ISSN 2278-3091
Volume 9, No.1.3, 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse2391.32020.pdf
https://doi.org/10.30534/ijatcse/2020/2391.32020

Java Plum Leaf Classification Using a Mobile Application with Inception Network

James Arnold E. Nogra
Cebu Institute of Technology – University, jamesnogra@gmail.com

ABSTRACT

In the Philippines, the Java Plum leaf is sometimes used as an alternative to the paper used to cover tobacco cigarettes. But, not all leaves can be used as a tobacco roll cover. Currently, human experts, usually in their late 50's and 60's can do the Java Plum leaf classification task with high accuracy. With the help of an inception-based network, this leaf classification task would be easier and faster. This is true not only to experts but also to workers who have started to learn the leaf classification. The study tested four different inception models with a varying number of channels. The final inception model chosen that has a high accuracy rate but has fewer parameters is the model with 128×128×64 dimensions after the first inception module, 64×64×128 dimensions after the second inception module, 32×32×256 dimensions after the third inception module, and 1×1×256 dimensions after the first fully connected layer. This inception network has a 94.09% accuracy using the training data and 90.52% accuracy using the validation data. This model is then used to create a mobile app that classifies a Java Plum leaf into three categories. The app is then used by the people who are doing this leaf classification manually. After a few months of using it, the users of the app were asked to answer five usability questionnaires. The average score of the mobile app obtained from the usability questionnaire is 4.93 out of 5. This signifies that the mobile app helped the users in the Java Plum leaf classification.

Key words: Convolutional Neural Network, Inception Network, Java Plum, Leaf Classification, Mobile App

1. INTRODUCTION

1.1 Background of the Study

Java Plum (Eugenia Jambolana) is a tree that is usually found in Southeast Asia. This tree bears violet-colored fruits that are edible. The seed extract of this tree is also a natural antioxidant [1]. Aside from its sweet fruits, the leaves of this tree are sometimes used as an alternative to the paper used to roll tobacco. In the Philippines, tobacco rolled in Java Plum leaves is a commodity. Local markets are selling this kind of alternative cigarette usually for a lower price than commercial ones. An expert in the leaf classification is needed to get the appropriate leaf for the cigarette roll. A leaf that is too dry might burn quickly and a leaf that is too green or fresh might not burn. These human experts in leaf classification are of older age. The younger people doing the same task doesn't have the same accuracy as the ones who are doing this for many years.

For leaf classification problems, neural networks are one of the best solutions out there [2]. Because there are thousands of types of plants and trees, classification some of it usually needs automation. Studies in leaf classification most of the time use a convolutional neural network (CNN) because this type of neural network detects features rather than processing the entire image [3]. CNN has been revered as the standard for image classification because of its feature extraction capabilities. However, CNN has a downside in terms of the saliency of the object being classified. A full image of a dog and a dog in the background will appear different and will be classified as two different classes [4]. This is because the same filter size is applied in every convolution layer. A similar study that classifies the Java Plum leaf using the Inception Model V3 achieved a 91.2% accuracy [5].

The issue with CNN's with the same filter size applied in each layer can be solved by an inception network. The Inception V3 is a widely recognized image recognition model with an accuracy of 78% using the ImageNet dataset [6]. This image recognition model has been widely used in many image recognition or pattern analysis. When the trained model is exported or saved, the file size using this model is very large. There are other versions of inception networks but the Inception V3 is one of the most popular.

Most of the image classification models run on a console or as a backend so to make the classification work with a user-friendly interface, an appropriate frontend must be made. Examples of a frontend that are commonly used by these models are web or mobile. Mobile apps that have machine learning backends are gaining popularity nowadays. These machine learning apps include image classification that runs under CNN models or Inception models. Mobile apps, especially in the Android PlayStore and Apple AppStore, allows users to review apps. These reviews are crucial in the improvement or future updates of the app. Sometimes, reviews are even used to determine what features to add in the next release.

1.2 Objectives of the Study

The main goal of this study is to automate the classification of Eugenia Jambolana (Java Plum) leaves using a mobile app. The mobile app must also be easy to use to the users who are doing this leaf classification manually. The mobile app must be powered by an inception network so that it can classify varying sizes of leaves. For the backend that processes the classification, the inception model must be able to classify the leaves with accuracy higher than 80%. Before the mobile app can be developed, the best and most efficient inception network must be determined through a series of tests.

