66	0.6	0.23	0.24
-0.03	-1.2	-0.66	0.1	0.37	0.11
0.05	-0.79	0.05	-0.33	-0.2	-0.18
0.41	-1.06	0.16	-0.33	-0.2	-0.18
0.41	-1.06	0.16	1.36	-1.32	1.26
-1.27	-1.5	-0.15	-1.25	-1.99	0.22
-0.45	-0.18	-0.07	-0.13	-0.01	0.04
-0.45	-0.18	-0.07	-0.13	-0.01	0.04
-0.5	-0.64	0.44	0.81	-1.99	-0.55
0.06	-1.05	-0.59	0.81	-1.99	-0.55

TABLE 3
THE CONFUSION MATRIX OF ACTIVITIES FOR CLASSIFIER

Activity	Walking	Jogging	Running	Hopping	Using stairs	Idle
Walking	2991	64	5	0	33	5
Jogging	16	1784	32	2	0	2
Running	11	19	1531	9	5	0
Hopping	24	8	2	14	1366	8
Using stairs	24	8	2	14	1366	8
Idle	3	0	0	0	8	1679

The feature data have been grouped into six clusters to distinguish the six activities of the target. Six features are extracted from each axis of the accelerometer. In total, 18 features are used to detect the activity of each class. As a result, in this study, 108 fuzzy membership functions are generated to distinguish the output activities. Some samples of fuzzy membership functions for the first output of the target are shown in Figure 4, Figure 5, and Figure 6. The Gaussian membership functions have been chosen automatically by fuzzy inference system (FIS) in contrast with the fuzzy logic algorithm.

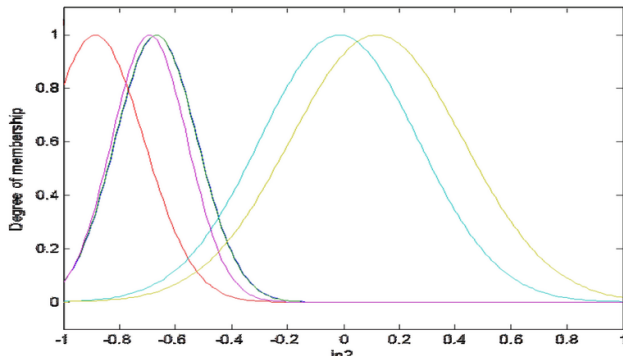


Fig. 4. Fuzzy membership of Input 2 for the first class

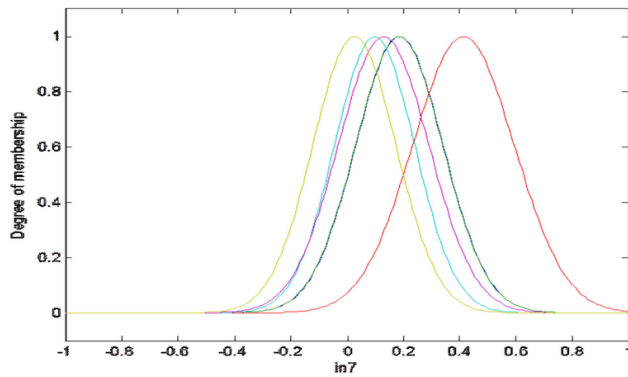


Fig. 5. Fuzzy membership of Input 7 for the first class

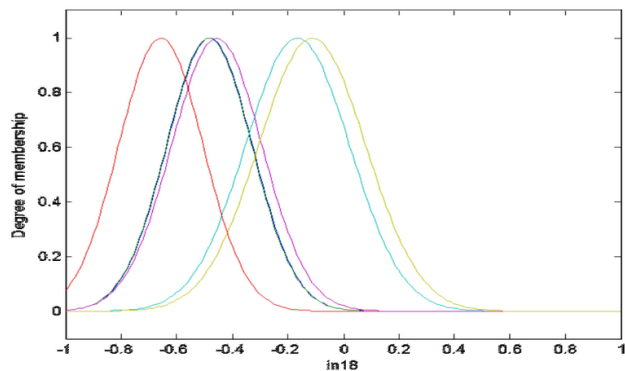


Fig. 6. Fuzzy membership of Input 18 for the first class

V. EXPERIMENT AND ANALYSIS OF RESULTS

To detect the activities the features data have been clustered to the number of activities by the fuzzy clustering algorithm. In the next step, the output of FCM has passed through a full node connection neural network to classify the human activities. The pattern recognition network has 18 neurons in the hidden layer, the training function of the network has been selected by Levenberg-Marquardt, and the performance function was chosen by mean square error. The features data of FCM have been divided randomly in the training ratio of 70%, the validation ratio of 15%, and the remained 15% for the test of the algorithm.

The result of confusion matrix and ROC curve are shown in Table III and Figure 7 respectively. The ROC curves which are closer to the left part of the plot have better classification accuracy. The confusion table shows that the most confused about activities are between jogging and walking, and using stairs and walking. The simulation shows that the accuracy of this classification is 97.2%. The output performance of the algorithm is shown in Figure 8. In the performance of the simulation, after 30 epochs the training stopped due to an increase of validation error against the minimum mean square error (MSE) parameter. The confusion table of the classifier shows that maximum errors are between jogging with walking, running with jogging, and using stairs with walking activities.

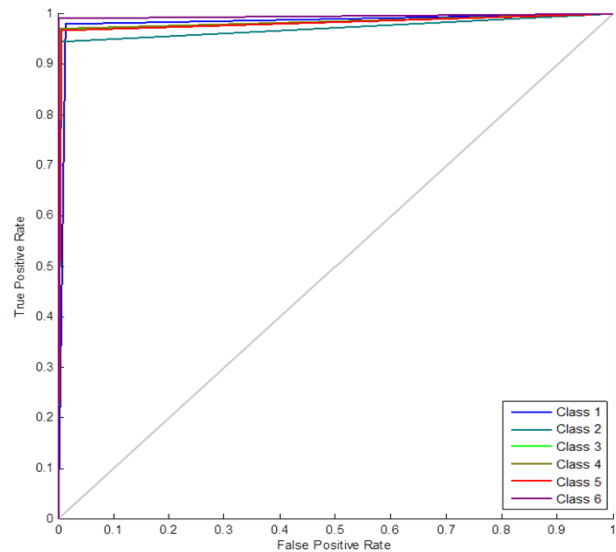


Fig. 7. ROC curve of classification

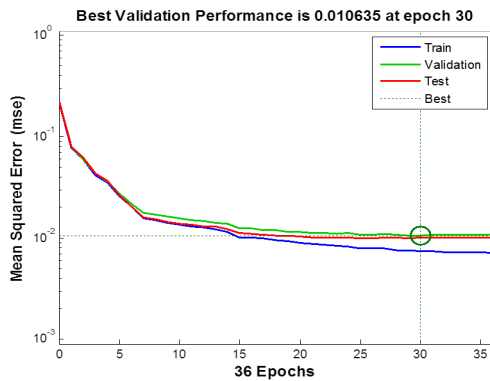


Fig. 8. Performance of simulation model

VI. CONCLUSION AND FUTURE WORK

In this study, the hybrid fuzzy clustering with the neural network algorithm is used to classify and recognize the human activities. The fuzzy c-means is used to generate the fuzzy inference system to develop the ANFIS based method.

Using the hybrid fuzzy cluster algorithm proves that the developed ANFIS model is more robust and faster than classic algorithms with an acceptable accuracy to compare of related works in the relevant area.

In the future works, the hybrid Fuzzy Clustering Neural Network algorithm could be compared to another type of hybrid fuzzy cluster algorithms such as Ant colony optimization, Differential evolution, or Particle swarm optimization.

ACKNOWLEDGEMENT

The author would like to thank the Intelligent Systems Laboratory of De La Salle University for providing the facilities in pursuing this research study.

REFERENCES

- [1] J. W. Lockhart, T. Pulickal, and G. M. Weiss, "Applications of mobile activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 1054–1058.
- [2] C. P. Dadula and E. P. Dadios, "Fuzzy logic system for abnormal audio event detection using mel frequency cepstral coefficients," *Journal of advanced computational intelligence and intelligent informatics*, vol. 21, no. 2, pp. 205–210, 2017.
- [3] C. Dadula and E. Dadios, "Event detection using adaptive neuro fuzzy inference system for a public transport vehicle," in *11th International Conference of the Eastern Asia Society for Transportation Studies*, 2016.
- [4] E. P. Dadios, J. J. C. Biliran, R.-R. G. Garcia, D. Johnson, and A. R. B. Valencia, "Humanoid robot: Design and fuzzy logic control technique for its intelligent behaviors," in *Fuzzy Logic-Controls, Concepts, Theories and Applications*. InTech, 2012.
- [5] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
- [6] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Pattern Recognition and Image Analysis*. Springer, 2011, pp. 289–296.
- [7] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [8] G. Castellano, C. Castiello, A. M. Fanelli, and L. Jain, "Evolutionary neuro-fuzzy systems and applications," in *Advances in Evolutionary Computing for System Design*. Springer, 2007, pp. 11–45.
- [9] S. Kar, S. Das, and P. K. Ghosh, "Applications of neuro fuzzy systems: A brief review and future outline," *Applied Soft Computing*, vol. 15, pp. 243–259, 2014.
- [10] C.-H. Chen and Y.-Y. Liao, "An efficient cluster-based tribes optimization algorithm for functional-link-based neurofuzzy inference systems," *Applied Soft Computing*, vol. 13, no. 5, pp. 2261–2271, 2013.
- [11] W. S. Liew, M. Seera, C. K. Loo, and E. Lim, "Affect classification using genetic-optimized ensembles of fuzzy ARTMAPs," *Applied Soft Computing*, vol. 27, pp. 53–63, 2015.
- [12] C.-J. Lin and Y.-J. Xu, "Design of neuro-fuzzy systems using a hybrid evolutionary learning algorithm," *Journal of information science and engineering*, vol. 23, no. 2, pp. 463–477, 2007.
- [13] Y.-T. Liu, Y.-Y. Lin, T.-Y. Hsieh, S.-L. Wu, and C.-T. Lin, "A global optimized neuro-fuzzy system using artificial bee colony evolutionary algorithm," in *Intelligent Systems and Applications: Proceedings of the International Computer Symposium (ICS) Held at Taichung, Taiwan, December 1214*, 2014, vol. 274. IOS Press, 2015, p. 140.
- [14] Y. O' zbay, R. Ceylan, and B. Karlik, "A fuzzy clustering neural network architecture for classification of ECG arrhythmias," *Computers in Biology and Medicine*, vol. 36, no. 4, pp. 376–388, 2006.
- [15] D.-W. Kim, K. H. Lee, and D. Lee, "A novel initialization scheme for the fuzzy c-means algorithm for color clustering," *Pattern Recognition Letters*, vol. 25, no. 2, pp. 227–237, 2004.
- [16] S. A. Mingoti and J. O. Lima, "Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms," *European journal of operational research*, vol. 174, no. 3, pp. 1742–1759, 2006.
- [17] A. Mukhopadhyay and U. Maulik, "Towards improving fuzzy clustering using support vector machine: Application to gene expression data," *Pattern Recognition*, vol. 42, no. 11, pp. 2744–2763, 2009.
- [18] I. B. Aydilek and A. Arslan, "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm," *Information Sciences*, vol. 233, pp. 25–35, 2013.
- [19] J.-S. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378–406, 1995.
- [20] "MATLAB and fuzzy logic Toolbox Release 2015b, The MathWorks, Inc., Natick, Massachusetts, United States."
- [21] T. A. Runkler and C. Katz, "Fuzzy clustering by particle swarm optimization," in *Fuzzy Systems, 2006 IEEE International Conference on IEEE*, 2006, pp. 601–608.
- [22] M. Huang, Z. Xia, H. Wang, Q. Zeng, and Q. Wang, "The range of the value for the fuzzifier of the fuzzy c-means algorithm," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2280–2284, 2012.

Implementation of k-Nearest Neighbor for PC-Based Character Recognition of Philippine Vehicle Standard License Plate

Klariz Donna Mae B. Bundal, Arlyn P. Lauron, Jan Jeffrey R. Camiña and Cristina P. Dadula

Abstract—This study focused on the development of a PC-based licensed plate recognition system using Visual Basic programming language. A system that is able to recognize Philippines' currently used standard vehicle plate numbers using EmguCV image processing and K-Nearest Neighbor machine learning algorithm. The system accepts image as an input or a snapshot of the image from video of a moving vehicle. There were twenty-two (22) unique images of a vehicle in acquired in 3 different positions: upright position (UP), skewed to right position (SR), and skewed to left position (SL). Image processing techniques were applied to the images such as grayscale conversion, Gaussian blurring, and thresholding. Another processing is the detection of plate number area. Optical character recognition is applied to this area where the characters in the image were segmented and individually recognized. The output equivalent characters and the cropped region of the plate number area are displayed on the user interface. The results showed that in: UP, the system recognition accuracy is 83.12%; in SR, it is 39.97%, and 46.21% in SL. The best system accuracy rate was obtained when the captured image of the vehicle is in the upright position which is 83.12%. For better performance, future works may consider the use of exact font style, different angle and position of the license plates, and different lighting conditions of sample license plates for training

Keywords: *EmguCV, PC-Based, KNN Machine Learning Algorithm, Blob Analysis*

I. INTRODUCTION

Originally, in modern technology, licensed plate recognition (LPR) systems are developed for security

Klariz Donna Mae B. Bundal, Arlyn P. Lauron, Jan Jeffrey R. Camiña and Cristina P. Dadula, Mindanao State University, General Santos City, Philippines (e-mail: cris_dadula@yahoo.com)

purposes and automation. LPRs are integrated into intelligent transportation systems and become possible to automatically monitor motorway collection, analyze traffic and intersection roads, improve law enforcements, and many more [1]. It is believed that currently, more than half a billion cars have their vehicle identification number (VIN) also known as license plate number as their primary identifier. The vehicle's identification number is the identifying code for a specific automobile. VIN serves as the car's primary "fingerprint" as no two or more vehicle has the same identification number. A VIN can be used to track recalls, registrations, warranty claims, thefts and insurance coverage [1].

Our country is confronted with problems regarding traffic violations in road intersection. Common road accidents in everywhere are attributed to the collision of vehicles, pedestrian, or a collision with an object that would result to death, disability and damage to property. These road accidents were caused by driver's errors (26%), over speeding (18%), mechanical defect (12%), drinking spree before driving (1%), and damaged roads (5%) [2]. The common causes of crashes in the intersections of streets according to studies are drivers' negligence and recklessness where drivers are running through red lights, ignoring the yield and stop signs [3]. Beating red light (traffic signal light which means to STOP) is one of the minor problems we are facing today and most of the time, we tend to ignore it, not knowing that it might be a cause of an enormous accident that can probably result to a sudden death of victims. According to Department of Transportation and Communications – Land Transportations Office (DOTC-LTO) of the Republic of the Philippines, it is a crime against the law to disregard traffic signs, and violators shall pay a fine of P1, 000.00 [4]. With a numerous amount of accidents and traffic rule violations, identification of vehicles has become a task of prime importance. With various optical character recognition (OCR) techniques, any valuable information such as vehicle license plate number is obtainable from captured images [5].

A means of technology such as, CCTV camera, road intersections incidents can now be monitored with the help of a video analyst. They can interpret and review CCTV footages [6]. They can help track those who beat traffic lights that might cause serious accidents and report it immediately to authority in order for them to stop tolerating these behaviors and receive proper sanction. Thus, developing a PC-Based automatic license plate recognition is the goal of this study which attempts to automate the identification of vehicles. The system could probably help lessen the hours spent for investigations as well as strengthen the enforcement of law on traffic lights to avoid road accidents. Also, this study makes use of the *EmguCV* Library references for Microsoft Visual Studio Community 2017 - Windows Forms Application (.NET Framework), Visual Basic as its language, and K-Nearest Neighbor as character recognition training machine.

II. PLATE NUMBER DETECTION

Vehicle detection is necessary to perform license plate recognition systems from a video input. An easy way to do vehicle detection is by using a background subtraction algorithm and the tracking of the vehicle can be achieved by using blob tracker algorithm (cvBlob or OpenCVBlobsLib) or blob detection [7]. The cvBlob library provides some methods to get the centroid, the track and the ID of the moving objects. A bounding box can also be set to draw on the frame surface, and also the centroid and angle of the tracked object. [8]. After vehicle tracking, the centroid of the moving object will be checked if it has crossed the region of interest or virtual line in the video. So when a vehicle passes through the virtual line, a frame capture will be executed in the system.

Images taken enter image processing through a series of algorithms which provide an alpha numeric conversion of the characters on the input image into text form [9]. The system localizes the plate number region and looks for the characteristics that would indicate that the chosen object is a license plate. Hence, the searching would continue to process until such time that a series of characters would be recognized by the system by using OCR [10], [11].

III. METHODOLOGY

The system developed accepts both image and video as input and gives an output of the vehicle's licensed plate number in both image and text format.

A. Equipment Used

This study makes the use of personal computer for

software application development, steel tape for measuring the distance between a capturing device and a vehicle plate number, and a digital camera as a capturing device.

Preparation of PC Software Application

Free edition of Microsoft Visual Studio 2017 community from <https://www.visualstudio.com/vs/> was downloaded and installed with default options of the IDE. Also downloaded and installed for free were the latest version of *EmguCV* executable installer without *cuda* support, and *libemgucv-windows-universal-3.0.0.2157.exe*. Visual Basic is the programming language and *EmguCV* is a suitable library that wraps *OpenCV* in .NET form [12] [13].

GUIs Form Layout

Visual Basic – Windows Forms Application was selected for starting up a *New Project*. A total of four (4) forms were created in Visual Basic.

- License Plate Recognition – Character Training (KNN Algorithm)
- License Plate Recognition – Testing the Trained System
- License Plate Recognition – Image of Vehicle with License Plate (Input)
- License Plate Recognition – Video Clip of Moving Vehicle (Input)

B. Image and Video Acquisition

Images of vehicle with standard license plate were captured using digital camera from selected parking lots around General Santos City. A total of twenty-two (22) unique vehicles were captured in three (3) different positions as shown in Figure 1: Group A, Group B, and Group C. These were captured in upright position, skewed to right position, and skewed to left position, respectively. The images were properly cropped and scaled.



Fig. 1. Different Capturing Positions, a) Group A; b) Group B; c) Group C

The video inputs used in this study were recorded from two (2) selected places in General Santos City, one (1) video per location was considered or testing. Only one vehicle per video was captured and tested for the system. A snapshot

from each video is shown in Figure 2, wherein *Input 1* was obtained from a university main entrance, and *Input 2* was from a mall parking lot entrance.

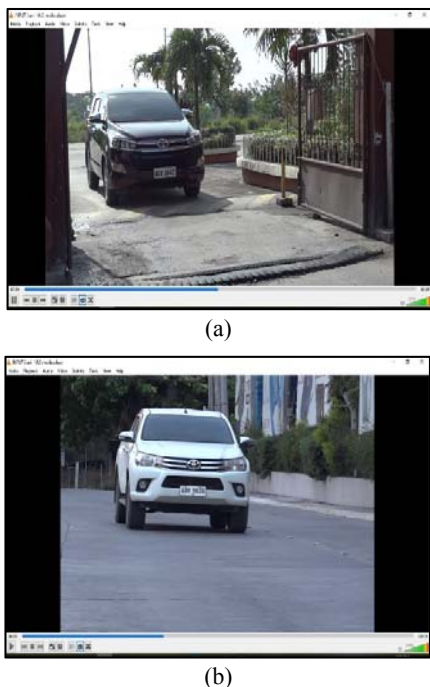


Fig. 2. Snapshot from each Video, a) Input 1 and b) Input 2

The recognition of characters of the vehicle’s license plates were separated for the image input and the video input. The outputs were properly tabulated for better viewing of results and comparisons.

C. Preparation of Machine Learning Method

KNN machine learning algorithm was settled on $k = 1$ or finding the single nearest neighbor producing optimal result because of the relatively small size of training data set. KNN algorithm is a simple algorithm which stores all available cases and classifies new cases [14].

The training was done by encoding a set of English alphabet and numbers with font similar to the font used in license plate, and saving this file in image or jpg format, a sample is shown in Figure 3. The characters on the number plate must have uniform fonts so that the OCR for number plate recognition would be less complex as compared to other methods [15]. OCR processes recognizes both handwritten and encoded/printed characters depending upon the quality of the subject (characters: letters and numbers) to be recognized [16]. The training set were encoded in six (6) batches including: four batches of *License Plate Font* from <http://www.fontspace.com/dave-hansen/license-plate>; one batch for *ARCADE R Font* from <https://fonts2u.com/arcade-r.font> and; one batch also for *GL NummernsChild*

Eng Font from <http://www.fontspace.com/gutenberg-labo/gl-nummernschild>. The training of character recognition was done and saved as a first Visual Basic – Windows Forms Application GUI project for Visual Studio entitled: *License Plate Recognition – Character Training (KNN Algorithm*, as stated in section B, GUIs Form Layout.



Fig. 3. Characters in Image Format

For testing and verification if the trained system can characterize letters and number, a set of sample text images shown Figure 4 was used as input to the second *Windows Forms Application project, License Plate Recognition – Testing the Trained System*. Since the characters were encoded, OCR was easily performed

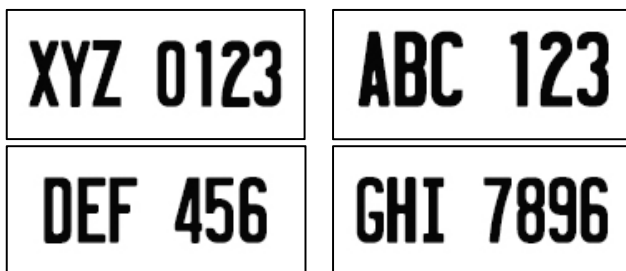


Fig. 4. Set of Images of Characters

D. Optical Character Recognition (OCR) and Image Processing

The character recognition process is shown in Figure 5. Given the captured image of the vehicle, the system automatically searches for possible plate number region. This region undergoes the optical character recognition process shown in Figure 6. If the system recognizes characters, then system decides that the plate number area was found and it displays the recognized characters as the plate number of the vehicle in text format as an output. Otherwise, it keeps on searching for possible plate region until certain characters were recognized.

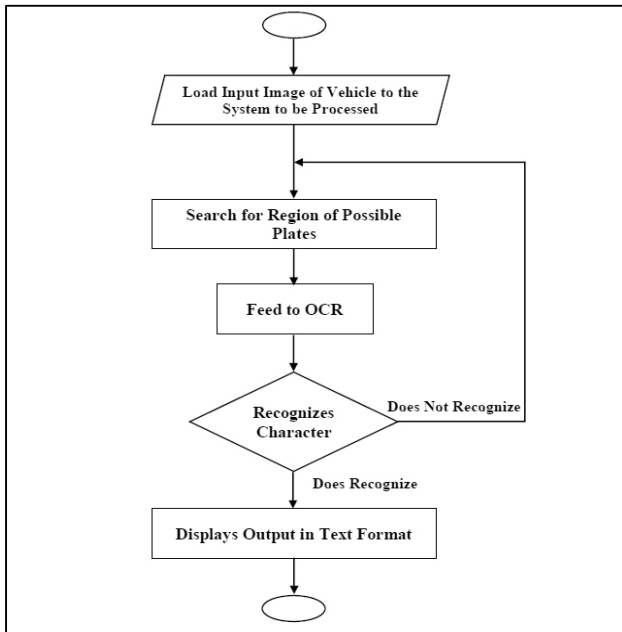


Fig. 5. Character Recognition System Flowchart

The process of OCR involves the application of image processing techniques. Image processing is a method to perform some operations in an image, in order to get an enhanced image or extract some useful information from it [17]. In image processing, the images are converted as desired, for instance, to have a simpler processing of the image, an image is converted from red-green-blue (RGB) layers to gray scale layer [10]. In Figure 6, the image acquired need to undergo processing such as noise removal, image thresholding, edge detection segmentation, feature extraction and classification. The OCR cannot perform segmentation and classification without KNN training. An overview of KNN is shown in Figure 7. The output of the system are the characters composed of letters and numbers in text format.

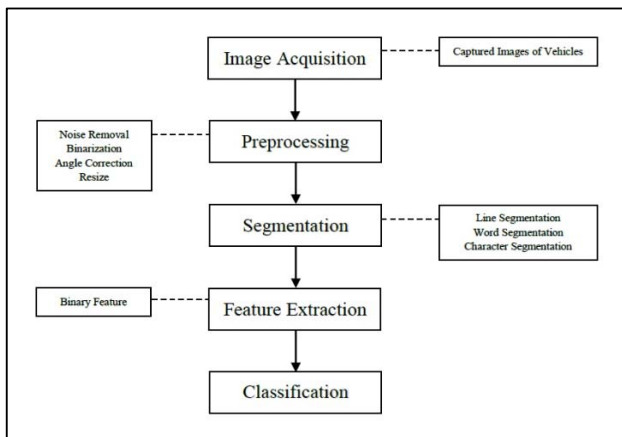


Fig. 6. OCR Processing

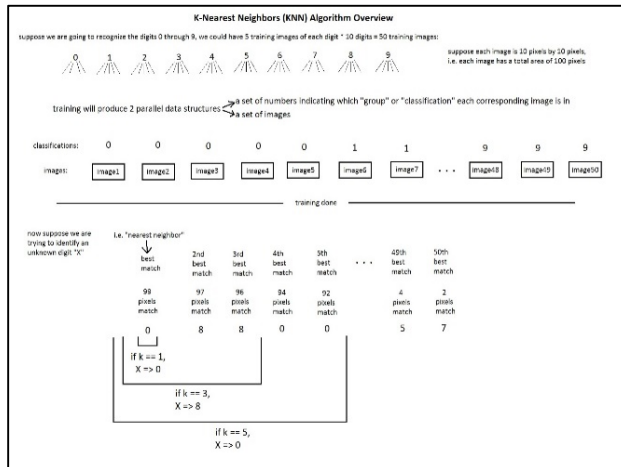


Fig. 7. KNN Overview [18] [19]

There were 5 class definition in Visual Basic for the image processing as described in Figure 8: *Preprocessing*, *PossiblePlates*, *DetectPlates*, *PossibleCharacters*, and *DetectCharacters*. The concept of pre-processing method involves converting the original image to grayscale and finding its edges for easy detection of plate region. Finding possible license plate involves localizing possible plate numbers. Several image contouring was done until a possible plate number was located, this helps the system to easily detect and localize the plate number. In license plate detection, the license plate detected was extracted and cropped to be processed again for classifying or defining characters. The cropped region was processed again with techniques like gray scaling, thresholding, and several contouring until certain characters were detected. The segmented characters were localized and loaded to KNN trained system for recognition. The characters defined were converted to a text format and was displayed in the textbox of the form as license plate characters.

Preprocess	Possible Plates	Detect Plates	Possible Characters	Detect Characters
Original Image	Find possible plate	Detect/Localized plate region	Find possible characters	Detect characters (Load KNN Training)
Grayscale Threshold	Contouring until detection	Extracted and cropped plate	Contouring until characters detected	Extract characters
				Text output

Fig. 8. Image Processing.

E. Video Input Processing

Video processing is a particular case of signal processing, which often employs video filters and where the input and output signals are video files or video streams [6]. The last *Visual Basic – Windows Forms Application* project made was the *License Plate Recognition – Video Clip of Moving Vehicle (Input)* which uses video as input. The video undergoes video processing such as *blob analysis*. The system creates a virtual

line in the image frame that can be used to determine if a vehicle passes through. If the system detects the presence of a vehicle, it captures an image frame and subject that frame to image processing. The process is similar to the works of references [20], [8], and [21].

F. Evaluation of System's Performance

The performance of the system is evaluated using the following parameters: *plate number correctness* and *recognition accuracy*. The *plate number correctness* is expressed as the weighted score of the plate, $s_w(\mathbf{P}^R)$, defined as:

$$s_w(\mathbf{P}^R) = \frac{|\{P_i^R | P_i^R = P_i^C\}|}{|\{P_i^R\}|} \times 100\% = \frac{m}{n} \times 100\% \quad (1)$$

Where \mathbf{P}^R is plate recognized by a system, and \mathbf{P}^C is the correct one, m = number of correctly recognized characters and n = total number of characters in a plate. For example, if the plate “ABC 1234” has been recognized as “ABC 1284”, the weighted correctness score s_w is 6/7 or 85.7%. The *system accuracy* A , was calculated from the equation

$$A = \frac{\sum \text{weighted score of the license plate}}{\text{number of total plates}} \quad (2)$$

III. RESULT & DISCUSSION

Results of the images that underwent image processing are shown in the following subsections.

A. Character Recognition Training and Preliminary Testing

OCR were performed on typed-text input in .png format as a valid input with the letters of the English alphabet (capital/uppercase letters only) and digits 0 – 9 as classes of characters [22]. Figure 9(a) shows the interface that opens the image file used for training. During training, the letters in the image were selected and was surrounded by a red rectangular shaped outline one by one as the user inputs the corresponding letter as shown in Figure 9(b). When the selection of all letters were finished, the system notifies that the training was completed as shown in Figure 10. A sample of the result of the tests after the training was completed is shown in Figure 11(a) and (b), the system recognized all the characters correctly.

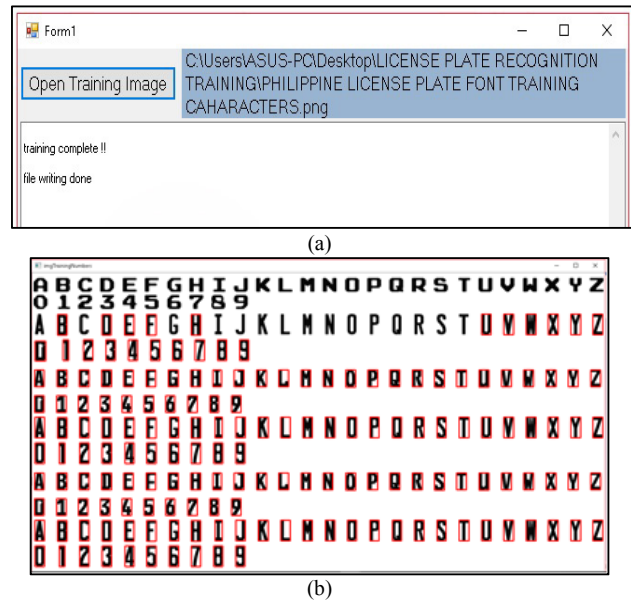


Fig. 9. Character identification

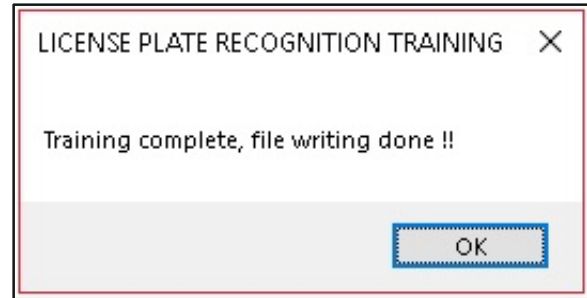
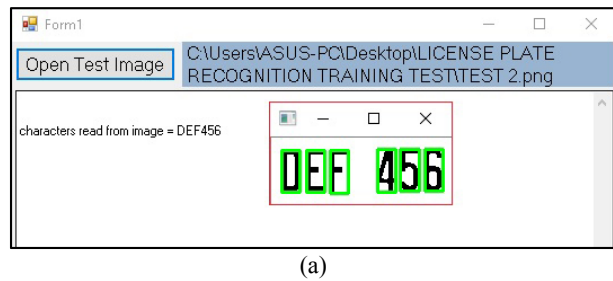
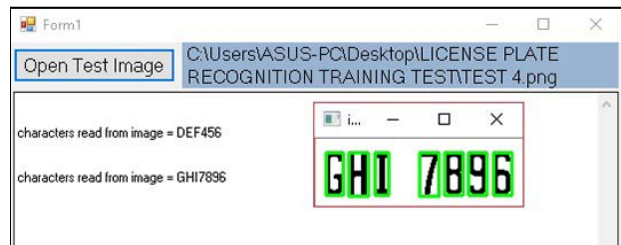


Fig. 10. Notification that the training was completed and that the file writing was done.



