International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Deep Learning-Based Embedded System for Carabao Mango (Mangifera Indica L.) Sorting

Ryan Joshua Liwag, Kevin Jeff Cepria, Anfernee Rapio, Karlos Leo Castillo, Melvin Cabatuan

Abstract*: This paper presents the design and development of an embedded system for 'Carabao' or Philippine mango sorting utilizing deep learning techniques. In particular, the proposed system initially takes as input a top view image of the mango, which is consequently rolled over to evaluate every sides. The input images were processed by Single Shot MultiBox Detector (SSD) MobileNet for mango detection and Multi-Task Learning Convolutional Neural Network (MTL-CNN) for classification/sorting ripeness and basic quality, running on an embedded computer, i.e. Raspberry Pi 3. Our dataset consisting of 2800 mango images derived from about 270 distinct mango fruits were annotated for multiple classification tasks, namely, basic quality (defective or good) and ripeness (green, semi-ripe, and ripe). The mango detection results achieved a total precision score of 0.92 and a mean average precision (mAP) of over 0.8 in the final checkpoint. The basic quality classification accuracy results were 0.98 and 0.92, respectively, for defective and good quality, while the ripeness classification for green, ripe, and semi-ripe were 1.0, 1.0, and 0.91, respectively. Overall, the results demonstrated the feasibility of our proposed embedded system for image-based Carabao mango sorting using deep learning techniques.

***Index Terms*: deep learning, embedded system, deep artificial neural network, multi-task learning, region-based convolutional neural network, image-based mango sorting.**

I. INTRODUCTION

Mangoes are well known throughout the world as one of the sweetest tropical fruits. Before mangoes are exported from tropical countries, i.e. Philippines, farmers and/or merchants segregate the fruits based on different factors, such as quality, and degree of ripeness. Although mangoes can be picked immediately from the tree when physically mature (but unripe), this usually leads to a ripened fruit of substandard quality [1]. However, if the mango is harvested when fully ripe, it is more susceptible to handling and/or transportation damages, deterioration, and may start rotting before it reaches the consumer. Thus, in practice, the degree of ripening the mangoes in the tree before harvest depends on the target market - if the target is local market, then the mangoes may be harvested when almost fully ripe, otherwise, the mangoes are harvested earlier and ripened postharvest. Therefore, besides basic fruit quality, classifying the degree of ripeness is a significant task in the mango production/distribution process.

Recently, deep artificial neural networks are making major advances in solving problems, i.e. image/speech recognition etc. [2][3]. Artificial neural networks are computational models that mimics the human brain, with various interdisciplinary applications, e.g. agriculture/aquaculture [4][5][6][7], healthcare [8][9][10], face detection [11], voice recognition [12], electronic communication [13], forensics [14], etc. In this study, deep artificial neural network was applied for mango detection and quality & ripeness classification tasks in an embedded system environment. Although similar deep learning approaches have been proposed before, i.e. fruit detection in orchards, using Faster Region based Convolutional Neural Network (R-CNN) [15], and vision-based fruit detection system utilizing multi-modal Faster R-CNN model [16], the proposed system utilizes Single Shot MultiBox Detector (SSD), which is generally faster than Faster R-CNN. Furthermore, the proposed system is specifically designed for the embedded platform with a corresponding customized hardware component lacking in the past approaches, which were purely computational.

Multi task learning (MTL) is a technique that has led to the success of many applications in machine learning and have been beneficial to computer vision, throughout the past years [17]. When there is more than one loss function being optimized, it can be considered a multi task learning, in which domain-specific information contained in the training signals of related tasks improves generalization [18]. In other words, when the model is training on a specific task X, it will learn a good representation of task X and ignore the corresponding noise, thus, as the model learns different tasks with each having different noise patterns, this leads to learning a more general data representation. Fig. 1 shows the MTL approach.

