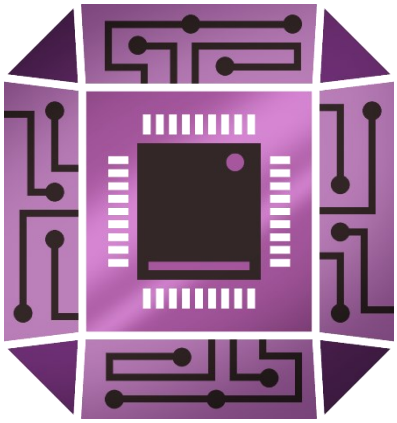

CLUSTER SIX

COMPUTER SCIENCE

The color violet is synonymous with power, and the gem design is inspired by printed circuit boards often found in electronic devices, wherein a microchip is integrated into the middle. Hence, the gem is representative of a computer, in that it requires a power source and circuitry to function. Through scientific analysis of data done mainly through the power of computers, studies under this cluster aim to derive information that transcends electronic applications.

These studies fall under the Industry, Energy, and Emerging Technology (IEET) Research Development Agenda, in line with goals to increase state-of-the-art information and communications technology (ICT) research and the utilization of computer-based systems in various industries and scientific fields.

BASED ON: Harmonized National Research and Development Agenda (HNRDA)

Publiscience Vol. 4 Issue 1

Developing a neural network that uses satellite imagery to estimate carbon dioxide emissions in the Philippines

EWEMIZ C. INSIGNE, KIRSTEN DIANNE G. DELMO, NICO ANDREI R. SERRATO, and GERALD U. SALAZAR

Philippine Science High School Western Visayas Campus - Department of Science and Technology (DOST-PSHSWVC), Brgy. Bito-on, Jaro, Iloilo City 5000, Philippines

Article Info

Submitted: May 14, 2021
Approved: Jul 21, 2021
Published: Aug 30, 2021

Keywords:
artificial intelligence
carbon dioxide
satellite imagery
neural network
data science

Abstract

In recent years, satellite imagery has become a popular subject of studies involving artificial intelligence. AI and satellite imagery were used together to estimate real-life variables in various fields, such as predicting poverty levels, detecting oil spills, and analyzing coal mine areas. With the rising importance of climate change, this study aimed to develop a convolutional neural network trained with daytime and nighttime data through a transfer learning approach to predict carbon dioxide emissions in the Philippines. The developed scripts were adapted and based on code from public GitHub repositories and existing research. The trained model was concluded to have high predictive power ($r^2 = 0.74$), and can be used as a starting point for more fully developed software and models that can serve as an alternative method of collecting CO_2 emissions data.

Introduction. - Data science uses statistics, artificial intelligence (AI), and available data from various sources to process into valuable information [1]. AI in data science looks for patterns, predicts outcomes, and offers evidence-based information using the data that it was given [2]. More and more AI-based tools and technologies are beginning to emerge as datasets begin to be more readily available online and in other publicly-available sources. A specific type of AI, the convolutional neural network (CNN), is mainly used for the analysis of images. It takes an input and extracts features from the image to be analyzed by neurons – an interconnected system of cells that takes input and multiplies it by a specific weight that constantly changes until the CNN can correctly predict an outcome.

Recent studies were able to use satellite imagery to predict poverty levels in an area [3,4,5]. Other studies utilized satellite imagery for purposes such as detecting oil spills, analyzing coal mine areas, and detecting clouds [6,7,8]. The effectiveness of combining satellite imagery and AI and show how it has many potential uses in various fields.

With the increase of CO_2 concentration in the atmosphere, the need for "the quantification of the spatial distribution of CO_2 in the atmosphere" was felt by researchers in the last decade [9]. The main focus of currently available literature on estimating CO_2 emissions focus on mathematical models that don't utilize AI. [9,10]. The correlation of nightlights and CO_2 emissions using satellite imagery has also been proven by studies, where highly-lit points were characterized to be highly urbanized and were seen to have high CO_2 emissions [9,11]. While using AI (specifically neural networks) to estimate CO_2 emissions is not an uncommon method [12], this study differs in that it develops a CNN that estimates CO_2 emissions in specific 1km x 1km areas within the Philippines, and utilizes a transfer learning approach. Transfer learning is a method that utilizes the learning extracted from one problem or variable (such as nightlights data) and uses it to make better predictions for related data.

The objective of this paper is to see how effective the use of AI with satellite imagery can be for the estimation of real-life variables, such as CO_2 emissions, in order to prove its versatility as a tool to be used in a wide range of applications in various fields. As a tool specifically used to estimate CO_2 emissions, it could serve as an alternative method of collecting such data. Thus, the goal of this study is to develop an AI—specifically a CNN—that uses satellite imagery to estimate CO_2 emissions in the Philippines. The specific objectives are enumerated below. They also serve as an outline for the succeeding Methods section.

(i) Design an algorithm for the CNN model;

(ii) Collect latest (at the time of development) satellite imagery and CO_2 emissions data;

(iii) Code the scripts for data preprocessing, CNN model training, and evaluating the model's estimations using Python 3.8;

(iv) Train the CNN using the collected data; and

How to cite this article:
CSE: Insigne EC, Delmo KDG, Serrato NAR, Salazar GU. 2021. Developing a neural network that uses satellite imagery to estimate carbon emissions in the Philippines. Publiscience. 4(1): 88–92.
APA: Insigne, E.C., Delmo, K.D.G., Serrato, N.A.R., & Salazar, G.U. (2021). Developing a neural network that uses satellite imagery to estimate carbon emissions in the Philippines. *Publiscience*, *4*(1), 88–92.

For supplementary data, contact: publiscience@wvc.pshs.edu.ph.

(v) Train a ridge regression model using the CNN's extracted features for evaluation and data visualization of results

Methods. - This study consisted of five phases: *Design*, *Data Collection* (for data pre-processing), *Development*, *Training*, and *Evaluation*. The *Design* phase was for the creation of the algorithm for the CNN model; *Data Collection* phase was for the collection of images and nightlight values for pre-processing used for training the developed program; *Development* phase was for the coding and debugging of the scripts; *Training* phase was for the tuning and fitting of the model, and; *Evaluation* was for the training of the ridge regression model for evaluation and data visualization of results.

Design phase. During the *Design* phase, a flowchart and the equivalent pseudocode of the program's process were made to be used as a basis for the creation of our actual program during the *Development* phase. The flowchart presented the various parts of the process in an human-readable format that can then be converted to code. This phase employed the use of open-source codes by Jean et al. [4] and Tingzon et al. [5] that were available on the authors' GitHub repositories.

Data Collection phase. In order to properly train the AI, certain data were gathered and pre-processed. This study used the Open-Data Inventory for Anthropogenic Carbon dioxide (ODIAC) of the National Institute for Environmental Studies (NIES) Japan for the year 2015 [13]. The data points for the Philippines were filtered out, as seen in Figure 1. Daytime satellite images were then collected through a script that downloads 1 km by 1 km Google Static Maps API images according to the given data points from the CO_2 emissions data. Nighttime lights data was collected from the Visible Infrared Imaging Radiometer Suite (VIIRS) Day/Night Band (DNB) data of the Earth Observation Group, Payne Institute for Public Policy [14].

Development phase. The method used in the developed scripts were adapted from the transfer learning approach used by Jean et al. [4] Daytime satellite image features were extracted by the CNN to estimate the CO_2 emission level in an area.

The scripts were developed based on the algorithm made in the design phase as well as open-source scripts and codes from similar studies publicly available on GitHub [15,16,17]. Python 3.8 was the primary programming language used for development, and the scripts were coded in JupyterLab 1.1.4 and JupyterNotebook 6.0.1.

The first three scripts were used for the pre-processing of data. The first script filtered out the CO_2 emission data points for the Philippines. The second script downloaded the daytime satellite images for each datapoint with a zoom level of 13, and the third script compiled the corresponding nighttime radiance for the coordinates of each CO_2 emission datapoint.

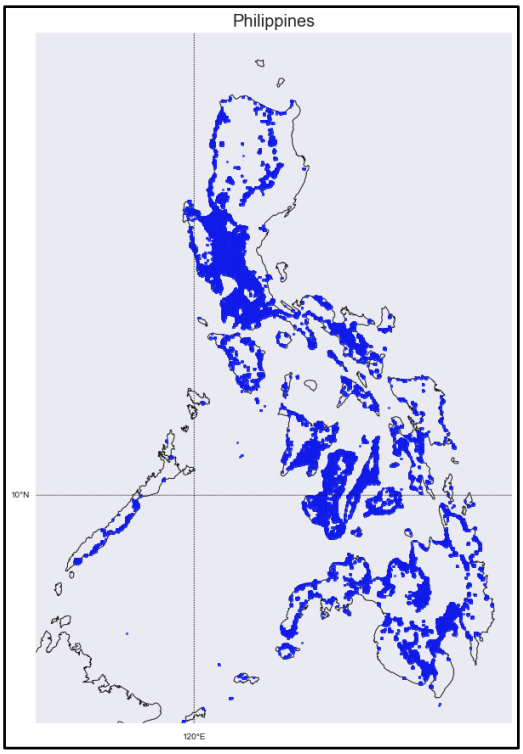

Figure 1. Philippine CO_2 emission data points. The data points for the Philippines were filtered out from the ODIAC dataset.

The first script also created a dataframe with the latitude and longitude values, the corresponding CO_2 emission. These values were then exported into a comma-separated values (.csv) file.