(a)



(b)

Fig. 11. Testing the System

B. Image Processing of License Plate

The figures shown in Figure 12 are sample of the system’s main interface with an image input file was loaded. The system detects the plate region, the one enclosed by a red box, the segmented image in the lower right corner, and above it is the recognized characters of the license plate in text format. The system saved the results including

the cropped licensed plate region and its gray scaled output. Table 1 shows sample of the saved results: original cropped license plate region, result of thresholding, result of segmentation, recognized characters in text format. The characters in red font color indicates an incorrect character recognition. Note that the table also shows the results for all three different positions.



Fig. 12. License Plate Recognition, a) Upright Position; b) Skewed to Right Position; c) Skewed to Left Position

TABLE 1
IMAGE SAMPLE OF CROPPED LICENCE PLATE REGION

Original image (input)	Gray scaling/Tresholding	Character Segmentation	Text output
Group A: Uright position			
			KAG4891
			APAJJ78
Group B: Skewed to right position			
			E86NV1
			APA117O
Group C: Skewed to left position			
			KK0489
			APA1198

C. License Plate Recognition for Image Input

D. The results of the license plate recognition tests for the three groups of images are tabulated in Table II, III and IV. Each row shows the input, output, the number of characters correctly recognize, and the weighted score computed using

Equation 1. A character indicated in red color indicates an incorrect character recognition. The accuracy of the system for each group of images was calculated using the Equation 2 and they are summarized in Table V.

TABLE 2
LICENSE PLATE WEIGHTED SCORE FOR GROUP A: UPRIGHT POSITION

Plate No.	Input	Output	No. of Correctly Recognized Characters	Weighted Score
1	AAG 4891	KAG4891	6	85.71%
2	APA 1178	APAJJ78	5	71.43%
3	ABO 6206	AB06206	5	71.43%
4	ABH 7344	ABH7344	7	100.00%
5	AOA 2299	A0A2299	6	85.71%
6	ABC 5018	ABC5018	7	100.00%
7	ADO 9171	AD09171	5	71.43%
8	AKA 5591	AKA5591	7	100.00%
9	AOA 1583	AOA1583	7	100.00%
10	AAS 4083	AAS4083	7	100.00%
11	ACU 9594	ACU9594	7	100.00%
12	ABA 8101	ADA810J	4	57.14%
13	ADP 4510	AOP450	5	71.43%
14	AOA 2665	A0A2665	6	85.71%
15	ADP 4557	AOP4667	4	57.14%
16	A0A 2665	A0A2665	6	85.71%
17	AHA 1624	AHA1624	7	100.00%
18	AOA 1875	A0A1B75	5	71.43%
19	AOA 2092	A0A2092	5	71.43%
20	AAG 5979	AA05979	6	85.71%
21	ABT 3242	ADT3242	6	85.71%
22	A0A 1879	A0A1079	5	71.43%
Average Accuracy				83.12%

TABLE 3
 LICENSE PLATE WEIGHTED SCORE FOR GROUP B: SKEWED TO RIGHT POSITION

Plate No.	Input	Output	No. of Correctly Recognized Characters	Weighted Score
1	AAG 4891	E86NV1	1	14.29%
2	APA 1178	APA117O	6	85.71%
3	ABO 6206	AD06206	5	71.43%
4	ABH 7344	ASW93AA	1	14.29%
5	AOA 2299	A0324--	1	14.29%
6	ABC 5018	A-C5018	6	85.71%
7	ADO 9171	Y1SOS171	3	42.86%
8	AKA 5591	-JA5591	4	57.14%
9	AOA 1583	A0A1583	7	100.00%
10	AAS 4083	JJS4O83	4	57.14%
11	ACU 9594	ACU97--	4	57.14%
12	ABA 8101	ADA81O1	5	71.43%
13	ADP 4510	JOP45-O	5	71.43%
14	AOA 2665	K0A26-5	5	71.43%
15	ADP 4557	L116A--	0	00.00%
16	A0A 2665	74Z	0	00.00%
17	AHA 1624	--	0	00.00%
18	AOA 1875	1ZY4277	1	14.29%
19	AOA 2092	--	0	00.00%
20	AAG 5979	--	0	00.00%
21	ABT 3242	--	0	00.00%
22	A0A 1879	--9A97-	2	28.57%
Average Accuracy				38.97%

TABLE 4
 LICENSE PLATE WEIGHTED SCORE FOR GROUP C: SKEWED TO LEFT POSITION

Plate No.	Input	Output	No. of Correctly Recognized Characters	Weighted Score
1	AAG 4891	KK04891	4	57.14%
2	APA 1178	APA1198	6	85.71%
3	ABO 6206	AD06206	5	71.43%
4	ABH 7344	ABH7344	7	100.00%
5	AOA 2299	A0A22S9	5	71.43%
6	ABC 5018	A8C50J8	5	71.43%
7	ADO 9171	ABD9171	7	100.00%
8	AKA 5591	AKA5591	7	100.00%
9	AOA 1583	A0A1583	6	85.71%
10	AAS 4083	LLS4083	5	71.43%
11	ACU 9594	ALIAGA2	1	14.29%
12	ABA 8101	A0A2454	6	85.71%
13	ADP 4510	--	0	00.00%
14	AOA 2665	--	0	00.00%
15	ADP 4557	--	0	00.00%
16	A0A 2665	--	0	00.00%
17	AHA 1624	--	0	00.00%
18	AOA 1875	--	0	00.00%
19	AOA 2092	--	0	00.00%
20	AAG 5979	A3D---3	3	42.86%
21	ABT 3242	KDT3242	5	71.43%
22	A0A 1879	--	0	00.00%
Average Accuracy				46.21%

TABLE 5
SYSTEM ACCURACY OF THREE GROUPS

Group	Total number of detected plates	Average Weighted score/ Group
A	22	83.12%
B	19	38.97%
C	19	46.21%

It can be observed in Table II that the system detected the plate numbers of all input images where vehicle are captured in the upright position. The system recognition accuracy is 83.12%. In the case of the skewed vehicle position as shown in Table IV and V, there were cases that the system failed to detect the plate number area such as plate nos. 17, 19, 20, and 2 of Table IV; and plate nos. 13 to 19, and 22 in Table IV. The system recognition accuracy is 38.97% for *skewed to right* position, and 46.21% for *skewed to left* position. This is because the plate numbers of the vehicle in SL or SR position were not clear enough. Incorrect recognition also occurred when the plate number was not properly cropped such as the case in the first entry of Table I, portion of the first character was slightly cropped, the system recognized letter *K* instead of letter *A*. Another observation in Table II, the system failed to distinguished slightly similar characters such as: character *J* was misidentified as *I* in plate no. 2, character *O* was misidentified as *0* and vice versa in plate no.3, and character *B* was misidentified as *D* in plate no. 7. Same goes with plate nos. 12 to 22.

A number of considerable errors were also observed as shown in Table III and IV. In Table III, the characters *AAG 489* in plate no. 1 were incorrectly identified as *E86NV*, and the characters *ABH7344* in plate no. 4 were incorrectly identified as *ASW93AA*. In Table IV, the characters *ACU9594* were incorrectly identified as *ALLAGA2*. Due to these errors, the system recognition accuracy for SR and SL group of images is very low, 38.97% and 46.21%, respectively.

D. License Plate Recognition for Video Input

The two video clips fed to the system are shown is shown in Figure 13. The system captured frame and processed it repeatedly, and displayed the output of the system while playing the input video. It was observed many times that the output is incorrect. The procedure slowed down the whole process of license plate character recognition. However, after several attempts of locating plate number, the system finally locate the plate number area, segment, and recognize the characters correctly. In addition, the system successfully displayed the correct plate number in text format.

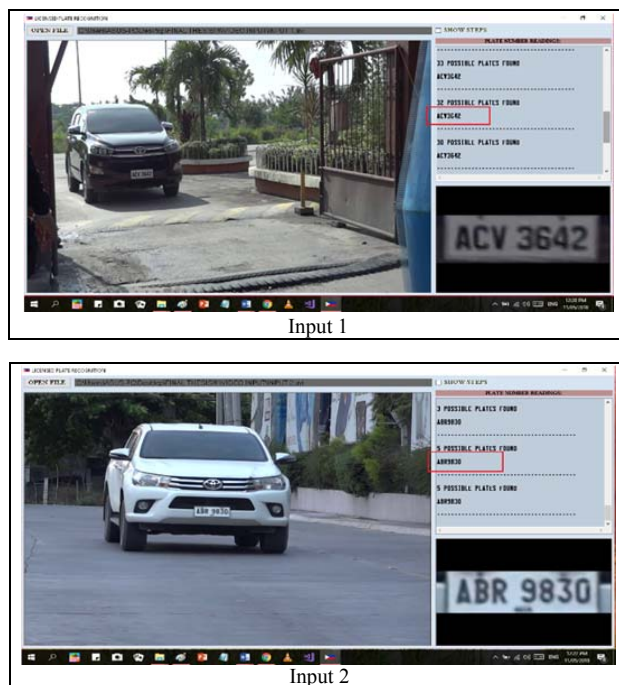


Fig. 13. Loading Video Input

V. CONCLUSION & RECOMMENDATIONS

The designed automatic license plate recognition system was successfully implemented in Visual Basic using EmguCV image processing library and KNN machine learning algorithm. It was able to recognized characters in the license plate using image or video input. The tests using two video inputs with one vehicle each was successful. The system was able to locate and recognize all the characters in the license plate accurately as displayed in the output. The system's performance using image inputs in three groups of images: UP, SR, and SL, is 83.12%, 38.97%, and 46.21%, respectively. The best performance is when the vehicle is in the upright position in the captured image that is 83.12%. Incorrect recognition in the UP position is due the similarity of the characters such as O and 0, J and I, B and D. Considerable errors in SL and SR positions are due mainly to the poor quality of the captured image of the vehicle.

For the improvement of the system, it recommended to use exact font style of license plate in training the system, and capture samples images at different angle positions and lighting conditions. Obtain samples as many as possible. Also, the system can be improved by adding more features to the interface like time stamp or database and work for the methods to make the detection real-time.

REFERENCES

- [1] V. Gigante and S. B. Jolcanain, "Development of a Motorcycle License Plate Number Recognition System," pp. 1, 13, March 2014.
- [2] A. Tamayo, "Occurrence of Traffic Accidents in the Philippines: An Application of Poisson Regression Analysis," *SSRN Electronic Journal*, p. 10.2139/ssrn.1438478, July 2009.
- [3] J. Dr. Balcanao, "Would you say that accidents on Philippine roads are caused mostly by errant drivers?," Phil Star Global, Benguet, 2011.
- [4] E. Tipan, "DOTC, LTO to impose stiffer fines for traffic violations," AUTO INDUSTRY NEWS, Quezon, 2014.
- [5] R. K. C. Billones, A. A. Bandala, E. Sybingco, L. A. Gan Lim, A. D. Fillone and E. P. Dadios, "Vehicle Detection and Tracking using Corner Feature Points and Artificial Neural Networks for a Vission Based Contactless Apprehension System," *Computing Conference 2017*, p. 688, 2017.
- [6] "Video processing," 1 July 2017. [Online]. Available: https://en.wikipedia.org/wiki/Video_processing.
- [7] R. K. Billones, A. Bandala, E. Sybingco, L. Gan Lim and E. P. Dadios, "Intelligent System Architecture for a Vision-Based Contactless Apprehension for Traffic Violations," p. 1, 2017.
- [8] A. Sobral, "simple_vehicle_counting," 12 04 2017. [Online]. Available: https://github.com/andrewssobral/simple_vehicle_counting.
- [9] License Plate Recognition, "How License Plates Recognition Works," 2010. [Online]. Available: www.licenseplatesrecognition.com/how-lpr-works. [Accessed May 2018].
- [10] S. Bold and B. Sosorbaram, "Smart License Plate Recognition Using Optical Character Recognition Based on," *International Journal on Recent and Innovation Trends in Computing and Communication*, pp. 92-96, 2017.
- [11] V. Ong and D. Suhartono, "USING K-NEAREST NEIGHBOR IN OPTICAL CHARACTER RECOGNITION," *ComTech Vol. 7 No. 1*, pp. 53-65, 2016.
- [12] S. Shi, *Emgu CV Essentials*, Birmingham B3 2PB, UK: PACKT Publishing Ltd., 2013.
- [13] S. Saxena and M. S. Tiwari, "Starting with Emgu CV," 19 September 2013. [Online]. Available: <https://social.technet.microsoft.com/wiki/contents/articles/15385.starting-with-emgu-cv.aspx>.
- [14] S. Dr. Sayad, "K Nearest Neighbor," 2010. [Online]. Available: <http://chem-eng.utoronto.ca/~datamining/>.
- [15] K. Kaur and V. Banga, "NUMBER PLATE RECOGNITION USING OCR TECHNIQUE," *IJRET: International Journal of Research in Engineering and Technology*, pp. 286-290, 2013.
- [16] L. Eikvil, *OCR Optical Character Recognition*, P.B. 114 Blindern: Norsk Regnesentral, 1993.
- [17] G. Anbarjafari, "Digital Image Processing," 2014. [Online]. Available: <https://sisu.ut.ee/imageprocessing/book/1>.
- [18] C. Dahms, "OpenCV_3_KNN_Character_Recognition_Emgu_CV_3_Visual_Basic," 9 December 2015. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_KNN_Character_Recognition_Emgu_CV_3_Visual_Basic/blob/master/DocsAndPresentation/KNN_Overview.png.
- [19] C. Dahms, "OpenCV_3_License_Plate_Recognition_Emgu_CV_3_Visual_Basic," 20 July 2017. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_License_Plate_Recognition_Emgu_CV_3_Visual_Basic.
- [20] C. Dahms, "OpenCV_3_Car_Counting_Visual_Basic," 20 February 2016. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_Car_Counting_Visual_Basic.
- [21] E. P. Dr. Dadios, A. Dr. Fillone, E. Engr. Sybingco, L. G. Dr. Lim and A. Dr. Bandala, "Contactless Apprehension of Traffic Violators on 24-hours Basis All-Vehicle Detection System (CATCH-ALL)," p. 18, 2015.
- [22] J. Nguyen, "Introduction to Optical Character," p. 9, 2014.

Real-Time Vehicle Classification Using MobileNet

Reagan L. Galvez,^{1,*} Melvin K. Cabatuan,² and Argel A. Bandala²

Abstract—classification is an important part of vision systems and has several applications like autonomous cars and surveillance. This is a challenging task because computers see images differently from humans. This paper used the MobileNet model for training the data and tested it on an Android device. This model is lightweight and efficient compared with previous developed models. This was inspired by the sample code from Google Codelabs. Experiment results show that the Android application can accurately classify the type of vehicle in real time.

Keywords: convolutional neural network, deep learning, MobileNet, vehicle classification

I. INTRODUCTION

Intelligent transport systems (ITSs) are developed to provide safe travel and ensure effective transportation [1]. To implement this, the ITS needs access to data like the type of vehicle. This will help authorities to identify criminals quickly. It can also be used to apprehend violators like trucks [2], during certain periods of time (ex., rush hour) when trucks are not allowed on the road. Vehicle classification maybe a simple problem for a human, but for a computer, it's a complex problem. That's why many researchers are exploring image processing [3] to solve this problem with high accuracy by creating different architectures/models. There are many techniques that can be used to classify images like artificial neural networks, decision tree, support vector machine, and fuzzy measure [4]. Currently, convolutional neural networks (CNNs) are used in image classification because they provide accurate performance in computer vision tasks. There are many pre-trained CNN based models that can be used for image classification like AlexNet [5], VGG16, VGG19 [6], ResNet50 [7], InceptionV2, InceptionV3 [8], Xception

[9], and DenseNet [10]. On the other hand, models like R-CNN [11], Fast R-CNN [12], YOLO [13], YOLO9000 [14], SSD [15], and MobileNet [16] are commonly used object classification and detection models. The focus of this paper is to utilize MobileNet for object classification using android platform.

This paper is organized as follows: section 2 discusses the concept about MobileNet and its architecture. Section 3 describes the experiment setup and the dataset used for training. Section 4 shows the performance of the MobileNet model in vehicle classification and its deployment in an Android platform.

II. MOBILENET

MobileNet is an efficient model designed for mobile and embedded vision applications [16]. It uses depthwise separable convolutions to build lightweight and efficient deep neural networks. The depthwise convolution applies a single filter to each input channel. To combine the output of depthwise convolution, pointwise convolution applies 1×1 convolution. The combination of two convolutions results to a depthwise separable convolution. Two layers are formed by depthwise separable convolution. The first layer is used for filtering, and the second layer is used for combining. This is called factorization, and as a result, it reduces the model size and computation.

Table 1 shows the MobileNet architecture. The model consists of 28 layers. These layers are combinations of alternating depthwise and pointwise convolutions. Every layer is followed by batch normalization and rectified linear unit (ReLU) [17] function except the fully connected layer which is followed by a softmax classifier that gives actual probabilities in each class. Batch normalization speeds up the training [18], and the ReLU function is 0 for negative values and grows linearly for positive values. The notations s_1 and s_2 are the number of strides. Stride controls how the filter convolves around an input volume.

Reagan L. Galvezla, Bulacan State University, Malolos City, Philippines (e-mail: reangalvez@gmail.com)

Melvin K. Cabatuan and Argel A. Bandala, De La Salle University–Manila, Philippines

TABLE 1
MOBILENET ARCHITECTURE

Layer	Type/Stride	Input Size
1	Conv /s2	224 × 224 × 3
2	Conv dw /s1	112 × 112 × 32
3	Conv /s1	112 × 112 × 32
4	Conv dw /s2	112 × 112 × 64
5	Conv /s1	56 × 56 × 64
6	Conv dw /s1	56 × 56 × 128
7	Conv /s1	56 × 56 × 128
8	Conv dw /s2	56 × 56 × 128
9	Conv /s1	28 × 28 × 128
10	Conv dw /s1	28 × 28 × 256
11	Conv /s1	28 × 28 × 256
12	Conv dw /s2	28 × 28 × 256
13	Conv /s1	14 × 14 × 256
14–23	Conv dw/ s1 Conv /s1	14 × 14 × 512 14 × 14 × 512
24	Conv dw /s2	14 × 14 × 512
25	Conv /s1	7 × 7 × 512
26	Conv dw /s2	7 × 7 × 1024
27	Conv /s1	7 × 7 × 1024
	Avg. Pool /s1	7 × 7 × 1024
28	FC /s1	1 × 1 × 1024
	Softmax /s1	1 × 1 × 1000

III. EXPERIMENT SETUP

The effectiveness of MobileNet in vehicle classification was tested in an android application. It is a simple camera application that runs a TensorFlow image recognition program to identify vehicles. This was inspired by the code from “TensorFlow for Poets 2” of Google Codelabs [19]. TensorFlow mobile was used to run the MobileNet and integrated it to mobile application. The Anaconda Prompt was used to initiate command in training and testing the model.

The first step in training the data is to install dependencies like TensorFlow. This is an open-source library for high-performance calculation, which allows easy deployment in different platforms and supports deep learning applications [20]. Figure 1 shows the block diagram of the vehicle classification structure. Next is to collect the data that will be used for training. The dataset is composed of five types of vehicles such as pickup, SUV, sedan, van, and truck. These images came from ImageNet, an image database. Table 2 shows the number of images per category of vehicle.

TABLE 2
DATASET

Vehicle Type	Number of Images
Pickup	400
SUV	400
Sedan	400
Van	400
Truck	400
Total	2000

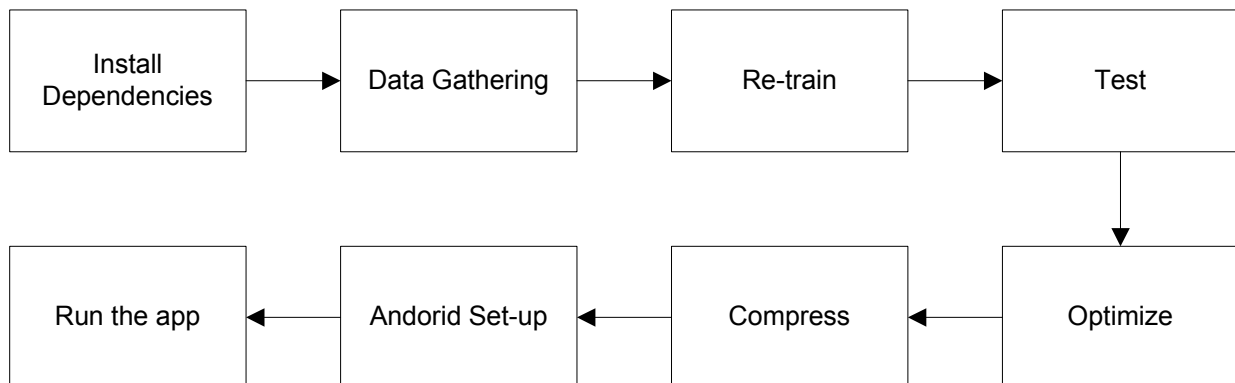


Fig.1. Block diagram.

Next is to re-train the data using the MobileNet model. The output of this is a .pb file; this is a single model file that contains graph variables frozen as constants. The input images can be configured using 4 different input sizes such as 128, 160, 192, and 224 pixels. The width multiplier (α) of the model can also be set such as 1.0, 0.75, 0.5, and 0.25. The width multiplier makes the model small and faster by reducing computational cost and number of parameters.

Then, test the model by using random image and compute its accuracy. After this, optimize the model by removing some nodes that are not needed in a given set of input and output. Next is to compress the model by quantizing the network weights to make it ideal for mobile applications. After compression, the model is ready to upload in an Android platform.

IV. RESULTS

A. Image Classification Performance

The performance of the MobileNet model was tested by calculating its validation accuracy, cross entropy, and evaluation time. Validation accuracy is the precision on a randomly selected group of images from a different set. Cross entropy is a loss function that shows how well the learning process is progressing. The ideal value of cross entropy is 0. Evaluation time is the time it takes to classify the test image. The total number of steps used in training the model was 2000 steps. This is enough to see if the model is learning. Table 3 shows the accuracy and cross entropy using different values of width multiplier (α) in MobileNet. It shows that when the MobileNet's width multiplier (α) decreases, the validation accuracy also decreases and the cross entropy increases.

TABLE 3
ACCURACY AND CROSS ENTROPY

Input Size	Width Multiplier (α)	Accuracy (%)	Cross Entropy
224	1.0	84.83	0.541196
224	0.75	84.36	0.554476
224	0.5	81.99	0.713274
224	0.25	72.04	2.38165

Figure 2 shows the training (orange) and validation accuracy (blue) during the training. This graph was from MobileNet using $\alpha = 1.0$. As shown, the model was not learning anymore. In this case, the training can be stopped. The average validation accuracy is equal to 84.83%.

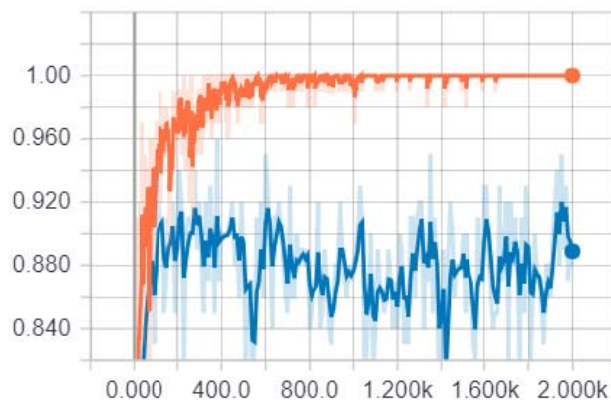


Fig. 2. Training steps vs. accuracy.

Figure 3 shows the training (orange) and validation cross entropy (blue) during the training. The ideal value of cross entropy is 0 because this is a loss function. The average cross entropy is equal to 0.541196.

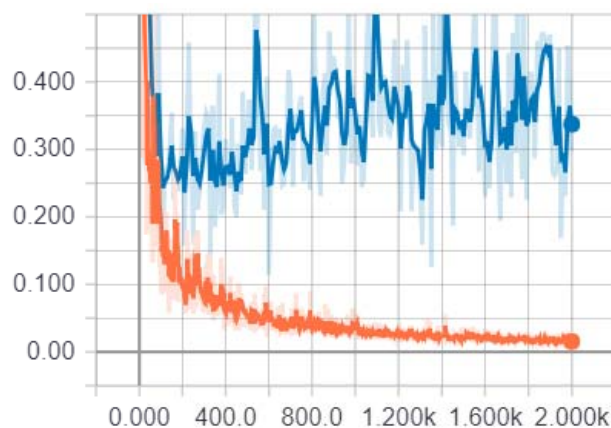


Fig. 3. Training steps vs. cross entropy.

Table 4 shows the time comparisons using different width multiplier (α) in the model. As the width multiplier (α) decreases, the time of evaluation also decreases. This is because the model became lightweight and has fewer computations.

Table 5 shows the model size comparisons for each different width multiplier (α) in the model. Using $\alpha = 1.0$, there was 32.50% size reduction from the optimized to the compressed model. This compression can be useful in deploying the model for mobile applications because it can be downloaded easily.

Figure 4 shows the bubble chart of evaluation time versus accuracy. The size of the bubble is proportional to the size of the model. As we can see, there is a trade-off between accuracy and speed. If we need a faster model, $\alpha = 0.25$ can be used, but the accuracy will decrease. For real-time application, $\alpha = 1.0$ can still be used.

TABLE 4
EVALUATION TIME COMPARISONS

Input Size	Width Multiplier (α)	Time (Re-Trained) (s)	Time (Optimized) (s)	Time (Compressed) (s)
224	1.0	2.163	2.023	2.061
224	0.75	1.756	1.776	1.701
224	0.5	1.352	1.341	1.359
224	0.25	1.122	1.121	1.121

TABLE 5
MODEL SIZE COMPARISONS

Input Size	Width Multiplier (α)	Size (Re-Trained) (KB)	Size (Optimized) (KB)	Size (Compressed) (KB)
224	1.0	15,489	15,485	5,033
224	0.75	9,490	9,486	3,174
224	0.5	4,916	4,913	1,786
224	0.25	1,766	1,763	708

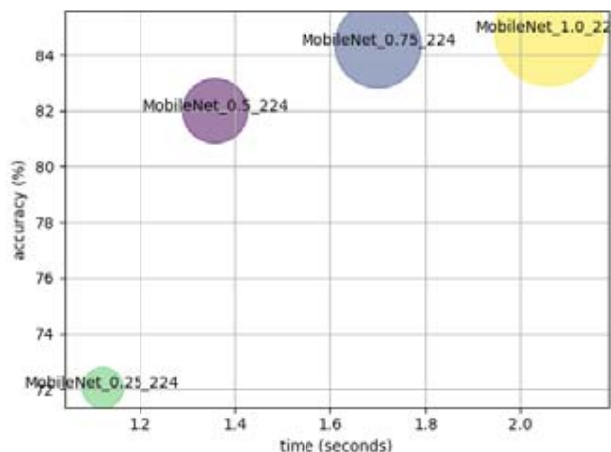


Fig. 4. Evaluation time vs. accuracy.

B. Deployment in Android Platform

The trained model was deployed in an Android platform to test its performance in real-time application. TensorFlow mobile was used to prepare the model for mobile deployment. Figure 5 shows the screenshot from the TensorFlow application using a Samsung Galaxy S7 Flat phone. The testing images were randomly selected to test if the application can detect unseen data. The actual images in the figure were both sedan and correctly identified as sedan.

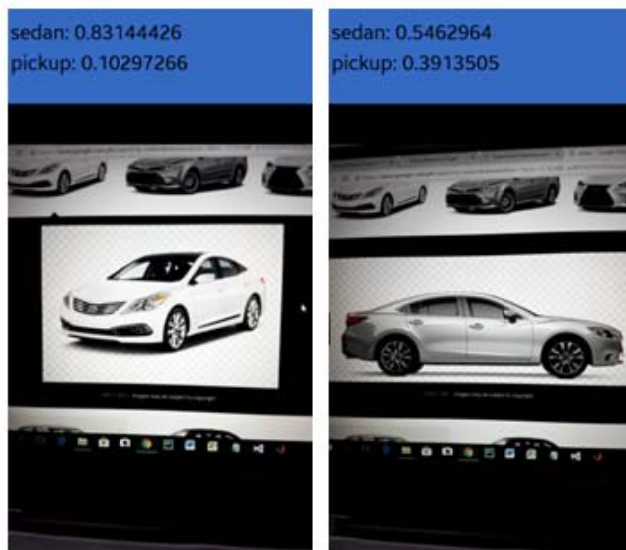


Fig. 5. Screenshot from TensorFlow application (sedan).

Another set of images was tested as shown in Figure 6. These images were a truck and correctly identified as truck by the TensorFlow application.

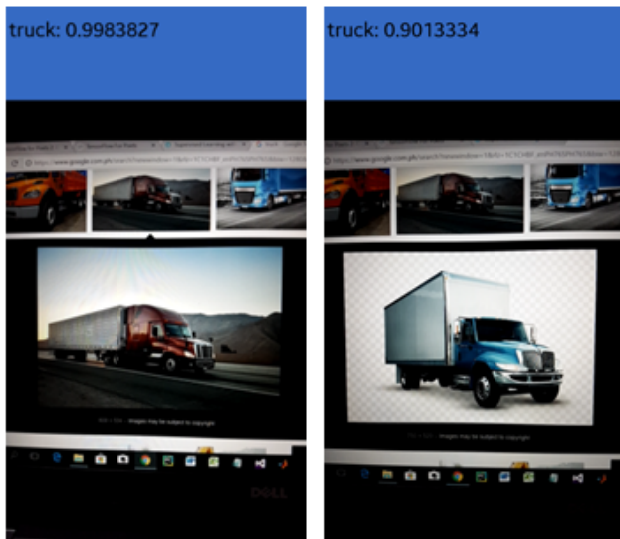


Fig. 6. Screenshot from TensorFlow application (truck).

Figure 7 shows an image of an SUV. The TensorFlow application failed to classify the first image as SUV; it was predicted as sedan. This is due to the similar features of an SUV and a sedan.

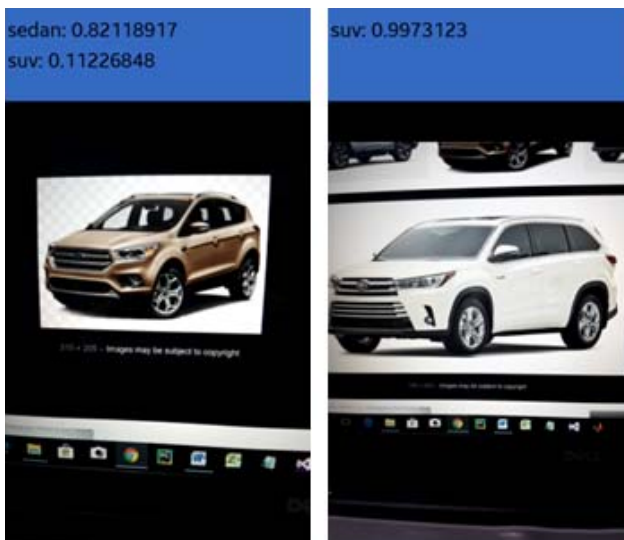


Fig. 7. Screenshot from TensorFlow application (SUV).