1.3 Significance of the Study

Classifying Eugenia Jambolana (Java Plum) leaf is crucial to small business owners who process these leaves for cigarette-like products. The leaf of the Java Plum is a cheap alternative to the paper roll used to cover tobacco leaves. This is the reason why a few small and family-owned businesses are treating these leaves as a commodity. Selecting the best leaf to be used as a cover roll alternative can be hard especially to workers who are new to the classification task. With the help of a mobile app and machine learning, classification can be automated or can be done by anyone.

2. REVIEW OF RELATED LITERATURE

Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) are popular deep learning image classifiers that are widely used. These neural network models achieve high accuracy in image classification or recognition. LSTM networks are just a modified version of a Recurrent Neural Network (RNN). An example of an LSTM network used for image classification is the one used in Baybáyin handwriting recognition [7]. This LSTM model achieved an accuracy of 92.9% accuracy using 9700 handwritten images. CNN is far more widely used in image classification problems. This is because this type of model looks for patterns in images [8]. A combination of CNN and RNN is also used for image recognition with higher accuracy. In one study of the combination of these two neural networks, using the CIFAR-10 dataset, this neural network scored higher accuracy than traditional CNNs [9].

Many of the researches on image classification revolve on deep neural networks but there are other techniques who can do this task with high accuracy. These techniques include K-nearest neighbors (KNN) and Support Vector Machine (SVM). One of these techniques was used to provide better tools and discriminant power for utilizing functional imaging in clinical prognosis [10]. Even though neural networks are dominating the research ground in image classification or recognition, these classical techniques can still compete in terms of accuracy.

A similar study was done to classify the Java Plum leaves into three classes. The model that was used in the Inception V3. This model got a 92.9% accuracy in terms of validation data accuracy [5]. It got a high accuracy because the photos used to train the model are taken in a controlled environment. The images that were used to train the CNN model are almost of the same size and were taken at the same angle. In the real world, these leaf images would have a different size in a photo and are taken at a different angle. For this kind of data set for training, an inception model would be more suitable. An inception model will try to process the image using different sizes of filters during convolution and will concatenate the output [11].

Because neural networks are lightweight in terms of file size, many mobile applications are using this technique as a feature. One such application is sketching an image and then the neural network will return a list of images saved in the device resembling the sketch [12]. Currently, mobile phones, even the ones that are affordable are powerful enough to do some simple tasks based on neural networks [13].

3. METHODOLOGY

Before the mobile app "Lomboy Leaf Classifier" can be developed, an inception network must be made first. There are numerous open-source convolutional neural networks and inception networks such as Inception V3, AlexNet, ResNet, and VGGNet but this study used a custom one. One reason is that some of these models are large (in terms of output file size and the number of hidden units) and are not suitable for mobile computing. Another reason is that the model only classifies three types of leaves and a smaller network would be more suitable.

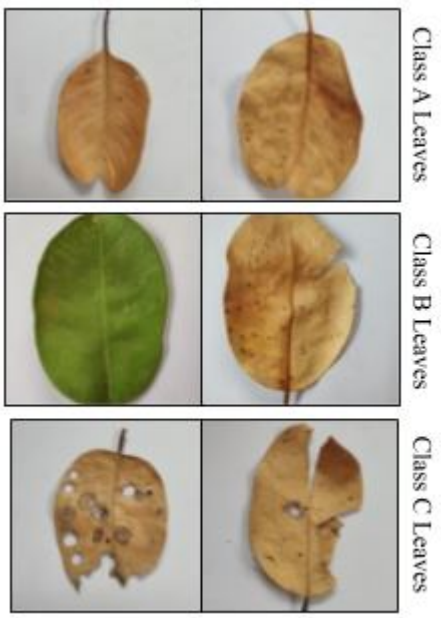

Figure 1: Sample images of the classification of the Java Plum Leaf.

The small business owner who specializes in the Java Plum leaf classification states that they classify the leaves into three classes as shown in Figure 1. Class A are leaves that are ready to be used as a cigarette roll cover. Class B leaves

are leaves that need to be further processed in order to be accepted or used. These processes include drying the leaf more or flattening it. After either of the processes, the leaf can still be rejected if it can't be classified as a Class A. For the Class C leaf, it's always rejected and thrown out. According to the business owner, only the Class C leaf is easy to identify while Class A and B need human experts.