Revised Manuscript Received on 30 July 2019.
* Correspondence Author
Ryan Joshua Liwag*, Electronics and Communications Engineering Department, De La Salle University, Manila, Philippines.
Kevin Jeff Cepria, Electronics and Communications Engineering Department, De La Salle University, Manila, Philippines.
Anfernee Rapio, Electronics and Communications Engineering Department, De La Salle University, Manila, Philippines.
Karlos Leo Castillo, Electronics and Communications Engineering Department, De La Salle University, Manila, Philippines.
Melvin Cabatuan, Electronics and Communications Engineering Department, De La Salle University, Manila, Philippines.

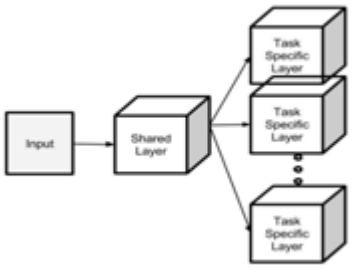

Fig. 1.Multi-task learning (MTL) approach.

MTL also provides implicit data augmentation, by effectively increasing the sample size for each specific feature. MTL can also help models prevent overfitting, as training on a single task X has the risk of overfitting on the specific task, compared to learning tasks X and Y together enables the model to build a better representation through averaging the noise patterns. In this study, we utilized Multi-Task Learning Convolutional Neural Network (MTL-CNN) for classification of the degree of ripeness and basic quality, running on an embedded computer, i.e. Raspberry Pi 3 Model B.

The rest of the paper is organized as follows – section II discusses the methodology, while section III presents the data and results. Finally, this paper is concluded in section IV.

II. METHODOLOGY

The methodology in this study is inspired by evolutionary and incremental prototyping in software development where a prototype is immediately built (both hardware and software), constantly refined, and integrated into a whole system.

A. Proposed Embedded System for Mango Sorting

The proposed mango sorting embedded system is composed of the following main components, namely, camera, embedded computer, display, Gizduino, motor driver and motor, mounted in the fabricated conveyor-style platform, as shown in Fig. 2. The main processing unit for the proposed system is the embedded computer, Raspberry Pi 3 Model B, connected to camera (top of platform) and the display monitor. A ring light is placed on the camera to eliminate shaded areas around the mango. The camera is then placed at the center of the gap between the rollers, the expected position where the mango would be placed by the user also called as the examination area. The camera is suspended 30 centimeters above the mango to minimize the background. The rate of image capture is synchronized with the movement of the motor. For every 180 degrees of angle that the roller makes, a single image is taken at certain intervals covering the two sides of the mango. The rollers are operated by the motor and would revolve only once for the computer to have an image of all sides of the mangoes. Finally, the output of the system is displayed in the monitor after processing and the examination area would then be ready for the next mango.