V. CONCLUSION

The implementation of real-time vehicle classification was successfully tested using the Android platform. The experiment result shows that the MobileNet can classify vehicle type up to 84.83% accuracy. TensorFlow mobile is a good deep learning solution for a mobile platform like Android. Although in testing an SUV image, the

model cannot consistently classify the image, other types of vehicles were accurately identified. This is due to the physical structure of the SUV that is not totally different from the other types of vehicles because some SUVs are similar to a sedan, van, and pickup. As a solution, it is recommended to increase the dataset and remove some images that are confusing.

For future work, the application of MobileNet can be upgraded to track and identify vehicle type using CCTV videos in real-time.

ACKNOWLEDGMENTS

The author would like to thank the Department of Science and Technology—Engineering Research and Development for Technology (DOST-ERDT) and De La Salle University for the financial support while doing this research.

REFERENCES

- [1] A. Ferdowsi, U. Challita and W. Saad, "Deep Learning for Reliable Mobile Edge Analytics in Intelligent Transportation Systems," *arXiv:1712.04135v1*, December 2017.
- [2] R. K. C. Billones, A. A. Bandala, E. Sybingco, L. A. G. Lim and E. P. Dadios, "Intelligent system architecture for a vision-based contactless apprehension of traffic violations," *2016 IEEE Region 10 Conference (TENCON)*, pp. 1871 - 1874, 2016.
- [3] R. K. C. Billones, A. A. Bandala, E. Sybingco, L. A. G. Lim, A. D. Fillone and E. P. Dadios, "Vehicle Detection and Tracking using Corner Feature Points and Artificial Neural Networks for a Visionbased Contactless Apprehension System," *Computing Conference 2017*, pp. 688-691, 2017.
- [4] P. Kamavisdar, S. Saluja and S. Agrawal, "A Survey on Image Classification Approaches and Techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. II, no. 1, pp. 1005-1009, January 2013.
- [5] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in neural information processing systems*, 2012.
- [6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," in *ICLR*, 2015.
- [7] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference Publication*, pp. 1-12, 2015.
- [8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," *arXiv:1409.4842v1*, pp. 1-12, September 2014.
- [9] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *IEEE Conference Publication*, pp. 1-8, 2017.

- [10] G. Huang, Z. Liu, L. Van der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [11] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv:1311.2524v5 [cs.CV]*, pp. 1-21, 2014.
- [12] R. Girshick, "Fast R-CNN," *arXiv:1504.08083v2 [cs.CV]*, 2015.
- [13] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv:1506.02640v5 [cs.CV]*, pp. 1-10, 2016.
- [14] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *arXiv:1612.08242v1 [cs.CV]*, pp. 1-9, 2016.
- [15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," *arXiv:1512.02325v5*, pp. 1-17, 2016.
- [16] A. G. Howard, M. Zhu, B. Chen and D. Kalenichenko, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision," 2017.
- [17] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010.
- [18] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv:1502.03167v3*, pp. 1-11, March 2015.
- [19] Codelabs, Google, "Tensorflow for Poets 2," December 2017. [Online]. Available: <https://github.com/googlecodelabs/tensorflow-for-poets-2>.
- [20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng, "TensorFlow: A system for large-scale machine learning," *arXiv:1605.08695v2*, pp. 1-18, May 2016.

The Contributors

Ariel James T. Abarquez was born in General Santos City, South Cotabato in June 29, 1997. He received his bachelor's degree in Electronics Engineering (ECE) from Mindanao State University – GSC on June 2018. He was a scholar of Darlene Antonino Foundation. He conducted his On-the-Job Training at the Civil Aviation Authority of the Philippines (CAAP) under the Air Navigation Services (ANS) department. He was a member of the Junior Institute of Electronics Engineers of the Philippines (JIECEP), Organization of Electronics and Electrical Engineering Students and Engineering Students Organization.

Ryann A. Alimuin is a full-time faculty of the Electronics Engineering Department of Technological Institute of the Philippines – Quezon City and is currently taking his Ph.D in Electronics Engineering at the De La Salle University – Manila. He Received his BSECE in TIP-QC (2007) and MEng in TUP-Mla (2012). He is a registered Professional Electronics Engineer and a licensed professional teacher major in Mathematics. He is a consultant of several industries for CCTV systems layout, Fire Detection and Safety, PABX and Structured Cabling. His research works are in the field of industrial and power electronics, power management systems, as well as, intelligence systems for precision agriculture. Currently, he is engaged in research works for smart farming, RF groundwave propagations and face vector analysis for surveillance systems.

Ramtin Aminpour is a PhD candidate in Electronics and Communications Engineering in De La Salle University. He received his Bachelors of Electronics Engineering in Azad University of Tabriz in 1998. He received his Master of Digital Electronics Engineering in Iran University Science and Technology, Tehran, in 2008. His current research works are in the field of artificial intelligence, data mining, and machine learning. He is a member of the Institute of Electrical and Electronics Engineers (IEEE)

Argel A. Bandala is an Associate Professor and Research Faculty of the Electronics Engineering Department at De La Salle University. He received his Master of Science in Electronics and Communications Engineering in

year 2012 and Doctor of Philosophy in Electronics and Communications Engineering in year 2015 at De La Salle University. He is the current Vice-Chair, The Institute of Electrical and Electronics Engineers (IEEE) Philippines Section and secretary of Computational Intelligence Society Philippine Chapter. He is also a member of IEEE Robotics and Automation Society. His main works are “Implementation of Varied Particle Container for Smoothed Particle Hydrodynamics-Based Aggregation for Unmanned Aerial Vehicle Quadrotor Swarm” and “Swarming Algorithm for Unmanned Aerial Vehicle (UAV) Quadrotors – Swarm Behavior for Aggregation, Foraging, Formation, and Tracking.

Jomari Joseph A. Barrera is an instructor in the Department of Computer Science and Technology at Visayas State University (VSU), Visca, Baybay City. He finished his Bachelor of Science in Computer Science in 2012 and pursuing his master's degree in Computer Science at Cebu Institute of Technology - University (CIT-U). His field of interests are data structures and algorithms, computer vision and intelligent systems.

Klariz Donna Mae B. Bundal was born in Aborlan Medicare Hospital, Poblacion, Aborlan, Palawan on August 15, 1996. She is the second from the eldest of five children of Adon C. Bundal and Evangeline B. Bundal. She started her preparatory school at Western Philippines University – Basic Elementary Laboratory School, finished next her elementary level at San Juan Elementary School and secondary at Western Philippines University – Agricultural Science High School. She received her degree of Bachelor of Science in Electronics Engineering from Mindanao State University – General Santos City Campus on June 2018. She was a consistent Department of Science and Technology – Science Education Institute (DOST-SEI) scholar under R.A. 7687 Scholarship Program. Spent five years in service as an officer of the Organization of Electrical and Electronics Engineering Student (OEECES – MSU GSC) and member of Junior Institute of Electronics Engineers of the Philippines (JIECEP – SOCCSKSARGEN) and finishes her On the Job Training at Wireless Link Technologies Incorporated, General Santos City branch.

Melvin K. Cabatuan received the B.Sc. degree in Electronics and Communications Engineering (ECE) from Cebu Institute of Technology, Cebu, Philippines, in 2004; MS degree in Engineering from Nara Institute of Science and Technology (NAIST), Nara, Japan, in 2010; and Ph.D. degree in ECE from De La Salle University (DLSU) - Manila in 2016. He joined the Electronics and Communications Engineering department of DLSU in 2011, where he is currently an Assistant Professor. His current research interest involves Machine/Deep Learning, Computer Graphics, and Computer Vision applications to Health Informatics, Education Technology, Game/Graphic Content Development, and Mobile Computing. He is knowledgeable in programming languages - C/Cpp, Java, and MatLab/Octave scripting. He has more than 5-year experience in computer vision application development with OpenCV library in the mobile framework.

Elmer P. Dadios is a University Fellow and Professor at De La Salle University. He is also the president, Neuronemch, Inc. In 1996, he received his Doctor of Philosophy from Loughborough University. In 1997, he was an Exchange Scientist in Japan Society for the Promotion of Science, Tokyo Institute of Technology. He served as the Director of Engineering Graduate School, De La Salle University in 1998-1999 and Director of School of Engineering, De La Salle University in 2003-2004. He is also the General Chair for HNICEM in 2003, 2005, 2007, 2009, 2011, 2013. His main works are "Fuzzy Logic – Controls, Concepts, Theories and Applications," ISBN: 978-95351-0396-7, 2012. "Fuzzy Logic – Algorithms, Techniques and Implementations," ISBN: 978-953-510393-6, 2012. "Fuzzy Logic – Emerging Technologies and Applications," ISBN: 978-953-51-03370, 2012. His research interests includes; Robotics, Mechatronics, Automation, Intelligent Systems, Neural Networks, Fuzzy Logic, Genetic Algorithms, Evolutionary Computation and IT. He is a Senior Member in The Institute of Electrical and Electronics Engineers (IEEE) and founder and current of Chair of IEEE Computational Intelligence Society, Philippines. He is a member of IEEE Region 10 Executive Committee and founder and President of The Mechatronics and Robotics Society of the Philippines.

Cristina P. Dadula is a professor of Electrical and Electronics Engineering Department of Mindanao State University, General Santos City. She earned her PhD in Electronics and Communications Engineering degree at De La Salle University, Manila, Philippines. She finished her Bachelor of Science in Electronics and Communications Engineering, and Master of Science in Computer Applications at Mindanao State University-Iligan Institute of Technology. Her research interests include design and implementation of artificial intelligent algorithms, digital signal processing, robotics, and control systems.

Reagan L. Galvez received his Bachelor of Science in Electronics and Communications Engineering from Bulacan State University (BSU) in 2010. He received his Master of Science in Electronics and Communications Engineering from De La Salle University (DLSU) in 2016. He is currently an instructor in BSU, while taking his Doctor of Philosophy in Electronics and Communications Engineering at DLSU. His research interests are in the field of artificial intelligence, robotics and machine learning. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), and Institute of Electronics Engineers of the Philippines (IECEP).

Sittie Alyzza M. Garcia was born on Tacloban City, Leyte, Philippines in May 21, 1996. She earned her bachelor's degree in Electronics Engineering from the Mindanao State University – General Santos last June 2018. She was also a scholar of Department of Science and Technology – Science Education Institute under Republic Act No. 10612. A member of Junior Institute of Electronics Engineers of the Philippines, Organization of Electrical and Electronics Engineering Students, and Engineering Students Organization in MSU-GSC.

Arlyn P. Lauron was born in General Santos City, Philippines, in July 06, 1997. She received her bachelor's degree in Electronics Engineering from Mindanao State University-GSC on June 2018. She was also a scholar of Department of Science and Technology – Science Education Institute under Republic Act 7687. She conducted her On-the-Job Training at the Civil Aviation Authority of the Philippines-GSC under the Air Navigation Services department. She was member of Junior Institute of Electronics Engineers of the Philippines and an officer of her college.

Elmer A. Maravillas is a Full Professor, Chair of Computer Science, Coordinator of Doctor in Information Technology Program and Consultant of Research and Development Coordinating Office (RDCO) at Cebu Institute of Technology - University (CIT-U), Cebu City. He received his BS degree in Mechanical Engineering in 1981 from Western Mindanao State University (WMSU), MS Computer Science in 1998 and PhD in Mechanical Engineering in 2006 from De La Salle University-Manila. He worked as Mechanical Project Engineer and Power Plant Engineer of M&S Company Inc. from 1982 to 1984. He served as Department Head of Computer Engineering Department in WMSU from 1984 to 2003 and RDCO in CIT-U from 2006 to 2007. His main works include modeling and simulations, fuzzy logic systems, neural networks, intelligent systems and expert systems.

Jonah Flor V. Oraño is an instructor in the Department of Computer Science and Technology at Visayas State University (VSU), Visca, Baybay City. She completed her Bachelor of Science in Computer Science degree in 2006 at VSU and MS in Information Technology in 2012 at University of San Carlos (USC), Cebu City. She is currently pursuing Doctor in Information Technology at Cebu Institute of Technology - University (CIT-U). Her research interests are computer vision, machine learning, fuzzy logic systems, and data mining.

Guidelines for Contributors

1. The Journal on Computational Innovations and Engineering Applications (JCIEA) aims to promote the development of new and creative ideas on the use of technology in solving problems in the field of computational applications, computational intelligence, electronics and information and communications technology (ICT), manufacturing engineering, energy and environment, robotics, control and automation, and all their related fields. Manuscript submissions should, therefore, be in pursuit of the same goal and within the related fields.
2. JCIEA only accepts manuscripts written in English. The responsibility for copyediting manuscripts, as well as obtaining reproduction permissions for the use of graphics and other materials from their references, will fall on the author.
3. Authors must also remember to cite all references and ensure that their paper submission has not been previously published or is undergoing peer review for another publication.
4. Manuscripts should include a unique title, an abstract, some keywords, an introduction and discussion of the study, a presentation and discussion of results, and a conclusion. Authors may also include an acknowledgement of funding organizations or consultants, if needed.
5. Manuscripts may be sent to *jciea.dlsu@gmail.com* or *jciea@dlsu.edu.ph* as either an MS Word file (*.doc or *.docx) or a LaTeX file (*.tex), including its supporting files.
6. Manuscripts in either file format should have the following features:
 - Single-spaced, two-column format with 1-inch margin on all sides on letter-sized template
 - Font to be used is Times New Roman, size 11
 - Graphs (*.eps, *.svg), tables (*.csv), and images (*.jpg, *.png) should be saved and sent apart from the MS Word file.
 - Citations and references should be submitted in IEEE or APA format.
 - Submission of these references in a BibTeX format is preferred.
7. Manuscript should be eight to twelve (8–12) pages long, including all figures, tables, and references. Manuscripts exceeding the 12-page limit will require permissions from the editors.
8. Authors must include their full names and affiliations in the manuscript. They may include a 150- to 200-word biography to be included in the back portion of the journal.

Editorial Board

Editor-in-Chief

Prof. Elmer Dadios

De La Salle University, Manila, Philippines
elmer.dadios@dlsu.edu.ph

Managing Editor

Robert Kerwin Billones

De La Salle University, Manila, Philippines
robert.billones@dlsu.edu.ph

Editors

Prof. Abdoullah A. Afjeh

University of Toledo, USA

Prof. Marcelo Ang

National University of Singapore, Singapore

Prof. Kathleen Aviso

De La Salle University, Manila, Philippines

Prof. Argel Bandala

De La Salle University, Manila, Philippines

Prof. John-John Cabibihan

Qatar University, Qatar

Prof. Anthony SF Chiu

De La Salle University, Manila, Philippines

Prof. Kukjin Chun

Seoul National University, South Korea

Prof. Joel Cuello

University of Arizona, USA

Prof. Alvin Culaba

De La Salle University, Manila, Philippines

Prof. Eryk Dutkiewicz

University of Technology Sydney, Australia

Prof. Alexis Fillone

De La Salle University, Manila, Philippines

Prof. Laurence Gan Lim

De La Salle University, Manila, Philippines

Prof. Noel Gunay

Mindanao State University, General Santos City

Prof. Kaoru Hirota

Tokyo Institute of Technology, Japan
Japan Society for Promotions of Science, China

Prof. Rodrigo Jamisola, Jr.

*Botswana International University of Science
and Technology*

Prof. Oussama Khatib

Stanford University, USA

Prof. Ioan Marinescu

University of Toledo, USA

Prof. Janina Mazierska

James Cook University, Australia

Prof. Raouf Naguib

BIOCORE, International U.K.
Liverpool Hope University, U.K.

Prof. Yong-Jin Park

Waseda University, Japan

Dr. Nguyen Thi Quynh

*Royal Melbourne Institute of Technology (RMIT) University,
Vietnam*

Prof. Raymond Sison

De La Salle University, Manila, Philippines

Prof. Raymond Girard Tan

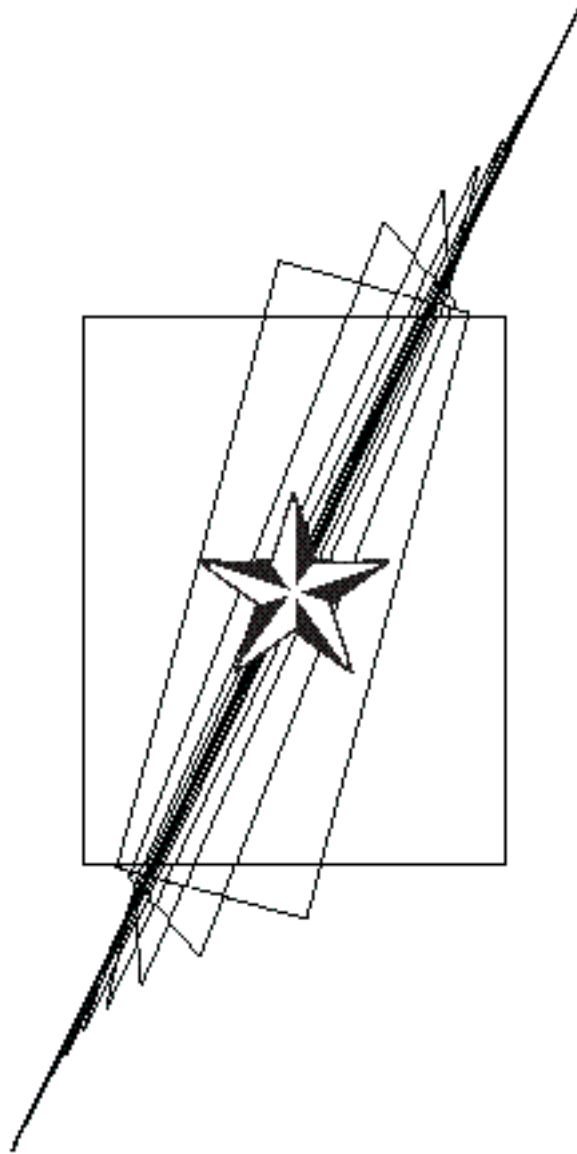
De La Salle University, Manila, Philippines

Dr. Ryan Vicerra

De La Salle University, Manila, Philippines

Prof. Lawrence Wong

National University of Singapore, Singapore



JOURNAL OF COMPUTATIONAL
INNOVATIONS AND
ENGINEERING APPLICATIONS

Volume 3 Number 1
JULY 2018

The **Journal of Computational Innovations and Engineering Applications (JCIEA)** is a peer-reviewed, open access journal of De La Salle University, Manila. The JCIEA aims to promote the development of new and creative ideas on the use of technology in solving different problems in different fields of our daily lives. The JCIEA solicits high quality papers containing original contributions in all areas of theory and applications of Engineering and Computing including but not limited to: Computational Applications, Computational Intelligence, Electronics and Information and Communications Technology (ICT), Manufacturing Engineering, Energy and Environment, Robotics, Control and Automation, and all their related fields. The JCIEA editorial board is comprised of experts from around the world who are proactively pushing for the development of research in these fields.

Annual Subscription Rates: Foreign libraries and institutions: US\$60 (airmail). Individuals: US\$50 (airmail). Philippine domestic subscription rates for libraries and institutions: Php1,800, individuals: Php1,300. Please contact Ms. Joanne Castañares for subscription details: telefax: (632) 523-4281, e-mail: dlsupublishinghouse@dlsu.edu.ph

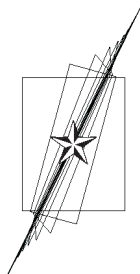
Copyright © 2017 by De La Salle University

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without written permission from the copyright owner.

ISSN 2507-9174

Published by De La Salle University Publishing House
2401 Taft Avenue, Manila 0922 Philippines
Telephone: (63 2) 523-4281 / 524-2611 loc 271
Fax: (63 2) 523-4281
Email: dlsupublishinghouse@dlsu.edu.ph
Website: <http://www.dlsu.edu.ph/offices/publishing-house/journals.asp>

The De La Salle University Publishing House is the publications office of De La Salle University, Manila, Philippines.



JOURNAL OF COMPUTATIONAL INNOVATIONS AND ENGINEERING APPLICATIONS

Table of Contents

From the Editor

Elmer P. Dadios
Editor-in-Chief

Research Articles

- Jackfruit Phytophthora Palmivora (Butler) Disease Recognizer Using Naïve Bayes Classifier 1
Jonah Flor V. Oraño, Jomari Joseph A. Barrera, and Elmer A. Maravillas
- ReTeSoil: A Temperature, Relative Humidity, and Soil Moisture Monitoring System 8
Using GSM with Blynk
Sittie Alyzza M. Garcia, Ariel James T. Abarquez, and Cristina P. Dadula
- A LabVIEW-Based Target Optimization Genetic Algorithm for Biological Predators 15
Ryann A. Alimuin, Elmer P. Dadios, and Argel A. Bandala
- Hybrid Sensor Based Fuzzy Clustering Neural Network Classification for 21
Human Activity Recognition
Ramtin Aminpour and Elmer P. Dadios
- Implementation of k-Nearest Neighbor (KNN) for PC-Based Character Recognition of 27
Currently Used Philippine Vehicle Standard Licensed Plate
*Klariz Donna Mae B. Bundal, Arlyn P. Lauron, Jan Jeffrey R. Camiña,
and Cristina P. Dadula*
- Real-Time Vehicle Classification Using MobileNet 38
Reagan L. Galvez, Melvin K. Cabatuan, and Argel A. Bandala

The Contributors

Guidelines for Contributors

From the Editor

The Journal of Computational Innovations and Engineering Applications (JCIEA) is a peer-reviewed and abstracted journal published twice a year by De La Salle University, Manila, Philippines. JCIEA aims to promote and facilitate the dissemination of quality research outputs that can push for the growth of the nation's research productivity.

In its third volume first issue, six articles were selected to provide valuable references for researchers and practitioners in the field of agricultural crop and livestock monitoring and control, crop disease detection, environmental condition assessment, wireless sensor networks, vision systems, intelligent transport systems, fuzzy logic control, and human activity recognition.

The first article is "Jackfruit Phytophthora Palmivora (Butler) Disease Recognizer Using Naïve Bayes Classifier". This paper presents a technique to detect the presence of *P. palmivora* disease in jackfruit trunk using Naïve Bayes classifier. Based on the result, the classifier achieved 94% accuracy in detecting the disease incidence.

The second article is "ReTeSoil: A Temperature, Relative Humidity, and Soil Moisture Monitoring System Using GSM with Blynk". This monitoring system was developed based on the temperature, relative humidity, and soil moisture content of a plant to minimize the occurrence of excessive and irregular irrigation that leads to certain problems like shortage in soil nutritive elements and decrease in productivity.

The third article is "A LabVIEW-Based Target Optimization Genetic Algorithm for Biological Predators". In the livestock industry, one of the major causes of destructive loss is reptilian predators; these are hunters that feed on fowls and their by-product. The aim of this research paper is to provide a solution for determining the locus (gene position) of a target such as predators and take consideration of some parameters such as speed of movement and location status.

The fourth article is "Hybrid Sensor Based Fuzzy Clustering Neural Network Classification for Human Activity Recognition". In this study, the fuzzy c-means has been considered by the ANFIS model to produce the fuzzy inference system (FIS) to make the classification with the neural network algorithm to detect the six major human activities. The results of the experiments show that the 97.2% accuracy could be acceptable in the field of study and the clustering structure could make the simulation more robust and faster.

The fifth article is "Implementation of k-Nearest Neighbor (KNN) for PC-Based Character Recognition of Currently Used Philippine Vehicle Standard Licensed Plate". This study focused on the development of a PC-based license plate recognition system using visual basic language. A system that can recognize the Philippines' currently standard vehicle plate numbers using EmguCV methods in image processing and KNN machine learning algorithm in recognizing characters.

The sixth article is "Real-time Vehicle Classification Using MobileNet". Image classification is an important part of vision systems and has several applications like autonomous cars and surveillance. This is a challenging task because computer sees images differently from humans. This paper used MobileNet model for training the data and tested it on android device. This model is lightweight and efficient compared with previous developed models.

The JCIEA editorial board expresses their warmest thanks and deepest gratitude to the distinguished authors for their outstanding contribution to JCIEA second volume first issue. They likewise express profound appreciation to the peer reviewers for their assistance and cooperation.

Original research outputs are most welcome to JCIEA. There is no publication fee in this journal, and the research papers are assured of fair and fast peer review process. For further information, please visit www.dlsu.edu.ph/offices/publishinghouse/journals.asp.

Prof. Elmer P. Dadios, PhD
Editor-in-Chief, JCIEA

Jackfruit *Phytophthora Palmivora* (Butler) Disease Recognizer Using Naïve Bayes Classifier

Jonah Flor V. Oraño,^{1,*} Jomari Joseph A. Barrera,² and Elmer A. Maravillas,³

Abstract

Abstract— This paper presents a technique to detect the presence of *P. palmivora* disease in jackfruit trunk using Naïve Bayes classifier. In this study, 200 sample images of jackfruit trunk were used, which were divided into two sets: for training and for testing. Each set contains 50 images for healthy and 50 images for disease infected. The input images were subjected to image pre-processing such as cropping, scaling, and brightness and contrast adjustment. Then, the images were segmented into two regions using color masking. Texture features such as angular second moment (uniformity) and sum of squares (variance) were also extracted from the images. Next, Naïve Bayes classifier was used to classify whether the jackfruit is infected with the disease or not. Finally, the performance of the classifier was evaluated by computing the overall accuracy of the system. Based on the result, the classifier achieved 94% accuracy in detecting the disease incidence. Moreover, this rate can be further improved by adding texture features and by applying other classification algorithms.

Keywords: *Phytophthora palmivora*, Naïve Bayes classification, texture features, image processing

I. INTRODUCTION

Jackfruit, which is scientifically known as *Artocarpus heterophyllus* Lam and locally known as “nangka” or “langka,” is one of the most widely grown fruit crops in the Philippines, which produces the largest edible fruit that may weigh as much as 50 kg [1]. It is an emerging industry in the country especially in Eastern Visayas, where it is marketed as the “flagship fruit.” It is a multipurpose

tropical fruit tree which can be the source of food, timber, fodder, dyes, latex, and medicinal and other value-added products [2]. In 2013, jackfruit was planted in a total area of 14,526 hectares (ha) with a total production of 46,080 metric tons (mt) [3]. A follow-up report in [4] shows that production of jackfruit amounted to about 44,605 mt in 2014, 43,666 mt in 2015, and 42,021 mt in 2016. It indicates that the country’s production continued to decline.

This decline can be attributed to a variety of reasons such as natural calamities, pests, and boring insects; however, damage caused by pathogens plays a significant role in crop reduction in both quality and quantity. To name one, *Phytophthora palmivora* (Butler), which is known as one of the most destructive genera of plant pathogens in temperate and tropical regions [5], was identified as the major cause of jackfruit production decline in southern Philippines, which affects 85% of jackfruit orchards in Leyte and Samar [6]. The occurrence of this disease hampered its production and threatened the livelihood of local farmers.

Jackfruit infected with *P. palmivora* shows symptoms including trunk cankers (Fig. 1), chlorosis and wilting of the foliage, root lesions, and tree death. As described by the authors in [7], stem cankers appeared firstly as wet lesions on the bark surfaces, often close to the insertion of large branches, but more frequently at trunk bases. A reddish-brown resin oozed from cracks in the bark. The wood tissues under the lesions showed cream to reddish brown discoloration. The infected areas enlarged, girdling the stems and causing severe decline of the trees.

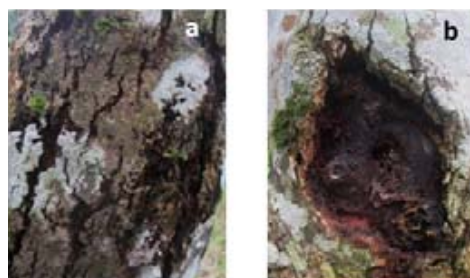


Fig. 1. Jackfruit infected with *P. palmivora* disease. a. Exterior tree trunk showing canker lesions. b. Exterior surface removed to show reddish color disease

¹Jonah Flor V. Oraño, Visayas State University, Visca Baybay City, Leyte, Philippines (e-mail: jforano@gmail.com)

²Jomari Joseph A. Barrera, Visayas State University, Visca Baybay City, Leyte, Philippines

³Elmer A. Maravillas CIT-University, N. Bacalso Ave., Cebu City, Philippines

If timely handling and proper management are not taken into consideration, this disease poses a serious threat to jackfruit yield and long-term viability of plantation. In this context, early and advanced disease detection is of utmost importance. Currently, jackfruit growers and domain experts identify the occurrence of the disease through naked-eye observation and laboratory tests. On the other hand, these could be time consuming and laborious especially in monitoring big jackfruit orchards. With the advent of technology, this process can be automated through image processing and machine learning techniques. These techniques have been applied to various agricultural applications such as to detect the incidence and severity of plant diseases, to determine plant varieties, and to identify the quality grading level of fruits and vegetables. For instance, in the study of [8], features such as color, morphology, and color coherent vector (CCV) were extracted and support vector machine (SVM) classification was used to determine incidence of pomegranate disease, in which the authors obtained an accuracy rate of 82%. Aside from SVM, other classification algorithms were proven to be effective, notably the Naïve Bayes [9] classifier wherein it outperforms the conventional classifiers as indicated in the studies of [10] and [11] in terms of classification accuracy.

There are other numerous studies pertaining to automated crop diseases identification [12] [13]; however, detection of *P. palmivora* disease occurrence in jackfruit does not exist yet. This could be beneficial in monitoring big farms and in the absence of domain experts, hence the conduct of this study.

II. METHODOLOGY

Figure 2 illustrates that the system architecture applied in this study consists of two main components: image processing and Naïve Bayes classification. In the following, details about each component were presented.

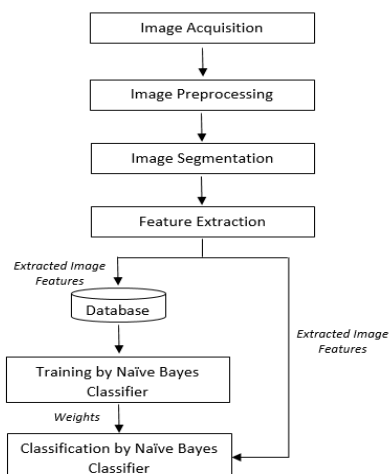


Fig. 2. System architecture of jackfruit *P. palmivora* disease recognizer.

A. Image Acquisition and Image Pre-Processing

The images of jackfruit trunk (healthy and infected) were captured using a 12-megapixel digital camera last July 24, 2017, and September 5, 2017, in two jackfruit farms located at Mahaplag, Leyte. Cropping was applied to images to remove unwanted details/objects and to emphasize the region of interest. Further pre-processing such as scaling into 600×600 and brightness and contrast adjustments were also performed.