For training the inception network, there were a total of 2945 Java Plum leaf images. Class A leaves have 1022 images, Class B leaves have 970 images, while Class C leaves have 953 images. When training the neural network model, of the 2945 leaf images, 80% (2356) of which was used for testing the model while 20% (589) was used to validate the model during the training process. All of the images were taken using a mobile phone. The mobile phones used to take the pictures of the leaves are Xiaomi Mi Mix 2S, Oukitel K10000, and LG V30.

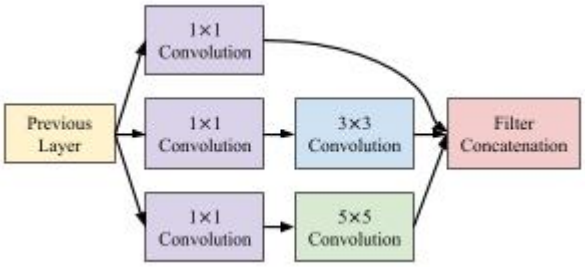

Figure 2: Inception module used in the leaf classification.

Shown in Figure 2, the inception module used implements dimension reduction to reduce the number of feature maps while retaining salient features [14]. Without the 1x1 convolutions before the 3x3 and 5x5 convolution, the number of feature maps would greatly increase.

Table 1: All of the inception networks that were tested for the leaf classification.

Model	Inception Module 1	Inception Module 2	Inception Module 3	Fully Connected 1
A	128 × 128 × 32	64 × 64 × 64	32 × 32 × 128	1 × 1 × 128
B	128 × 128 × 64	64 × 64 × 128	32 × 32 × 256	1 × 1 × 256
C	128 × 128 × 128	64 × 64 × 256	32 × 32 × 512	1 × 1 × 512
D	128 × 128 × 256	64 × 64 × 512	32 × 32 × 1024	1 × 1 × 1024

There were a total of 4 inception networks that were tested for this classification as shown in Table 1. The final model that will be used for the mobile app backend will be based on how well the model classifies the leaves. All of the models have three inception modules and two fully connected layers. The differences between the models tested are the number of channels they have in the inception modules and the first fully connected layer.

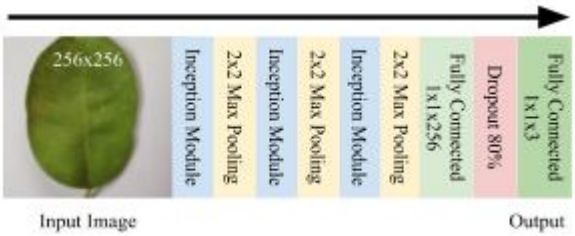

Figure 3: The whole neural network model which uses three inception modules, three max pooling layers, and two fully connected layers.

The entire neural network model shown in Figure 3 is composed of three inception modules. Every after the inception module is a max pooling layer with a 2x2 filter size. After the max pooling layers, the dimension of the image is halved. Also, in the first inception module, the number of channels is 64, on the second inception module, the number of channels increases to 128, and on the third inception module, the number of channels is 256. The first fully connected layer after the third inception module has dimensions of 1x1x256. The second fully connected layer has a dimension of 1x1x3 which equates to the three classes of the Java Plum leaf. For all of the convolution operations and the first fully connected layer, the activation function used is the Rectified linear unit (ReLU) shown in Equation (1). After the first fully connected layer, a dropout layer with 80% retention of units is implemented. The activation function used for the output layer is the Softmax function. This function is used to calculate the probability distribution from a vector of real numbers [15]. The loss function used to calculate the error of the training examples is the Categorical Cross-Entropy. The optimizer used in training the neural network is the RMSprop Optimizer which divides the learning rate by an exponentially decaying average of squared gradients [16]. All of the tested models use the Tensorflow framework like in convolution, max pooling, dropout, and fully connected operations. The research methods of this study have been influenced by [17], [18], and [19].

$$R(z) = max(0, z) \quad (1)$$

Equation (1) is the general formula of the Rectified linear unit (ReLU) where z is the output.