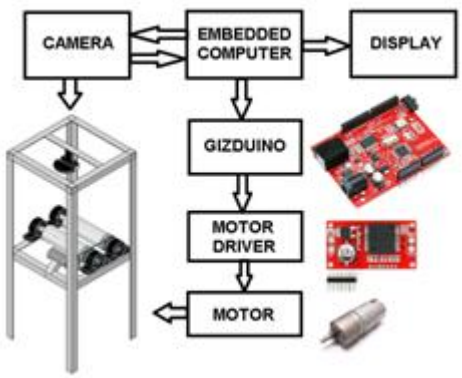

Fig. 2.Conceptual diagram of the proposed embedded mango sorting system hardware

The utilized Logitech C270 HD camera is capable of capturing HD videos for up to 720p resolution and pictures for up to three megapixel, sufficient for this study. The camera comes with a USB 2.0 connection suitable for use in Raspberry Pi. The camera on this research was utilized to capture images and videos of mangoes. The images and videos later on would be used as system input for the classification process. The Gizduino (Arduino clone) module was used as a controller to interface the weight sensor and motor controls with the main processing unit. The Gizduino has a single AVR (Automatic voltage regulator) microcontroller which is the ATMega168 which operates at 5V, with the recommended input into the Gizduino to be around 7-12V. The Gizduino module has 14 digital I/O pins which can be used, six of it are being used for Pulse Width Modulation (PWM). The I/O pins of the Gizduino module are limited to output current of up to 20mA, the 3.3V and 5V pin however can output up to 50mA. The module runs at a clock speed of 16 MHz and the ATmega168 contains 32kb of flash memory. Once the mango is placed in the middle of the rollers on its flat side, the Gizduino would send commands to the driver to deliver enough current to the motor so that it would rotate by 180 degrees. Once this is done, the Gizduino sends a signal to the computer to take a picture using the camera. The process happens again for the system to see the other side of the mango. The images are then processed by the software to determine the characteristics of the mango. The size, level of ripeness, and presence of defects of the mango are known, these characteristics are then displayed on the monitor. The user will then remove the mango from the examination area and place another one to be graded. The VNH2SP30 motor driver is the main motor driver for the DC Gear Motor setup. The motor driver itself is a full-bridge motor controller and is capable of delivering high current that would be used for bidirectional speed control of brushed DC Motor. The motor driver operates at 5.5 - 16 VDC and delivers continuous current of 5A or 14A with thermal management. Initially, the researchers preliminarily proposed a radically different setup involving a clear plastic platform where the mango would rest while being examined and two cameras, one suspended above the mango and another positioned below the mango so that all side of the mango would be visible.

Another camera would be focused on the stem area of the mango. However, initial tests showed that the plastic platform would distort the way the mango looks in the image

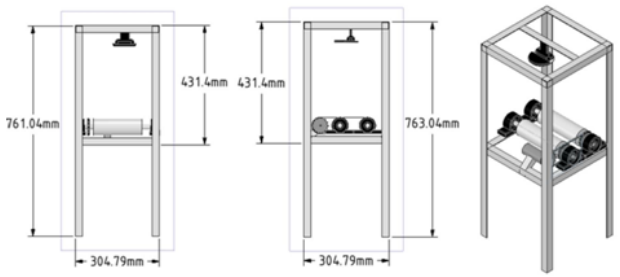

Fig. 3.Conveyor-style platform design

Fig. 4.Implementation of Conveyor-style platform

and can provide lighting artifacts that can hinder the proper characterization of the mango. An alternative to the plastic platform would be a metal mesh, which would remove the distortion problems that the researchers encountered in the previous material. The drawback of this platform would be that it would leave marks on the delicate skin of the fruit since the fruit would be pressed on to the mesh by its own weight. A new setup was then formulated that eliminated the need for a plastic platform, as shown in Fig. 3, 4. The number of cameras was reduced to one while the mango now rests on two fabricated stainless steel rollers. The rollers smoothened to ensure that its surface would not bruise the skin of the fruit, were spaced accordingly so that even the smallest Carabao mango would not slip through the gap while ensuring that the mango would remain in place while being rolled. The SGM37-550 DC Geared motor is connected by a chain and sprocket system to the two rollers to facilitate the rolling process. While this new setup is more complicated, it would eliminate the need for more cameras since all sides of the mango would have a chance to be seen by the camera during the operation of the system.

To integrate the software with the hardware, serial communication channel between the Python script and the Gizduino was utilized. The Python script will send a serial signal that will be passed to the Gizduino, which will in turn be sending the appropriate commands to the motor driver to turn the rollers a full 180 degrees. This process would allow the capture of images of each of the two sides of the mango.

Fig. 5.Sample mango image with augmented set of images

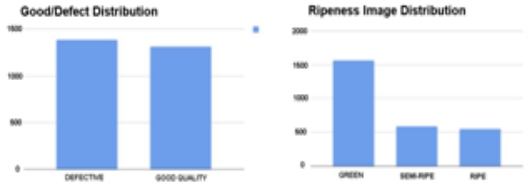

Fig. 6.Mango dataset distribution for quality and ripeness tasks.

B. Mango Dataset

Before training the deep artificial neural networks utilized in this study, a dataset composed of images of mango in different quality and states of ripeness was gathered. The quality were divided into two categories: defective and good quality (Fig. 8), while the states were separated into three classes: unripe (green), partially ripe (semi-ripe), and ripe (Fig. 7). The researchers acquired 2800 mango images derived from 270 distinct mangoes of varying state of ripeness and quality from local fruit market. The dataset distribution is shown in Fig. 6. The whole dataset was then divided into training and validation. The images were obtained from video shots of the mangoes rotated throughout the whole video to expose all areas of the mango to the camera. Each video lasts from five seconds to ten seconds. The researchers derived 10-20 images from each video.