The second script downloads 400 px by 400 px Google Static Maps API images at a zoom level of 13 (corresponding to an estimated 1 km by 1 km area) for each data point. A total of 79942 unique images were downloaded.

The third script collected the corresponding nightlight radiance for each point in the dataset. The values were then appended to the data frame created in the first script.

The fourth script was for the training. In its initial stages, the dataset and the corresponding images were divided into two folders: 80% of the images were used for training and the remaining 20% were used for validation/testing of the model. Three nighttime light intensity classes were obtained by fitting a mixture of three Gaussian distributions to the relative frequencies of the nighttime light intensity values.

Training phase. The training phase aimed to train the CNN with images so that it could create its own model in order to analyze patterns and make accurate estimations close to the given CO_2 data.

In the first part of the training, we fine-tuned a pre-trained model, VGG F, to estimate nighttime light intensity at various locations given the corresponding daytime satellite images. This pre-trained model network is an eight-layer deep convolutional neural network (DCNN), which had been originally designed and trained for image classification on ImageNet [18].

The training method and code were heavily adapted from Mather's Predicting Poverty repository on GitHub [15] and the Pytorch CNN Training Method [19].

After training, the model was then tested by making estimations using the testing data set.

Evaluation phase. Once the CNN was trained, the extracted features were then used to train a ridge regression model. It specifically used k-fold cross-validation. Python was also used in data visualization, specifically for the creation of the graphs.

Data Analysis. For data visualization and analysis, this study used Python 3.8 and different libraries that were available and used for statistical calculations and data visualization such as NumPy, Pandas, Scikit-learn, and Matplotlib.

This analysis utilized a cross-validation technique, which is a data resampling method that assesses the generalization ability of a predictive model and prevents overfitting. The *k*-fold cross-validation technique is much less prone to selection bias compared to other cross-validation methods. In *k*-fold cross-validation, the process starts out by dividing the dataset into given k subsets and uses *k-1* subsets as the training sets while the remaining set serves as the testing set. This cross-validation method is then repeated *k* times, using different testing sets from the original *k* subsets each time. The *k* value for this study was *5*, similar to that of Jean et al. [4].

Safety Procedure. As this study focused on AI, all processes were done on a computer. There were no major ethical issues dealt with. The data used is publicly available and the source codes are open-access.

Results and Discussion. - The aim of this study is to develop a convolutional neural network to analyze CO_2 emissions in the Philippines using satellite imagery, and see how effective the use of AI with satellite imagery can be for the estimation of real-life variables (such as CO2 emissions). This was done by creating the algorithm for the program, collecting data that would be used for pre-processing and training, developing the scripts used for the program, and then evaluating and visualizing the results. The evaluation phase specifically employed k-fold cross validation, with various libraries in Python such as Matplotlib used for data visualization.

The current results, shown in Figure 2, showcase strong predictive power using the model trained on a dataset with 79942 data points. With a r^2 value of 0.74, this means that the trained model fits the data well, with 74% of the total variation being accounted for.

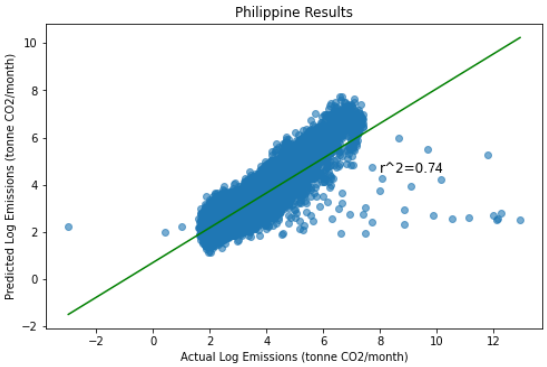

Figure 2. Philippine results. Predictions and reported r^2 value are from five-fold cross-validation. Green line shown is the line of best fit.

To assure the accuracy of this statistic, five random sampling trials were performed. Five thousand random points were selected from the data set and were processed through the final script to predict emissions. The summary of trials is presented in Table 1.

Table 1. Summary of five trials of 5000 randomly selected points.

Trial	R^2	Ridge Score (Validation)	Ridge Score (Training)
1	0.64	0.62	0.85
2	0.69	0.66	0.86
3	0.65	0.63	0.84
4	0.66	0.57	0.92
5	0.65	0.58	0.90
Ave	0.66	0.61	0.87

The average r^2 value of the five trials is 0.66, and the average ridge score for validation is 0.61. Compared to the final result's r^2 of 0.74 and ridge score for validation of 0.73, there is some difference. The small disparity can be explained by the fact that the trials only used 5000 data points, which affects the forward pass process within the predicting consumption script. Still, the results are notable as it shows that 5000 data points (6.7% of the original 79942) can already show high predictive power, explaining up to 69% of the variance.

The trials' ridge scores for training are much higher compared to their respective ridge scores for validation, which may show that there is some overfitting. In the final result, however, the difference between the training ridge score (0.78) and the validation ridge score (0.74) is much smaller, showing that a bigger dataset helps minimize overfitting.

The ODIAC dataset served as the proxy ground-truth data in this study but can also be used as a basis for a comparison of the model's performance. In a study by Chen et al. [20], the ODIAC dataset's predictions among 14 large cities were compared to emission inventory statistics. This revealed that in

some cities, especially in developing countries, the dataset overestimates. As Chen states, this overestimation could be due to the "poor correlation between nightlight intensity with human activity [...] in developing countries." This is similar to a problem Jean et al. [4] cites, in which areas with very low light levels (often in developing countries) often show little variation, leading to models incapable of distinguishing differences in economic activity. Jean et al. improved upon existing studies by using a transfer learning approach in their model, which allowed them to have better estimates in countries and areas with minimal nightlight data. This study applied a similar approach; it was based on the transfer learning approach by Jean et al., which utilized both daytime images and nighttime data to estimate CO_2 emissions, thus making predictions within the Philippines - especially in areas that are darker or with low luminosity values - more accurate.

Though the results cannot be directly compared, due to the models measuring different variables, a comparison may offer some valuable insight. It was found that the developed model of this study estimates CO_2 emissions better than how the models of Jean et al. ($r^2_{max} = 0.55$) and Tingzon et al. ($r^2 = 0.63$) predict economic status.

Limitations. One of the primary concerns in this study was the lack of "true" ground truth data. Directly-measured CO_2 emissions data in the Philippines could not be found available in any of the country's database agencies, despite many efforts. Because of this, the study decided to use the ODIAC dataset as a proxy for ground truth data, which, while still a valid substitute, may contain overestimations and underestimations in urban areas and rural areas respectively as it is based on space-based nighttime light data and individual power plant emission/location profiles.

Due to restraints in time and processing power, data points with no emissions (data points with a value of 0) were removed, which left the 79942 data points which were then used for the rest of the study. It is recommended that in future studies and developments, all data points, including those with 0-emission values, should be included.

A heatmap couldn't be done in this study due to the lack of time and experience on the part of the researchers.

Conclusion. - Through the development of the scripts, adapted from and based on the publicly available GitHub repositories of Jatin Mather (2016) and Jean et al. (2016), this study was able to conclude that the developed convolutional neural network model used to estimate CO2 emissions in the Philippines has strong predictive power. It improves upon ODIAC, an existing CO_2 dataset, by looking at both daytime and nighttime images and data through transfer learning. The developed scripts and CNN model can be used as a starting point for more fully developed software and models that can serve as an alternative method of collecting CO_2 emissions data.

Recommendations. - Though the scripts and algorithms may prove themselves to be valid processes with strong predictive power, concerns may arise if the dataset itself has some issues - such as overestimations and underestimations - similar to that of ODIAC. It is recommended that, if there is access to a cleaner dataset of direct measurements from the Philippines rather than estimations, future researchers may want to utilize that instead.

This model is limited to data in 2015. It is possible that this model could be used for more recent years, but further research is required in order to confirm so.

A direct comparison between the accuracy of the ODIAC dataset and the accuracy of the developed model (compared to existing emission inventories) through graphs or a heatmap may also give additional insights.

Acknowledgment. - The researchers would like to thank Mr. Neal Jean of Stanford University and Thinking Machines Data Science Inc. for their invaluable time, knowledge, and support that was extended to this study.

References

[1] Van der Aalst WP. 2018. Data Science in Action. In: Process Mining. Berlin: Springer Berlin. p. 3–23.

[2] Dinov I. 2019. Data Science and Predictive Analytics: Biomedical and Health Applications Using R. S.l.: Springer.

[3] Xie M, Jean N, Burke M, Lobell D, Ermon S. 2016. Transfer learning from deep features for remote sensing and poverty mapping. In: Schuurmans D, Wellman MP, editors. c2016. AAAI'16. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence; 2016 Feb 12-17; Phoenix, Arizona. Palo Alto (CA): AAAI Press. p. 3929–3935.

[4] Jean N, Burke M, Xie M, Davis WM, Lobell DB, Ermon S. 2016. Combining satellite imagery and machine learning to predict poverty. Science 353: 790–794.

[5] Tingzon I, Orden A, Sy S, Sekara V, Weber I, Fatehkia M, Herranz MG, Kim D. 2019. Mapping Poverty in the Philippines Using Machine Learning, Satellite Imagery, and Crowd-sourced Geospatial Information. International Conference on Machine Learning AI for Social Good Workshop.

[6] Zhi-Ming D, Li-Xia G, Jian-Kui Z, Xue-Bin Z. 2015. Oil-Spills Detection in Net-Sar Radar Images Using Support Vector Machine. The Open Automation and Control Systems Journal 7(1): 1958–1962.