After selecting the most accurate and efficient model, a mobile app was developed. The app was coded using Phonegap Cordova using HTML, CSS, and JavaScript. The inception model is not stored locally on the device and is stored in a remote server. The database used to store the classified leaves is MySQL. The app is named "Lomboy Leaf Classifier" and is not available in both Android PlayStore and iOS AppStore. The mobile app only runs on Android devices due to budget limitations during development. The app is not yet publicly available because this app is custom made for a specific client. The client also provided the sample training images that's why they can solely own the mobile app. All of the classifications of the Java Plum leaves were based on how their experts classified it.

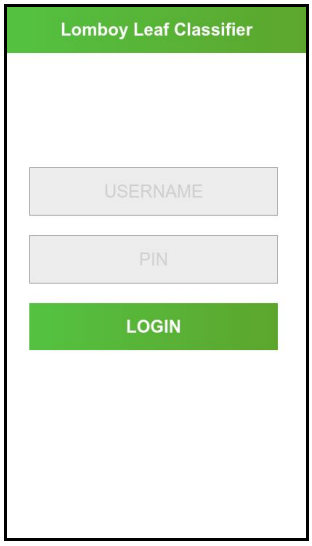

Figure 4: Login page of the mobile app.

Users of the app need to log in as shown in Figure 4 in order for the classified leaves to be tagged as theirs. The login screen also prevents outsiders from using the app. The main users of this app is a small business that specializes in making custom cigarettes. One of their products is cigarettes that have a Java Plum leaf cover roll instead of paper.

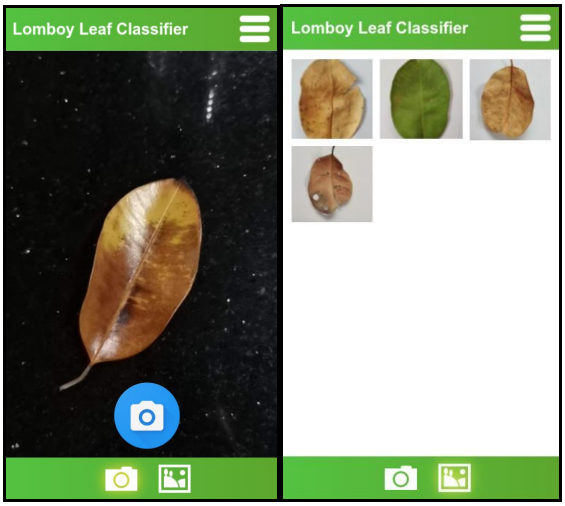

Figure 5: Users of the app can classify Java Plum leaves by either taking photos or browsing images in their device gallery.

After logging in, users will have an option to take a picture of a leaf or browse from the gallery of their device to select which leaf to be classified as shown in Figure 5. When taking the image, the native camera resolution doesn't matter because all of the images are resized to 256×256 before being processed by the inception network. Same for images that were already taken and were imported from the device gallery.

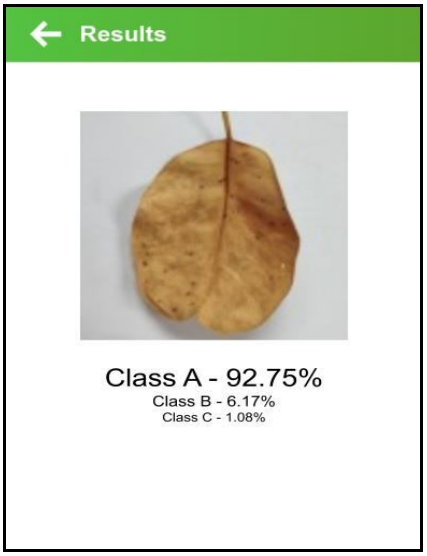

Figure 6: Screenshot of the mobile app that shows the result of the image being classified.

After selecting or taking a photo of a leaf image, the mobile app will redirect to the results screen as shown in Figure 6. There will be three results based on the Softmax results of the last fully connected layer of the neural network. The three results correspond to the three types of leaves of the Java Plum tree.

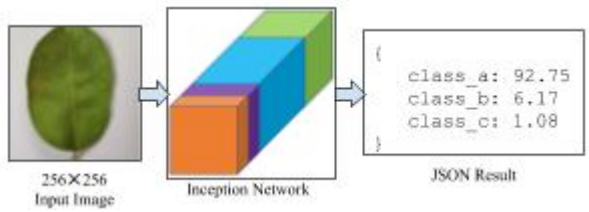

Figure 7: Schematic diagram of the inception network used as a backend service.