As the dataset contains only relatively few images and since these images were obtained from videos, several images would display similar data thus making the neural network algorithm to simply "memorize" the data rather than completely learn the patterns appropriate to that classification which would lead to overfitting. Thus, data augmentation was employed to supplement the lack of images in the dataset by modifying several parameters of the images such as rotation, width shifting, height shifting, zoom and channel shifting, as shown in Fig. 5. Modifying the zoom, width and height condition of the images by cropping and resizing avoids the distortion of the mango within the frame of the picture but allows the creation of new images that contain unique data points for the neural network to learn.

C. Deep Learning-based Detection/Classification

The proposed system involves several parts working together as a whole. Each sub-module of the pipeline are distinct and their performance can be evaluated independently.

As shown in Fig. 9, a mango image is captured

Fig. 7.Sample unripe, partially ripe, and ripe mangoes (from left to right)

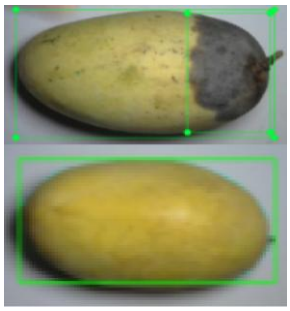

Fig. 8. Sample defective and good quality mangoes in dataset. Note: All Ground truth annotations for the mango dataset were performed using LabelImg graphical image annotation tool (github.com/tzutalin/labelImg).

and taken as input to the mango detection sub-module which is Single Shot MultiBox Detector (SSD) [19] – MobileNet [20] and then, the proposed multitask learning convolutional neural network (MTL-CNN)[21] accepts the cropped mango image for ripeness and quality classification. The original SSD utilizes VGG [22], however, initial tests showed that this is slow and memory intensive in embedded system environment such as in the proposed system, thus, a faster feature extractor, e.g. MobileNet, was considered for the proposed system. SSD-MobileNet model was chosen because despite having the lowest score among the COCO [23] models, it was shown to have achieved the fastest speed among the models [24].

TABLE I. SSD-MOBILENET MANGO DETECTION TRAINING CONFIGURATION

Parameter	Description
Model	SSD mobile-net v1 COCO
Training Dataset	2600
Evaluation Dataset	200
Image Input Resize	300 x 300
Batch Sizes	24
Training Steps	27,000
Learning Rate	0.004
Decay Steps	1000
Decay Rate	0.95
Image Augmentations	Random Horizontal Flip and Random cropping with fixed aspect ratio

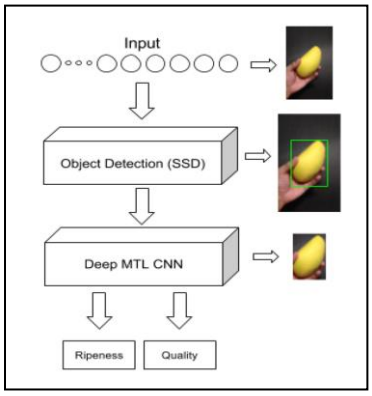

Fig. 9.Confusion Matrix using the Filipino Database

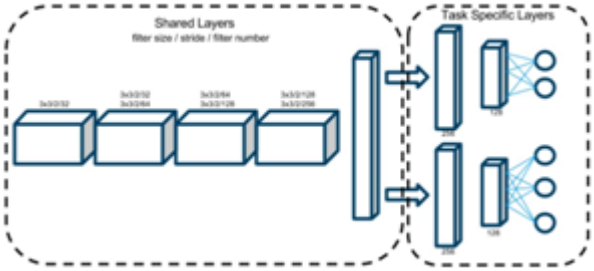

Fig. 10. Proposed Mango MTL-CNN Classification model

After model selection, the authors utilized the Google Cloud platform to train the model. After having uploaded the model and the dataset into cloud storage buckets, the training parameters of the model was configured as shown in Table I. It took many iterations and trial-and-error in configuring an optimal learning rate as having too high of a learning does not allow the model to converge to a minima, and having too low of a learning rate takes the much longer to train. Basic image augmentations are included to increase variety in training data.

