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	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.

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.
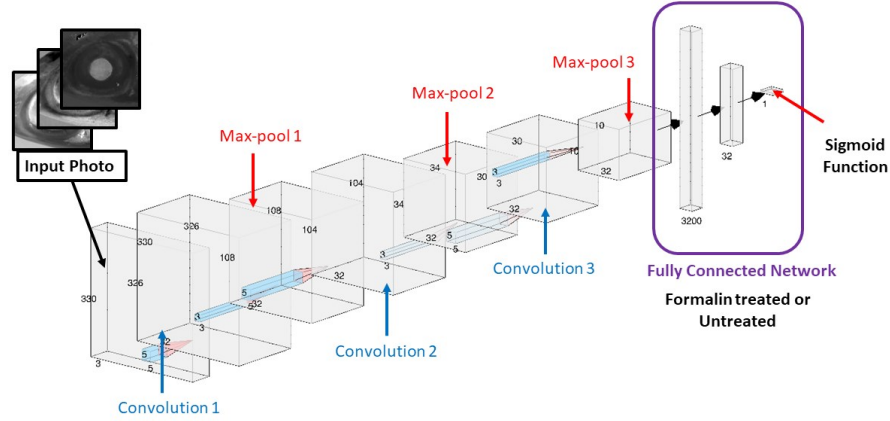

Figure 1. The sssNet-SVM algorithm diagram.

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:

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

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.

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

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

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

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 $\rho_f = 0.99$ is also initialized with the Keras module.

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

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 α 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 1×10^{-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.
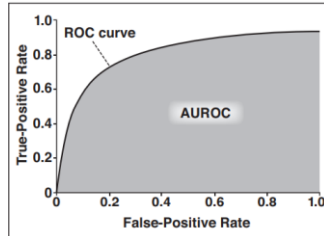

Figure 2. AUROC curve [14].

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.
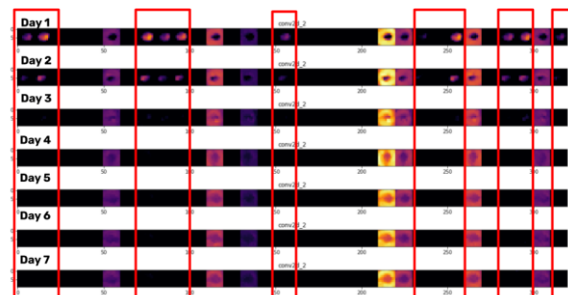

Figure 4. Final feature extraction layers of formalin-treated *C. chanos* sample #28, left-eye, in seven days.

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.
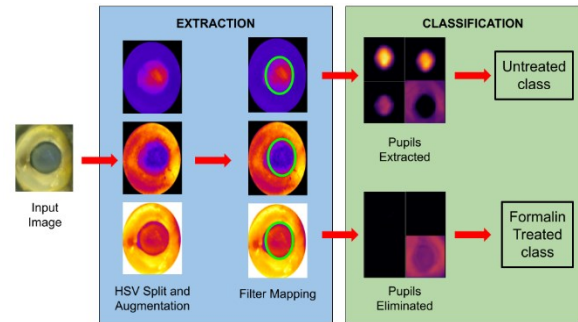

Figure 5. Framework of sssNet-SVM algorithm in classifying the *C. chanos* dataset.

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.

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.