The mobile app was developed in a laptop with 16 gigabytes of RAM, 256 gigabytes of storage space, and an Intel Core i7-7700HQ processor. The same device is used for all of the training of the inception models. The backend service as shown in Figure 7 is running in a remote server with resources of a quad-core CPU with 2.3GHz and 2 gigabytes of memory. The backend service takes an input image and resize it to 256×256×3 and then outputs a JavaScript Object Notation (JSON) for the classification results.

Table 2: Usability questionnaire was given to the users of the "lomboy leaf classifier" mobile app.

Measurement Item Number	Measurement
1	The mobile app is easy to navigate.
2	Leaf classification is faster using the app.
3	Screen layouts and icons help.
4	The results of the classification are accurate.
5	The leaf classification is fast enough.

Two months after the business owner started using the mobile app, a usability questionnaire shown in Table 2 was given to the users. For their business, there were 8 users that were actively using the mobile app. These 8 users were given the usability questionnaires. The users can answer from 1 to 5 where 5 is the highest or strongly agree while 1 strongly disagrees.

4. RESULTS AND DISCUSSION

A total of five training per inception model was done. The average of the accuracy of the training for the five tests was recorded. These averages were the determining factor whether the inception model will be used as the backend of the "Lomboy Leaf Classifier" mobile app.

Table 3: The average training and validation accuracies of the four models that were tested after training it five times.

Model	Average Training Data Accuracy	Average Validation Data Accuracy
A	77.14	74.89
B	94.09	90.52
C	95.82	91.05
D	94.69	89.78

In Table 3, Model B, C, and D have almost the same average accuracy rates from the five tests that were done. The accuracy of Model A is not high enough for the leaf classification task. The final model that was selected was Model B. Model B has fewer parameters than Model C and D. This makes the leaf classification faster and more efficient. Model B takes an input of 256×256×3. After the first inception module and max pooling layer, the dimension becomes 128×128×64. Then after the 2nd inception module and max pooling layer, the dimension output is 64×64×128. After the third and final inception module and max pooling layer, the dimension output is 32×32×256. Then after this, the first fully connected layer transforms the output into 1×1×256. At the last fully connected layer, the output becomes 1×1×3.

Table 4: The average training results of model b.

Epochs	Average Training Data Accuracy	Average Validation Data Accuracy
25	45.90	47.24
50	62.57	60.15
75	88.01	83.68
100	94.09	90.52

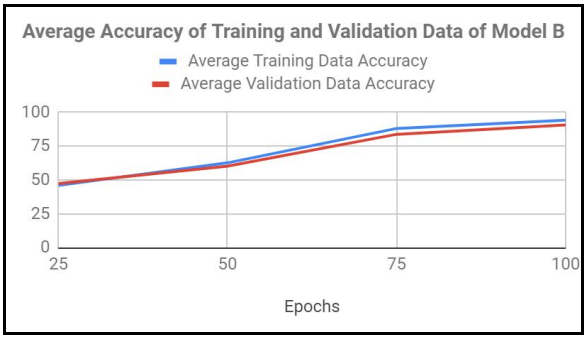

Figure 8: Progress of the averages of the accuracy rates using the training and validation data for 100 epochs.

In Table 4 and Figure 8, Model B that was tested was able to reach an accuracy rate of 90.52 after 100 epochs. Because of this, Model B will be used in the backend of the "Lomboy Leaf Classifier" mobile app. Model B has also the lowest number of parameters after Model A but has one of the highest accuracy rates for both validation and training data.

Table 5: Responses from the 8 users of the mobile app on the usability questionnaire.

Measurement Item Number	Average Score
1	5.00
2	4.88
3	5.00
4	4.75
5	5.00
Average	**4.93**

From the responses of the users of the mobile app, it can be inferred that the mobile app helped them in their task as shown in Table 5. They all strongly agreed that the mobile app is easy to use, the screen layouts and icons help them, and the leaf classification itself is fast enough. The usability questionnaire that got the lowest score is the question about the accuracy of the classification which only got 4.75 out of 5.0. The average usability score of the app for all the 5 questions across all the users is 4.93 out of 5.0. This means that the mobile app is useful for them and for the business.