After the image is passed through the object detection process and the mango is successfully localized. The mango image is then cropped and fed into the mango classification submodule (Fig. 9). Initially, the authors attempted to train on the MobileNet model, but it resulted to a low accuracy of below 90 percent across all classes. Thus, the researchers propose a VGG inspired MTL-CNN model for classifying the mango's ripeness and basic quality, as shown in Fig. 10. This design uses the MTL concept of hard parameter sharing, where there are no soft weight sharing in between the task specific layers. Each task specific layer will optimize its weights to classify their target class, which is ripeness or quality. Calculated softmax loss for each task specific layer is added together to form a combined loss which is where the models optimizer will be looking to minimize. The training parameters for the proposed MTL-CNN model is shown in Table II.

TABLE II. PROPOSED MANGO CLASSIFICATION TRAINING CONFIGURATION

Parameter	Description
Input Shape	1 x 50 x 50 x 3
Number of images per epoch	2700
Learning Rate	0.001
Image Input Resize	24
Batch Sizes	24
Optimizer	Adam Optimizer
Image augmentations	Shearing, Cropping, Tilting

D. Evaluation Metric

For the mango detection submodule, 200 images were withheld from training data as testing data. The system was evaluated based on its mean Average Precision (mAP), as in (1). The mAP metric is a common evaluation used in computer vision and object detection task (supported by Google's object detection API).

$$mAP = \frac{1}{n}\sum_{n} AP_n \qquad (1)$$

where AP represents the Average Precision value for a given topic (in information retrieval) from the evaluation set of n topics. For object detection, AP is defined in the popular PASCAL visual object classes challenge [25].

For the mango classification problem, one hundred distinct mango images were used to test the accuracy of ripeness/ quality classification pipeline. This made the system unbiased test on a completely distinct data which demonstrated the model's ability to generalize and classify. Accuracy can be directly derived from the confusion matrix, which will be presented in the following section.

Finally, to test the program execution time, we set a timing code to record the start of each loop when classifying images. This allowed mean time calculation for execution of the SSD-MobileNet and MTL-CNN models.

III. RESULTS AND DISCUSSION

A. Mango Detection Accuracy Results

The mango detection model (SSD MobileNet) trained in Google Cloud platform was evaluated on 200 mango images, and achieved a total precision score of 0.92 and a mean average precision (mAP) of over 0.8 in the final checkpoint, as shown in Fig. 11.

Fig. 11. **SSD MobileNet Mango detection performance evaluated on 200 test images (2700 steps) .**

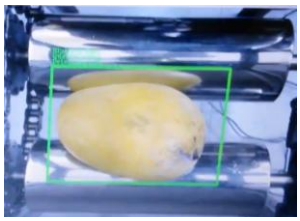

Fig. 12. **Sample Mango Detection in the Proposed System**

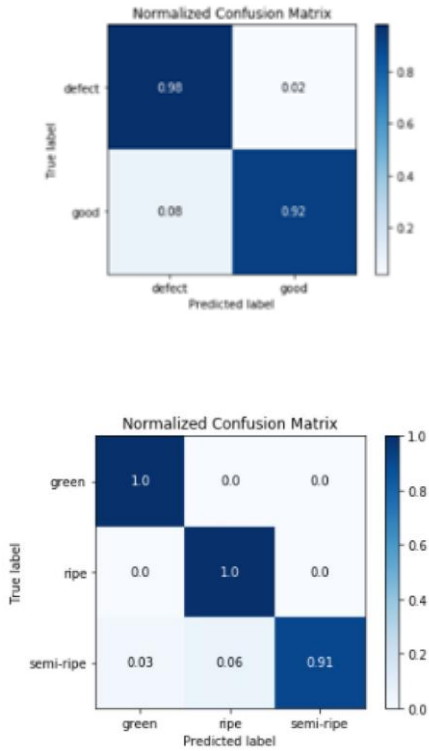

Fig. 13. **Confusion Matrices for the Mango classification task. Top: Mango quality; Bottom: Mango Ripeness**