B. Image Segmentation and Feature Extraction

Features such as color and texture were extracted from the images. For the color features, images were segmented into two regions using color masking based on the formula shown in Eq. (1). Threshold values that indicate the color characteristics of the infected region are reflected in Figure 3.

$$f(C_i) = \begin{cases} R_1, & \text{if } \angle_i(\text{hue}(C_i)) < 10^\circ \mid \text{brightness}(C_i) < 0.0625 \\ R_2, & \text{otherwise} \end{cases} \quad (1)$$

where

- C_i is color
- R_1 is infected region
- R_2 is not infected region

Hue is an angle between 0° and 360° . Equations (2) and (3) show the computations for hue angle and brightness [14].

$$\text{hue}(C_i) = \begin{cases} 0 & \text{if max} = \text{min} \\ (60^\circ \times \frac{g-b}{\text{max}-\text{min}} + 0^\circ) \bmod 360^\circ, & \text{if max} = r \\ 60^\circ \times \frac{b-r}{\text{max}-\text{min}} + 120^\circ, & \text{if max} = g \\ 60^\circ \times \frac{r-g}{\text{max}-\text{min}} + 240^\circ, & \text{if max} = b \end{cases} \quad (2)$$

$$\text{brightness}(C_i) = \frac{1}{2} (\text{max} + \text{min}) \quad (3)$$

The angle of incidence (\angle) is used to compute the distance between hue of red and the hue of C_i .

$$\angle_i(\theta) = \min(360^\circ - \theta, \theta) \quad (4)$$

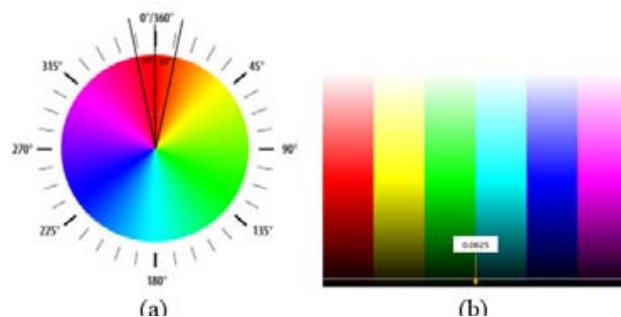


Fig. 3. Color space for (a) hue and (b) brightness.

Furthermore, for the texture features, two of the 14 descriptors defined by Haralick [15] from GLCM were extracted. These include angular second moment (Eq. 5) and sum of squares: variance (Eq. 6). GLCM produces features which describe well the relationship of adjacency among pixels in a texture image.

$$\text{Angular Second Moment (ASM)} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} p(i,j)^2 \tag{5}$$

$$\text{Sum of Squares: Variance} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - \mu)^2 p(i,j) \tag{6}$$

The extracted features values were then categorized based on the defined range of values for a specific category level shown in Table 1 to Table 3. These features from all images in the training set were stored in the database to be used in the classification process.

TABLE 1
CATEGORY OF VALUES FOR THE MASKED REGIONS

Category	Values
Very low	>0 to <=10
Low	>10 to <=20
Moderate	>20 to <= 35
High	>35 to <= 55
Very high	>55

TABLE 2
CATEGORY OF VALUES FOR ASM

Category	Values
Very low	<= 0.00105200
Low	>0.00105200 to <=0.00203048
Moderate	>0.00203048 to <=0.00300897
High	>0.00300897 to <=0.00398745
Very high	>0.00398745

TABLE 3
CATEGORY OF VALUES FOR VARIANCE

Category	Values
Very low	<=58.05314194
Low	>58.05314194 to <=92.45157227
Moderate	>92.45157227 to <=126.85000259
High	>126.85000259 to <=161.24843291
Very high	>161.24843291

C. Training and Classification Phase

During the training phase, a data set of 100 images was used, which is composed of 50 healthy trunks and 50 infected trunks.

Class values for Naïve Bayes classifier were defined, which include: not infected and infected. The probabilities for each attribute/feature conditional on the class value were computed. Then, product rule was applied to obtain a joint conditional probability for the attributes while Bayes' rule in Eq. (7) was used to derive the conditional probabilities for each class, wherein the class with the highest probability was considered as the outcome of the prediction [16].

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{7}$$

where

- $P(c|x)$ —posterior probability
- $P(x|c)$ —likelihood
- $P(c)$ —class prior probability
- $P(x)$ —predictor prior probability

III. EXPERIMENTAL RESULTS

The graphical interface for the training phase is shown in Figure 4. It illustrates that once the image is loaded into the system, the extracted feature values and its equivalent category level are displayed. The user then needs to indicate the expert's classification on the image before clicking the save button. These values will be saved into the database and will be used for the training process.



Fig. 4. Features extraction result of the system.

The difference between healthy jackfruit trunks and those with *P. palmivora* disease infection is indicated in Table 4.

TABLE 4
SAMPLE IMAGES OF INFECTED AND NOT INFECTED
JACKFRUIT TRUNKS

INFECTED	NOT INFECTED
	
	
	

Table 5 illustrates the training set for the system after extracting features from all 100 images. It contains columns for extracted numerical values, its equivalent category, and its expected classification values.

Apart from 100 jackfruit trunk images used during the training phase, a different set of 100 images was used for the testing phase. These images were loaded into the system and were then converted into features form. For each image, the predicted class label obtained by Naïve Bayes classifier was saved into the database table alongside with its values (Table 6).

The system can now be used to determine whether the image of the jackfruit trunk shows occurrence of the disease or not. Figure 5 illustrates the decision of the classifier based on the features extracted from the given image.

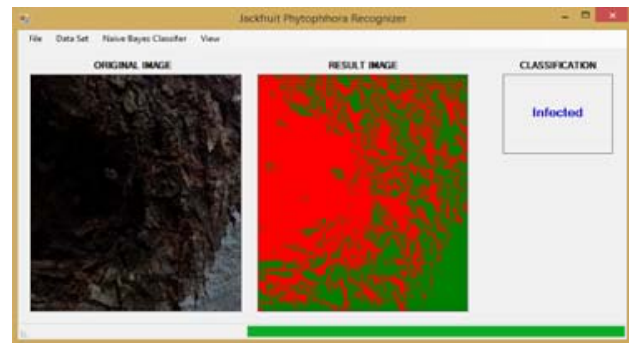


Fig.5. Recognition result of the system.

TABLE 5
TRAINING DATASET

Sample No.	Mask 1	Mask 2	ASM	Variance	Mask 1 Category	Mask 2 Category	ASM Category	Variance Category	Expected Result
1	34.76	65.24	0.00051	55.11715	Moderate	Very High	Very Low	Very Low	Infected
2	48.62	51.38	0.00252	23.65471	High	High	Moderate	Very Low	Infected
3	36.17	63.83	0.00033	46.04551	High	Very High	Very Low	Very Low	Infected
4	22.99	70.01	0.00108	30.03209	Moderate	Very High	Low	Very Low	Infected
5	59.31	40.69	0.00497	23.97449	Very High	High	Very High	Very Low	Infected
6	18.80	81.20	0.00027	54.99160	Low	Very High	Very Low	Very Low	Infected
7	4.22	95.78	0.00011	140.87128	Very Low	Very High	Very Low	High	Not Infected
8	3.15	96.85	0.00018	124.68015	Very Low	Very High	Very Low	Moderate	Not Infected
9	10.31	89.69	0.00012	134.12725	Low	Very High	Very Low	High	Not Infected
10	7.22	92.78	0.00016	169.53530	Very Low	Very High	Very Low	Very High	Not Infected
11	12.50	87.50	0.0009	178.21163	Low	Very High	Very Low	Very High	Not Infected
:	:	:	:	:	:	:	:	:	:
100	0.16	99.84	0.00025	78.19798	Very Low	Very High	Very Low	Low	Not Infected

TABLE 6
TESTING DATASET

Sample No.	Mask 1	Mask 2	ASM	Variance	Mask 1 Category	Mask 2 Category	ASM Category	Variance Category	Expected Result	Actual Result
1	43.57	56.43	0.00031	36.31061	High	Very High	Very Low	Very Low	Infected	Infected
2	33.09	66.91	0.00008	31.70688	Moderate	Very High	Very Low	Very Low	Infected	Infected
3	30.21	69.79	0.00043	71.98274	Moderate	Very High	Very Low	Low	Infected	Infected
4	19.57	80.43	0.00012	103.2699	Low	Very High	Very Low	Moderate	Infected	Not Infected
5	16.54	83.46	0.00007	105.0948	Low	Very High	Very Low	Moderate	Infected	Not Infected
6	16.04	83.96	0.00021	67.90312	Low	Very High	Very Low	Low	Infected	Infected
7	13.71	86.29	0.00030	84.07898	Low	Very High	Very Low	Low	Infected	Infected
8	12.3	87.70	0.00018	160.4463	Low	Very High	Very Low	High	Not Infected	Not Infected
9	10.15	89.85	0.00023	96.82886	Low	Very High	Very Low	Moderate	Infected	Not Infected
10	9.82	90.18	0.00012	166.3141	Very Low	Very High	Very Low	Very High	Not Infected	Not Infected
11	7.28	92.72	0.00016	118.4576	Very Low	Very High	Very Low	Moderate	Not Infected	Not Infected
:	:	:	:	:	:	:	:	:	:	:
100	7.02	92.98	0.00015	107.681	Very Low	Very High	Very Low	Moderate	Not Infected	Not Infected

A. Discussions

Table 7 shows the classifier created from the training set where NI means not infected and I means infected. The probabilities are computed as the number of samples of a class having the value (category) for a feature divided by the frequency of a class. In this case, the probability of not infected p(NI) is 0.50 and the probability of infected p(I) is 0.50 as well.

TABLE 7
PROBABILITY OF FEATURE CONDITIONAL ON THE CLASS

Category	Features							
	Mask 1		Mask 2		ASM		Variance	
	NI	I	NI	I	NI	I	NI	I
Very low	.47	.00	.00	.00	.50	.43	.00	.44
Low	.03	.20	.00	.00	.00	.05	.01	.06
Moderate	.00	.22	.00	.00	.00	.01	.11	.00
High	.00	.07	.00	.02	.00	.00	.34	.00
Very high	.00	.01	.50	.48	.00	.01	.04	.00

Table 8 below contains the values for all the extracted features from a sample image to be classified as not infected or infected.

TABLE 8
SAMPLE EXTRACTED VALUES TO BE CLASSIFIED AS INFECTED OR NOT INFECTED

Feature	Value
Mask 1	Low
Mask 2	Very high
ASM	Very low
Variance	Low

For the classification as not infected, the posterior is given by Eq. (8).

$$\begin{aligned}
 & \text{posterior(NotInfected)} \\
 &= \frac{p(\text{Mask 1|NotInfected}) p(\text{Mask 2|NotInfected}) p(\text{ASM|NotInfected}) p(\text{Variance|NotInfected}) p(\text{NotInfected})}{\text{predictor}} \quad (8)
 \end{aligned}$$

While for the classification as infected, the posterior is given by Eq. (9).

$$\begin{aligned}
 & \text{posterior(Infected)} \\
 &= \frac{p(\text{Mask 1|Infected}) p(\text{Mask 2|Infected}) p(\text{ASM|Infected}) p(\text{Variance|Infected}) p(\text{Infected})}{\text{predictor}} \quad (9)
 \end{aligned}$$

where

$$\begin{aligned}
 \text{predictor} = & p(\text{Mask 1|NotInfected}) p(\text{Mask 2|NotInfected}) p(\text{ASM|NotInfected}) \\
 & p(\text{Variance|NotInfected}) p(\text{NotInfected}) + \\
 & p(\text{Mask 1|Infected}) p(\text{Mask 2|Infected}) p(\text{ASM|Infected}) \\
 & p(\text{Variance|Infected}) p(\text{Infected})
 \end{aligned}$$

However, given the sample, the predictor is a constant and thus scales both posteriors equally. In this case, it does not affect the classification and can be ignored.

Therefore, considering the posterior numerator of Not Infected, which was calculated as

$$\begin{aligned} \text{posterior}(\text{NotInfected}) \\ = .03 \times .50 \times .50 \times .01 \times .50 = .00004 \end{aligned}$$

and the posterior numerator of Infected, which was calculated as

$$\begin{aligned} \text{posterior}(\text{Infected}) \\ = .20 \times .48 \times .43 \times .06 \times .50 = .00012 \end{aligned}$$

it can be predicted that the sample is infected.

For each image, the predicted class label obtained by Naïve Bayes classifier was then compared to the actual class label specified by the domain expert. The overall results are displayed in the confusion matrix (Table 9) to emphasize how many images from the total of each class are accurately predicted.

Then detection accuracy for each class was compute as shown in Eq. (10).

$$\frac{\text{sum of correctly predicted class label}}{\text{total number of predictions}} \quad (10)$$

While the overall accuracy of the system was calculated using Eq. (11).

$$\frac{\text{sum of correct classification}}{\text{total number of classification}} \quad (10)$$

TABLE 9
CONFUSION MATRIX

Class	Not Infected	Infected	Detection Accuracy
Not Infected	50	0	100.00%
Infected	6	44	88.00%
Overall Accuracy			94.00%

The result shows that the detection accuracy of infected class is lower (88%) compared to the not infected class (100%). However, it is notable that the system was able to achieve a high overall accuracy rate of 94%.

IV. CONCLUSION AND RECOMMENDATION

The overall accuracy rate of the system, which is 94%, indicates that the application of Naïve Bayes classifier on the extracted color and texture features can significantly support an accurate detection of the *P. palmivora* disease.

The recognition accuracy rate can be further improved by using more high-resolution images, applying other image processing techniques, adding more features, and applying other classification algorithms such as artificial neural network, C4.5 classifier, SVM, etc. It would also be better that the system be able to classify the severity of the disease infection for a basis of appropriate disease management strategies.

REFERENCES

- [1] "Jakfruit in the Philippines, Part 1," March 1985. [Online]. Available: <http://rfcarchives.org.au>.
- [2] C. R. Elevitch and H. I. Manner, "Artocarpus heterophyllum (jackfruit)," April 2006. [Online]. Available: www.traditionaltree.org.
- [3] "Philippines: Jackfruit planted area and production 2008-2013," [Online]. Available: <https://www.statista.com/>.
- [4] "CountrySTAT Philippines," 10 November 2017. [Online]. Available: <http://countrystat.psa.gov.ph>. [Accessed 18 April 2018].
- [5] A. Drenth and D. Guest, "Diversity and Management of Phytophthora," *ACIAR Monograph series no. 114. Australian Centre for International Agricultural Research, Canberra, 2004*.
- [6] L. Borines, V. Palermo, G. Guadalquiver, C. Dwyer, A. Drenth, R. Daniel and D. Guest, "Jackfruit decline caused by Phytophthora palmivora (Butler)," *Australasian Plant Pathology*, vol. 42, pp. 123-129, 2013.
- [7] M. Van Tri, N. Van Hoa, N. Minh Chau, A. Pane, R. Faedda, A. De Patrizio, L. Schena, C. H. Olsson, S. A. Wright, M. Ramstedt and S. O. Cacciola, "Decline of jackfruit (*Artocarpus heterophyllum*) incited by Phytophthora," *Phytopathologia Mediterranea*, pp. 9-14, 2015.
- [8] M. Bhange and H. Hingoliwala, "Smart Farming: Pomegranate Disease Detection Using Image Processing," *Procedia Computer Science*, pp. 280-288, 2015.
- [9] M. M. Roomi and S. Saranya, "Bayesian Classification of Fabrics Using Binary Cooccurrence Matrix," *International Journal of Information Sciences and Techniques (IJIST)*, vol. 2, no. 1, pp. 1-9, 2012.
- [10] D.-C. Park, "Image Classification Using Naive Bayes Classifier," *International Journal of Computer Science and Electronics Engineering (IJCSEE)*, vol. 4, no. 3, pp. 135-139, 2016.
- [11] M. Mursalin, M. M. Hossain, M. K. Noman and M. S. Azam, "Performance Analysis among Different Classifier Including Naive Bayes, Support Vector Machine and C4.5 for Automatic Weeds Classification," *Global Journal of Computer Science and Technology Graphics & Vision*, vol. 13, no. 3, 2013.
- [12] A. Fahrrozi, S. Madenda, Ernastuti and D. Kerami, "Wood Classification Based on Edge Detections and Texture Features Selection," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 5, p. 2016, 2167-2175.

- [13] S. Arivazhagan, R. N. Shebiah, S. Ananthi and S. V. Varthini, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features," *Agric Eng Int: CIGR Journal*, vol. 15, no. 1, pp. 211-217, 2013.
- [14] M. K. Agoston, *Computer Graphics and Geometric Modeling: Implementation and Algorithms*, Springer, 2005.
- [15] R. Haralick, K. Shanmugam and L. Dinstein, "Textural Features for Image Classification," *IEEE Transaction on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610-621, 1973.
- [16] M. K. Leung, "Naive Bayesian Classifier," 28 November 2007. [Online]. Available: cis.poly.edu/~mleung/FRE7851/f07/naiveBayesianClassifier.pdf. [Accessed 2018].

ReTeSoil: A Temperature, Relative Humidity, and Soil Moisture Monitoring System Using GSM With Blynk

Sittie Alyzza M. Garcia, Ariel James T. Abarquez , and Cristina P. Dadula

Abstract—This monitoring system was developed based on the temperature, relative humidity, and soil moisture content of a plant to minimize the occurrence of excessive and irregular irrigation that leads to certain problems like shortage in soil nutritive elements and decrease in productivity. The system is composed of two nodes: the sensor node and the base node, and BLYNK mobile application program. The sensor node consists of sensors and a transmitter for the sending of data to base node. The base node is responsible for accepting data from the sensor node and it consists of a receiver and a Global System for Mobile communication (GSM) module to relay the received data to the BLYNK cloud. The system used RF communication for sensor to base node data transmission and Global Packet Radio Service (GPRS) for base node to BLYNK cloud data transmission. The BLYNK application program is used to monitor and display the data from the sensors, and sends email notification when the measured value of at least one of the sensors is below or above the set limit. All functionalities of the application program were working. The maximum operating distance for the RF module was 35 m when both receiver and transmitter modules have antenna installed. In addition, it was observed that GPRS communication was not stable, a noticeable time delay was experienced in displaying sensor data to the application. Hence, the use of more stable communication like Wi-Fi is highly recommended to avoid some communication issues.

Keywords: monitoring temperature, relative humidity, and soil moisture; GSM, GPRS, RF communications, Wi-Fi, wireless sensor network.

Sittie Alyzza M. Garcia, Ariel James T. Abarquez , and Cristina P. Dadula, Mindanao State University, General Santos City, Philippines (e-mail: garciasittie21@gmail.com, ajs.abarquez97@gmail.com, cris_dadula@yahoo.com)

I. INTRODUCTION

Excessive and irregular irrigation is not good for both plants and soil. This will result to some serious problems like shortage in soil nutritive elements, decreased productivity, or increase in salinity. Thus, causing the soil nutrients to be submerged and remain in the subsoil while the groundwater comes up to the surface, the groundwater undergoes evaporation and forms salts on the soil [1]. This leads to decrease in quality and productivity of the soil. To avoid these problems, some systems were designed to remotely monitor the humidity of the soil. Whenever the humidity reached a certain level, a motor pump is turned on to deliver the required water [2]. There is also a monitoring system designed for precision agriculture implemented using wireless sensor nodes. These sensors are spread through the field to periodically collect and relay soil data to the processing centers [3]. Gerard Rudolph Mendez et al. [4] have performed temperature and soil humidity monitoring by using Wi-Fi communication; for sensor node, they used the WSN802G module. Sensor measurements are collected and stored in the server for further analysis. Kishore Babu and Saggam Divyasri [5] have made a similar system which used GPRS for transmitting data instead of WiFi. Nurul Fahmi et al. [6] also implemented a wireless sensor network for the monitoring of a precision agriculture system. They made a simple prototype that used temperature, humidity, pressure, and soil moisture sensors. This prototype can monitor environment status through a website and even on smartphone. In 2013, H.A. Mansour et al. [7] have studied the effect of automatic control on closed circuit drip irrigation system as a modified irrigation system on yellow corn crop vegetative and yield parameters.

This research aims to develop a microcontroller-based system for monitoring of essential plant growth factors that can be remotely accessed by the user wirelessly through an application program. The objectives of the study are: 1) to develop a system prototype that uses two main sensors (DHT11 humidity and temperature sensor, and soil moisture sensor), microcontroller, RF module and

GSM module, 2) to use BLYNK to build an application for displaying data, 3) to transmit data from the sensor node to the base node using RF communication, 4) to transmit data from the base node to the BLYNK cloud using GPRS, and lastly 5) to test the functionality of the RF communication range, stability of connection of the GSM module, and BLYNK cloud data transmission.

II. AGRICULTURAL MONITORING AND CONTROL SYSTEMS USING WIRELESS SENSOR NETWORK (WSN)

The study discussed in [3] demonstrate the design and development of a Wi-Fi-based WSN capable of intelligent monitoring of agricultural environment conditions using a pre-programmed control and management system that can be updated as required. The WSN has three components which are the sensor nodes, access point, and central station. The sensor node is capable of data collection of parameters such as relative humidity, air pressure, temperature, soil moisture, presence of light, and nutrient levels. On the other hand, the study conducted in [4] used WSN to aid farmers for real-time monitoring of different agricultural areas. A simple prototype for precision agriculture was developed that monitors humidity, soil moisture, temperature, and atmospheric pressure. The study discussed in [5] shows the effect of automatic control closed circuits drip irrigation system on yellow corn crop. This was conducted in the Al-Hasa region of Saudi Arabia. A microcontroller-based irrigation system was discussed in [6]. The researcher wants to develop a system that can work constantly in a remote location, even under abnormal conditions. The system monitors the soil humidity and delivers water whenever the humidity reaches a certain level. The study discussed in [7], developed a distributed wireless sensor network which is powered by photovoltaic panels. Soil moisture and temperature sensors are placed in the plant's roots to gather information. Sensor measurements are sent into a gateway to display values in a web application. These measurements are also used to trigger actuators for control of some system variables, such as water quantity. The system also used a communication link that used cellular-Internet interface for data transmission that allows for remote inspection and irrigation scheduling.

III. METHODOLOGY

This section presents the design and development of both the software and hardware components of the system. For the software application, BLYNK was used to develop a user interface and mobile application to display the data

coming from the sensor node. It is an Internet of Things (IoT) platform suitable only for smartphones [8]. Meanwhile, Arduino IDE was used for the development of algorithm needed by the hardware component of the system. Arduino IDE is an open-source software that is compatible with the application and board used in the system [9]. This allows for the hardware component to interact with the user application.

The system has two circuits: the base node and sensor node, as shown in Figure 1. The sensor node is composed of an Arduino Uno board, sensors (namely, DHT11, the temperature and humidity sensor, and the soil moisture sensor), and RF transmitter module to send the data to the base node. This base node is composed of an Arduino Uno board, a GSM module, and an RF receiver module. Sensor node will send data to the base node through RF communication using a TX/RX RF module. Moreover, the data received by the base node will also be sent to the BLYNK cloud by the GSM module so that the developed application can access the data anytime and anywhere.

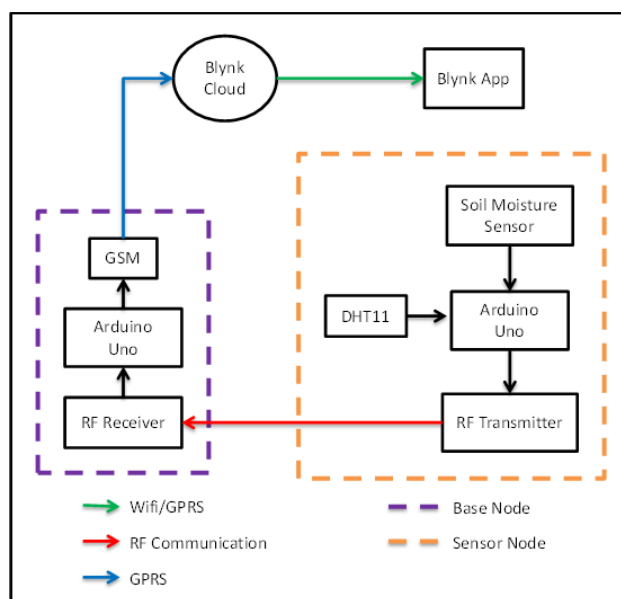


Fig. 1. Conceptual system framework.

A. Hardware Development

The project's hardware development consists of two circuitries: the sensor node and the base node construction.

Sensor Node

Sensor node is responsible for remote measurements and transmission of data [10]. It consists of Arduino Uno, RF transmitter module, DHT11, soil moisture sensor, and a 9V battery. The DHT11 sensor will gather the data for the temperature and relative humidity. It can also determine the soil's moisture using soil moisture sensor. Once all the data

are gathered, the RF transmitter will transmit all the data to the base node. Figure 2 shows the schematic diagram of the sensor node.

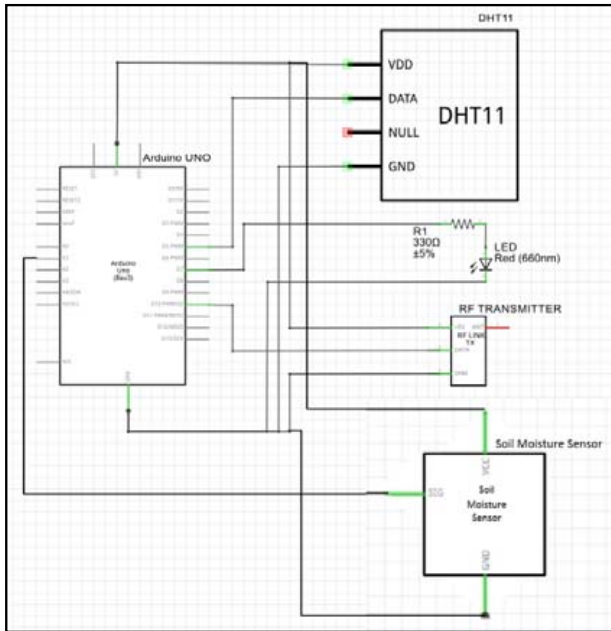


Fig. 2. Sensor node schematic diagram.

Base Node

The transmitted data coming from the sensor node will be received by the base node. It consists of Arduino Uno, GSM module, RF receiver module, and a 9V battery. When the data are transmitted from sensor node to the RF receiver module, the data will be forwarded to the BLYNK cloud by the GSM module so that the developed application can access the data anytime and anywhere.

The RF module that was used in the system operates in 433 MHz frequency band. This means that the RF transmitter and receiver will operate at the same frequency [11]. RF modules are commonly used in wireless communication because of their stability and wide coverage but is very prone to multipath fading and other error sources [12]. Another module that was used is the GSM sim900a module [13]. This module is capable of data communications between the base node and the BLYNK server. Figure 3 shows the schematic diagram of the base node.

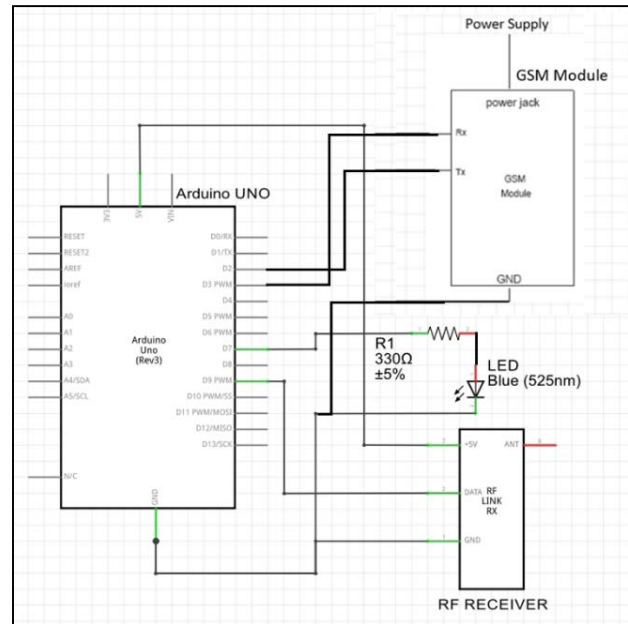


Fig. 3. Base node schematic diagram.

B. Software Application Development using BLYNK

BLYNK application was used in the software development. This application has the necessary functions in developing the system. Figure 4 shows how Blynk will be setup. It used a drag-and-drop feature in building the application and can communicate with Arduino IDE just by installing the needed libraries. The functions include the monitoring of the following sensors: (1) soil moisture, (2) temperature, (3) humidity, and (4) the *Timeline* for the real-time data of the sensors. With these in mind, the main layout of the BLYNK application should contain three gauges for every data of the sensors, SuperChart for the timeline of real-time data, and a notification and email widget. The notification function is used if sensor readings are above the set limit. However, to be able to monitor the sensors with these gauges and the SuperChart, it is then necessary for the BLYNK application to be able to communicate with the Arduino UNO board via the GSM Sim900a module. To do so, the GSM Sim900a module must have a data connection to access the BLYNK server, to monitor the data from the sensors which will be displayed by the BLYNK application. The codes are modified to make sure that the gauge and SuperChart can get the data from the sensors to monitor the soil moisture temperature, and relative humidity. The codes include the communication between the application and the Arduino. Figure 5 shows the layout of the BLYNK application.



Fig. 4. BLYNK setup [14].

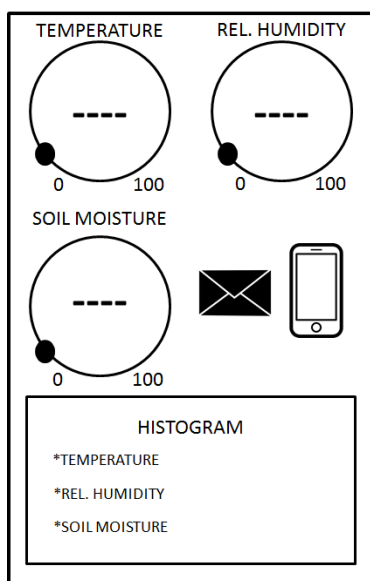


Figure 5. Application layout.

C. Hardware and Software Integration

After the necessary codes are finalized, testing whether the current codes are suitable to complete the anticipated functionalities of the hardware takes place. The researchers used the Arduino IDE Serial Monitor to check the initialization and connectivity of the GSM module and the BLYNK cloud, as shown in Figure 6. This initialization of the GSM module takes time depending on signal strength and where the node is being set up. Figure 7 shows that the BLYNK application is already connected to the Arduino IDE. It also displays the data from the sensors.

D. Distance Test

The transmission of data from the sensor node to base node was also tested for different distances (from 0 to 45 m) with the presence or absence of the antenna of the transmitter module and receiver module.

```

Initializing modem...
[15335]

v0.5.2 on Arduino Uno

[15342] Modem init...
[16247] Connecting to network...
[16559] Network: Globe Telecom-PH
[16559] Connecting to internet.globe.com.ph ...
[23806] Connected to GPRS
[24086] Connecting to blynk-cloud.com:80
[27984] Ready (ping: 836ms).
[60425] Connecting to blynk-cloud.com:80
[62874] Ready (ping: 852ms).
[96156] Connecting to blynk-cloud.com:80
    
```

Fig. 6. The initialization of the GSM module, network connection, Access Point Name (APN), Global Packet Radio System (GPRS) status, and BLYNK cloud connection.

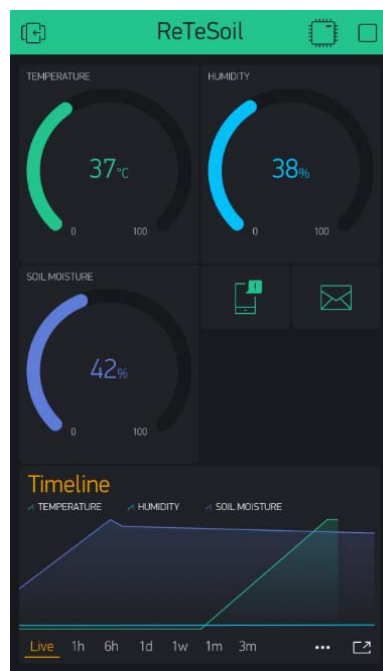


Fig. 7. Displaying of data using BLYNK application.