[7] Demirel N, Emil MK, Duzgun HS. 2011. Surface coal mine area monitoring using multi-temporal high-resolution satellite imagery. International Journal of Coal Geology 86:3–11.

[8] Bai T, Li D, Sun K, Chen Y, Li W. 2016. Cloud Detection for High-Resolution Satellite Imagery Using Machine Learning and Multi-Feature Fusion. Remote Sensing 8: 715.

[9] Ghosh T, Elvidge CD, Sutton PC, Baugh KE, Ziskin D, Tuttle BT. 2010. Creating a Global Grid of Distributed Fossil Fuel CO2 Emissions from Nighttime Satellite Imagery. Energies 3: 1895–1913.

[10] Makido Y, Dhakal S, Yamagata Y. 2012. Relationship between urban form and CO2 emissions: Evidence from fifty Japanese cities. Urban Climate 2: 55–67.

[11] Rayner PJ, Raupach MR, Paget M, Peylin P, Koffi E. 2010. A new global gridded data set of CO2 emissions from fossil fuel combustion: Methodology and evaluation. Journal of Geophysical Research 115.

[12] Abdullah L, Mohd Pauzi H. 2015. METHODS IN FORECASTING CARBON DIOXIDE EMISSIONS: A DECADE REVIEW. Jurnal Teknologi 75.

[13] Tomohiro O, Shamil M. 2015. ODIAC Fossil Fuel CO2 Emissions Dataset (ODIAC v1.7). Center for Global Environmental Research, National Institute for Environmental Studies, Downloaded from https://bit.ly/2SPGE5O.

[14] Elvidge, Christopher D., Kimberly Baugh, Mikhail Zhizhin, Feng Chi Hsu, and Tilottama Ghosh. "VIIRS night-time lights." International Journal of Remote Sensing 38, no. 21 (2017): 5860–5879.

[15] Mather, J. 2019. Predicting poverty replication. https://bit.ly/34IeUmm

[16] Tingzon I, Orden A, Sy S, Sekara V, Weber I, Fatehkia M, Herranz M, Kim D. 2018. Mapping Poverty in the Philippines Using Machine Learning, Satellite Imagery, and Crowd-sourced Geospatial Information. https://bit.ly/3fNvSGp.

[17] Jean N, Burke M, Xie M, Davis WM, Lobell DB, Ermon S. 2016. Combining satellite imagery and machine learning to predict poverty. https://bit.ly/3vN79HL.

[18] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. 2009. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition.

[19] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems 32 https://bit.ly/3gbmoUq.

[20] Chen J, Zhao F, Zeng N, Oda T. 2020. Comparing a global high-resolution downscaled fossil fuel CO2 emission dataset to local inventory-based estimates over 14 global cities. Carbon Balance and Management 15.

[21] Oxford University Press. 2020. Overfitting. In: Lexico. [accessed 2020 Dec 10]. https://bit.ly/3uH04Hb.

[22] Brownlee J. 2019. Overfitting and Underfitting With Machine Learning Algorithms. Machine Learning Mastery. [accessed 2020 Dec 10]. https://bit.ly/34Irm5K.

[23] Ying X. 2019. An Overview of Overfitting and its Solutions. Journal of Physics: Conference Series 1168: 022022.

Evaluating the ripening stages of *Musa acuminata* × *balbisiana* (Saba) using 2D-image analysis

ILYSSA MARIE D. MOSO, SEAN RAFAEL L. PUDA, PIA O. TIMAAN, and MARIA DEANNA B. JOLITO

Philippine Science High School Western Visayas Campus - Department of Science and Technology (DOST-PSHSWVC), Brgy. Bito-on, Jaro, Iloilo City 5000, Philippines

Article Info

Submitted: May 12, 2021
Approved: Aug 11, 2021
Published: Aug 30, 2021

Keywords:
banana
image analysis
Python
ripening stage
SVM

Abstract

Saba banana is one of the Philippines' main export products but has a fast ripening process which affects its quality and marketability. Most instruments utilized in measuring banana quality are destructive. However, few have used 2D-image analysis in assessing the ripeness of the banana. This paper presents the evaluation of the ripening stage of Saba banana using 2D-image analysis as an alternative, non-destructive method. The 120 banana images were pre-processed by obtaining the mask to remove the background, retaining only the region of interest, which was further converted into HSV and Grayscale. Ninety of the processed images were utilized as training data sets and the rest as testing data sets for the Support Vector Machine (SVM) where the overall agreement is 70%. The researchers recommend that future studies increase the texture features of the Gray Level Co-occurrence Matrix (GLCM) Analysis to improve the performance of the SVM.

Introduction. - Bananas are the top traded fruit worldwide, and it is one of the top fruit exports of the Philippines [1]. Therefore, it is important to ensure the quality (appearance, texture, flavor, and nutritive values) of the banana being exported and consumed locally because it affects their profitability [2].

There are four major cultivars of bananas in the Philippines and *Musa acuminata x balbisiana* (Saba) is one of the country's main products [22]. It is rich in vitamins and minerals and is also included in many Filipino dishes. *M. acuminata x balbisiana* is also used as a supplement or an alternative staple food from rice and corn in rural areas [3]. There was also a lack of research articles involving the analysis of the different ripening stages of this specific variety of banana using 2D image analysis. Bananas tend to be eaten at their mature stage which is why they are usually harvested at the mature green stage and remain without significant changes depending on the temperature, humidity, and age of the banana at the time of harvest. This is because the ripening process of bananas is irreversible once it starts because it brings about physical and chemical changes to the bananas [4]. It can be observed with the alterations in the fruit texture, the changes in the peel color (usually from green to yellow), the appearance of brown spots, and the synthesis of volatile components [5]. Bananas are known to have seven stages in the ripening process, as seen in Table 1, which are usually assessed through visual comparison of the colors of the peels [6]. As the banana ripens, its shape, size, color, and texture change which are observed by producers to decide when to harvest, transport, and sell the fruit [7]. The ability to identify maturity will help farmers optimize the harvesting phase [8].

Table 1. The stages of banana ripening based on peel color [6].

Ripening stage		Banana peel color
Unripe	**1**	Dark Green
	2	Light green, traces of yellow
Ripe	**3**	More green than yellow
	4	More yellow than green
	5	Green tip and yellow
	6	All yellow
Overripe	**7**	Yellow, flecked with brown

The usual methods for determining fruit quality involve destructive and time-consuming measures such as evaluation of dry matter content, total soluble solids content, sugar content, and juice acidity. Thus, non-destructive alternatives such as image analysis are encouraged [12]. As manual inspections tend to be tedious, subjective, and non-uniform, image analysis using different image processing techniques has been

How to cite this article:
CSE: Moso IMD, Puda SRL, Timaan PO. 2021. Evaluating the ripening stages of *Musa acuminata x balbisiana* (Saba) using 2-D image analysis. Publiscience. 4(1): 93–98.
APA: Moso, I.M.D., Puda, S.R.L., & Timaan, P.O. (2021). Evaluating the ripening stages of *Musa acuminata x balbisiana* (Saba) using 2-D image analysis. *Publiscience*, *4*(1), 93–98.

For supplementary data, contact: publiscience@wvc.pshs.edu.ph.

used to assess and evaluate the ripening process of bananas. Techniques such as the use of colorimeters [11], feature extraction and texture analysis [7], imaging and spectroscopy [12], color histogram [7], and many others have been used over the past few years. Many algorithms have also been utilized for the purpose of identifying the different ripening stages of bananas such as the use of the Gray Level Co-occurrence Matrix (GLCM) for processing; Support Vector Machine (SVM) and K-nearest Neighbor (KNN) for recognition [13]; Fuzzy Color Histogram (FCH) and Movement Imagination (MI) methods for feature extraction [14]; Artificial Neural Network (ANN) to increase quality detection [15], and many others have been proven to be effective and usable in real-life applications. There are also new and efficient techniques that are based on the Hue Saturation Value (HSV) colorspace, development of brown spots, and texture analysis of the banana. According to Tichkule and Gawali [20], non-destructive applications that can use texture analysis techniques on the products are good alternatives for effective food quality assessment because they contain information on the color and geometric structure of the fruit.

The results of this study will benefit researchers who aim to further enhance the use of 2D-image analysis as a non-destructive assessor of quality.

This study aimed to predict the ripening stages (unripe, ripe, and overripe) of *Musa acuminata x balbisiana* (Saba) using 2D-image analysis. Specifically, it aimed to:

(i) analyze the ripe, unripe, overripe stages of Saba banana by applying the Hue Saturation Value (HSV) and Grey-level Co-occurrence Matrix (GLCM) feature extraction techniques.

(ii) compare the HSV of the unripe, ripe, and overripe Saba bananas using histograms.

(iii) determine the different ripening stages of the bananas using the SVM.

Methods. - The Saba bananas were acquired from the local farmers in Leganes, Iloilo, Philippines. The samples were grouped into ripe, unripe, and overripe stages by. This research utilized 120 banana samples overall, 40 for each ripening stage. From each ripening stage, ten of the images were processed while the rest were used to train the SVM. The images were converted into binary images [7]. Next, the images underwent image segmentation which produced the original picture without the background [8]. For the feature extraction, the HSV color space was used to categorize each pixel based on its color and brightness and the GLCM was used in measuring the texture of the image. The final process utilized an SVM to differentiate the ripe, unripe, and overripe bananas [7].