B. Mango Classification Accuracy Results

Evaluating the mango classification task involves testing the model on data that it has not seen previously, i.e. test set of 100 images. Of the 100 images, 35 show a green mango, 33 show a semi-ripe mango, and 32 show a ripe mango. In terms of presence of defect, 40 images show a mango with defects on its skin while 60 images show a mango with smooth skin.

The confusion matrices in Fig. 13 are the result of applying the classification model on the testing dataset. It can be seen that the model was able to classify above 90 percent on each of the two factors of sorting. The model has correctly identified all green and ripe mangoes, while the accuracy for semi-ripe is at 91 percent. In addition, 98 percent of the defective mangoes and 92

percent of the good mangoes were correctly identified. Overall, the model has succeeded in accurately classifying ripeness and quality.

C. Latency/Processing Time

The average run time of SSD-MobileNet mango detection is 0.13 s while the MTL-CNN mango classification model has an average of 0.0049 s for 100 iterations.

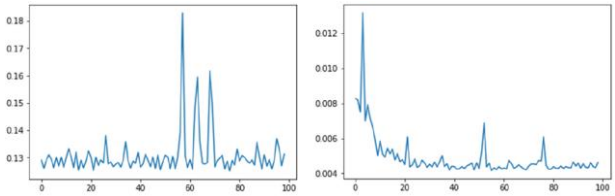

Fig. 14. **SSD-MobileNet) (left) and MTL-CNN (right) execution times in s across 100 test images**

IV. CONCLUSION

Overall, the results demonstrated the feasibility of our proposed embedded system for image-based Carabao mango sorting using deep learning techniques, particularly SSD-MobileNet for mango detection, and Multi-Task Learning Convolutional Neural Network (MTL-CNN) for classification of ripeness and basic quality. Furthermore, the average run-time latency is acceptable even with the embedded environment computational limitations. Therefore, future research may focus on the application of the proposed embedded system in actual mango production and packaging.