IV. RESULTS AND DISCUSSION

Figure 8 shows the final version of the BLYNK application, which has an interface that allows the user to monitor sensor data, through the GSM module connection.

A. Final Version of the BLYNK Application

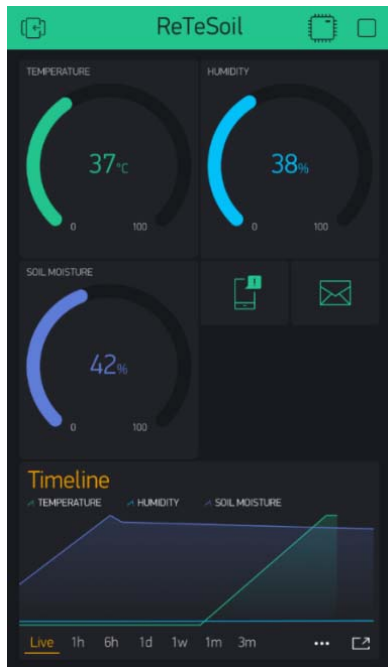


Fig. 8. BLYNK application main interface.

To monitor the data from the sensor node, the three gauges display the numeric data of the sensors, which are the DHT11 sensor (temperature and relative humidity sensor) and soil moisture sensor. The SuperChart shows the live and historical data of the sensors. The email widget allows the user to send email from the hardware to any email address if the sensor readings are above the set limit. The push notification widget allows the user to send push notification from the hardware to the user's device in case the hardware went offline and when the data from the sensors are above the set limit [15].

B. Initializing the GSM Modem

This section shows the results in the Arduino IDE serial monitor when the GSM module is being initialized, as shown in Figure 9. The initialization of the GSM module to connect to the data services of the network takes a bit longer depending upon the signal strength or the area where the base node is being set up. After the initialization, establishment of connectivity between the application and the module was initiated. Some problems encountered about the initialization of the GSM module include: 1) there were times that the module did not initialize properly and proceeded on restarting to re-initialize the module, and 2) there were also times that the researchers need to unplug the power supply and then plug it again.

```

Initializing modem...
[15335]

  _ _ _ _ _
 / _ \ / / _ _ _ / /
/_ _ \ / / / / _ \ \
/_ _ \ / \ / / / \ \
  / _ \ / v0.5.2 on Arduino Uno

[15342] Modem init...
[16247] Connecting to network...
[16559] Network: Globe Telecom-PH
[16559] Connecting to internet.globe.com.ph ...
[23806] Connected to GPRS
[24086] Connecting to blynk-cloud.com:80
[27984] Ready (ping: 836ms).
[60425] Connecting to blynk-cloud.com:80
[62874] Ready (ping: 852ms).
[96156] Connecting to blynk-cloud.com:80

```

Fig. 9. Arduino IDE Serial Monitor—GSM modem initializing.

C. Establishment of Connection between the BLYNK Application and the GSM Module

This section shows the application when the GSM module is connected to the BLYNK application. Figure 10 shows the Arduino's response to establish the connection.

```

Initializing modem...
[15335]

  _ _ _ _ _
 / _ \ / / _ _ _ / /
/_ _ \ / / / / _ \ \
/_ _ \ / \ / / / \ \
  / _ \ / v0.5.2 on Arduino Uno

[15342] Modem init...
[16247] Connecting to network...
[16559] Network: Globe Telecom-PH
[16559] Connecting to internet.globe.com.ph ...
[23806] Connected to GPRS
[24086] Connecting to blynk-cloud.com:80
[27984] Ready (ping: 836ms).
[60425] Connecting to blynk-cloud.com:80
[62874] Ready (ping: 852ms).
[96156] Connecting to blynk-cloud.com:80

```

Fig. 10. BLYNK is connected to the GSM module.

The researchers encountered some stability problems with the establishment of connection between the application and the GSM module. Figure 10 shows how the connection was established for a couple of seconds, and then it will be disconnected, and connected again afterwards.

D. Distance Test for RF Communication

Table 1 shows the summary of all the processes that can be performed by the RF module as mentioned above. Note that this evaluation was done while performing a distance test. As such, the table below shows the results for a distance test of up to 45 m.

TABLE 1
RF COMMUNICATION RANGE TESTING SUMMARY

Distance	TX With Antenna	RX With Antenna	Remarks
0 m	NO	NO	Transmit data
5 m	NO	NO	Failed
5 m	YES	NO	Transmit data
10 m	YES	NO	Transmit data
15 m	YES	NO	Failed
15 m	YES	YES	Transmit data
20 m	YES	YES	Transmit data
25 m	YES	YES	Transmit data
30 m	YES	YES	Transmit data
35 m	YES	YES	Transmit data
40 m	YES	YES	Failed
45 m	YES	YES	Failed

During testing, the transmission stability was poor when the distance was about on its limit (40 m). The base node, most of the time, can receive data as long as the receiver antenna was installed. Otherwise, the connection was not stable, causing it to poorly receive the arriving data. With the antenna installed, the transmission was successful up to 35 m.

E. Displaying of Data using BLYNK Application

The application successfully established the necessary hardware-to-hardware communication, hardware-to-software communication, hence it successfully displayed the data coming from the sensor node as shown in Figure 11. The researchers set the threshold value of each gauges, as shown in Table 2. So if the reading value of the temperature, for example, is above the threshold value, notification will appear, as shown in Figure 12.

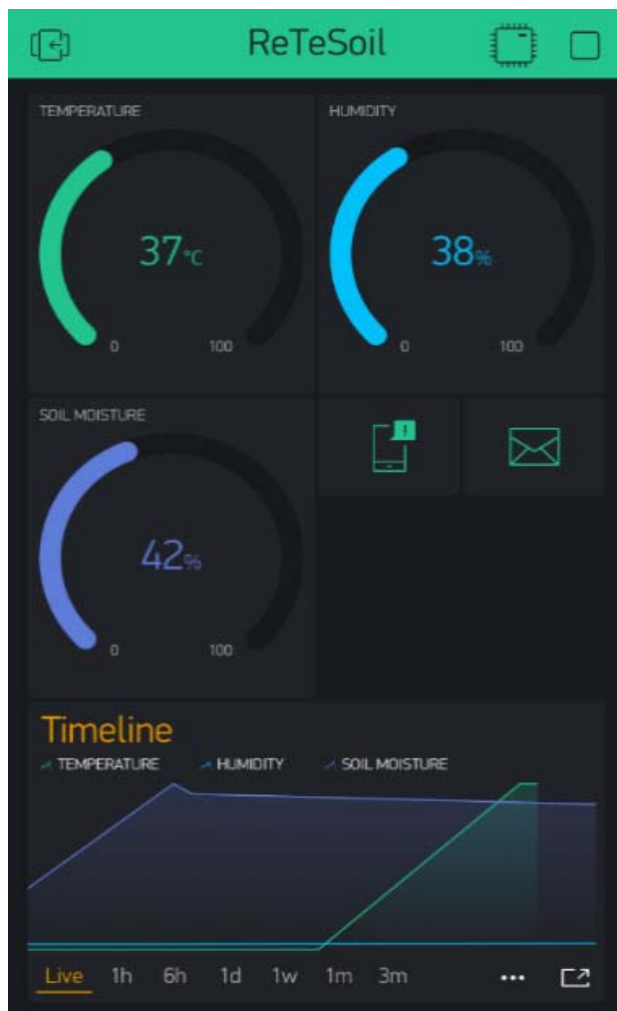


Fig. 11. Gauges reading.

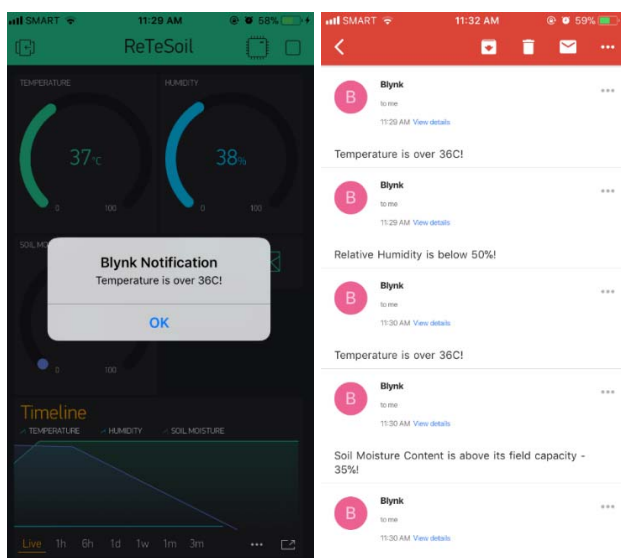


Fig. 12. Push and email notification.

TABLE 2
THRESHOLD SUMMARY

	Threshold Value
Temperature	36°C
Relative Humidity	50%
Soil Moisture	30% and 70%

The threshold value can be altered by changing the values in the source code. Once the values are set and the prototype is ready, it cannot be changed since the microcontroller was enclosed in a box.

V. CONCLUSION AND RECOMMENDATIONS

The researchers implemented a wireless monitoring system using BLYNK that can monitor temperature, relative humidity, and soil moisture content of the plant. With this, the sensors' data coming from the sensor node are displayed in the application with push notification and push email that allow users to remotely monitor the environment conditions in an area, if the threshold values are above or below the set limit. The researchers observed that the RF communication used in the base node connection has a maximum operating distance of 35 meters when both modules have antenna. Within this operating distance, data can be transmitted to the base node without any issues except when the distance is about its maximum limit (35 m). The initialization of GSM modem and connection to GPRS for base node and the BLYNK application communication was not stable. Delay was observed, it took a couple of seconds or minutes depending upon the signal strength of the network.

Future works may consider the improvement of the stability of data transmission from base node to BLYNK cloud by using Wi-Fi and/or other more stable communication. The use of a higher frequency operating RF modules can also be considered for wider range compare to 433-MHz RF modules that covers only 35 m, as well as, improving the hardware for both base node and sensor node. The use of 5-12 volts of power source with at least 1 Ampere rating to power up the Arduino board and GSM modem for the base node is highly recommended. This is also applicable for the sensor node since RF transmitting module consumes a lot of power when transmitting data remotely. Lastly, the use of a high-quality module and sensors can also be considered to avoid too much calibration.

REFERENCES

- [1] "Excessive and Irregular Irrigation Has the Following Effects," [Online]. Available: <http://www.biologydiscussion.com/articles/excessive-and-irregular-irrigation-has-the-following-effects/2403>. [Accessed 4 January 2018].
- [2] S. R. Kumbhar and A. P. Ghatule, "Microcontroller based Controlled Irrigation System for Plantation," p. 4, 2013.
- [3] H. Sahota, R. Kumar and A. Kamal, "A Wireless Sensor Network for Precision Agriculture and its Performance," p. 18, 2011.
- [4] G. R. Mendez, "A WiFi based Smart Wireless Sensor Network for an Agriculture Environment," Palmerston North, 2012.
- [5] S. Divyasri, and K. Babu, "Automated Irrigation System using a Wireless Sensor Network and GPRS Module," *International Journal of Innovative Technologies*, pp. 1307-1317, 2015.
- [6] N. Fhami, S. Huda, E. Prayitno, M. Rasyid, M. C. Roziqin and M. U. Pamenang, "A Prototype of Monitoring Precision Agriculture System Based on WSN," p. 6, 2017.
- [7] H. Mansour and Y. El-Melhem, "Impact the Automatic Control of Closed Circuits Drip Irrigation Systems on Yellow Corn Growth and Yield," p. 10, 2013.
- [8] M. S. Shah, U. A. Shaikh, and H. S. Doshi, "Internet of Things (IoT): Integration of Blynk for Domestic Usability," *Vishwakarma Journal of Engineering Research*, pp. 149-157, 2017.
- [9] Arduino, [Online]. Available: <http://arduino.cc>. [Accessed 6 May 2018].
- [10] A. Terzis, A. Szalay, and Q. Wang, "A Novel Soil Measuring Wireless Sensor Network," in *IEEE Instrumentation and Measurement Technology Conference Proceedings*, Austin, 2010.
- [11] S. M. A. Alim, M. S. Islam, K. B. R. Kawshik, F. Amid and S. Islam, "433 MHz (Wireless RF) Communication between Two Arduino UNO," *American Journal of Engineering Research*, pp. 358-362, 2016.
- [12] M. Maroti and A. Ledeczki, "Wireless Sensor Node Localization," 2011.
- [13] M. N. Jivani, "GSM Based Home Automation System Using App Inventor," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, pp. 12121-12128, 2014.
- [14] "Kickstarter," [Online]. Available: <http://create.arduino.cc/projecthub/vijayvikram11844/first-project-arduino-blynk-a5a297>.
- [15] Blynk, [Online]. Available: <http://docs.blynk.cc/>. [Accessed 6 May 2018].

A LabVIEW-Based Target Optimization Genetic Algorithm for Biological Predators

Ryann Alimuin,^{1,*} Elmer Dadios,² and Argel Bandala²

Abstract—Initial projection of a continuously repositioning target is a setback in genetic algorithm; a GA program needs constant input sampling to predict and declare a targets status. Another difficulty is the incorporation of GA to hardware and software which considered as the most important tool in sensor integration. Familiarity in programming is essential in utilizing the NI LabVIEW and NI myDAQ environment. The aim of this research paper is to provide a solution for determining the locus (gene position) of a target through distinctly employed multiple sensors which employs low-frequency (LF) ground-wave oscillations as its signal sources. The targets' position as well as the speed is continuously monitored through virtual instrument (VI) software; the user will be able to visually analyze the constant system mutation plots and the number of completed generations. Upon completion on the number of generations, the plot points can be imported to a spreadsheet for further analysis. The multiplatform software will be able to plot the response under real-time circumstances.

Keywords: LabVIEW, locus, mutation, low-frequency, multiplatform

I. INTRODUCTION

Today, multiple programming methods are applied to perform different tasks; one of them is the genetic algorithm (GA). GA in artificial intelligence is a type of evolutionary computer algorithm in which symbols (often called “genes” or “chromosomes”) representing diverse solutions undergo a process called crossover or sometimes addressed as breeding. This process involves recombination of possible genes and multiple mutations at a specific rate. GA mimics the way evolution acts and allows us to improve the performance of controllers or

adapt them to different systems [1]. GA is a probabilistic behavior, global searching, and optimization algorithm which is primarily intended to solve complex problems [2].

In comparison with common classical algorithms, GA has loads of advantages, such as a widely viable solution, searching in small and large groups, assistance without further information on the target, internal heuristic random search, parallel computing, etc. [3]. GA is a process for solving both forced and unrestrained optimization problems that is based on natural selection; the mutation process drives biological evolution [4]. It is mainly an overall arbitrary search and optimization method which aims to emulate the natural biological evolution [5].

In the livestock industry, one of the major causes of destructive loss is reptilian predators; these are hunters that feed on fowls and their by-product. This research aims to employ this method under a different target such as predators and take consideration of some parameters such as speed of movement and location status. Through LabVIEW, a programming platform which is widely utilized due to its simple graphical environment yet able to adopt to diverse interface on various hardware and software [6], the user is enabled to analyze the result through point-plotting approach. Unlike other object-oriented programming, LabVIEW allows wiring graphical objects in block diagrams which function as various utilities and modules [7]. The NI DAQ (Data Acquisition) hardware is an external module interface of LabVIEW which consists of several modules that can imitate analog-to-digital converters (ADC), signal conditioners, isolators, filters, and interfacing circuits [8].

II. PRIOR RELATED WORK

LabVIEW GA is a virtual instrument that integrates fitness evaluation, mutations, crossovers and selection [9] through prediction modelling, determination of different signals coming from multiple sources where attenuation and interferences can occur can be used [10].

Animal behavioral studies have been recently being studied especially on the use of sound pressure [11], [12] especially on common reptiles such as snakes through

¹Ryann Alimuin, Technological Institute of the Philippines, Quezon City, Philippines (e-mail: ryann.alimuin@tip.edu.ph)

²Elmer Dadios and Argel Bandala, De La Salle University–Manila, Philippines

vibration stimulation [13]. At 150–450 Hz in the range of 65–75 dBre at 20 mPa, Young and Aguiar observed a substantial decrease in the rattlesnake's *C. atrox* bodily movement and tongue flicking whilst showing a significant rise in numbers of head twitches and tail rattles [11]. Young and Morain added that olfactory-denervated and temporarily blinded Saharan sand vipers' (*Cerastes cerastes*) striking distance, angle, and accuracy were significantly reduced in capturing a free-running prey while employing only target vibrations [13]. Using GA, animal modeling rationalizes non-random mating and complex data organizations which utilizes parental phenotypes and offspring [14].

III. COMPONENTS AND SYSTEM ARCHITECTURE REQUIREMENTS

The system composition is a LabVIEW Virtual Instrument (VI) that employs a block diagram, front panel, and a DAQmx assistant which serves as a medium for acquiring data. The intent of the experiment is to record and plot multiple generations of data coming from the motion sensors. The sets of data that are sent to the computer are the position as well as the speed of the target. The program will be able to predict the succeeding point location where the target would move through a series of consecutive mutations and generations.

A. The following are the hardware components of the GA system:

1. Motion sensors/transducers
2. Signal transceivers/transducers
3. NI myDAQ
4. Personal computer/laptop
5. NI LabVIEW
6. Dual polarity power supply

B. The following are the computer hardware specifications necessary to operate the GA structure:

1. Pentium 4M processor or higher
2. 1 GB of RAM or higher
3. 32- or 64-bit Windows 7/8/8.1/10
4. Free 20-GB hard drive storage for system software requirements
5. Screen resolution of at least 1024×768 pixels

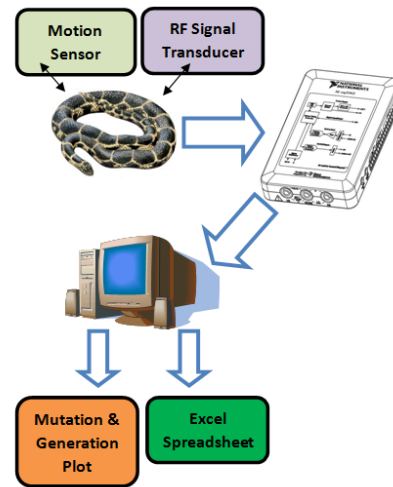


Fig. 1. System block diagram.

The system as shown in Figure 1 consists of two pre-positioned transducers, a motion sensor for detecting the rate of the specimen movement, and an RF signal transducer comprised of a ground-wave oscillator for quantifying the position of the target.

The transducers continuously monitor the movement of the target such as change in speed and position and send it to the data acquisition hardware. The GA virtual instrument takes the samples (parents) and generates consecutive mutations within the user assigned number of iterations. The VI will display the optimum value of mutation and plots the activity. The accumulated results can be analyzed through an integrated spreadsheet.

The LabVIEW VI is composed of the following:

1. Waveform chart for signal analysis
2. Generation progress monitor
3. Value indicator
4. Generation control
5. Mutation probability control
6. Interrupter button
7. Time estimator
8. Time generation monitor
9. "Save on spreadsheet" button

The hardware and software system requirements stated are based upon the minimum requirements needed to run LabVIEW 2012 to 2015 and NI myDAQ. Higher specification computers can increase the boot speed of the program as well as the necessary modules; this will also affect the effectiveness of system simulation.

IV. HARDWARE AND SOFTWARE INTEGRATION

The hardware and software integration is implemented through the following procedures:

- A motion sensor is installed in a defined location with minimal irregular geographic terrain.
- An RF signal transducer is utilized to monitor the location of the specimen.
- The transducers are connected to an NI MyDAQ signal acquisition hardware to integrate LabVIEW.
- The LabVIEW program acquires the samples and applies genetic algorithm to perform mutations.
- The user defines the number of generations that will be iterated by the program.
- LabVIEW GA plots each performed mutation and restarts after completing the defined generations.

The circuit in Figure 2 is the RF oscillator that serves as an input to the signal transmitter. It is comprised of multiple square wave generators and sinusoid converters. The waveforms produced are generally combined by a masking stage and are used to generate ground wave oscillations.

Using roulette selection [15], let the population size be p , the population members be c_i for $i \leq p$, and the fitness of chromosome be written $f(c)$ and the wheel size W by

$$W = \sum_i^p f(c_i)$$

Selecting a chromosome,

$$j < \sum_i^k f(c_i)$$

where each $f(c)$ slots corresponds to each chromosome. Assigning slot counts according to

$$f'(c) = F - f(c) + s$$

where

$$F = \max f(c_i)$$

The fitness function is given by with if and only if x is a solution.

The loose fitness function satisfies with if x is a solution where the sum of any member of fitness functions with non-negative weights is also a fitness function.

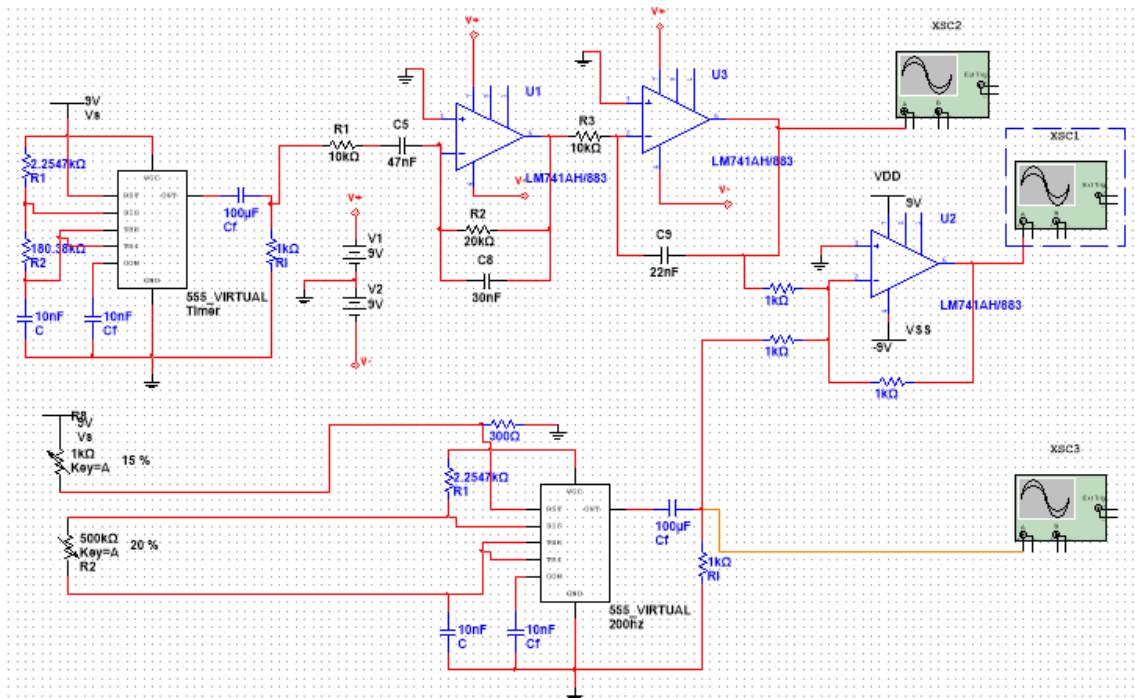


Fig. 2. RF oscillator schematic diagram.

V. SYSTEM FUNCTIONALITY DEMONSTRATION

The block diagram, as shown in Figure 3, illustrates the main VI program of the GA system. It is also composed of multiple sub-VIs which perform the mutation of each input sample. The DAQ Assistant serves as the interface between the external experiment proper and the software. The speed of iterations varies depending upon the capacity of the simulating CPU.

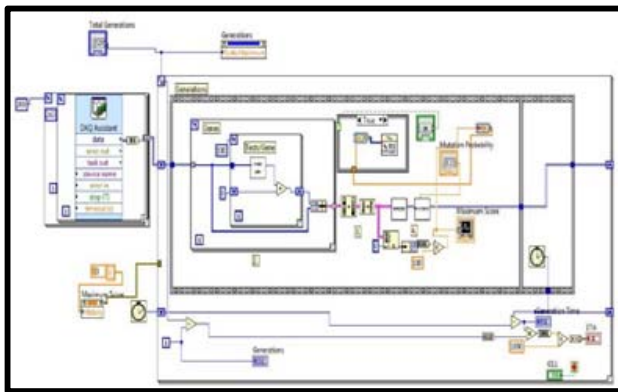


Fig. 3. GA system block diagram.

The virtual instrument is simulated and displays the plot on the number of mutations and completed generations. The optimum mutation probability is 0.006. The most effective number of generations should be at least 100 before it reaches the best fitness.

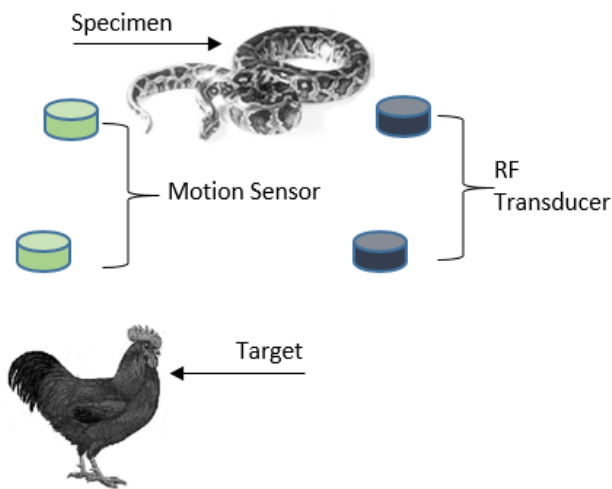


Fig. 4. Specimen and target setup.

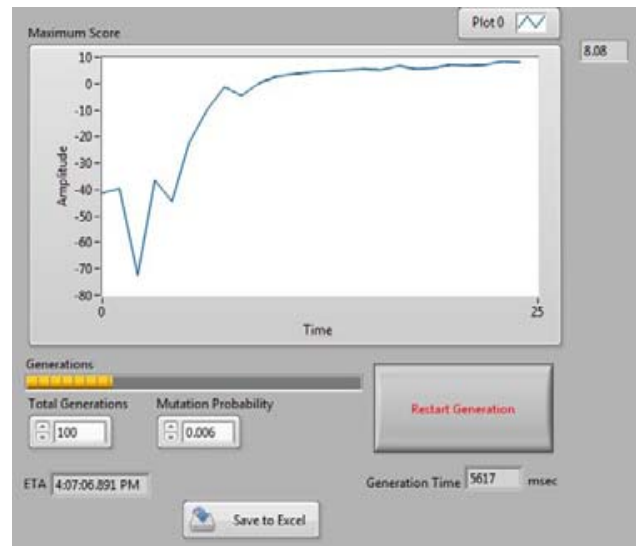


Fig. 5. First-generation plot.

Figure 5 shows the plot of the first generation. First mutations are low, yet the best fit value generated by the VI reaches an almost constant score.

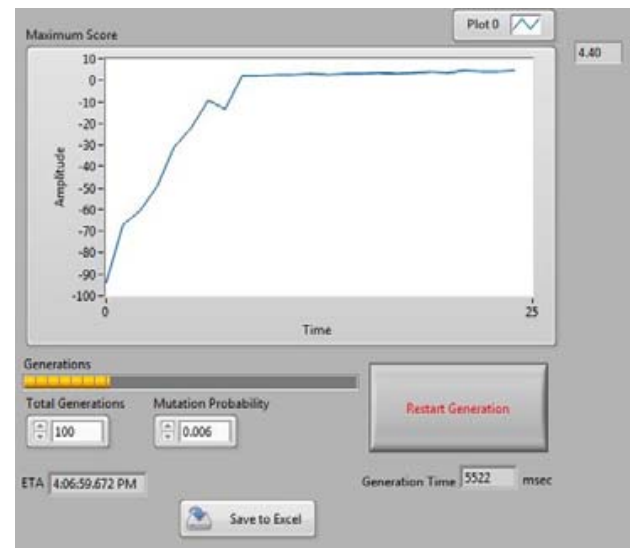


Fig. 6. Second-generation plot.

Figure 6 shows the plot of the second generation. First mutations are relatively lower than the first, yet the optimally fit value generated still reaches an almost constant score.

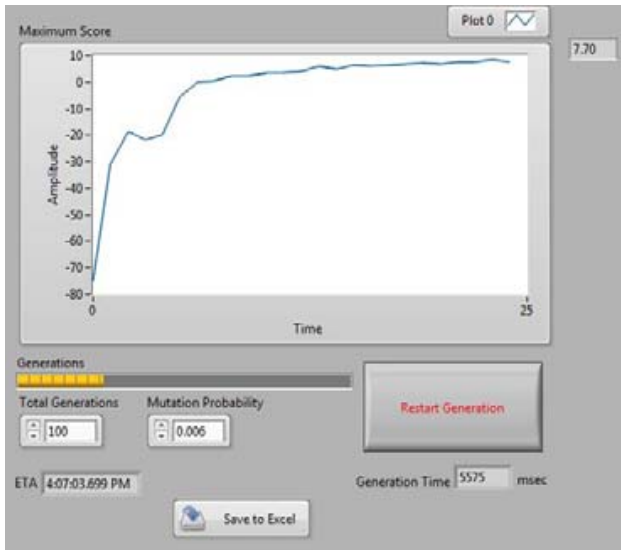


Fig. 7. Third-generation plot.

In Figure 7, the first mutations are low, but while finding the best fitness, the program still arrived at the optimum score which is nearly constant.

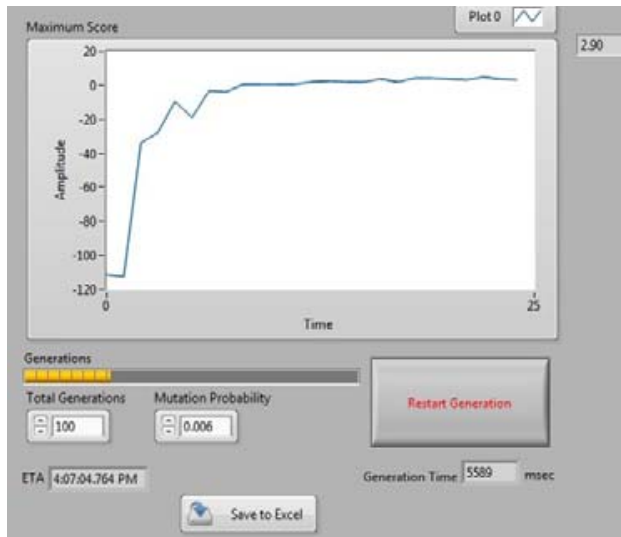


Fig. 8. Fourth-generation plot.

In the fourth generation as shown in Figure 8, the first samples started extremely low but still managed to recover and attained a maximum score throughout the generation.

Figure 9 illustrates the four generations of mutations done by the system. As shown, multiple iterations are repeated for mutation of the acquired samples. At the start of the experiment, mutations are low; however, the system continuous to attempt a prediction on how the specimen would succeed in reaching its target. The VI requires at least 11 ms before it settles for the best fitness outcome.

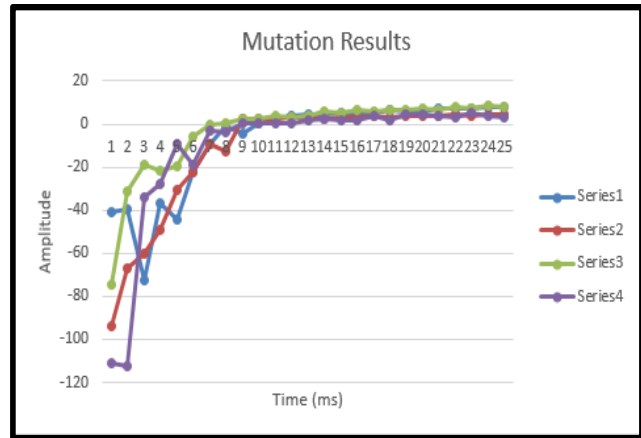


Fig. 9. Mutation results.