Program Implementation. The Python programming language of version 3.7.10 was implemented through Google Colaboratory which was mounted onto specific Google Drive folders where the acquired images were uploaded.

Sample Collection. The Saba banana samples collected in Leganes, Iloilo were identified and grouped by a qualified banana farmer. Six hands of bananas which were classified as unripe, ripe, and overripe were collected. Unripe bananas are green (stage 1 - stage 2), ripe bananas are usually yellow (stage 5 - stage 6) and overripe bananas are yellow with brown spots (stage 7) [8]. Forty (40) individual bananas were chosen from each ripening stage having 120 samples in total.

Image Acquisition. Images of the samples were acquired inside an enclosed black cardboard box by taking photos using a mobile phone model Vivo 1806 with a 9.8MP camera which was attached to a phone stand 19 cm above the surface of the black cloth. The picture was taken in a room with a temperature of 302°C [17] and with one major source of white light being an LED ring light [12]. Overall, 120 pictures were taken and transferred from the mobile phone to a Google Drive folder in JPEG format [18].

Table 2. The camera control settings used during image acquisition.

Variable	Settings
Image Size	2160 x 4560
Magnification	1.0x
Flash	No Flash
Image Type	JPEG
Aperture	f/2.0

Table 3. The specifications of the LED Ring light used during image acquisition.

Variable	Settings
Brand	SANYK
Model	10 inches live fill light
Outer Ring Size	26cm
Light	Cold Light
Dimmable	Yes
Color Temperature	2700-7000K
Color Rendering Index	RA/CRI:80
Overall lumens	600-1300LM
Number of Lamp Beads	120PCSA
Working Power Supply	DC 5V/1A
Power	12W/24W
Lamp Bead Model	2835 LED
Light Angle	120

Table 4. Raw images of three *Musa acuminata x balbisiana* samples from each ripening stage (unripe, ripe, and overripe).

Samples	Unripe	Ripe	Overripe
1			
2			
3			

Projected Area Estimation. The bananas were cut into six planes in the longitudinal axis. Each perpendicular cut was measured using a Vernier caliper, while the internal and external lengths of the bananas were measured using a flexible ruler [10]. The area of the banana was calculated by solving for the summation of the individual elements: the mean value of the ring thickness, area of the sectoral frustum, and center of the curvature.

Pre-processing. The images were converted into the HSV colorspace. Afterwards, it was converted into grayscale and further converted into binary images which served as segmentation masks [19].

Image Segmentation. Using the segmentation masks, the RGB banana images were separated from the unwanted regions where only the regions of interest were kept [8] which are the areas of the images that were used for analysis [20].

Feature Extraction. The study focused on measuring the sample images' color and texture features. The HSV method was used to evaluate the color, amount of color, and brightness of the images. Afterwards, the HSV values were extracted and plotted into a histogram for each ripening stage where the data was analyzed [7]. The GLCM was used to investigate the texture, specifically, the contrast and homogeneity of the pixels [7,12].

Recognition. The SVM model was trained using 90 testing images with 30 from each ripening stage. The SVM was then used in order to predict the ripening stages of the Saba bananas based on the data sets from the feature extraction process [7].

Data Analysis. The classification of bananas based on their ripening stages was analyzed using the HSV and GLCM for color and texture features, respectively. For the HSV extraction, the data obtained were from the colors of the pixels generated from the HSV color space. These pixels were plotted on a histogram. The GLCM extraction focused on obtaining the value for the contrast [8] and homogeneity [12] using the following formulas:

$$\text{Contrast} = \frac{\sigma}{(\alpha_4)^{0.25}}, \text{ and } \alpha_4 = \frac{\mu_4}{\sigma^4}$$

Where μ_4 is the fourth moment about the mean, and σ^2 is the variance.

$$\text{Homogeneity} = \sum_{i=1}^{K}\sum_{j=1}^{K}\frac{P_{ij}}{1+|i-j|}$$

Where K is the size of the co-occurrence matrix, P_{ij} is the estimate of the probability of a pair of points having values that satisfy an operator, and i and j are values of intensity.

The contrast and homogeneity values were then placed on a matrix and were sorted into pairs according to their banana sample [21]. Using Microsoft Excel, the mean and standard deviation of the texture values were calculated. The SVM was used to compare its predictions with the actual ripening stages of the samples.

Safety procedure. The mandated safety protocols of the LGUs for the COVID-19 were followed during the conduct of the data gathering by the researchers.

Results and Discussion. - Figures 1, 2, and 3 show the average hue, saturation, and intensity values from the ten segmented unripe, ripe, and overripe banana images, respectively, which were obtained from the HSV color feature extraction process. The hue values for the unripe stage are between the range of 90 to 110, while the ripe and overripe stages have hue values in between the range of 50 to 70 with the ripe banana samples having the highest frequency of pixels and the overripe bananas having the lowest frequency of pixels. For the saturation, ripe bananas had the highest saturation value being 0.6 - 0.8 while unripe bananas had the lowest saturation ranging only from 0.4 - 0.6. Lastly, for the intensity, the unripe bananas have the most diverse range of brightness from 50 to 250 while the overripe bananas had the least diverse range of brightness from 150 to 200.

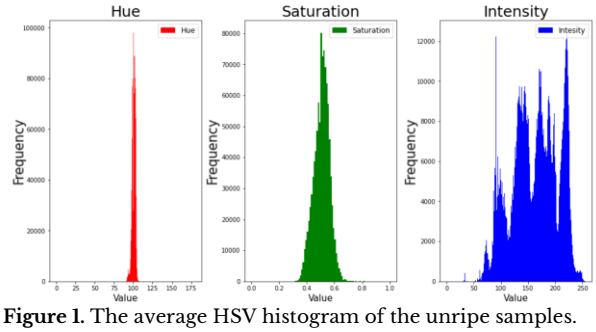

Figure 1. The average HSV histogram of the unripe samples.

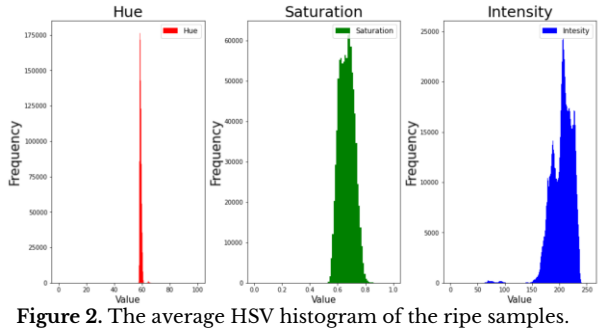

Figure 2. The average HSV histogram of the ripe samples.

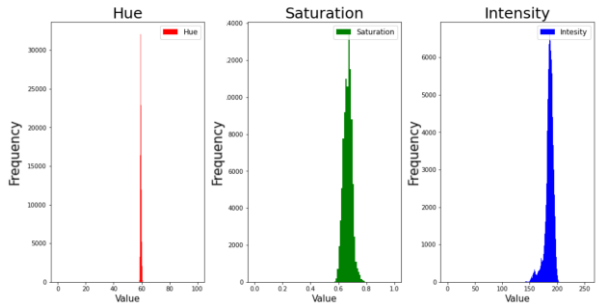

Figure 3. The average HSV histogram of the overripe samples.

The contrast and homogeneity values were obtained using Python in Google Colaboratory, meanwhile, Microsoft Excel was used to calculate the mean of the values and their standard deviation. Normalization of data was done to obtain the best performance for the system with values between zero and one [23]. From Table 4, the overripe bananas have obtained the lowest normalized texture feature value for contrast and homogeneity. The unripe and ripe bananas obtained the highest normalized values for contrast and homogeneity, respectively.

Table 5. The normalized feature vector for the unripe, ripe, and overripe Saba bananas using GLCM Feature Extraction.

Texture Feature	Contrast	STD. DEV	Homogene-ity	STD. DEV
Unripe	0.22	0.24	0.25	0.20
Ripe	0.21	0.25	0.29	0.12
Overripe	0.19	0.27	0.21	0.25

Figures 4, 5, and 6 are histograms that present the average red, green, and blue values of the ten unripe, ripe, and overripe banana samples that were analyzed during the HSV color feature extraction process, respectively. On the right side of the plots, the unripe samples have more green pixels than red pixels as evidenced by the higher peak around the 150-color range, while the ripe samples were the other way around having a stronger peak towards the 250-color range. On the other hand, the overripe samples have much more red, green, and blue pixels on the left side of the plot.

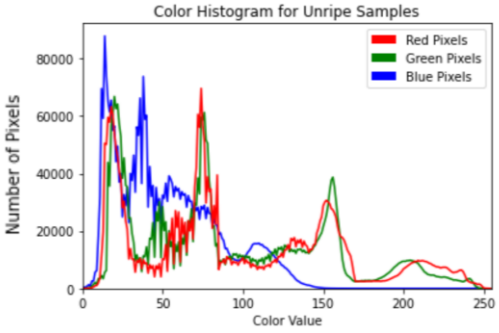

Figure 4. The average color histogram for the unripe samples.

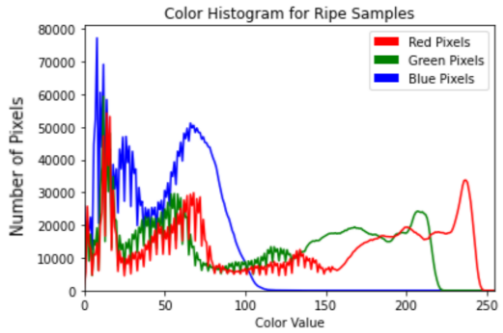

Figure 5. The average color histogram for the ripe samples.