5. CONCLUSION

After designing four inception networks with different number of channels for the Java Plum leaf classification, it can be concluded that the most accurate and most efficient, having the least parameters, is the model that has 128×128×64 dimensions after the first inception module, 64×64×128 dimensions after the second inception module, 32×32×256 dimensions after the third inception module, and 1×1×256 dimensions after the first fully connected layer. This model was able to classify 2945 leaf images with an accuracy rate of 94.69% using the training data and 90.52% using the validation data. This model was then used as a backend service for a mobile app that classifies Java Plum leaves. The users of the mobile app strongly agree that the app helped them in the leaf classification task. Using a 5-

question usability tool, the app managed to score an average of 4.93 out of 5.0. This signifies that the users of the app find the tool very helpful in their tasks.

REFERENCES

1. R. Rohadi et al., **Methanolic extract of Java Plum (Syzygium cumini Linn.) seeds as a natural antioxidant on lipid oxidation of oil-in-water emulsions**, *International Food Research Journal*. 24. 1636-1643, 2017
2. C.S.Sumathi & A.V.Senthil Kumar, **Neural Network based Plant Identification using Leaf Characteristics Fusion**, *International Journal of Computer Applications*, (0975 – 8887) Volume 89 – No.5, March 2014 https://doi.org/10.5120/15499-4141
3. D. Jaswal, S. Vishvanathan, S. Kp, **Image Classification Using Convolutional Neural Networks**, *International Journal of Scientific and Engineering Research*, 5(6):1661-1668, June 2014
4. V. Srivastava, & B. Biswas, **CNN-based salient features in HSI image semantic target prediction**, *Connection Science*, pp. 1-19, (2019), 10.1080/09540091.2019.1650330.
5. J. Nogra, **Classifying Java plum (Eugenia jambolana) Leaf for Tobacco Cigarette Wrapper using Convolutional Neural Network**, *Journal of Science, Engineering and Technology*, 6:96-103, 2018
6. C. Szegedy et al., **Going deeper with convolutions**, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015 https://doi.org/10.1109/CVPR.2015.7298594
7. J. Nogra, C. Sta Romana, E. Maravillas, **LSTM Neural Networks for Baybáyin Handwriting Recognition**, *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, Singapore, 2019, pp. 62-66
8. W. Rawat & Z. Wang, **Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review**, *Neural Computation 2017* 29:9, 2352-2449, September 2017
9. [9] Q. Yin, R. Zhang, X. Shao, **CNN and RNN mixed model for image classification**, *MATEC Web of Conferences* 277(4):02001, January 2019
10. E. Naqa, et al, **Exploring feature-based approaches in PET images for predicting cancer treatment outcomes**, *Pattern recognition*, 42(6), 1162–1171. doi:10.1016/j.patcog.2008.08.011, 2009
11. M. Haris, M. Widyanto, and H. Nobuhara, **Inception learning super-resolution**, Appl. Opt. 56, 6043-6048, 2017
12. N. Hewahi & S. Rashid, **Image Finder Mobile Application Based on Neural Networks**, *BRAIN: Broad Research in Artificial Intelligence and Neuroscience*. 8. 5, 2017
13. Y. Deng, **Deep Learning on Mobile Devices-A Review**, 2019
14. C. Szegedy, **Going deeper with convolutions**, arXiv preprint arXiv: 1409.4842 2014
15. C. Nwankpa et al., **Activation Functions: Comparison of Trends in Practice and Research for Deep Learning**, arXiv:1811.03378v1 [cs.LG] 8, November 2018
16. S. Ruder, **An overview of gradient descent optimization algorithms∗**, arXiv:1609.04747v2 [cs.LG] 15, June 2017
17. G. Alcober, T. Revano, & M. Garcia, **E-Safety in the Use of Social Networking Application**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 1.2 2020, https://doi.org/10.30534/ijatcse/2020/1291.22020
18. R. Dellosa, **An Efficient Position Estimation of Indoor Positioning System Based on Dynamic Time Warping**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 1.2 2020, https://doi.org/10.30534/ijatcse/2020/0491.22020
19. J. Victoriano & L. Lacatan, **A Geospatial Analysis and Kernel Density Estimation of River Quality Parameter in Bulacan, Philippines**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 1.2 2020, https://doi.org/10.30534/ijatcse/2020/1191.22020