REFERENCES

1. J.K. Brecht, and E.M. Yahia, “Harvesting and Postharvest Technology of Mango. Handbook of Mango Fruit: Production, Postharvest Science, Processing Technology and Nutrition”, John Wiley & Sons Ltd., pp. 105, 2017.
2. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, 521(7553), pp.436, 2015.
3. Y. Bengio, I. Goodfellow, and A. Courville, “Deep learning,” vol. 1, MIT press, 2017.
4. A. Kamilaris, and F.X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and electronics in agriculture*, 147, pp.70-90, 2018.
5. R.C. Gustilo, E.P. Dadios, “Machine vision support system for monitoring water quality in a small scale tiger prawn aquaculture”, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 20(1), pp. 111-116, 2016.
6. R.C. Gustilo, E.P. Dadios, “ Behavioural response analysis using vision engineering (BRAVENet)”, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 21(2), pp. 211-220, 2017.
7. A.D.M. Africa, J.C.C.A. Aguilar, C.M.S. Lim, P.A.A. Pacheco, and S.E.C. Rodrin, "Automated aquaculture system that regulates Ph, temperature and ammonia," *In 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management* (HNICEM), pp. 1-6, IEEE, 2017.
8. A.D.M. Africa, J. Velasco, "Development of a urine strip analyzer using artificial neural network using an android phone," *ARPN Journal of Engineering and Applied Sciences*, 12, pp. 1706-1713, 2017.
9. P.J.M. Loresco, and A.D. Africa, "ECG Print-out Features Extraction Using Spatial-Oriented Image Processing Techniques," *Journal of Telecommunication, Electronic and Computer Engineering* (JTEC), 10(1-5), pp.15-20, 2018.
10. A.D.M. Africa, "A Rough Set Data model with presbyopic and hypermetrope parameters for eye vision diagnostics," *In 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, pp. 1-5, IEEE, 2015.
11. S.D.F. Hilado, E.P. Dadios, R.C. Gustilo, “Face detection using neural networks with skin segmentation”, Proceedings of the 2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems, CIS 2011, pp. 261-265, 2011.
12. S.G.E. Brucal, A.D.M. Africa, and E.P. Dadios, "Female Voice Recognition Using Artificial Neural Networks and MATLAB Voicebox Toolbox," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(1-4), pp.133-138, 2018.
13. L. Villanueva, R.C. Gustilo, “Artificial neural network based antenna sensitivity assignments for chaotic Internet Service Provider network architecture” *International Journal of Engineering and Technology* (UAE), 7(2), pp. 14-17, 2018.
14. J. Del Espiritu, G. Rolluqui, R.C. Gustilo, “Neural network based partial fingerprint recognition as support for forensics”, *8th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management*, HNICEM 2015.
15. S. Bargoti, and J. Underwood, “Deep fruit detection in orchards,” *In 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3626-3633, IEEE, 2017.
16. I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, “Deepfruits: A fruit detection system using deep neural networks,” *Sensors*, 16(8), pp. 1222, 2016.
17. M. Long, Z. Cao, J. Wang, and S.Y. Philip, “Learning multiple tasks with multilinear relationship networks,” *In Advances in neural information processing systems*, pp. 1594-1603, 2017.
18. C. Lee, S. Jung, K. Kim, and G.G. Lee, “Hybrid approach to robust dialog management using agenda and dialog examples,” *Computer Speech & Language*, 24(4), pp.609-631, 2010.
19. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A.C. Berg, “Ssd: Single shot multibox detector,” In *European conference on computer vision*, pp. 21-37, Springer, Cham, 2016.
20. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv* preprint arXiv:1704.04861.
21. S. Ruder, “An overview of multi-task learning in deep neural networks,” arXiv preprint arXiv:1706.05098, 2017.
22. K. Simonyan, and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
23. T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, "Microsoft coco: Common objects in context," In European conference on computer vision, pp. 740-755, Springer, Cham, 2014.
24. V. Rathod and N. Wu, "TensorFlow detection model zoo", 2017.
25. M. Everingham, L. Van Gool, C.K. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes challenge 2007 (VOC2007) results”, 2007.

AUTHORS PROFILE

Ryan Joshua Liwag received the Bachelor of Science degree in Computer Engineering from De La Salle University (DLSU), Manila, Philippines, in 2018. He is presently working as analytics advisory senior analyst at Accenture in National Capital Region, Philippines.

Kevin Jeff Cepria received the Bachelor of Science degree in Computer Engineering from De La Salle University (DLSU), Manila, Philippines, in 2018. He is presently working as software engineer at SCI Ventures Inc. in National Capital Region, Philippines.

Anfernee Rapio received the Bachelor of Science degree in Computer Engineering from De La Salle University (DLSU), Manila, Philippines, in 2018. He is presently working as React Developer at White Cloak Technologies, Inc. in National Capital Region, Philippines.

Karlos Leo Castillo received the Bachelor of Science degree in Computer Engineering from De La Salle University (DLSU), Manila, Philippines, in 2018. He is presently working as package solution consultant at IBM Solutions Delivery Co., Ltd. in National Capital Region, Philippines.

Melvin Cabatuan, Ph.D. received the Bachelor of Science degree (cum laude) in Electronics and Communications engineering from Cebu Institute of Technology, Cebu, Philippines, in 2004; Master of Engineering from Nara Institute of Science and Technology (NAIST), Ikoma, Nara, Japan, in 2010; and Ph.D. in Electronics and Communications engineering from De La Salle University (DLSU), Manila, Philippines, in 2016. He joined the Electronics & Communications Engineering Department of DLSU in 2011, where he is currently an Associate Professor. His research interest involves machine vision, health informatics, and machine/deep learning applications to various engineering challenges.