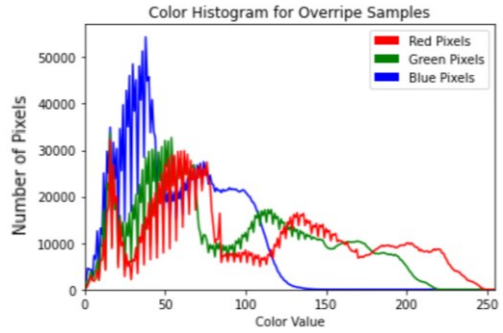

Figure 6. The average color histogram for the overripe samples.

The average values of the histograms in Figures 4, 5, and 6 have a high frequency of dark blue pixels as evidenced by the strong peaks of the blue line reaching over 50,000 pixels from the 0-50 range. This indicates that there may be high amounts of shadows that were included in the images during the HSV color extraction. However, the frequency of the blue pixels in all three histograms was reduced to zero as the color value rose, which means that there were no light blue pixels in the banana images. In contrast to the reduced amount of blue pixels, the right side of the histograms has higher amounts of red and green pixels.

Specifically, the unripe samples had more green pixels than the ripe and overripe samples while the ripe sample had more red pixels than the unripe and overripe samples. Moreover, the unripe samples have the least amount of red and green pixels on the ride side of the plot but have higher values of red, green, and blue pixels on the left side.

An SVM is a supervised learning method that analyzes data and recognizes patterns that are useful in data classification [24]. Random forest was used to

classify the images based on the SVM model. The classification was conducted in Colaboratory using python as a programming language.

Table 6. The confusion matrix resulting from the classification of the ripening stages of the Saba bananas using an SVM Classifier.

Actual Stage	Predicted Stage			Sensitivity (%)
	1	2	3	
Unripe	*5*	2	3	*50*
Ripe	0	**10**	0	100
Overripe	0	4	**6**	60
Precision (%)	100	62.5	66.7	Overall Correctness = **70%**

In Table *5*, the confusion matrix presented was used to evaluate the SVM system. The total samples that were processed in the system were 30 bananas, ten from each ripening stage. The system has an overall agreement of 70%. The system classified 10 out of the 10 samples of the ripe bananas correctly (100% sensitivity). Analyzing the errors, 4 out of 10 samples of overripe bananas were misclassified as ripe. This means that the banana samples between the ripe and overripe stages have a similar texture and color features. In a study conducted by Mazen and Nashat (2019), they had made an SVM that achieved an overall agreement of 96.6%. In their study, they used another image feature to differentiate between the different ripening stages—Ripening Factor (RF)—which was not included in this study. RF is the banana's ripeness factor where the total area of the brown spots is divided by the total area of the banana [10]. The maximum value of precision, which is a proportion of the predicted positive ripening stage that was correctly identified, in the classifier is from the unripe stage with 100%. The maximum value of sensitivity, which is the ability of the prediction model to select the instance of a certain ripening stage from the dataset, is reached by the ripe stage with 100%. Table *5* proves this fact because 10 out of 10 banana samples were classified in the correct stage.

Limitations. The limitations of this study include the color range threshold for the image segmentation as the study only used the visual chart provided by Soltani et al. (2019) due to the lack of previous literature involving Saba bananas. Furthermore, only the basic color values of yellow, green, and brown were used for the image segmentation process due to a lack of previous literature regarding the color ranges of Saba bananas.

Conclusion. - The color and texture features from the 90 images that were used to train SVM were able to achieve a 70% overall agreement in classifying the 30 Saba banana samples into unripe, ripe, and overripe classes. Nevertheless, the SVM was still able to garner a 100% accuracy rating for the ripe banana images. Hence, 2D-image analysis has the potential to be a non-destructive alternative in classifying the ripening stages of Saba banana as well as other similar fruits in the food industry, but can always be improved further in the future.

Recommendations. - It is recommended to use the official or actual color ranges of the different stages of bananas depending on their species, as this study was not able to find the color chart for the species used in this study -Saba banana. Moreover, it is recommended to apply more than two texture features for the GLCM analysis such as correlation, energy, entropy, dissimilarity, and other image features such as the Ripeness factor to achieve the best performance for the SVM [23]. Additionally, it is also advisable to use classifiers other than the SVM such as Artificial Neural Network [23] to attain a suitable classifier with the best performance depending on the research design process. Lastly, the researchers emphasize the importance of adding more samples to increase the number of images for the training dataset to also increase the performance of the system [24].

Acknowledgement. - The researchers would like to extend their gratitude to the OIC-Municipal Agriculturist for providing their knowledge in selecting a source for the saba bananas as well as the banana farmers who assisted us in the collection of the samples. The group also extends their appreciation to the external consultants for sharing their expertise in this research.

References

[1] [DA] Department of Agriculture. 2018 Dec. Philippine banana industry roadmap. Diliman (QC): DA. *52* p.

[2] Elmasry G, Wang N, Elsayed A, Ngadi M. 2007. Hyperspectral imaging for nondestructive determination of some quality attributes for strawberry. J Food Eng. 81(1): 98–107. doi: 10.1016/j.jfoodeng.2006.10.016.

[3] Calica GB, Lingbawan KR. 2019. Value chain analysis of processing Cardava banana chips in the Philippines: Luzon case. IJSRM. 7(1): 941–946. doi: 10.18535/ijsrm/v7i1.em01.

[4] Dadzie BK, Orchard JE. 1997. Routine post-harvest screening of banana/plantain hybrids criteria and methods. Rome: IPGRI. (INIBAP Technical Guidelines; No. 2).

[5] Drury R, Hörtensteiner S, Donnison I, Bird CR, Seymour GB. 1999. Chlorophyll catabolism and gene expression in the peel of ripening banana fruits. Physiol Plant. 107(1): 32–38. doi: https://doi.org/10.1034/j.1399-3054.1999.100105.x.

[6] Li M, Slaughter DC, Thompson JF. 1997. Optical chlorophyll sensing system for banana ripening. Postharvest Biol Tech. 12(3): 273–283.

[7] Nayak AM, Manjesh R, Dhanusha MS. 2019. Fruit recognition using image processing. IJERT. 7(8): 1–6.

[8] Mazen FMA, Nashat AA. 2019. Ripeness Classification of Bananas Using an Artificial Neural Network. Arab J Sci Eng. 44(8): 6901–6910. doi: 10.1007/s13369-018-03695-5.

[9] Soltani M, Alimardani R, Omid M. 2010. A New Mathematical Modeling of Banana Fruit and Comparison with Actual Values of Dimensional Properties. Mod Appl Sci. 4(8): 14–113.

[10] Mendoza F, Aguilera JM. 2006. Application of Image Analysis for Classification of Ripening Bananas. J Food Sci. 69(9). doi: 10.1111/j.1365-2621.2004.tb09932.x.

[11] Santoyo-Mora M, Sancen-Plaza A, Espinosa-Calderon A, Barranco-Gutierrez AI, Prado-Olivarez J. 2019. Nondestructive quantification of the ripening process in banana (Musa AAB Simmonds) using multispectral imaging. Sensors. 2019: 1–12. doi: 10.1155/2019/6742896.

[12] Shukla D. Recognition of fruits using hybrid features and machine learning. 2016. 2016 International Conference on Computing, Analytics and Security Trends (CAST). doi: 10.1109/cast.2016.7915046.

[13] Fachrurrozi EM, Fiqih A, Saputra BR, Algani R, Primanita A. 2017. Content based image retrieval for multi-objects fruits recognition using k-means and k-nearest neighbor. 2017 International Conference on Data and Software Engineering (ICoDSE). doi: 10.1109/icodse.2017.8285855.

[14] Yogesh, Dubey AK. 2016. Fruit defect detection based on speeded up robust feature technique. 2016. 5th International Conference on Reliability. Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO).

[15] Tichkule SK, Gawali DH. 2016. Plant diseases detection using image processing techniques. 2016 Online International Conference on Green Engineering and Technologies (IC-GET).

[16] Mohapatra A, Shanmugasundaram S, Malm-athanraj R. 2017. Grading of ripening stages of red banana using dielectric properties changes and image processing approach. Computers and Electronics in Agriculture. 143(1): 100–110. doi: 10.1016/j.compag.2017.10.010.

[17] Saad H, Ismail AP, Othman N, Jusoh MH, Naim NF, Ahmad NA. 2009. Recognizing the ripeness of bananas using an artificial neural network based on histogram approach. 2009 IEEE International Conference on Signal and Image Processing Applications. 536–541.

[18] Miljković O. 2009. Image pre-processing tool. Kragujevac J. Math. 32: 97–107.

[19] Mande A, Gurav G, Ajgaonkar K, Ombase P, Bagul V. 2018. Detection of Fruit Ripeness Using Image Processing. Common Comput Inf Sci. 545–555. doi: 10.1007/978-981-13-1813-9_54.

[20] Verma M, Raman B, Murala S. 2015. Local extrema co-occurrence pattern for color and texture image retrieval. Neurocomputing. 165: 255–269. doi: 10.1016/j.neucom.2015.03.015.

[21] Olaniyi EO, Adekunle AA, Odekuoye T, Khashman A. 2017. Automatic system for grading bananas using GLCM texture feature extraction and neural network arbitrations. J Food Process Eng. 40(6). doi: 10.1111/jfpe.12575.

[22] Other crops: Volume of production, by region and by PROVINCE, by quarter and semester, 2010-2020. PX-Web.