VI. CONCLUSIONS AND RECOMMENDATIONS

The point plotting GA system was able to execute locus plots through the reception of sensor data. Mutations of several samples on the target specimen’s velocity and position were obtained and monitored through NI LabVIEW as well as the importation on a spreadsheet summary for further analysis and assessment. The VI software can be modified to perform other tasks which allows gathering of other parameters of moving samples.

The system structure can be enhanced through several software and hardware modifications. These enhancements will improve genetic mutations on movement predictions, namely,

- Increase in CPU processing speed by utilizing greater hardware specifications,
- Replacement of a higher sensitivity reception transducers,
- Modifications on the LabVIEW program for generation mutations, and
- Placement of transmitter and receiver transducers on terrains with less inclinations as well as geographical obstructions.

ACKNOWLEDGEMENTS

The research proponents express their deep gratitude to the Technological Institute of the Philippines–Quezon City and De La Salle University for the collaborative research and experimentation. We would also acknowledge their permission on the use of relevant equipment and apparatuses for the design.

REFERENCES

- [1] P. Ponce, F. Ramirez and V. Medina, "A Novel Neuro-Fuzzy Controller Genetically Enhanced Using LabVIEW," *IEEE*, pp. 1559-1565, 2008.
- [2] X. Li, Y. Zhang, S. Wang and G. Zhai, "Analog Circuits Fault Diagnosis by GA-RBF Neural Network and Virtual Instruments," *IEEE*, pp. 236-239, 2012.
- [3] J. Li and M. Zhu, "An Implementation of Genetic Algorithm in Matlab," *IEEE*, pp. 638-641, 2010.
- [4] S. Jaganathan, A. Sekar and W. Gao, "Formulation of Loss minimization Problem Using Genetic Algorithm and Line-Flow-based Equations," *IEEE*, pp. 1-6, 2008.
- [5] C. Zhang and J. Chen, "Methods of Optimization of Milling Parameters Based on Genetic Algorithm," in *The Ninth International Conference on Electronic Measurement & Instruments*, Nanjing, China, 2009.
- [6] P. Mishra, V. Kumar, K. P. S. Rana, S. S. Nair and J. Kumar, "Cuckoo Search Implementation in LabVIEW," *IEEE*, pp. 1-6, 2016.
- [7] R. Turley and M. Wright, "Developing Engine Test Software in LabVIEW," *IEEE*, pp. 575-579, 1997.
- [8] M. Murugappan, R. Thirumani, M. I. Omar and S. Murugappan, "Development of Cost Effective ECG Data Acquisition System for Clinical Applications using LabVIEW," in *IEEE*, Kuala Lumpur, Malaysia, 2014.
- [9] V. Kumar, K. P. S. Rana, A. Kumar, R. Sharma, P. Mishra and S. Nair, "Development of a Genetic Algorithm Toolkit in LabVIEW," in *Third International Conference on Soft Computing for Problem Solving*, India, 2014.
- [10] S. K. Bhargav, B. V. Ramana, K. C. B. Rao and P. S. Kumar, "Prediction and Analysis of Electromagnetic Interference in Radio Frequency Propagation for Transmitter - Receiver pair using LabVIEW," in *International Conference on Electromagnetic Interference Compatibility and Workshop*, India, 2016.
- [11] R. Shine, "Reticulated pythons in Sumatra: biology, harvesting and sustainability," *Elsevier*, 1999.
- [12] B. A. Young and A. Aguiar, "Response of western diamondback rattlesnakes *Crotalus atrox* to airborne sounds," *Journal of Experimental Biology*, pp. 205, 3087-3092, 2002.
- [13] B. A. Young and Harris, "Auditory sensitivity of the northern pacific rattlesnake, *Crotalus viridis oreganus* : do behavioral responses conform to physiological performance?," *Northwest Science*, pp. 80, 218-223, 2006.
- [14] J. Odegard, T. H. Meuwissen, B. Heringstad and P. Madsen, "A simple algorithm to estimate genetic variance in an animal threshold model using Bayesian inference," *BioMed Central*, pp. 1-7, 2010.
- [15] R. Booth, D. Bormotov and A. Borovik, "Genetic Algorithm and Equations in Free Groups and Semi Groups," *American Mathematical Society*, pp. 63-80, 2003.

Hybrid Sensor Based Fuzzy Clustering Neural Network Classification for Human Activity Recognition

Ramtin Aminpour¹ and Elmer Dadios²

Abstract—The smartphone is going to become an all-purpose gadget for the human life and all of them at least armed with accelerometer sensor. In this study, the fuzzy c-means has been considered in the ANFIS model to produce the fuzzy inference system (FIS) to make the classification with the neural network algorithm to detect the six major human activities. The data were taken in real life with the accelerometer sensor of a smartphone. The results of the experiments show that the 97.2% accuracy could be acceptable in the field of study and the clustering structure could make the simulation more robust and faster.

Keywords: Fuzzy clustering, Neural network, Human activity recognition

I. INTRODUCTION

Human activity recognition is widely used in human-computer interaction, fitness tracking, and maintenance of elderly people [1]–[4]. The data from activities could be gathered by camera sensors, audio sensors, and embedded sensors [5]. Nowadays, smartphones are personal importance in the societies. All of the smartphones have internal sensors to collect data with low power consumption and powerful processors. One of the most useful embedded sensors of the smartphones is the accelerometer which could be used to collect extensive data on human activities [6].

In this study, a Neuro-Fuzzy inference system combined with the fully-connected layer neural network has been used to classify the human activities. Integration of neural networks and fuzzy logic systems could be a

hybrid approach to the model of a system. The study of the architecture is shown in Figure 1.

II. RELATED WORK

Applications of computational intelligence have been used in many fields based on neural network, fuzzy logic, genetic algorithm, and hybrid approaches of these algorithms. NeuroFuzzy has been proposed by Jang [7] with the concept of integration with human and learning capability of the structure of the neural network. Neuro-Fuzzy systems could be powerful solutions in many applications [8], [9]. Neural network and fuzzy logic are dynamic with the ability of parallel processing to approximate the input and output functions.

One of the problems of fuzzy design is the difficulty to determine the number of rules and also the number of membership functions of every rule. Hybrid algorithms can optimize the systems to the trade of this problem [10], [11]. Hybrid algorithms have been used for fine-tuning and learning of parameters of the neural network and fuzzy logic. Lin [12] proposed a Takagi Sugeno (TS) type of fuzzy model with a hybrid learning algorithmic rule. The approach was used to modify the mean and the deviation of the membership functions. A combined Takagi Sugeno type Neuro-Fuzzy system has been done with the bee colony algorithm for parameter optimization [13].

A comparative study for classification of ECG signals with MultiLayer backPropagation learning (MLP) has been done by Ozbay [14]. They introduced a Fuzzy Clustering Method (FCM) based neural network, which shows faster and better accuracy to compare of ordinary MLP architecture.

Kim et al. [15] have developed the FCM algorithm for the color clustering problems. Mingoti [16] performed a clustering algorithm based on Self-Organizing Map (SOM) neural network and FCM and they found that the performance of the algorithm was improved in the presence of outliers. A hybrid study of Support Vector Machine (SVM) and FCM for gene dataset has been done

¹ Ramtin Aminpour, Department of Electronics and Communications Engineering, De La Salle University, Manila, Philippines (email: ramtin.aminpour@dlsu.edu.ph)

² Elmer Dadios, Department of Manufacturing Engineering and Management, De La Salle University, Manila, Philippines (email: elmer.dadios@dlsu.edu.ph)

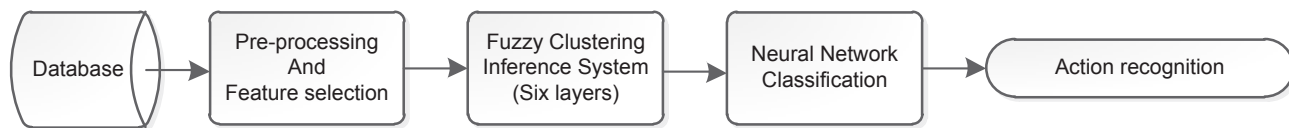


Fig. 1. Architecture of the FCM neural network model

by Mukhopadhyay [17]. Aydilek [18] proposed a hybrid approach of Genetic Algorithm (GA) with FCM and SVM to estimate missing values and optimization of the size of a cluster.

The hybrid algorithm could be an optimal way of a solution for system design. The learning algorithm of Neuro-Fuzzy is based on a steep descent optimization like the backpropagation algorithm. Similar to other gradient methods, steep descent optimization cannot prove to converge to the global solution. On the other hand, the parameters of the membership functions cannot be tuned in and modified. In these situations, optimization with hybrid algorithms could give an efficient result.

III. DATASET DESCRIPTION AND FEATURE EXTRACTION

An Android smartphone has been used to collect the data in this study. The accelerometer has been collected the data with a sample frequency rate of 50 Hz from seven adult persons to detect six activities of walking, jogging, running, jumping, using stairs, and standing. The smartphone was placed in the front pocket of the volunteers. Each activity was repeated five times during a period of 30 seconds. After collecting the data, pre-processing of the data by a low-pass filter has been done to clean the raw data from missing data, noise reduction, and outlier detection.

To detect the activities, proper feature set must be extracted from the raw data. A sliding window with the length of 2.5 seconds is used to separate the data. The acceleration sensor of the smartphone has three dimensions in which four features from time domain and two features of frequency domain have been extracted. In total 18 features have been selected to extract from X, Y, and Z axes of the accelerometer. These features are shown in Table I. In the Table x_i are the features, N is the total number of features, and FFT is the Fast Fourier Transform of the features.

The data might have a different scale because of the gait, the height and the weight of volunteers. For this reason, the

TABLE 1
FEATURES EXTRACTION FROM ACCELEROMETER OF SMARTPHONE

Features	Math explanation
Mean Absolute Value	$\frac{1}{N} \sum_{i=1}^N x_i$ (1)
Variance	$\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$ (2)
Root Mean Square	$\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i)^2}$ (3)
Skewness	$\frac{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^3}{\sigma^{3/2}}$ (4)
Dominant frequency	$Max(FFT)^2$ (5)
Energy	$\sum (FFT)^2$ (6)

data must be normalized to zero means before using into the classification algorithm.

$$f_{normalized(i)} = \frac{f_{raw(i)} - \mu}{\sigma} \quad (7)$$

Here μ is mean and σ is the standard deviation of the features data. All the raw data replaced with normalized data to build the features matrix.

IV. CLASSIFICATION MODEL ARCHITECTURE

The structure of the adaptive neuro-fuzzy framework model (ANFIS) consists of input variables and output variables with a Takagi-Sugeno type of rule-based. The type of this network is an adaptive, multi-layer, and feed-forward [7]. The rule sets are evaluated by linguistic variables, a linear combination between input values, and a constant parameter which is employed to obtain the result of the rules.

In the first layer of this network, input variables are mapped into fuzzy sets through the process of fuzzification and generate membership grades with a membership function such as triangular or sigmoid. The next layer is working as a fuzzy conjunction to combine the fuzzy sets on each input. Another layer calculates the ratio of the rule that fire to the sum of all firing strength and then multiplied by the function of TKS fuzzy rules and extract the sum of all outputs of each rule. Finally, the crisp fuzzy output is calculated during the process of defuzzification by the weighted average method.

The ANFIS has two types of parameters, linear parameters on the consequent part and nonlinear parameters in the commence part. To optimize these variables, ANFIS has several strategies such as gradient descent, steepest descent, and hybrid technique [7]. In the consequent part, the output of each layer is transferred toward the last layer and the parameters are adjusted by the least squares method. In the premise part, the error signals are transferred to the first layer and update the parameters by the gradient descent method. The processes are done with the limited number of iterations during the epochs. The error between the output and also the target goes to minimize by weight adjustment of the connections throughout the learning method [19].

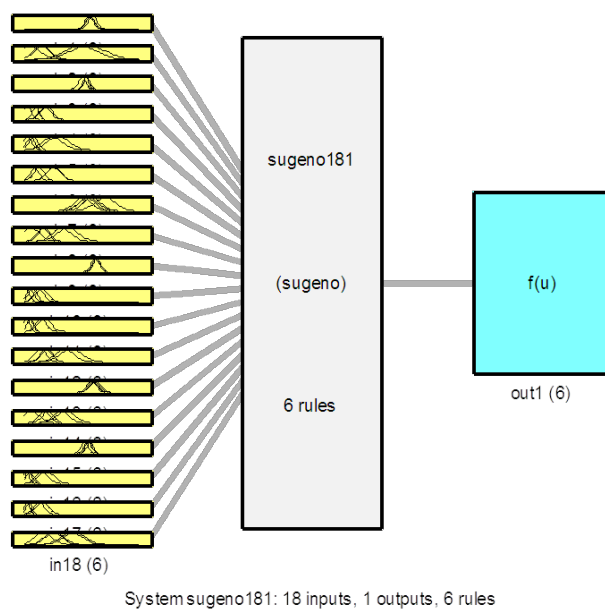


Fig. 2. Sugeno neuro fuzzy clustering model

The Sugeno Neuro-fuzzy clustering for this study is shown in Figure 2.

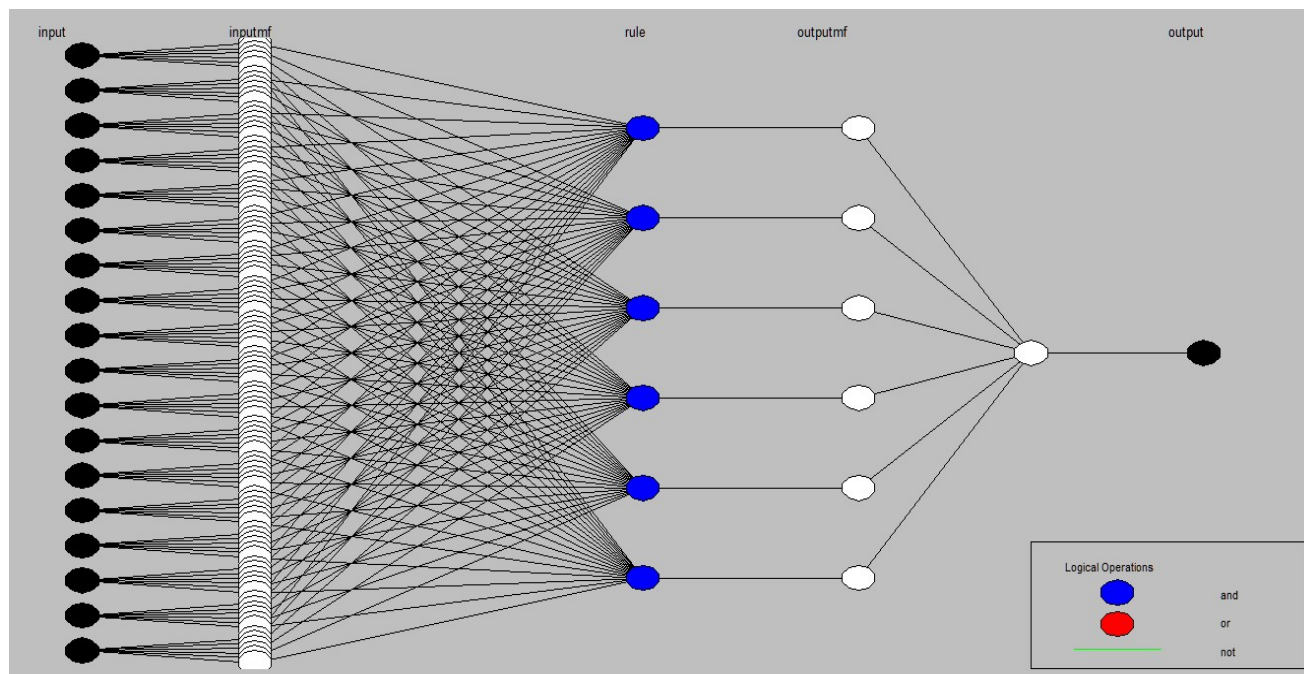


Fig. 3. ANFIS structure model

A. ANFIS Model Building

In this study, the data have been separated into three parts. Seventy percent have been used during the training of the algorithm, 15% as a checking data to prevent of the over-fitting of the model, and remaining 15% for testing of the model to check the predicted ability of the algorithm. During the training phase of FCM, the parameters are determined automatically in the specific epochs to minimize the checking error. Samples of the dataset for walking and running activities are given in Table II.

There are three methods to generate the fuzzy inference system (FIS) structure in ANFIS, grid partition, subtractive clustering, and fuzzy c-means clustering [20]. The fuzzy clustering method and the neural network which have been used as a classification are explained below. Because of six activities, the simulation has six different outputs. The ANFIS structure of one of the outputs of this study is shown in Figure 3.

B. Fuzzy Clustering Method

Fuzzy inference system (FIS) could be generated by fuzzy c-means (FCM) clustering. The FCM structure

can generate the rules from the behavior of the data and determine the number of rules and membership functions of the input and output variables of the algorithm. The number of clusters can be set in the FCM algorithm. The bigger cluster radius generates fewer clusters and fewer rules during the process of generating FIS. In the Takagi Sugeno structure, the membership functions for input and output are set with Gaussian and linear, respectively.

The steps of a fuzzy clustering algorithm are as follows [21], [22]:

- 1) Initialize the number of clusters.
- 2) The fuzzification step which is selecting a metric Euclidean norm and the weighting metric.
- 3) Initialize the cluster prototype and iterative counter.
- 4) Calculate the partition matrix.
- 5) Update the fuzzy cluster centers.
- 6) If the norm of cluster centers was smaller than epsilon then stop the algorithm otherwise, repeat step two up to four.

TABLE 2
SAMPLES OF DATASET FOR TWO ACTIVITIES

<i>Walking_x</i>	<i>Walking_y</i>	<i>Walking_z</i>	<i>Running_x</i>	<i>Running_y</i>	<i>Running_z</i>
-0.03	-1.2	-0.66	0.6	0.23	0.24
-0.03	-1.2	-0.66	0.1	0.37	0.11
0.05	-0.79	0.05	-0.33	-0.2	-0.18
0.41	-1.06	0.16	-0.33	-0.2	-0.18
0.41	-1.06	0.16	1.36	-1.32	1.26
-1.27	-1.5	-0.15	-1.25	-1.99	0.22
-0.45	-0.18	-0.07	-0.13	-0.01	0.04
-0.45	-0.18	-0.07	-0.13	-0.01	0.04
-0.5	-0.64	0.44	0.81	-1.99	-0.55
0.06	-1.05	-0.59	0.81	-1.99	-0.55

TABLE 3
THE CONFUSION MATRIX OF ACTIVITIES FOR CLASSIFIER

Activity	Walking	Jogging	Running	Hopping	Using stairs	Idle
Walking	2991	64	5	0	33	5
Jogging	16	1784	32	2	0	2
Running	11	19	1531	9	5	0
Hopping	24	8	2	14	1366	8
Using stairs	24	8	2	14	1366	8
Idle	3	0	0	0	8	1679

The feature data have been grouped into six clusters to distinguish the six activities of the target. Six features are extracted from each axis of the accelerometer. In total, 18 features are used to detect the activity of each class. As a result, in this study, 108 fuzzy membership functions are generated to distinguish the output activities. Some samples of fuzzy membership functions for the first output of the target are shown in Figure 4, Figure 5, and Figure 6. The Gaussian membership functions have been chosen automatically by fuzzy inference system (FIS) in contrast with the fuzzy logic algorithm.

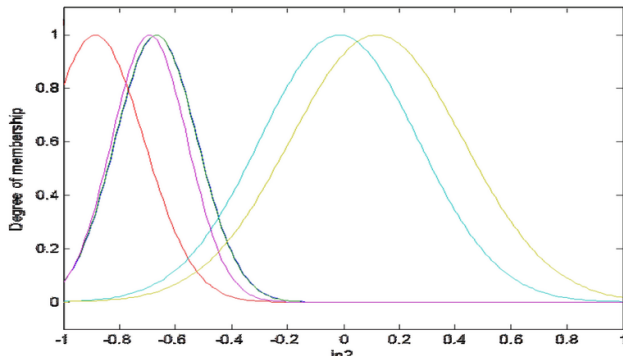


Fig. 4. Fuzzy membership of Input 2 for the first class

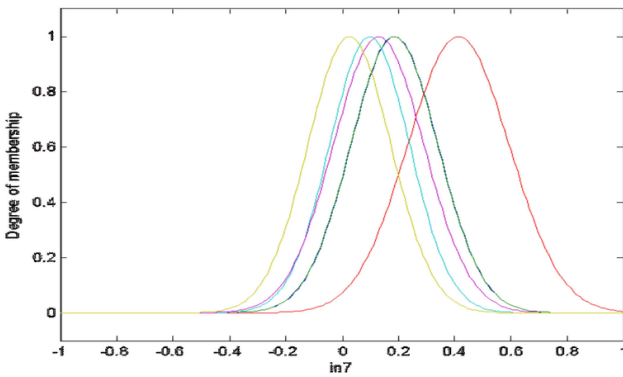


Fig. 5. Fuzzy membership of Input 7 for the first class

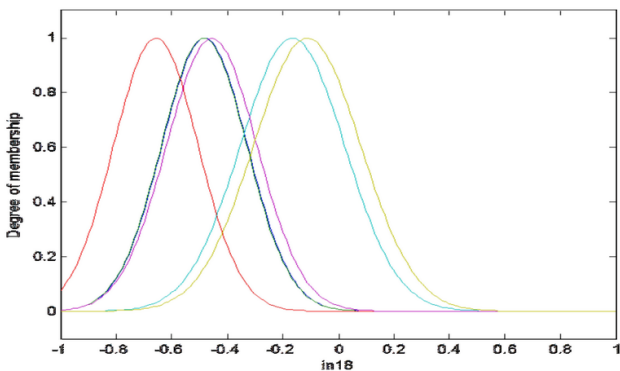


Fig. 6. Fuzzy membership of Input 18 for the first class

V. EXPERIMENT AND ANALYSIS OF RESULTS

To detect the activities the features data have been clustered to the number of activities by the fuzzy clustering algorithm. In the next step, the output of FCM has passed through a full node connection neural network to classify the human activities. The pattern recognition network has 18 neurons in the hidden layer, the training function of the network has been selected by Levenberg-Marquardt, and the performance function was chosen by mean square error. The features data of FCM have been divided randomly in the training ratio of 70%, the validation ratio of 15%, and the remained 15% for the test of the algorithm.

The result of confusion matrix and ROC curve are shown in Table III and Figure 7 respectively. The ROC curves which are closer to the left part of the plot have better classification accuracy. The confusion table shows that the most confused about activities are between jogging and walking, and using stairs and walking. The simulation shows that the accuracy of this classification is 97.2%. The output performance of the algorithm is shown in Figure 8. In the performance of the simulation, after 30 epochs the training stopped due to an increase of validation error against the minimum mean square error (MSE) parameter. The confusion table of the classifier shows that maximum errors are between jogging with walking, running with jogging, and using stairs with walking activities.

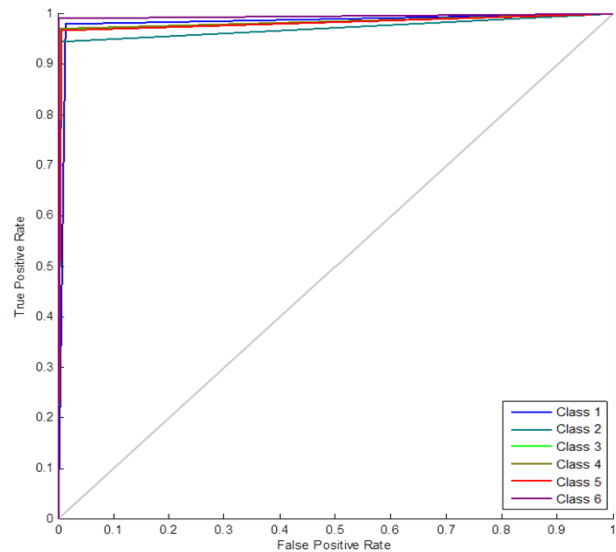


Fig. 7. ROC curve of classification

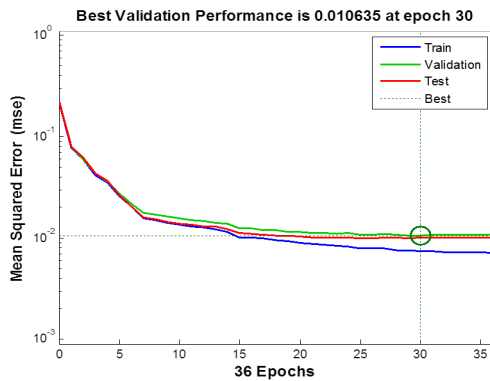


Fig. 8. Performance of simulation model

VI. CONCLUSION AND FUTURE WORK

In this study, the hybrid fuzzy clustering with the neural network algorithm is used to classify and recognize the human activities. The fuzzy c-means is used to generate the fuzzy inference system to develop the ANFIS based method.

Using the hybrid fuzzy cluster algorithm proves that the developed ANFIS model is more robust and faster than classic algorithms with an acceptable accuracy to compare of related works in the relevant area.

In the future works, the hybrid Fuzzy Clustering Neural Network algorithm could be compared to another type of hybrid fuzzy cluster algorithms such as Ant colony optimization, Differential evolution, or Particle swarm optimization.

ACKNOWLEDGEMENT

The author would like to thank the Intelligent Systems Laboratory of De La Salle University for providing the facilities in pursuing this research study.

REFERENCES

- [1] J. W. Lockhart, T. Pulickal, and G. M. Weiss, "Applications of mobile activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 1054–1058.
- [2] C. P. Dadula and E. P. Dadios, "Fuzzy logic system for abnormal audio event detection using mel frequency cepstral coefficients," *Journal of advanced computational intelligence and intelligent informatics*, vol. 21, no. 2, pp. 205–210, 2017.
- [3] C. Dadula and E. Dadios, "Event detection using adaptive neuro fuzzy inference system for a public transport vehicle," in *11th International Conference of the Eastern Asia Society for Transportation Studies*, 2016.
- [4] E. P. Dadios, J. J. C. Biliran, R.-R. G. Garcia, D. Johnson, and A. R. B. Valencia, "Humanoid robot: Design and fuzzy logic control technique for its intelligent behaviors," in *Fuzzy Logic-Controls, Concepts, Theories and Applications*. InTech, 2012.
- [5] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
- [6] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Pattern Recognition and Image Analysis*. Springer, 2011, pp. 289–296.
- [7] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [8] G. Castellano, C. Castiello, A. M. Fanelli, and L. Jain, "Evolutionary neuro-fuzzy systems and applications," in *Advances in Evolutionary Computing for System Design*. Springer, 2007, pp. 11–45.
- [9] S. Kar, S. Das, and P. K. Ghosh, "Applications of neuro fuzzy systems: A brief review and future outline," *Applied Soft Computing*, vol. 15, pp. 243–259, 2014.
- [10] C.-H. Chen and Y.-Y. Liao, "An efficient cluster-based tribes optimization algorithm for functional-link-based neurofuzzy inference systems," *Applied Soft Computing*, vol. 13, no. 5, pp. 2261–2271, 2013.
- [11] W. S. Liew, M. Seera, C. K. Loo, and E. Lim, "Affect classification using genetic-optimized ensembles of fuzzy ARTMAPs," *Applied Soft Computing*, vol. 27, pp. 53–63, 2015.
- [12] C.-J. Lin and Y.-J. Xu, "Design of neuro-fuzzy systems using a hybrid evolutionary learning algorithm," *Journal of information science and engineering*, vol. 23, no. 2, pp. 463–477, 2007.
- [13] Y.-T. Liu, Y.-Y. Lin, T.-Y. Hsieh, S.-L. Wu, and C.-T. Lin, "A global optimized neuro-fuzzy system using artificial bee colony evolutionary algorithm," in *Intelligent Systems and Applications: Proceedings of the International Computer Symposium (ICS) Held at Taichung, Taiwan, December 1214*, 2014, vol. 274. IOS Press, 2015, p. 140.
- [14] Y. O' zbay, R. Ceylan, and B. Karlik, "A fuzzy clustering neural network architecture for classification of ECG arrhythmias," *Computers in Biology and Medicine*, vol. 36, no. 4, pp. 376–388, 2006.
- [15] D.-W. Kim, K. H. Lee, and D. Lee, "A novel initialization scheme for the fuzzy c-means algorithm for color clustering," *Pattern Recognition Letters*, vol. 25, no. 2, pp. 227–237, 2004.
- [16] S. A. Mingoti and J. O. Lima, "Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms," *European journal of operational research*, vol. 174, no. 3, pp. 1742–1759, 2006.
- [17] A. Mukhopadhyay and U. Maulik, "Towards improving fuzzy clustering using support vector machine: Application to gene expression data," *Pattern Recognition*, vol. 42, no. 11, pp. 2744–2763, 2009.
- [18] I. B. Aydilek and A. Arslan, "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm," *Information Sciences*, vol. 233, pp. 25–35, 2013.
- [19] J.-S. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378–406, 1995.
- [20] "MATLAB and fuzzy logic Toolbox Release 2015b, The MathWorks, Inc., Natick, Massachusetts, United States."
- [21] T. A. Runkler and C. Katz, "Fuzzy clustering by particle swarm optimization," in *Fuzzy Systems, 2006 IEEE International Conference on IEEE*, 2006, pp. 601–608.
- [22] M. Huang, Z. Xia, H. Wang, Q. Zeng, and Q. Wang, "The range of the value for the fuzzifier of the fuzzy c-means algorithm," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2280–2284, 2012.

Implementation of k-Nearest Neighbor for PC-Based Character Recognition of Philippine Vehicle Standard License Plate

Klariz Donna Mae B. Bundal, Arlyn P. Lauron, Jan Jeffrey R. Camiña and Cristina P. Dadula

Abstract—This study focused on the development of a PC-based licensed plate recognition system using Visual Basic programming language. A system that is able to recognize Philippines' currently used standard vehicle plate numbers using EmguCV image processing and K-Nearest Neighbor machine learning algorithm. The system accepts image as an input or a snapshot of the image from video of a moving vehicle. There were twenty-two (22) unique images of a vehicle in acquired in 3 different positions: upright position (UP), skewed to right position (SR), and skewed to left position (SL). Image processing techniques were applied to the images such as grayscale conversion, Gaussian blurring, and thresholding. Another processing is the detection of plate number area. Optical character recognition is applied to this area where the characters in the image were segmented and individually recognized. The output equivalent characters and the cropped region of the plate number area are displayed on the user interface. The results showed that in: UP, the system recognition accuracy is 83.12%; in SR, it is 39.97%, and 46.21% in SL. The best system accuracy rate was obtained when the captured image of the vehicle is in the upright position which is 83.12%. For better performance, future works may consider the use of exact font style, different angle and position of the license plates, and different lighting conditions of sample license plates for training

Keywords: *EmguCV, PC-Based, KNN Machine Learning Algorithm, Blob Analysis*

I. INTRODUCTION

Originally, in modern technology, licensed plate recognition (LPR) systems are developed for security

Klariz Donna Mae B. Bundal, Arlyn P. Lauron, Jan Jeffrey R. Camiña and Cristina P. Dadula, Mindanao State University, General Santos City, Philippines (e-mail: cris_dadula@yahoo.com)

purposes and automation. LPRs are integrated into intelligent transportation systems and become possible to automatically monitor motorway collection, analyze traffic and intersection roads, improve law enforcements, and many more [1]. It is believed that currently, more than half a billion cars have their vehicle identification number (VIN) also known as license plate number as their primary identifier. The vehicle's identification number is the identifying code for a specific automobile. VIN serves as the car's primary "fingerprint" as no two or more vehicle has the same identification number. A VIN can be used to track recalls, registrations, warranty claims, thefts and insurance coverage [1].