[23] Burges CJC. 1998. A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov. 2(2): 121–167.

[24] Zhu X, Vondrick C, Fowlkes CC, Ramanan D. 2015. Do we need more training data?. Int J Comput Vis. 119(1): 76–92. doi: 10.1007/s11263-015-0812-2.

Using a sssNet Convolutional Neural Network (CNN) with Support Vector Machine (SVM) algorithm to identify formalin presence in images of eyes of *Chanos chanos* (milkfish)

THOREENZ P. SOLDEVILLA, DOMINIC NATHANIEL ANTHONY A. REDANIEL, LANCE CHRISTIAN V. SY, and MARIA MILAGROSA A. NULLA

Philippine Science High School Western Visayas Campus - Department of Science and Technology (DOST-PSHSWVC), Brgy. Bito-on, Jaro, Iloilo City 5000, Philippines

Article Info

Submitted: Apr 30, 2021
Approved: Jun 21, 2021
Published: Aug 30, 2021

Keywords:
Chanos chanos
formalin
image processing
neural network
SVM

Abstract

Using image processing, eye turbidity of formalin-treated *Chanos chanos* (milkfish) was statistically proven to be significantly different from those untreated. However, automation of such processes was yet to be explored. This study aims to use a sssNet Convolutional Neural Network (CNN) with Support Vector Machine (SVM) algorithm to identify formalin presence in milkfish. Ninety percent of 420 formalin-treated milkfish images and 420 untreated images, each with an indicated day of image capture, were subjected to feature extraction and classification using sssNet-SVM. The remaining 10% of the dataset was used to validate the algorithm's performance. The algorithm garnered 98.16 to 99.15% validation accuracy for identifying formalin presence. However, seven-day feature map analysis reveals that the algorithm struggles to determine formalin presence in treated samples using their images that were captured one or two days after the samples' dousing in formalin.

Introduction. - The Philippines is a fish-producing country that ranks 11th in global fishing production [1]. The main aquacultural produce of the country, *Chanos chanos*, locally known as *bangus* or milkfish, accounts for 2.4% of the national fisheries production [2]. Fish are highly perishable food, with storage times for tropical species ranging from 6-40 days [3]. Due to this limitation, various preservation techniques have been devised to prolong its freshness in order for such to be marketable for longer periods of time. One of the chemicals used to preserve fish is formalin [4,5], a solution consisting of 37% formaldehyde, a known respiratory disease enabler [6]. Formalin can also cause early protein denaturation which compromises fish quality [7]. Several studies used chemical analysis methods to detect the early presence of formalin in meat and fish such as spectrophotometry [8] and formalin rapid testing [9].

However, methods regarding chemical analysis are labor-intensive and time-consuming, while rapid test kits are not readily available in the market. This limitation was addressed by Cadorna et al. [8]. that used computer vision techniques such as image processing to detect formalin presence in milkfish by evaluating its eye, a method similar to most computer vision techniques that measure fish freshness. Image analysis has implied that the eye of formalin-treated fish became cloudy after a seven-day period as opposed to untreated fish which almost retained its appearance. The study then quantified the eye turbidity by capturing the image of the fish, splitting the channels into HSV (Hue, Saturation, and Value) components, and determining the intensity of each color space using an image processing tool. The study then found out that with values below 0.05 level of significance, the value components of the eye images +-of formalin treated and untreated *C. chanos* are significantly different. Such has opened the possibility of utilizing eye turbidity to be used for automation of formalin detection using imagery.

Since automation of fish quality [11] and classification of eye appearance [12] is possible, formalin detection in *C. chanos* can be done using supervised machine learning. Algorithms for classifying fish samples according to their quality are using two distinct methods – feature extraction and classification. These are done by first enhancing the images using methods such as blob extraction and border smoothing [11], as well as eye masking [12]. Then, the processed images are loaded into algorithms such as *k*-Nearest Neighbor, Support Vector Machine (SVM), or Feed-forward Artificial Neural Network (ANN). Hence, it is implied that feature extraction and classification algorithm methods for machine learning are always interdependent of each other. However, feature extraction is a tedious process that requires enhancing images manually before being analyzed or loaded into an algorithm. To be able to overcome this limitation, an algorithm that unifies feature extraction and classification shall be utilized.

How to cite this article:
CSE: Soldevilla TP, Redaniel DNAA, Sy LCV, Nulla MMA. 2021. Using a sssNet Convolutional Neural Network (CNN) with Support Vector Machine (SVM) algorithm to identify formalin presence in images of eyes of *Chanos chanos*. Publiscience. 4(1): 99–104.
APA: Soldevilla, T.P., Redaniel, D.N.A.A., Sy L.C.V., & Nulla, M.M.A. (2021). Using a sssNet Convolutional Neural Network (CNN) with Support Vector Machine (SVM) algorithm to identify formalin presence in images of eyes of *Chanos chanos*. *Publiscience, 4*(1), 99–104.

For supplementary data, contact: publiscience@wvc.pshs.edu.ph.

One such algorithm unexplored in determining fish quality is the sssNet algorithm with SVM, developed by Nagata et al. [13]. It is a type of Convolutional Neural Network (CNN) that enhances the features of the images by filtering important pixels in the image and removing unwanted features. The sssNet-SVM algorithm was validated by testing it with images of 7000 resin articles, 6000 of which have defects such as cracks, while the remaining 1000 images are of good quality. The algorithm only misclassified three images of resins of good quality and 13 images of defects. By implementing an algorithm with a unified extraction and classification such as sssNet-SVM, the need for tedious operation of manually extracting the important features of fish for analysis is eliminated. This proposed machine learning model composed of the algorithm trained using a reliable dataset would help those in the aquaculture sector to detect such a hazardous preservative faster and cheaper than conventional chemical analysis methods.

The aim of this study was to use the sssNet-SVM algorithm in identifying formalin presence in *Chanos chanos* (Milkfish) and specifically aimed to:

(i) Preprocess images of a *C. chanos* dataset to be suitable for algorithm training,

(ii) Extract features from the images using sssNet and classify them into formalin-treated and untreated classes using sigmoid Support Vector Machine,

(iii) Evaluate the confusion matrix table and percent accuracy of the image classifications, and

(iv) Determine differences in features between images of formalin-treated and untreated classes extracted by the optimized algorithm using heatmap visualization.

Methods. — The dataset created by Cadorna et al. [10], which consists of 420 images of eyes of *C. Chanos* that were treated with formalin, and 420 images of eyes of *C. Chanos* that were untreated with formalin, were used for training the algorithm. Before the images were loaded into the algorithm, the images were preprocessed. Tensorflow (version 2.3.1) powered by the Python programming language (version 3.8) was used to execute the methodology. The Anaconda distribution for Python was used, along with Jupyter as the Integrated Development Environment. The methodology is then divided into three parts, namely (1) image preprocessing, (2) algorithm building, and (3) algorithm training. The specifications for the hardware and software used to run the algorithm is stated in table 1.

Image Preprocessing. Images from the dataset were loaded using the *os* library in Python. Python then reads the images as sets of arrays with hexadecimal values that correspond to each pixel. Since Cadorna et al. [10] used HSV (Hue, Saturation, and Value) color space for assessment of the images, the dataset was converted from RGB (Red, Green, and Blue) color space into HSV. The pixel values were then normalized by dividing each pixel value by 255 which is the maximum hexadecimal value. Image augmentation was then used to artificially enlarge the dataset size by providing multiple instances for each image. Manipulation methods such as resize fit, rotate, stretching, skewing, zooming, flipping, and filling were used for data augmentation.

Table 1. Specifications of hardware and software used for the algorithm training.

Item	Specifications
desktop computer	Windows 10 Operating System, AMD Ryzen 5 3600 3.6 GHz processor, AMD RX 5700-XT GPU, 512 GB SSD with 330 MB/S read and write speeds, 1 TB HDD with 175 MB/S read and write speeds, and 32GB of RAM clocked at 3000MHz DDR4
Python	Version 3.8
Anaconda*	Version 4.0.15
Tensorflow	Version 2.3.0
Jupyter	Version 6.0.3.

**open source version*

Algorithm Building. As previously mentioned, the study used the sssNet-SVM algorithm as the classification method for the *C. chanos* images. The algorithm is a type of Artificial Neural Network (ANN). A more specific type of ANN, the Convolutional Neural Network (CNN), is a multilayer perceptron that deals with grid-structured data. Used often in computer vision, CNN outputs an input image by using a two-dimensional filter with a set of weights that would be multiplied to each input [15]. The resulting output would be an image with amplified features, just like the expected outputs from feature extractions. The study used the sssNet algorithm as the classification layer, as seen in Figure 1. The input image is characterized as a normalized HSV image with dimensions of 330 by 330 pixels with a depth of three representing the three color space channels. The image then entered the algorithm through a feature layer structure that consists of three instances of weighted filters, or convolutional layers with filter size 5 followed by a max-pooling layer of filter size 3. After the final max-pooling layer, the pixel values are flattened, or the pixel values are lined side by side, then fitted into 32 sets of outputs, then summarized and fitted into the SVM, which is a sigmoid function.

The sigmoid function is described as:

$$(1)\quad \Phi(y) = \frac{1}{1+e^{-y}}$$

The sigmoid function fitted the summarized outputs y into the probabilistic values of 1 to 0, with 1 being close to the formalin treated value and 0 being close to the untreated value [15].

Algorithm Training. In order for the algorithm to learn, such was optimized using backpropagation, a process of relearning the receptive filters of the feature extracting layers by (1) finding the loss function, (2)

finding the gradient descent, and (3) using an optimization equation to revise the filters.