Our country is confronted with problems regarding traffic violations in road intersection. Common road accidents in everywhere are attributed to the collision of vehicles, pedestrian, or a collision with an object that would result to death, disability and damage to property. These road accidents were caused by driver's errors (26%), over speeding (18%), mechanical defect (12%), drinking spree before driving (1%), and damaged roads (5%) [2]. The common causes of crashes in the intersections of streets according to studies are drivers' negligence and recklessness where drivers are running through red lights, ignoring the yield and stop signs [3]. Beating red light (traffic signal light which means to STOP) is one of the minor problems we are facing today and most of the time, we tend to ignore it, not knowing that it might be a cause of an enormous accident that can probably result to a sudden death of victims. According to Department of Transportation and Communications – Land Transportations Office (DOTC-LTO) of the Republic of the Philippines, it is a crime against the law to disregard traffic signs, and violators shall pay a fine of P1, 000.00 [4]. With a numerous amount of accidents and traffic rule violations, identification of vehicles has become a task of prime importance. With various optical character recognition (OCR) techniques, any valuable information such as vehicle license plate number is obtainable from captured images [5].

A means of technology such as, CCTV camera, road intersections incidents can now be monitored with the help of a video analyst. They can interpret and review CCTV footages [6]. They can help track those who beat traffic lights that might cause serious accidents and report it immediately to authority in order for them to stop tolerating these behaviors and receive proper sanction. Thus, developing a PC-Based automatic license plate recognition is the goal of this study which attempts to automate the identification of vehicles. The system could probably help lessen the hours spent for investigations as well as strengthen the enforcement of law on traffic lights to avoid road accidents. Also, this study makes use of the *EmguCV* Library references for Microsoft Visual Studio Community 2017 - Windows Forms Application (.NET Framework), Visual Basic as its language, and K-Nearest Neighbor as character recognition training machine.

II. PLATE NUMBER DETECTION

Vehicle detection is necessary to perform license plate recognition systems from a video input. An easy way to do vehicle detection is by using a background subtraction algorithm and the tracking of the vehicle can be achieved by using blob tracker algorithm (cvBlob or OpenCVBlobsLib) or blob detection [7]. The cvBlob library provides some methods to get the centroid, the track and the ID of the moving objects. A bounding box can also be set to draw on the frame surface, and also the centroid and angle of the tracked object. [8]. After vehicle tracking, the centroid of the moving object will be checked if it has crossed the region of interest or virtual line in the video. So when a vehicle passes through the virtual line, a frame capture will be executed in the system.

Images taken enter image processing through a series of algorithms which provide an alpha numeric conversion of the characters on the input image into text form [9]. The system localizes the plate number region and looks for the characteristics that would indicate that the chosen object is a license plate. Hence, the searching would continue to process until such time that a series of characters would be recognized by the system by using OCR [10], [11].

III. METHODOLOGY

The system developed accepts both image and video as input and gives an output of the vehicle's licensed plate number in both image and text format.

A. Equipment Used

This study makes the use of personal computer for

software application development, steel tape for measuring the distance between a capturing device and a vehicle plate number, and a digital camera as a capturing device.

Preparation of PC Software Application

Free edition of Microsoft Visual Studio 2017 community from <https://www.visualstudio.com/vs/> was downloaded and installed with default options of the IDE. Also downloaded and installed for free were the latest version of *EmguCV* executable installer without *cuda* support, and *libemgucv-windows-universal-3.0.0.2157.exe*. Visual Basic is the programming language and *EmguCV* is a suitable library that wraps *OpenCV* in .NET form [12] [13].

GUIs Form Layout

Visual Basic – Windows Forms Application was selected for starting up a *New Project*. A total of four (4) forms were created in Visual Basic.

- License Plate Recognition – Character Training (KNN Algorithm)
- License Plate Recognition – Testing the Trained System
- License Plate Recognition – Image of Vehicle with License Plate (Input)
- License Plate Recognition – Video Clip of Moving Vehicle (Input)

B. Image and Video Acquisition

Images of vehicle with standard license plate were captured using digital camera from selected parking lots around General Santos City. A total of twenty-two (22) unique vehicles were captured in three (3) different positions as shown in Figure 1: Group A, Group B, and Group C. These were captured in upright position, skewed to right position, and skewed to left position, respectively. The images were properly cropped and scaled.



Fig. 1. Different Capturing Positions, a) Group A; b) Group B; c) Group C

The video inputs used in this study were recorded from two (2) selected places in General Santos City, one (1) video per location was considered or testing. Only one vehicle per video was captured and tested for the system. A snapshot

from each video is shown in Figure 2, wherein *Input 1* was obtained from a university main entrance, and *Input 2* was from a mall parking lot entrance.

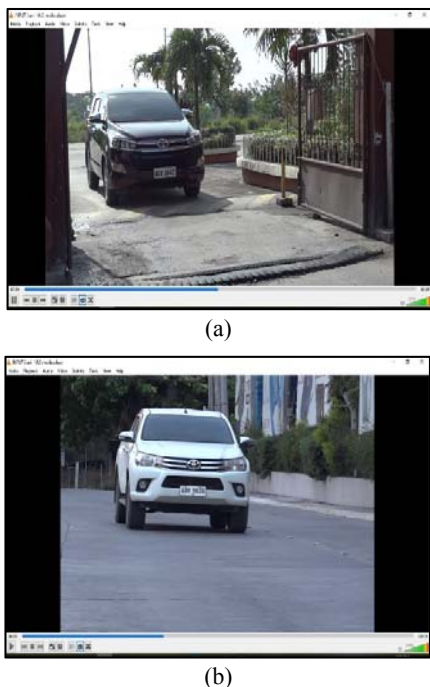


Fig. 2. Snapshot from each Video, a) Input 1 and b) Input 2

The recognition of characters of the vehicle’s license plates were separated for the image input and the video input. The outputs were properly tabulated for better viewing of results and comparisons.

C. Preparation of Machine Learning Method

KNN machine learning algorithm was settled on $k = 1$ or finding the single nearest neighbor producing optimal result because of the relatively small size of training data set. KNN algorithm is a simple algorithm which stores all available cases and classifies new cases [14].

The training was done by encoding a set of English alphabet and numbers with font similar to the font used in license plate, and saving this file in image or jpg format, a sample is shown in Figure 3. The characters on the number plate must have uniform fonts so that the OCR for number plate recognition would be less complex as compared to other methods [15]. OCR processes recognizes both handwritten and encoded/printed characters depending upon the quality of the subject (characters: letters and numbers) to be recognized [16]. The training set were encoded in six (6) batches including: four batches of *License Plate Font* from <http://www.fontspace.com/dave-hansen/license-plate>; one batch for *ARCADE R Font* from <https://fonts2u.com/arcade-r.font> and; one batch also for *GL NummernsChild*

Eng Font from <http://www.fontspace.com/gutenberg-labo/gl-nummernschild>. The training of character recognition was done and saved as a first Visual Basic – Windows Forms Application GUI project for Visual Studio entitled: *License Plate Recognition – Character Training (KNN Algorithm*, as stated in section B, GUIs Form Layout.



Fig. 3. Characters in Image Format

For testing and verification if the trained system can characterize letters and number, a set of sample text images shown Figure 4 was used as input to the second *Windows Forms Application project, License Plate Recognition – Testing the Trained System*. Since the characters were encoded, OCR was easily performed



Fig. 4. Set of Images of Characters

D. Optical Character Recognition (OCR) and Image Processing

The character recognition process is shown in Figure 5. Given the captured image of the vehicle, the system automatically searches for possible plate number region. This region undergoes the optical character recognition process shown in Figure 6. If the system recognizes characters, then system decides that the plate number area was found and it displays the recognized characters as the plate number of the vehicle in text format as an output. Otherwise, it keeps on searching for possible plate region until certain characters were recognized.

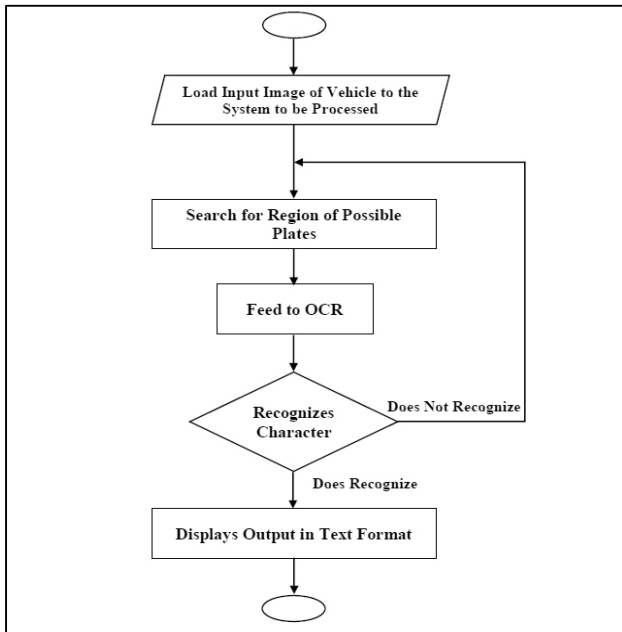


Fig. 5. Character Recognition System Flowchart

The process of OCR involves the application of image processing techniques. Image processing is a method to perform some operations in an image, in order to get an enhanced image or extract some useful information from it [17]. In image processing, the images are converted as desired, for instance, to have a simpler processing of the image, an image is converted from red-green-blue (RGB) layers to gray scale layer [10]. In Figure 6, the image acquired need to undergo processing such as noise removal, image thresholding, edge detection segmentation, feature extraction and classification. The OCR cannot perform segmentation and classification without KNN training. An overview of KNN is shown in Figure 7. The output of the system are the characters composed of letters and numbers in text format.

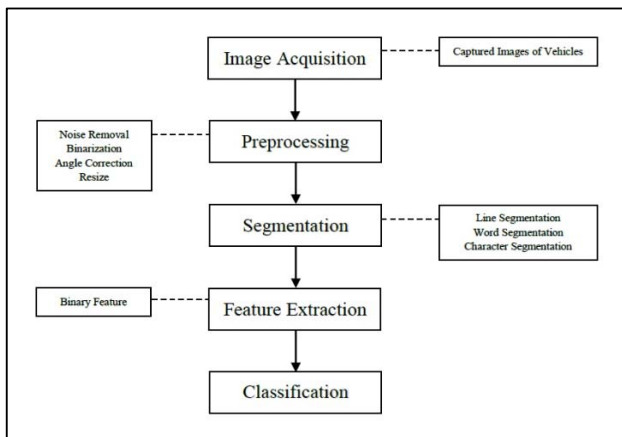


Fig. 6. OCR Processing

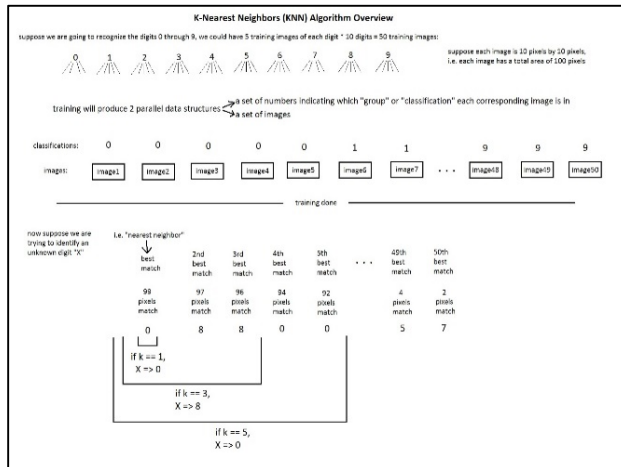


Fig. 7. KNN Overview [18] [19]

There were 5 class definition in Visual Basic for the image processing as described in Figure 8: *Preprocessing*, *PossiblePlates*, *DetectPlates*, *PossibleCharacters*, and *DetectCharacters*. The concept of pre-processing method involves converting the original image to grayscale and finding its edges for easy detection of plate region. Finding possible license plate involves localizing possible plate numbers. Several image contouring was done until a possible plate number was located, this helps the system to easily detect and localize the plate number. In license plate detection, the license plate detected was extracted and cropped to be processed again for classifying or defining characters. The cropped region was processed again with techniques like gray scaling, thresholding, and several contouring until certain characters were detected. The segmented characters were localized and loaded to KNN trained system for recognition. The characters defined were converted to a text format and was displayed in the textbox of the form as license plate characters.

Preprocess	Possible Plates	Detect Plates	Possible Characters	Detect Characters
Original Image	Find possible plate	Detect/Localized plate region	Find possible characters	Detect characters (Load KNN Training)
Grayscale Threshold	Contouring until detection	Extracted and cropped plate	Contouring until characters detected	Extract characters Text output

Fig. 8. Image Processing.

E. Video Input Processing

Video processing is a particular case of signal processing, which often employs video filters and where the input and output signals are video files or video streams [6]. The last *Visual Basic – Windows Forms Application* project made was the *License Plate Recognition – Video Clip of Moving Vehicle (Input)* which uses video as input. The video undergoes video processing such as *blob analysis*. The system creates a virtual

line in the image frame that can be used to determine if a vehicle passes through. If the system detects the presence of a vehicle, it captures an image frame and subject that frame to image processing. The process is similar to the works of references [20], [8], and [21].

F. Evaluation of System's Performance

The performance of the system is evaluated using the following parameters: *plate number correctness* and *recognition accuracy*. The *plate number correctness* is expressed as the weighted score of the plate, $s_w(\mathbf{P}^R)$, defined as:

$$s_w(\mathbf{P}^R) = \frac{|\{P_i^R | P_i^R = P_i^C\}|}{|\{P_i^R\}|} \times 100\% = \frac{m}{n} \times 100\% \quad (1)$$

Where \mathbf{P}^R is plate recognized by a system, and \mathbf{P}^C is the correct one, m = number of correctly recognized characters and n = total number of characters in a plate. For example, if the plate “ABC 1234” has been recognized as “ABC 1284”, the weighted correctness score s_w is 6/7 or 85.7%. The *system accuracy* A , was calculated from the equation

$$A = \frac{\sum \text{weighted score of the license plate}}{\text{number of total plates}} \quad (2)$$

III. RESULT & DISCUSSION

Results of the images that underwent image processing are shown in the following subsections.

A. Character Recognition Training and Preliminary Testing

OCR were performed on typed-text input in .png format as a valid input with the letters of the English alphabet (capital/uppercase letters only) and digits 0 – 9 as classes of characters [22]. Figure 9(a) shows the interface that opens the image file used for training. During training, the letters in the image were selected and was surrounded by a red rectangular shaped outline one by one as the user inputs the corresponding letter as shown in Figure 9(b). When the selection of all letters were finished, the system notifies that the training was completed as shown in Figure 10. A sample of the result of the tests after the training was completed is shown in Figure 11(a) and (b), the system recognized all the characters correctly.

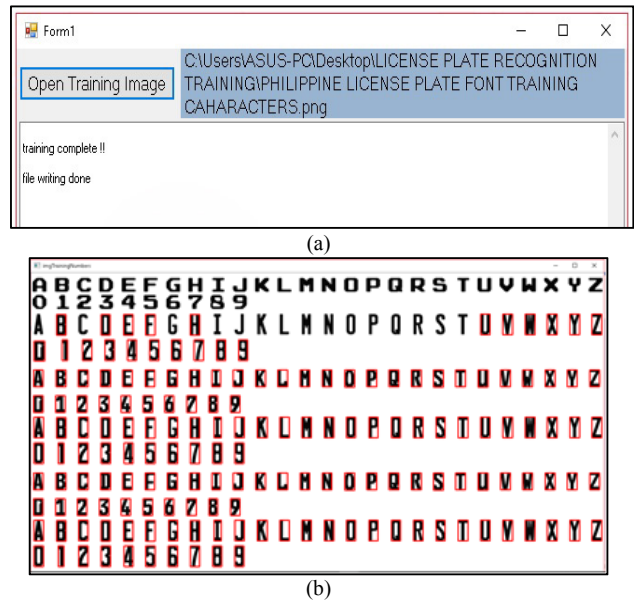


Fig. 9. Character identification

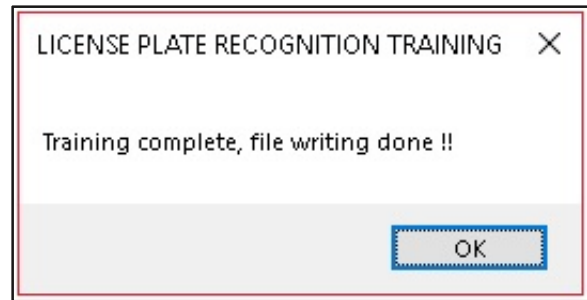


Fig. 10. Notification that the training was completed and that the file writing was done.

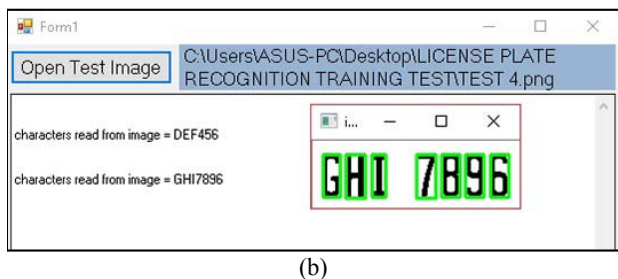
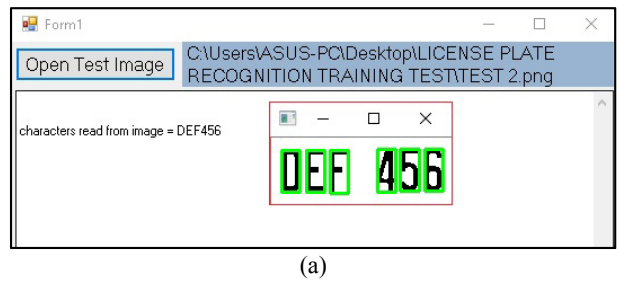


Fig. 11. Testing the System

B. Image Processing of License Plate

The figures shown in Figure 12 are sample of the system’s main interface with an image input file was loaded. The system detects the plate region, the one enclosed by a red box, the segmented image in the lower right corner, and above it is the recognized characters of the license plate in text format. The system saved the results including

the cropped licensed plate region and its gray scaled output. Table 1 shows sample of the saved results: original cropped license plate region, result of thresholding, result of segmentation, recognized characters in text format. The characters in red font color indicates an incorrect character recognition. Note that the table also shows the results for all three different positions.



Fig. 12. License Plate Recognition, a) Upright Position; b) Skewed to Right Position; c) Skewed to Left Position

TABLE 1
IMAGE SAMPLE OF CROPPED LICENCE PLATE REGION

Original image (input)	Gray scaling/Tresholding	Character Segmentation	Text output
Group A: Uright position			
			KAG4891
			APAJJ78
Group B: Skewed to right position			
			E86NV1
			APA117O
Group C: Skewed to left position			
			KK0489
			APA1198

C. License Plate Recognition for Image Input

D. The results of the license plate recognition tests for the three groups of images are tabulated in Table II, III and IV. Each row shows the input, output, the number of characters correctly recognize, and the weighted score computed using

Equation 1. A character indicated in red color indicates an incorrect character recognition. The accuracy of the system for each group of images was calculated using the Equation 2 and they are summarized in Table V.

TABLE 2
LICENSE PLATE WEIGHTED SCORE FOR GROUP A: UPRIGHT POSITION

Plate No.	Input	Output	No. of Correctly Recognized Characters	Weighted Score
1	AAG 4891	KAG4891	6	85.71%
2	APA 1178	APAJJ78	5	71.43%
3	ABO 6206	AB06206	5	71.43%
4	ABH 7344	ABH7344	7	100.00%
5	AOA 2299	A0A2299	6	85.71%
6	ABC 5018	ABC5018	7	100.00%
7	ADO 9171	AD09171	5	71.43%
8	AKA 5591	AKA5591	7	100.00%
9	AOA 1583	AOA1583	7	100.00%
10	AAS 4083	AAS4083	7	100.00%
11	ACU 9594	ACU9594	7	100.00%
12	ABA 8101	ADA810J	4	57.14%
13	ADP 4510	AOP450	5	71.43%
14	AOA 2665	A0A2665	6	85.71%
15	ADP 4557	AOP4667	4	57.14%
16	A0A 2665	A0A2665	6	85.71%
17	AHA 1624	AHA1624	7	100.00%
18	AOA 1875	A0A1B75	5	71.43%
19	AOA 2092	A0A2092	5	71.43%
20	AAG 5979	AA05979	6	85.71%
21	ABT 3242	ADT3242	6	85.71%
22	A0A 1879	A0A1079	5	71.43%
Average Accuracy				83.12%

TABLE 3
LICENSE PLATE WEIGHTED SCORE FOR GROUP B: SKEWED TO RIGHT POSITION

Plate No.	Input	Output	No. of Correctly Recognized Characters	Weighted Score
1	AAG 4891	E86NV1	1	14.29%
2	APA 1178	APA117O	6	85.71%
3	ABO 6206	AD06206	5	71.43%
4	ABH 7344	ASW93AA	1	14.29%
5	AOA 2299	A0324--	1	14.29%
6	ABC 5018	A-C5018	6	85.71%
7	ADO 9171	Y1SOS171	3	42.86%
8	AKA 5591	-JA5591	4	57.14%
9	AOA 1583	A0A1583	7	100.00%
10	AAS 4083	JJS4O83	4	57.14%
11	ACU 9594	ACU97--	4	57.14%
12	ABA 8101	ADA81O1	5	71.43%
13	ADP 4510	JOP45-O	5	71.43%
14	AOA 2665	K0A26-5	5	71.43%
15	ADP 4557	L116A--	0	00.00%
16	A0A 2665	74Z	0	00.00%
17	AHA 1624	--	0	00.00%
18	AOA 1875	1ZY4277	1	14.29%
19	AOA 2092	--	0	00.00%
20	AAG 5979	--	0	00.00%
21	ABT 3242	--	0	00.00%
22	A0A 1879	--9A97-	2	28.57%
Average Accuracy				38.97%

TABLE 4
 LICENSE PLATE WEIGHTED SCORE FOR GROUP C: SKEWED TO LEFT POSITION

Plate No.	Input	Output	No. of Correctly Recognized Characters	Weighted Score
1	AAG 4891	KK04891	4	57.14%
2	APA 1178	APA1198	6	85.71%
3	ABO 6206	AD06206	5	71.43%
4	ABH 7344	ABH7344	7	100.00%
5	AOA 2299	A0A22S9	5	71.43%
6	ABC 5018	A8C50J8	5	71.43%
7	ADO 9171	ABD9171	7	100.00%
8	AKA 5591	AKA5591	7	100.00%
9	AOA 1583	A0A1583	6	85.71%
10	AAS 4083	LLS4083	5	71.43%
11	ACU 9594	ALIAGA2	1	14.29%
12	ABA 8101	A0A2454	6	85.71%
13	ADP 4510	--	0	00.00%
14	AOA 2665	--	0	00.00%
15	ADP 4557	--	0	00.00%
16	A0A 2665	--	0	00.00%
17	AHA 1624	--	0	00.00%
18	AOA 1875	--	0	00.00%
19	AOA 2092	--	0	00.00%
20	AAG 5979	A3D---3	3	42.86%
21	ABT 3242	KDT3242	5	71.43%
22	A0A 1879	--	0	00.00%
Average Accuracy				46.21%

TABLE 5
SYSTEM ACCURACY OF THREE GROUPS

Group	Total number of detected plates	Average Weighted score/ Group
A	22	83.12%
B	19	38.97%
C	19	46.21%

It can be observed in Table II that the system detected the plate numbers of all input images where vehicle are captured in the upright position. The system recognition accuracy is 83.12%. In the case of the skewed vehicle position as shown in Table IV and V, there were cases that the system failed to detect the plate number area such as plate nos. 17, 19, 20, and 2 of Table IV; and plate nos. 13 to 19, and 22 in Table IV. The system recognition accuracy is 38.97% for *skewed to right* position, and 46.21% for *skewed to left* position. This is because the plate numbers of the vehicle in SL or SR position were not clear enough. Incorrect recognition also occurred when the plate number was not properly cropped such as the case in the first entry of Table I, portion of the first character was slightly cropped, the system recognized letter *K* instead of letter *A*. Another observation in Table II, the system failed to distinguished slightly similar characters such as: character *J* was misidentified as *I* in plate no. 2, character *O* was misidentified as *0* and vice versa in plate no.3, and character *B* was misidentified as *D* in plate no. 7. Same goes with plate nos. 12 to 22.

A number of considerable errors were also observed as shown in Table III and IV. In Table III, the characters *AAG 489* in plate no. 1 were incorrectly identified as *E86NV*, and the characters *ABH7344* in plate no. 4 were incorrectly identified as *ASW93AA*. In Table IV, the characters *ACU9594* were incorrectly identified as *ALLAGA2*. Due to these errors, the system recognition accuracy for SR and SL group of images is very low, 38.97% and 46.21%, respectively.

D. License Plate Recognition for Video Input

The two video clips fed to the system are shown is shown in Figure 13. The system captured frame and processed it repeatedly, and displayed the output of the system while playing the input video. It was observed many times that the output is incorrect. The procedure slowed down the whole process of license plate character recognition. However, after several attempts of locating plate number, the system finally locate the plate number area, segment, and recognize the characters correctly. In addition, the system successfully displayed the correct plate number in text format.

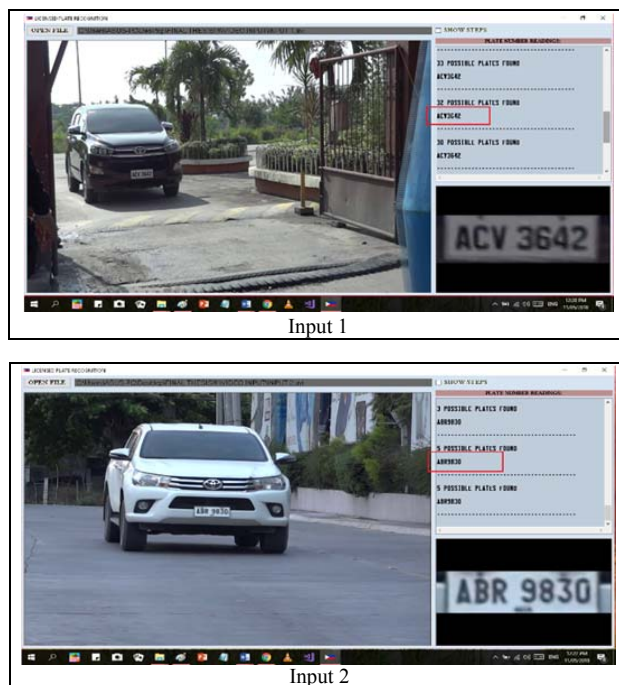


Fig. 13. Loading Video Input

V. CONCLUSION & RECOMMENDATIONS

The designed automatic license plate recognition system was successfully implemented in Visual Basic using EmguCV image processing library and KNN machine learning algorithm. It was able to recognized characters in the license plate using image or video input. The tests using two video inputs with one vehicle each was successful. The system was able to locate and recognize all the characters in the license plate accurately as displayed in the output. The system's performance using image inputs in three groups of images: UP, SR, and SL, is 83.12%, 38.97%, and 46.21%, respectively. The best performance is when the vehicle is in the upright position in the captured image that is 83.12%. Incorrect recognition in the UP position is due the similarity of the characters such as O and 0, J and I, B and D. Considerable errors in SL and SR positions are due mainly to the poor quality of the captured image of the vehicle.

For the improvement of the system, it recommended to use exact font style of license plate in training the system, and capture samples images at different angle positions and lighting conditions. Obtain samples as many as possible. Also, the system can be improved by adding more features to the interface like time stamp or database and work for the methods to make the detection real-time.

REFERENCES

- [1] V. Gigante and S. B. Jolcanain, "Development of a Motorcycle License Plate Number Recognition System," pp. 1, 13, March 2014.
- [2] A. Tamayo, "Occurrence of Traffic Accidents in the Philippines: An Application of Poisson Regression Analysis," *SSRN Electronic Journal*, p. 10.2139/ssrn.1438478, July 2009.
- [3] J. Dr. Balcanao, "Would you say that accidents on Philippine roads are caused mostly by errant drivers?," Phil Star Global, Benguet, 2011.
- [4] E. Tipan, "DOTC, LTO to impose stiffer fines for traffic violations," AUTO INDUSTRY NEWS, Quezon, 2014.
- [5] R. K. C. Billones, A. A. Bandala, E. Sybingco, L. A. Gan Lim, A. D. Fillone and E. P. Dadios, "Vehicle Detection and Tracking using Corner Feature Points and Artificial Neural Networks for a Vission Based Contactless Apprehension System," *Computing Conference 2017*, p. 688, 2017.
- [6] "Video processing," 1 July 2017. [Online]. Available: https://en.wikipedia.org/wiki/Video_processing.
- [7] R. K. Billones, A. Bandala, E. Sybingco, L. Gan Lim and E. P. Dadios, "Intelligent System Architecture for a Vision-Based Contactless Apprehension for Traffic Violations," p. 1, 2017.
- [8] A. Sobral, "simple_vehicle_counting," 12 04 2017. [Online]. Available: https://github.com/andrewssobral/simple_vehicle_counting.
- [9] License Plate Recognition, "How License Plates Recognition Works," 2010. [Online]. Available: www.licenseplatesrecognition.com/how-lpr-works. [Accessed May 2018].
- [10] S. Bold and B. Sosorbaram, "Smart License Plate Recognition Using Optical Character Recognition Based on," *International Journal on Recent and Innovation Trends in Computing and Communication*, pp. 92-96, 2017.
- [11] V. Ong and D. Suhartono, "USING K-NEAREST NEIGHBOR IN OPTICAL CHARACTER RECOGNITION," *ComTech Vol. 7 No. 1*, pp. 53-65, 2016.
- [12] S. Shi, *Emgu CV Essentials*, Birmingham B3 2PB, UK: PACKT Publishing Ltd., 2013.
- [13] S. Saxena and M. S. Tiwari, "Starting with Emgu CV," 19 September 2013. [Online]. Available: <https://social.technet.microsoft.com/wiki/contents/articles/15385.starting-with-emgu-cv.aspx>.
- [14] S. Dr. Sayad, "K Nearest Neighbor," 2010. [Online]. Available: <http://chem-eng.utoronto.ca/~datamining/>.
- [15] K. Kaur and V. Banga, "NUMBER PLATE RECOGNITION USING OCR TECHNIQUE," *IJRET: International Journal of Research in Engineering and Technology*, pp. 286-290, 2013.
- [16] L. Eikvil, *OCR Optical Character Recognition*, P.B. 114 Blindern: Norsk Regnesentral, 1993.
- [17] G. Anbarjafari, "Digital Image Processing," 2014. [Online]. Available: <https://sisu.ut.ee/imageprocessing/book/1>.
- [18] C. Dahms, "OpenCV_3_KNN_Character_Recognition_Emgu_CV_3_Visual_Basic," 9 December 2015. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_KNN_Character_Recognition_Emgu_CV_3_Visual_Basic/blob/master/DocsAndPresentation/KNN_Overview.png.
- [19] C. Dahms, "OpenCV_3_License_Plate_Recognition_Emgu_CV_3_Visual_Basic," 20 July 2017. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_License_Plate_Recognition_Emgu_CV_3_Visual_Basic.
- [20] C. Dahms, "OpenCV_3_Car_Counting_Visual_Basic," 20 February 2016. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_Car_Counting_Visual_Basic.
- [21] E. P. Dr. Dadios, A. Dr. Fillone, E. Engr. Sybingco, L. G. Dr. Lim and A. Dr. Bandala, "Contactless Apprehension of Traffic Violators on 24-hours Basis All-Vehicle Detection System (CATCH-ALL)," p. 18, 2015.
- [22] J. Nguyen, "Introduction to Optical Character," p. 9, 2014.