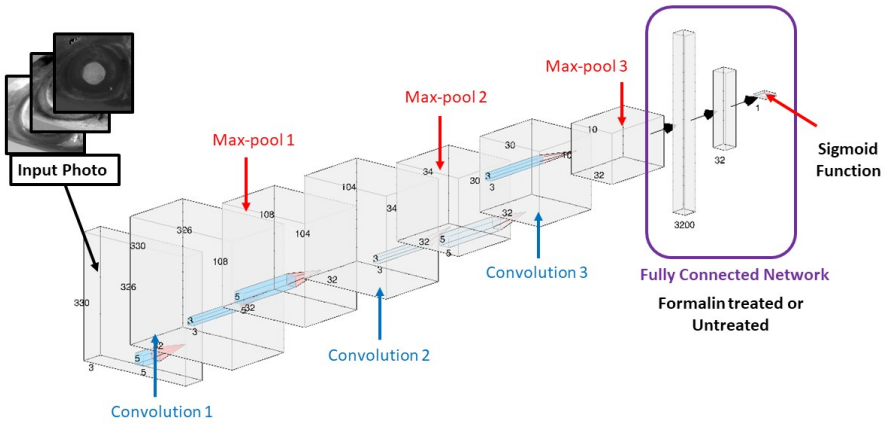

Figure 1. The sssNet-SVM algorithm diagram.

Furthermore, to expedite the training process, the images shall be fed to the algorithm in batches and shall automatically stop if training is sufficient using the callback function. Finding the loss function is done by plotting the loss values of the input data. The binary cross-entropy loss equation is used to find the difference between the calculated output value y and the actual label of the input images [15].

The cross-entropy loss equation is evaluated as:

$$(2)\ L = -\sum_{i=1}^{k} \quad y_i \log(\Phi y)$$

L as the negative summation of expected output data y_i multiplied by the logarithm of input data log Φy.

The loss values L are then plotted, in which a loss function is determined. Gradient descent method is then used to find the relative minimum of the loss [15]. The gradient method is noted as the derivative of loss L over the derivative of the filter weight w. After determining the gradient, the ADAM optimization equation takes note of the gradient then relearns the weights of the sssNet-SVM algorithm [15]. ADAM, short for Adaptive Moment Estimation, is an optimizer that tries to use learning decay rates characterized as ρ and momentum F_i in order for a model to find the minimum loss with accelerating speed. In the Keras module, ρ is initialized as 0.9.

$$(3.1)\ A_i \Leftarrow \rho A_i + (1-\rho)\left(\frac{\delta L}{\delta w_i}\right)^2 \forall_i$$

The first equation describes how the gradient descent is regulated with the learning decay rate.

$$(3.2)\ F_i \Leftarrow \rho_f F_i + (1-\rho_f)\left(\frac{\delta L}{\delta w_i}\right) \forall_i$$

The second equation, F_i is the momentum of the optimizer equation. Adding momentum to the optimizer increases the speed of the model approaching the minimum loss of the data. The learning decay rate ρ_f = 0.99 is also initialized with the Keras module.

$$(3.3)\ w_i \Leftarrow w_i - \frac{\alpha_t}{\sqrt{A_i}} F_i \forall_i$$

The last equation describes how a weight of a filter is adjusted according to A_i and F_i. w_i is replaced as the previous w_i subtracted by the ratio of learning rate a and the square root of A_i which is then multiplied by the momentum F_i. A_i in Keras is added with a stabilizer ε of 1 x 10^{-8} to avoid division by zero during the first model training with no pre-saved values.

Longer periods of training would result in fluctuating accuracy because the gradient descent value may be larger than a value's proximity to the relative minimum of the loss function. To cope with this limitation, a callback function, which detects when a model's accuracy is about to fluctuate or overfit, was implemented. In order for the algorithm to train faster, batches of images were fed in the algorithm, then the summation of the cross-entropy loss for each batch was evaluated instead of feeding the algorithm with images one-by-one then evaluating the loss values for each image. Ninety percent (90%) of the dataset consists of 756 images being fed into the algorithm. For each learning instance of the algorithm, or epoch, 18 batches of 42 images per batch were fed into the algorithm.

Data Analysis The data analysis part is divided into two parts, namely (1) AUROC analysis, and (2) heatmap visualization.

A Receiving Operating Characteristics curve or ROC is used by classification studies to assess the accuracy of the model. Once used in the medical field, ROCs are used to determine how deviant the values of the true positive and true negative values are. The computation of values shall be done using a Confusion Matrix Table [14]. Using training data obtained from the Cadorna et al. [10] dataset, an image's probability value as calculated by the sigmoid function was calculated by the algorithm if it is a false positive (FP), false negative (FN), true positive (TP) or true negative (TN). Next, the true positive rate was obtained by dividing TP by TP+FN. Then, the false-positive rate was obtained by subtracting the true positive rate from 1.0. A linear regression model of the true vs false positive rate was obtained to get the ROC curve. The Area Under ROC curve or AUROC for the epoch was then evaluated using the Riemann sum method as shown in Figure 2.

To test the ability of the optimal algorithm to identify formalin presence in *C. chanos* using unseen data, the remaining 10% of the Cadorna et al. [10] dataset was fed into the optimal algorithm. The confusion matrix table of the validation data was then evaluated.

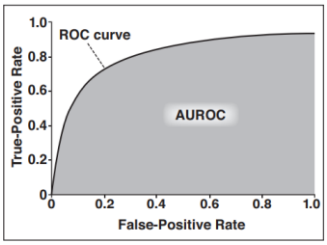

Figure 2. AUROC curve [14].

Lastly, to visualize how the convolutional and max-pooling layers of the optimal algorithm extract features from formalin-treated image classes and untreated image classes, the seven-day image capture of a randomly-picked sample from the formalin-treated class, as well as the seven-day image capture of a randomly-picked sample from the untreated class was obtained from the dataset. All images are then fed into the optimal algorithm, which were then visualized using a heatmap. The images are then compared to the heatmap of the unaltered HSV color spaces of the input image. The code for heatmap visualization was then used from a GitHub repository [16].

Results and Discussion. - Table 2 shows the accuracy for the sssNet-SVM algorithm tested in five trials using the same dataset showing a consistent accuracy above the 95% confidence threshold, ranging from 98.16 to 99.15%. AUROC curve percentage is calculated from the confusion matrix table, or evaluation of formalin-treated and untreated images which are either correctly classified or otherwise. Out of 84 validation images, trial 5 presents an askew confusion matrix with 13 images falsely classified as formalin-treated, unlike the other trials which falsely classified only 3 or 4 images as formalin-treated. For the untreated classes, all trials only falsely classified 0-4 images as untreated.

Table 2. Confusion matrix and AUROC curve percentages for each training trial of the sssNet-SVM algorithm

Trial	TF	FF	TU	FU	AUROC (%)
1	39	3	40	2	99.15
2	38	4	41	1	98.64
3	38	4	40	2	98.30
4	39	3	38	4	98.16
5	29	13	42	0	98.64

Legend:
TF: True Formalin-Treated class
FF: False Formalin-Treated class
TU: True Untreated class
FU: False Untreated class

Heatmap visualization is then applied into the feature extraction filters of the sssNet-SVM algorithm. Visualization of the extracted images is important in determining the receptive field that defines the significant difference between classes [17]. The receptive field in this algorithm pertains to the rectified pupils of the image samples. The pupils are said to be rectified if they are isolated from the eye of the fish and are then enlarged in the final layers of the algorithm.

The filter visualizations for the formalin-treated class and untreated classes were then compared with each other to see if differences in feature maps were exhibited by the algorithm in order for it to achieve high accuracy in classifying the dataset. Selection of feature maps for comparison used images taken from a single *C. chanos* sample which are captured daily, in a span of seven days. This is to examine if data for formalin-treated and untreated data are comparable irregardless of when the *C. chanos* sample was captured.

Figure 3 shows that regardless of the day the sample was taken, the pupil of the eye of untreated *C. chanos* is rectified. In contrast, Figure 4 shows that for images in day 1 and day 2 of the formalin-treated classes, the pupil is rectified. This is not the case for the few remaining filters in figure 4. Hence, we can say that irregardless of the onset of storage of untreated *C. chanos*, the algorithm easily determines it as an untreated class. On the other hand, formalin-treated fish can be easily determined by the algorithm as such if the fish was taken 3 days after the onset of treatment or later.

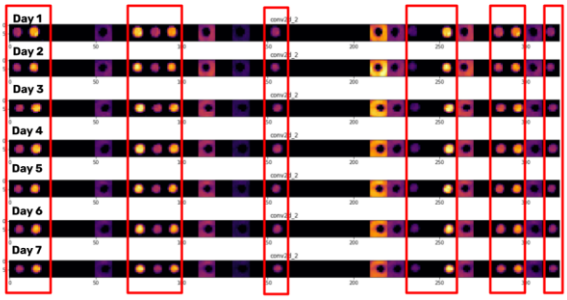

Figure 3. Final feature extraction layers of untreated *C. chanos* sample #28, left-eye, in seven days.