Real-Time Vehicle Classification Using MobileNet

Reagan L. Galvez,^{1,*} Melvin K. Cabatuan,² and Argel A. Bandala²

Abstract—classification is an important part of vision systems and has several applications like autonomous cars and surveillance. This is a challenging task because computers see images differently from humans. This paper used the MobileNet model for training the data and tested it on an Android device. This model is lightweight and efficient compared with previous developed models. This was inspired by the sample code from Google Codelabs. Experiment results show that the Android application can accurately classify the type of vehicle in real time.

Keywords: convolutional neural network, deep learning, MobileNet, vehicle classification

I. INTRODUCTION

Intelligent transport systems (ITSs) are developed to provide safe travel and ensure effective transportation [1]. To implement this, the ITS needs access to data like the type of vehicle. This will help authorities to identify criminals quickly. It can also be used to apprehend violators like trucks [2], during certain periods of time (ex., rush hour) when trucks are not allowed on the road. Vehicle classification maybe a simple problem for a human, but for a computer, it's a complex problem. That's why many researchers are exploring image processing [3] to solve this problem with high accuracy by creating different architectures/models. There are many techniques that can be used to classify images like artificial neural networks, decision tree, support vector machine, and fuzzy measure [4]. Currently, convolutional neural networks (CNNs) are used in image classification because they provide accurate performance in computer vision tasks. There are many pre-trained CNN based models that can be used for image classification like AlexNet [5], VGG16, VGG19 [6], ResNet50 [7], InceptionV2, InceptionV3 [8], Xception

[9], and DenseNet [10]. On the other hand, models like R-CNN [11], Fast R-CNN [12], YOLO [13], YOLO9000 [14], SSD [15], and MobileNet [16] are commonly used object classification and detection models. The focus of this paper is to utilize MobileNet for object classification using android platform.

This paper is organized as follows: section 2 discusses the concept about MobileNet and its architecture. Section 3 describes the experiment setup and the dataset used for training. Section 4 shows the performance of the MobileNet model in vehicle classification and its deployment in an Android platform.

II. MOBILENET

MobileNet is an efficient model designed for mobile and embedded vision applications [16]. It uses depthwise separable convolutions to build lightweight and efficient deep neural networks. The depthwise convolution applies a single filter to each input channel. To combine the output of depthwise convolution, pointwise convolution applies 1×1 convolution. The combination of two convolutions results to a depthwise separable convolution. Two layers are formed by depthwise separable convolution. The first layer is used for filtering, and the second layer is used for combining. This is called factorization, and as a result, it reduces the model size and computation.

Table 1 shows the MobileNet architecture. The model consists of 28 layers. These layers are combinations of alternating depthwise and pointwise convolutions. Every layer is followed by batch normalization and rectified linear unit (ReLU) [17] function except the fully connected layer which is followed by a softmax classifier that gives actual probabilities in each class. Batch normalization speeds up the training [18], and the ReLU function is 0 for negative values and grows linearly for positive values. The notations s_1 and s_2 are the number of strides. Stride controls how the filter convolves around an input volume.

Reagan L. Galvezla, Bulacan State University, Malolos City, Philippines (e-mail: reangalvez@gmail.com)

Melvin K. Cabatuan and Argel A. Bandala, De La Salle University–Manila, Philippines

TABLE 1
MOBILENET ARCHITECTURE

Layer	Type/Stride	Input Size
1	Conv /s2	224 × 224 × 3
2	Conv dw /s1	112 × 112 × 32
3	Conv /s1	112 × 112 × 32
4	Conv dw /s2	112 × 112 × 64
5	Conv /s1	56 × 56 × 64
6	Conv dw /s1	56 × 56 × 128
7	Conv /s1	56 × 56 × 128
8	Conv dw /s2	56 × 56 × 128
9	Conv /s1	28 × 28 × 128
10	Conv dw /s1	28 × 28 × 256
11	Conv /s1	28 × 28 × 256
12	Conv dw /s2	28 × 28 × 256
13	Conv /s1	14 × 14 × 256
14–23	Conv dw/ s1 Conv /s1	14 × 14 × 512 14 × 14 × 512
24	Conv dw /s2	14 × 14 × 512
25	Conv /s1	7 × 7 × 512
26	Conv dw /s2	7 × 7 × 1024
27	Conv /s1	7 × 7 × 1024
	Avg. Pool /s1	7 × 7 × 1024
28	FC /s1	1 × 1 × 1024
	Softmax /s1	1 × 1 × 1000

III. EXPERIMENT SETUP

The effectiveness of MobileNet in vehicle classification was tested in an android application. It is a simple camera application that runs a TensorFlow image recognition program to identify vehicles. This was inspired by the code from “TensorFlow for Poets 2” of Google Codelabs [19]. TensorFlow mobile was used to run the MobileNet and integrated it to mobile application. The Anaconda Prompt was used to initiate command in training and testing the model.

The first step in training the data is to install dependencies like TensorFlow. This is an open-source library for high-performance calculation, which allows easy deployment in different platforms and supports deep learning applications [20]. Figure 1 shows the block diagram of the vehicle classification structure. Next is to collect the data that will be used for training. The dataset is composed of five types of vehicles such as pickup, SUV, sedan, van, and truck. These images came from ImageNet, an image database. Table 2 shows the number of images per category of vehicle.

TABLE 2
DATASET

Vehicle Type	Number of Images
Pickup	400
SUV	400
Sedan	400
Van	400
Truck	400
Total	2000

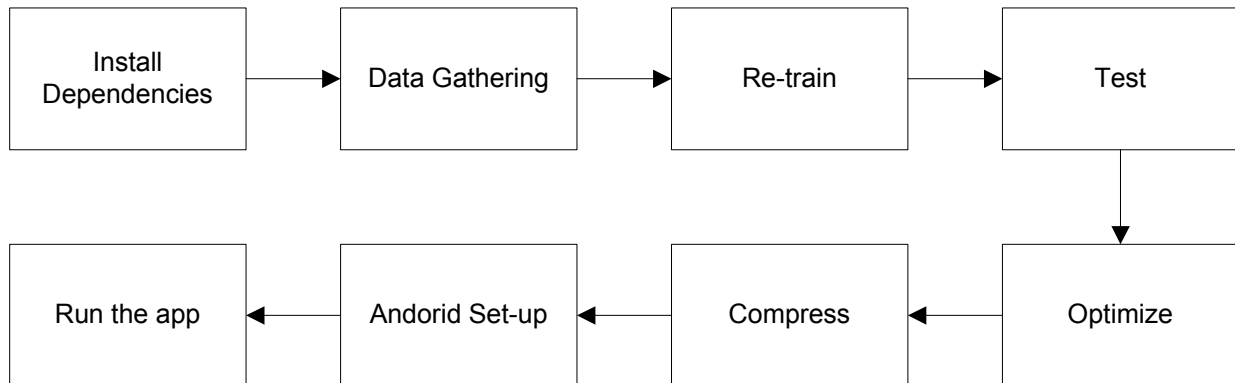


Fig.1. Block diagram.

Next is to re-train the data using the MobileNet model. The output of this is a .pb file; this is a single model file that contains graph variables frozen as constants. The input images can be configured using 4 different input sizes such as 128, 160, 192, and 224 pixels. The width multiplier (α) of the model can also be set such as 1.0, 0.75, 0.5, and 0.25. The width multiplier makes the model small and faster by reducing computational cost and number of parameters.

Then, test the model by using random image and compute its accuracy. After this, optimize the model by removing some nodes that are not needed in a given set of input and output. Next is to compress the model by quantizing the network weights to make it ideal for mobile applications. After compression, the model is ready to upload in an Android platform.

IV. RESULTS

A. Image Classification Performance

The performance of the MobileNet model was tested by calculating its validation accuracy, cross entropy, and evaluation time. Validation accuracy is the precision on a randomly selected group of images from a different set. Cross entropy is a loss function that shows how well the learning process is progressing. The ideal value of cross entropy is 0. Evaluation time is the time it takes to classify the test image. The total number of steps used in training the model was 2000 steps. This is enough to see if the model is learning. Table 3 shows the accuracy and cross entropy using different values of width multiplier (α) in MobileNet. It shows that when the MobileNet's width multiplier (α) decreases, the validation accuracy also decreases and the cross entropy increases.

TABLE 3
ACCURACY AND CROSS ENTROPY

Input Size	Width Multiplier (α)	Accuracy (%)	Cross Entropy
224	1.0	84.83	0.541196
224	0.75	84.36	0.554476
224	0.5	81.99	0.713274
224	0.25	72.04	2.38165

Figure 2 shows the training (orange) and validation accuracy (blue) during the training. This graph was from MobileNet using $\alpha = 1.0$. As shown, the model was not learning anymore. In this case, the training can be stopped. The average validation accuracy is equal to 84.83%.

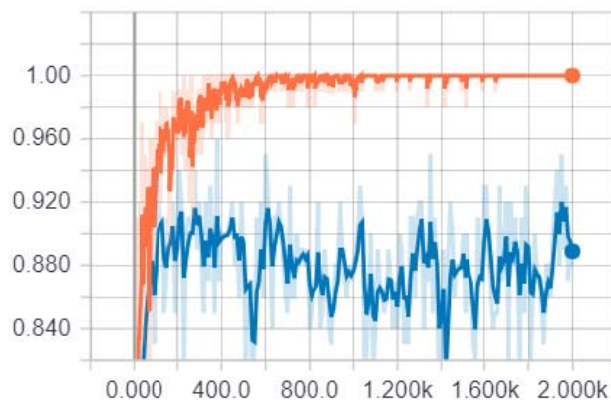


Fig. 2. Training steps vs. accuracy.

Figure 3 shows the training (orange) and validation cross entropy (blue) during the training. The ideal value of cross entropy is 0 because this is a loss function. The average cross entropy is equal to 0.541196.

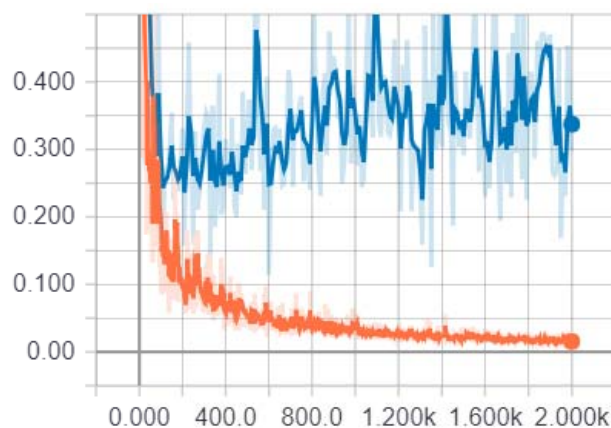


Fig. 3. Training steps vs. cross entropy.

Table 4 shows the time comparisons using different width multiplier (α) in the model. As the width multiplier (α) decreases, the time of evaluation also decreases. This is because the model became lightweight and has fewer computations.

Table 5 shows the model size comparisons for each different width multiplier (α) in the model. Using $\alpha = 1.0$, there was 32.50% size reduction from the optimized to the compressed model. This compression can be useful in deploying the model for mobile applications because it can be downloaded easily.

Figure 4 shows the bubble chart of evaluation time versus accuracy. The size of the bubble is proportional to the size of the model. As we can see, there is a trade-off between accuracy and speed. If we need a faster model, $\alpha = 0.25$ can be used, but the accuracy will decrease. For real-time application, $\alpha = 1.0$ can still be used.

TABLE 4
EVALUATION TIME COMPARISONS

Input Size	Width Multiplier (α)	Time (Re-Trained) (s)	Time (Optimized) (s)	Time (Compressed) (s)
224	1.0	2.163	2.023	2.061
224	0.75	1.756	1.776	1.701
224	0.5	1.352	1.341	1.359
224	0.25	1.122	1.121	1.121

TABLE 5
MODEL SIZE COMPARISONS

Input Size	Width Multiplier (α)	Size (Re-Trained) (KB)	Size (Optimized) (KB)	Size (Compressed) (KB)
224	1.0	15,489	15,485	5,033
224	0.75	9,490	9,486	3,174
224	0.5	4,916	4,913	1,786
224	0.25	1,766	1,763	708

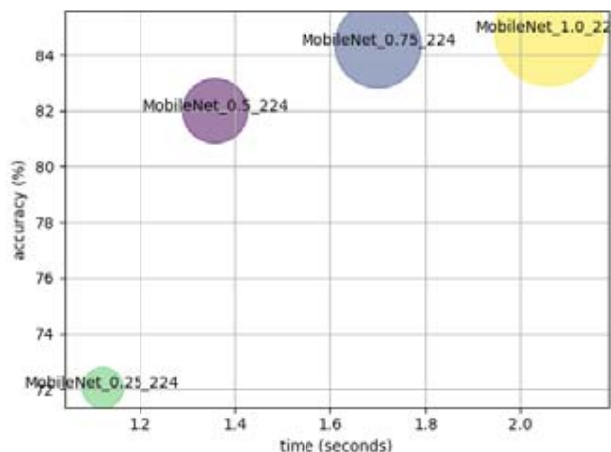


Fig. 4. Evaluation time vs. accuracy.

B. Deployment in Android Platform

The trained model was deployed in an Android platform to test its performance in real-time application. TensorFlow mobile was used to prepare the model for mobile deployment. Figure 5 shows the screenshot from the TensorFlow application using a Samsung Galaxy S7 Flat phone. The testing images were randomly selected to test if the application can detect unseen data. The actual images in the figure were both sedan and correctly identified as sedan.

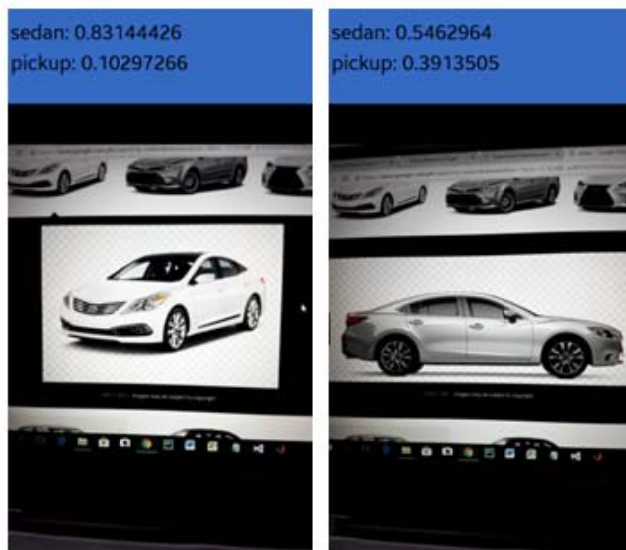


Fig. 5. Screenshot from TensorFlow application (sedan).

Another set of images was tested as shown in Figure 6. These images were a truck and correctly identified as truck by the TensorFlow application.

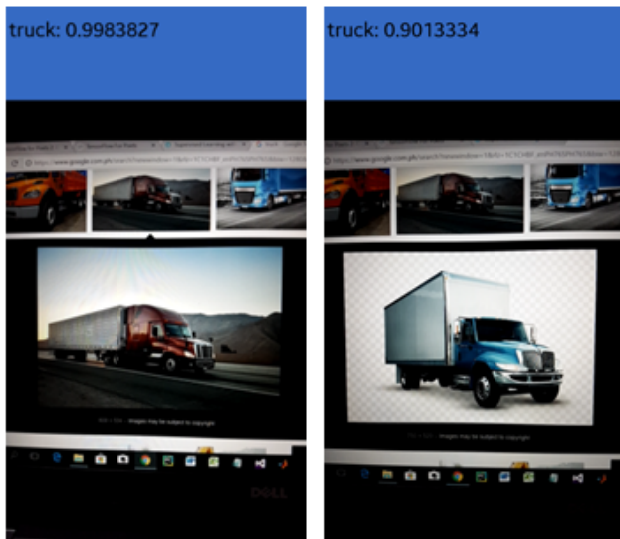


Fig. 6. Screenshot from TensorFlow application (truck).

Figure 7 shows an image of an SUV. The TensorFlow application failed to classify the first image as SUV; it was predicted as sedan. This is due to the similar features of an SUV and a sedan.

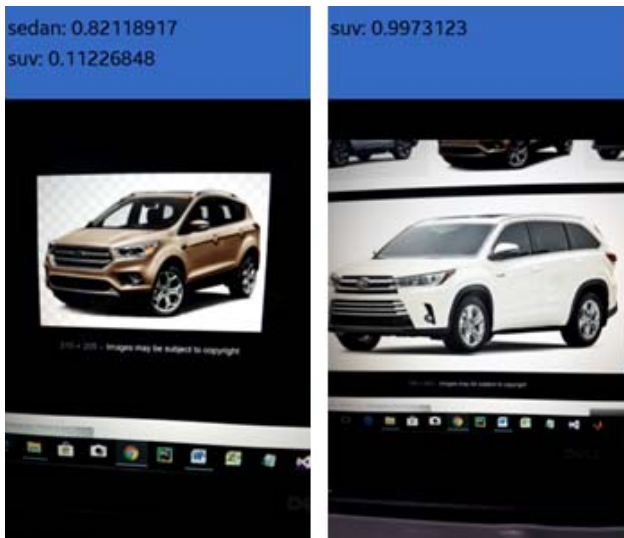


Fig. 7. Screenshot from TensorFlow application (SUV).

V. CONCLUSION

The implementation of real-time vehicle classification was successfully tested using the Android platform. The experiment result shows that the MobileNet can classify vehicle type up to 84.83% accuracy. TensorFlow mobile is a good deep learning solution for a mobile platform like Android. Although in testing an SUV image, the

model cannot consistently classify the image, other types of vehicles were accurately identified. This is due to the physical structure of the SUV that is not totally different from the other types of vehicles because some SUVs are similar to a sedan, van, and pickup. As a solution, it is recommended to increase the dataset and remove some images that are confusing.

For future work, the application of MobileNet can be upgraded to track and identify vehicle type using CCTV videos in real-time.

ACKNOWLEDGMENTS

The author would like to thank the Department of Science and Technology—Engineering Research and Development for Technology (DOST-ERDT) and De La Salle University for the financial support while doing this research.

REFERENCES

- [1] A. Ferdowsi, U. Challita and W. Saad, "Deep Learning for Reliable Mobile Edge Analytics in Intelligent Transportation Systems," *arXiv:1712.04135v1*, December 2017.
- [2] R. K. C. Billones, A. A. Bandala, E. Sybingco, L. A. G. Lim and E. P. Dadios, "Intelligent system architecture for a vision-based contactless apprehension of traffic violations," *2016 IEEE Region 10 Conference (TENCON)*, pp. 1871 - 1874, 2016.
- [3] R. K. C. Billones, A. A. Bandala, E. Sybingco, L. A. G. Lim, A. D. Fillone and E. P. Dadios, "Vehicle Detection and Tracking using Corner Feature Points and Artificial Neural Networks for a Visionbased Contactless Apprehension System," *Computing Conference 2017*, pp. 688-691, 2017.
- [4] P. Kamavisdar, S. Saluja and S. Agrawal, "A Survey on Image Classification Approaches and Techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. II, no. 1, pp. 1005-1009, January 2013.
- [5] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in neural information processing systems*, 2012.
- [6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," in *ICLR*, 2015.
- [7] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference Publication*, pp. 1-12, 2015.
- [8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," *arXiv:1409.4842v1*, pp. 1-12, September 2014.
- [9] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *IEEE Conference Publication*, pp. 1-8, 2017.

- [10] G. Huang, Z. Liu, L. Van der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [11] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv:1311.2524v5 [cs.CV]*, pp. 1-21, 2014.
- [12] R. Girshick, "Fast R-CNN," *arXiv:1504.08083v2 [cs.CV]*, 2015.
- [13] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv:1506.02640v5 [cs.CV]*, pp. 1-10, 2016.
- [14] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *arXiv:1612.08242v1 [cs.CV]*, pp. 1-9, 2016.
- [15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," *arXiv:1512.02325v5*, pp. 1-17, 2016.
- [16] A. G. Howard, M. Zhu, B. Chen and D. Kalenichenko, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision," 2017.
- [17] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010.
- [18] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv:1502.03167v3*, pp. 1-11, March 2015.
- [19] Codelabs, Google, "Tensorflow for Poets 2," December 2017. [Online]. Available: <https://github.com/googlecodelabs/tensorflow-for-poets-2>.
- [20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng, "TensorFlow: A system for large-scale machine learning," *arXiv:1605.08695v2*, pp. 1-18, May 2016.

The Contributors

Ariel James T. Abarquez was born in General Santos City, South Cotabato in June 29, 1997. He received his bachelor's degree in Electronics Engineering (ECE) from Mindanao State University – GSC on June 2018. He was a scholar of Darlene Antonino Foundation. He conducted his On-the-Job Training at the Civil Aviation Authority of the Philippines (CAAP) under the Air Navigation Services (ANS) department. He was a member of the Junior Institute of Electronics Engineers of the Philippines (JIECEP), Organization of Electronics and Electrical Engineering Students and Engineering Students Organization.

Ryann A. Alimuin is a full-time faculty of the Electronics Engineering Department of Technological Institute of the Philippines – Quezon City and is currently taking his Ph.D in Electronics Engineering at the De La Salle University – Manila. He Received his BSECE in TIP-QC (2007) and MEng in TUP-Mla (2012). He is a registered Professional Electronics Engineer and a licensed professional teacher major in Mathematics. He is a consultant of several industries for CCTV systems layout, Fire Detection and Safety, PABX and Structured Cabling. His research works are in the field of industrial and power electronics, power management systems, as well as, intelligence systems for precision agriculture. Currently, he is engaged in research works for smart farming, RF groundwave propagations and face vector analysis for surveillance systems.

Ramtin Aminpour is a PhD candidate in Electronics and Communications Engineering in De La Salle University. He received his Bachelors of Electronics Engineering in Azad University of Tabriz in 1998. He received his Master of Digital Electronics Engineering in Iran University Science and Technology, Tehran, in 2008. His current research works are in the field of artificial intelligence, data mining, and machine learning. He is a member of the Institute of Electrical and Electronics Engineers (IEEE)

Argel A. Bandala is an Associate Professor and Research Faculty of the Electronics Engineering Department at De La Salle University. He received his Master of Science in Electronics and Communications Engineering in

year 2012 and Doctor of Philosophy in Electronics and Communications Engineering in year 2015 at De La Salle University. He is the current Vice-Chair, The Institute of Electrical and Electronics Engineers (IEEE) Philippines Section and secretary of Computational Intelligence Society Philippine Chapter. He is also a member of IEEE Robotics and Automation Society. His main works are “Implementation of Varied Particle Container for Smoothed Particle Hydrodynamics-Based Aggregation for Unmanned Aerial Vehicle Quadrotor Swarm” and “Swarming Algorithm for Unmanned Aerial Vehicle (UAV) Quadrotors – Swarm Behavior for Aggregation, Foraging, Formation, and Tracking.

Jomari Joseph A. Barrera is an instructor in the Department of Computer Science and Technology at Visayas State University (VSU), Visca, Baybay City. He finished his Bachelor of Science in Computer Science in 2012 and pursuing his master's degree in Computer Science at Cebu Institute of Technology - University (CIT-U). His field of interests are data structures and algorithms, computer vision and intelligent systems.

Klariz Donna Mae B. Bundal was born in Aborlan Medicare Hospital, Poblacion, Aborlan, Palawan on August 15, 1996. She is the second from the eldest of five children of Adon C. Bundal and Evangeline B. Bundal. She started her preparatory school at Western Philippines University – Basic Elementary Laboratory School, finished next her elementary level at San Juan Elementary School and secondary at Western Philippines University – Agricultural Science High School. She received her degree of Bachelor of Science in Electronics Engineering from Mindanao State University – General Santos City Campus on June 2018. She was a consistent Department of Science and Technology – Science Education Institute (DOST-SEI) scholar under R.A. 7687 Scholarship Program. Spent five years in service as an officer of the Organization of Electrical and Electronics Engineering Student (OEECES – MSU GSC) and member of Junior Institute of Electronics Engineers of the Philippines (JIECEP – SOCCSKSARGEN) and finishes her On the Job Training at Wireless Link Technologies Incorporated, General Santos City branch.

Melvin K. Cabatuan received the B.Sc. degree in Electronics and Communications Engineering (ECE) from Cebu Institute of Technology, Cebu, Philippines, in 2004; MS degree in Engineering from Nara Institute of Science and Technology (NAIST), Nara, Japan, in 2010; and Ph.D. degree in ECE from De La Salle University (DLSU) - Manila in 2016. He joined the Electronics and Communications Engineering department of DLSU in 2011, where he is currently an Assistant Professor. His current research interest involves Machine/Deep Learning, Computer Graphics, and Computer Vision applications to Health Informatics, Education Technology, Game/Graphic Content Development, and Mobile Computing. He is knowledgeable in programming languages - C/Cpp, Java, and MatLab/Octave scripting. He has more than 5-year experience in computer vision application development with OpenCV library in the mobile framework.

Elmer P. Dadios is a University Fellow and Professor at De La Salle University. He is also the president, Neuronemch, Inc. In 1996, he received his Doctor of Philosophy from Loughborough University. In 1997, he was an Exchange Scientist in Japan Society for the Promotion of Science, Tokyo Institute of Technology. He served as the Director of Engineering Graduate School, De La Salle University in 1998-1999 and Director of School of Engineering, De La Salle University in 2003-2004. He is also the General Chair for HNICEM in 2003, 2005, 2007, 2009, 2011, 2013. His main works are "Fuzzy Logic – Controls, Concepts, Theories and Applications," ISBN: 978-95351-0396-7, 2012. "Fuzzy Logic – Algorithms, Techniques and Implementations," ISBN: 978-953-510393-6, 2012. "Fuzzy Logic – Emerging Technologies and Applications," ISBN: 978-953-51-03370, 2012. His research interests includes; Robotics, Mechatronics, Automation, Intelligent Systems, Neural Networks, Fuzzy Logic, Genetic Algorithms, Evolutionary Computation and IT. He is a Senior Member in The Institute of Electrical and Electronics Engineers (IEEE) and founder and current of Chair of IEEE Computational Intelligence Society, Philippines. He is a member of IEEE Region 10 Executive Committee and founder and President of The Mechatronics and Robotics Society of the Philippines.

Cristina P. Dadula is a professor of Electrical and Electronics Engineering Department of Mindanao State University, General Santos City. She earned her PhD in Electronics and Communications Engineering degree at De La Salle University, Manila, Philippines. She finished her Bachelor of Science in Electronics and Communications Engineering, and Master of Science in Computer Applications at Mindanao State University-Iligan Institute of Technology. Her research interests include design and implementation of artificial intelligent algorithms, digital signal processing, robotics, and control systems.

Reagan L. Galvez received his Bachelor of Science in Electronics and Communications Engineering from Bulacan State University (BSU) in 2010. He received his Master of Science in Electronics and Communications Engineering from De La Salle University (DLSU) in 2016. He is currently an instructor in BSU, while taking his Doctor of Philosophy in Electronics and Communications Engineering at DLSU. His research interests are in the field of artificial intelligence, robotics and machine learning. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), and Institute of Electronics Engineers of the Philippines (IECEP).

Sittie Alyzza M. Garcia was born on Tacloban City, Leyte, Philippines in May 21, 1996. She earned her bachelor's degree in Electronics Engineering from the Mindanao State University – General Santos last June 2018. She was also a scholar of Department of Science and Technology – Science Education Institute under Republic Act No. 10612. A member of Junior Institute of Electronics Engineers of the Philippines, Organization of Electrical and Electronics Engineering Students, and Engineering Students Organization in MSU-GSC.

Arlyn P. Lauron was born in General Santos City, Philippines, in July 06, 1997. She received her bachelor's degree in Electronics Engineering from Mindanao State University-GSC on June 2018. She was also a scholar of Department of Science and Technology – Science Education Institute under Republic Act 7687. She conducted her On-the-Job Training at the Civil Aviation Authority of the Philippines-GSC under the Air Navigation Services department. She was member of Junior Institute of Electronics Engineers of the Philippines and an officer of her college.

Elmer A. Maravillas is a Full Professor, Chair of Computer Science, Coordinator of Doctor in Information Technology Program and Consultant of Research and Development Coordinating Office (RDCO) at Cebu Institute of Technology - University (CIT-U), Cebu City. He received his BS degree in Mechanical Engineering in 1981 from Western Mindanao State University (WMSU), MS Computer Science in 1998 and PhD in Mechanical Engineering in 2006 from De La Salle University-Manila. He worked as Mechanical Project Engineer and Power Plant Engineer of M&S Company Inc. from 1982 to 1984. He served as Department Head of Computer Engineering Department in WMSU from 1984 to 2003 and RDCO in CIT-U from 2006 to 2007. His main works include modeling and simulations, fuzzy logic systems, neural networks, intelligent systems and expert systems.

Jonah Flor V. Oraño is an instructor in the Department of Computer Science and Technology at Visayas State University (VSU), Visca, Baybay City. She completed her Bachelor of Science in Computer Science degree in 2006 at VSU and MS in Information Technology in 2012 at University of San Carlos (USC), Cebu City. She is currently pursuing Doctor in Information Technology at Cebu Institute of Technology - University (CIT-U). Her research interests are computer vision, machine learning, fuzzy logic systems, and data mining.

Guidelines for Contributors

1. The Journal on Computational Innovations and Engineering Applications (JCIEA) aims to promote the development of new and creative ideas on the use of technology in solving problems in the field of computational applications, computational intelligence, electronics and information and communications technology (ICT), manufacturing engineering, energy and environment, robotics, control and automation, and all their related fields. Manuscript submissions should, therefore, be in pursuit of the same goal and within the related fields.
2. JCIEA only accepts manuscripts written in English. The responsibility for copyediting manuscripts, as well as obtaining reproduction permissions for the use of graphics and other materials from their references, will fall on the author.
3. Authors must also remember to cite all references and ensure that their paper submission has not been previously published or is undergoing peer review for another publication.
4. Manuscripts should include a unique title, an abstract, some keywords, an introduction and discussion of the study, a presentation and discussion of results, and a conclusion. Authors may also include an acknowledgement of funding organizations or consultants, if needed.
5. Manuscripts may be sent to *jciea.dlsu@gmail.com* or *jciea@dlsu.edu.ph* as either an MS Word file (*.doc or *.docx) or a LaTeX file (*.tex), including its supporting files.
6. Manuscripts in either file format should have the following features:
 - Single-spaced, two-column format with 1-inch margin on all sides on letter-sized template
 - Font to be used is Times New Roman, size 11
 - Graphs (*.eps, *.svg), tables (*.csv), and images (*.jpg, *.png) should be saved and sent apart from the MS Word file.
 - Citations and references should be submitted in IEEE or APA format.
 - Submission of these references in a BibTeX format is preferred.
7. Manuscript should be eight to twelve (8–12) pages long, including all figures, tables, and references. Manuscripts exceeding the 12-page limit will require permissions from the editors.
8. Authors must include their full names and affiliations in the manuscript. They may include a 150- to 200-word biography to be included in the back portion of the journal.