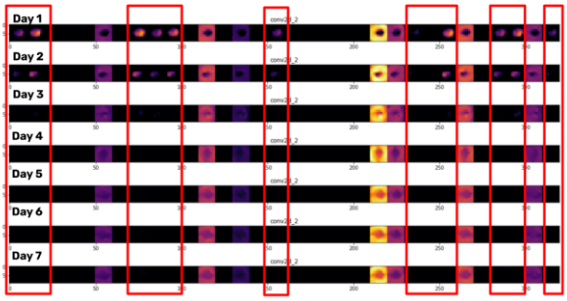

Figure 4. Final feature extraction layers of formalin-treated *C. chanos* sample #28, left-eye, in seven days.

Based on Figures 3 and 4, the framework of the algorithm, as shown in Figure 5, is generally described as starting from the input image is split into hue, saturation, and value channels. Then, the input image is augmented and extracted using the sssNet. If pupils are rectified, or present in the extracted image, the

fish sample in the picture is likely to be untreated with formalin; otherwise, the fish is likely to be doused with such substance. When training a CNN such as sssNet, the perception, or filter map of the algorithm is trained to focus on the specific parts of the image wherein classifications could produce significant differences between classes [17]. Eye turbidity of the image, the parameter that best describes the significant difference between the formalin-treated and untreated, is exhibited by the pupil, hence the algorithm focused on extracting features from the pupil.

Trials presented in Table 2 describe the algorithm training was run with the same process, however, the image augmentation is randomized for each trial. An image may be skewed 10% rightward or enlarged 5% before being fed in the algorithm. This is done to (1) artificially enlarge the limited dataset of 756 *C. chanos* images, and (2) take account of human bias in taking pictures. This randomized image augmentation affects the capability of the algorithm to detect formalin in fish samples if such algorithm is retrained, hence the study has done five trials to see the consistency of the algorithm.

While the samples consistently achieved above threshold AUROC curve of 95% confidence, the number of false classifications for the formalin-treated class greatly varied. This pertains to trial 5 classifying 13 out of 84 images falsely as formalin-treated as compared to trials 1-4 only having false classifications ranging from 3-4 images. Image augmentation is one of the reasons for such skewed results of trial 5 compared to other trials. Theoretically, if a formalin-treated image is randomly enlarged or skewed, its pupils would be beyond the bounds of the filter map of the algorithm. Hence, the pupil of an image is accidentally rectified or extracted by the algorithm. This implies that an algorithm's filter interpretability should be improved by training the filter to map the bounds of the region of interest [18], specifically of the pupil of the eye irrespective of how skewed, enlarged, or manipulated the image is.

Another reason for the false classifications that occurred was the presence of pupils. Days one and two of the untreated dataset exhibited rectified pupils. By basing on the framework in Figure 5, untreated images should not have pupils that are rectified. This is because the difference in the mean eye turbidity values for days one and two of formalin-treated and untreated classes of *C. chanos* is nearer as compared to the differences in these two classes observed beyond two days [10]. However, statistical analysis for each day of treatment is needed to support this claim.

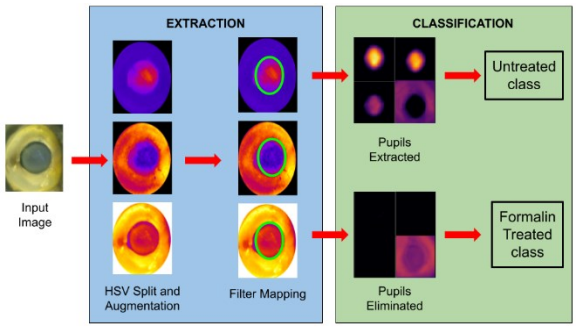

Figure 5. Framework of sssNet-SVM algorithm in classifying the *C. chanos* dataset.

Limitations. The qualitative description of describing the feature maps of the optimal algorithm partially supports the claim that the optimal algorithm is effective in classifying *C. chanos* images. A supposed quantitative assessment of the feature maps, such as calculating the receptive field of the algorithm's filters with respect to the training data [19] is beyond the capabilities of this study since such empirical method shall only be applicable if the sssNet-SVM algorithm was proven to be accurate. The researchers were unable to perform quantitative analysis for this claim.

Conclusion. - The sssNet-SVM algorithm developed by Nagata et al. [13] is accurate in identifying formalin presence in images of *Chanos chanos* with a 98.16-99.15% AUROC curve. However, due to the varied confusion matrix trials as well as pupils extracted found in feature maps of *C. chanos* images doused with formalin one or two days after immersion, the sssNet-SVM will be challenged to classify *C. chanos* doused with formalin if the image analyzed was captured one or two days after its supposed dousing. It is advised that the sssNet-SVM algorithm is to be used solely to delimit the number of commercially available *C. chanos* needed to be tested for formalin presence using spectro-photometry analysis.

Recommendations. - Using Grad-CAM analysis for quantification of feature maps is highly encouraged to measure the extent of feature maps between formalin-treated and untreated samples. A newer study may also want to integrate the algorithm into a mobile application to utilize a camera for in-site analysis.

Acknowledgment. - This study would not be done without the help of Mr. Rajo Christian G. Cadorna, Ms. Maxine P. Chan, and Mr. Juan Paulo Miguel I. Salmon, for the dataset that they provided was easily accessible and of high quality. We would also like to acknowledge their support in our research endeavors.

References

[1] [FAO]. Food and Agriculture Organization of the United Nations. 2020. The State of World Fisheries and Aquaculture 2020. Sustainability in action. Rome (Italy): FAO. Retrieved October 27, 2020, from: https://doi.org/10.4060/ca9229en.

[2] [PSA] Philippine Statistics Authority. 2020. Performance of Philippine Agriculture 2020. Manila (Philippines): Philippine Statistics Authority. [Internet]. Accessed October 20, 2020.

[3] Shawyer M, Pizzali AM. 2003. The use of ice on small fishing vessels (No. 436). FAO Fisheries.

[4] Uddin R, Wahid MI, Jasmeen T, Huda NH, Sutradhar KB. 2011. Detection of formalin in fish samples collected from Dhaka City, Bangladesh. Stamford Journal of Pharmaceutical Sciences. 4(1): 49–52.

[5] Villaflores RL. 2018. Effect of Formalin on the Sensory and Some Biochemical Attributes of Fish. IJHSS. 8(4): 82–85.

[6] [IARC] International Agency for Research on Cancer – Working Group on the Evaluation of Carcinogenic Risks to Humans. 2012. Chemical Agents and Related Occupations [Internet]. Lyon (FR): IARC; [cited 2021 May 15]. Available from: https://www.ncbi.nlm.nih.gov/books/NBK304416.

[7] Yeasmin T, Reza MS, Shikha FH, Khan MNA, Kamal M. 2010. Quality changes in formalin treated rohu fish (*Labeo rohita*, Hamilton) during ice storage conditions. Asian Journal of Agricultural Sciences. 2(4): 158–163.

[8] Jaman N, Hoque MS, Chakrabort, SC, Hoq ME, Seal HP. 2015. Determination of formaldehyde content by spectrophotometric method in some freshwater and marine fishes of Bangladesh. Intl. J. Fish. Aqua. Stud. 2(6): 94–98.

[9] Rahman S, Majumder MAA, Ahasan R, Ahmed SM, Das P, Rahman N. 2016. The extent and magnitude of formalin adulteration in fish sold in domestic markets of Bangladesh: a literature review. International Journal of Consumer Studies, 40(2): 152–159.

[10] Cadorna RCG, Chan MP, Salmon JPMI, Salazar GU. 2020. Measurement of eye turbidity of formalin-treated *Chanos chanos* (milkfish) using image analysis. Publiscience. 3(1): 84–89.

[11] Tolentino LKS, Orillo JWF, Aguacito PD, Colango EJM, Malit JRH, Marcelino JTG, Nadora AC, Odeza AJD. 2017. Fish freshness determination through support vector machine. Journal of Telecommunication, Electronic and Computer Engineering (JTEC). 9(2–5): 139–143.

[12] Navotas IC, Santos CNV, Balderrama EJM, Candido FEB, Villacanas AJE, Velasco JS. 2018. Fish identification and freshness classification through image processing using artificial neural network. ARPN Journal of Engineering and Applied Sciences. 13(18): 4912–4922.

[13] Nagata F, Tokuno K, Nakashima K, Otsuka A, Ikeda T, Ochi H, ... & Habib MK. 2019. Fusion Method of Convolutional Neural Network and Support Vector Machine for High Accuracy Anomaly Detection. In 2019 IEEE International Conference on Mechatronics and Automation (ICMA). pp. 970–975. IEEE.

[14] Yusuf M, Theophilous S, Adejoke J, Hassan AB. 2019. Web-Based Cataract Detection System Using Deep Convolutional Neural Network. In 2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf) pp. 1–7. IEEE.

[15] Aggarwal CC. 2018. Neural networks and deep learning. Switzerland: Springer Nature Switzerland AG; [accessed October 21, 2020].

[16] Moroney L, Vanderplas J. 2019. Colabs for ML Day Tokyo. San Francisco (CA): GitHub; [accessed 2020 October 18]. Available from: https://bit.ly/3feOzBe.

[17] Teow MY. 2017. Understanding convolutional neural networks using a minimal model for handwritten digit recognition. In: 2017 IEEE 2ndInternational Conference on Automatic Control and Intelligent Systems *(I2CACIS)* pp. 167–172. IEEE.

[18] Zhang Q, Wu YN, Zhu SC. 2018. Interpretable convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 18-22 June 2018; Salt Lake City, UT, USA. Computer Vision Foundation. p. 8827–8836.

[19] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. c2015. Object detectors emerge in deep scene CNNs. In: IEEE International Conference on Learning Representation; 7-9 May 2015; San Diego, CA, USA. ICLR.