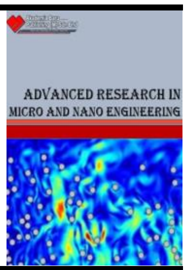
Penerbit Akademia Baru

Journal of Advanced Research in Micro and Nano Engineering

Journal homepage:
https://www.akademiabaru.com/submit/index.php/armne/index
ISSN: 2756-8210



Comparison of Conventional CNN Sequential API and Functional API for Microalgae Identification

Sri Dewi Hisham[1], Shaza Eva Mohamad[1,*], Mohd Ibrahim Shapiai[1], Koji Iwamoto[1], Aimi Alina Hussin[1], Norhayati Abdullah[1], Fazrena Nadia Md Akhir[1]

[1] Malaysia–Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 17 December 2023
Received in revised form 25 January 2024
Accepted 27 February 2024
Available online 30 March 2024

Keywords:
Deep learning; microalgae; identification; convolutional neural network; artificial intelligence

ABSTRACT

Microalgae is widely known for its application in producing biodiesel and other health supplements. However, microalgae are also the leading cause of harmful algae blooms that may affect consumers and sea wildlife. The current microalgae identification method requires professionals, resources, budgets, technologies, and time. Therefore, a novel approach to identifying microalgae has been produced by implementing deep learning, specifically the Convolutional Neural Network (CNN). Due to the blooming of research in the deep learning field for microalgae identification, this research aims to compare application programming interface (API) use and study its effects on the accuracy and loss of a model. Using a light microscope, the microalgae images' datasets are self-collected from the AlBio laboratory at the Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia. The microalgae species were *Acutodesmus obliquus*, *Monoraphidium* sp, *Spirullina* sp, *Tetradesmus deserticola*, and *Desmodesmus perforatus*. The architecture used to identify the microalgae in this research was the conventional CNN with different APIs, functional and sequential. The functional API resulted in 0.85 accuracies and a loss score of 3.77. On the other hand, the sequential scored 0.89 and a loss of 0.32. This study concluded that the sequential API was better than the functional API for a linear convolutional neural network. However, further improvement to the model could be applied by applying better hyperparameters and parameters to prevent underfitting and improve the model's accuracy.

1. Introduction

The unicellular species known as microalgae are found in freshwater, brackish water, and wastewater and range in size from microscopic to seaweed species [1] is one of the many eukaryotic microorganisms that exist in this world but have not been fully utilized. Only 167,381 of the estimated millions of species have been recorded in the AlgaeBase (*http://www.algaebase.org*)[1] database as of August 2022 [2]. However, the commercial value of microalgae production has rapidly increased over

* *Corresponding author.*
E-mail address: shaza@utm.my

https://doi.org/10.37934/armne.17.1.96104

the year [3]. Microalgae, also called "Green gold," which has a wide range of carbon compounds, can be used to make biofuels, cosmetics, and a variety of medicines [4]. Apart from providing sustainable and renewable source of energy such as biofuel produced from microalgae, this approach could ultimately serves as an alternative to fossil fuels, which the extensive consumption of these may be detrimental to environment and led to global climate change [5]. As a result, by 2018, Malaysia had made a significant investment of US$1.5 billion in the microalgae consumption sector [6]. Additionally, because microalgae have sustainable and renewable resources, they can be utilized to produce medications, and food ingredients.

Microalgae, however, can also negatively affect the environment, such as the harmful algae bloom (HAB). Harmful algal blooms (HAB), also called "Red Tide," have been happening in Malaysia for the past few years. This phenomenon has significantly impacted the survival rate of marine life, and some species may be harmful to consumers. For example, when consumed by humans or marine mammals, certain microalgae can accumulate in shellfish and release paralytic toxins that may harm other creatures. In addition, microalgae that produce toxins that attack fish gills cause fish kills. As a result, the fish dies because its ability to transport oxygen through its gills is reduced [7].

Despite the many advantages and disadvantages of microalgae, the microalgae identification process still largely relies on conventional methods, including research laboratories and experts with years of experience. Besides, traditional microalgae identification entails several steps. Microalgae samples must first be purified to create pure or single-species cultures. Microscopes for morphological analysis and DNA sequencing for molecular identification usually come next [8]. These procedures require highly technical expertise; the outcomes may differ depending on the expert's skill. These require a lot of resources, time, budget, technology, and technical knowledge.

In 2017, a new method to identify microalgae implied using deep learning, specifically Convolutional Neural Network (CNN), to identify microalgae [9]. However, due to the ongoing challenges and considering CNN model is still in early development, the reported studies exhibit inconsistent results in terms of accuracy and loss, which can be attributed to various factors, including the specific model used. To the best of our knowledge, there has been no research conducted to compare the performance of different Application Programming Interfaces (APIs) in the context of CNN architecture, namely the functional and sequential. Therefore, this study aims to revise the impact of APIs when constructing CNN architecture and determine which approach is optimal for a linear conventional CNN architecture.

2. Methodology

2.1 Data Collection

The dataset was collected at the Algal-Biomass (AlBio) laboratory at the Malaysia-Japan International Institute of Japan (MJIIT), Universiti Teknologi Malaysia (UTM). Five microalgae species, namely *Acutodesmus obliquus*, *Monoraphidium* sp., *Spirullina* sp., *Tetradesmus deserticola*, and *Desmodesmus perforatus* are locally isolated in Malaysia. The images are taken from the available light microscope (BH2-RFCA) connected to the AmScope camera. The light microscope and Amscope camera require a laptop installed with AmScope software as a medium to view and capture images of the microalgae species.

Figure 1 shows the images taken at all the available magnifications, which are 20X, 40X, and 100X. The idea of using the three magnifications (20X, 40X, and 100X) for data collection is to concatenate the images. Therefore, further improvising the model as the shape of the microalgae differs at each magnification. Hence, this may help the model accurately identify each species given at any image magnification. The dimensions of the images are 2048 x 1536 in .bmp format. A total of 1000 images were taken for this research.

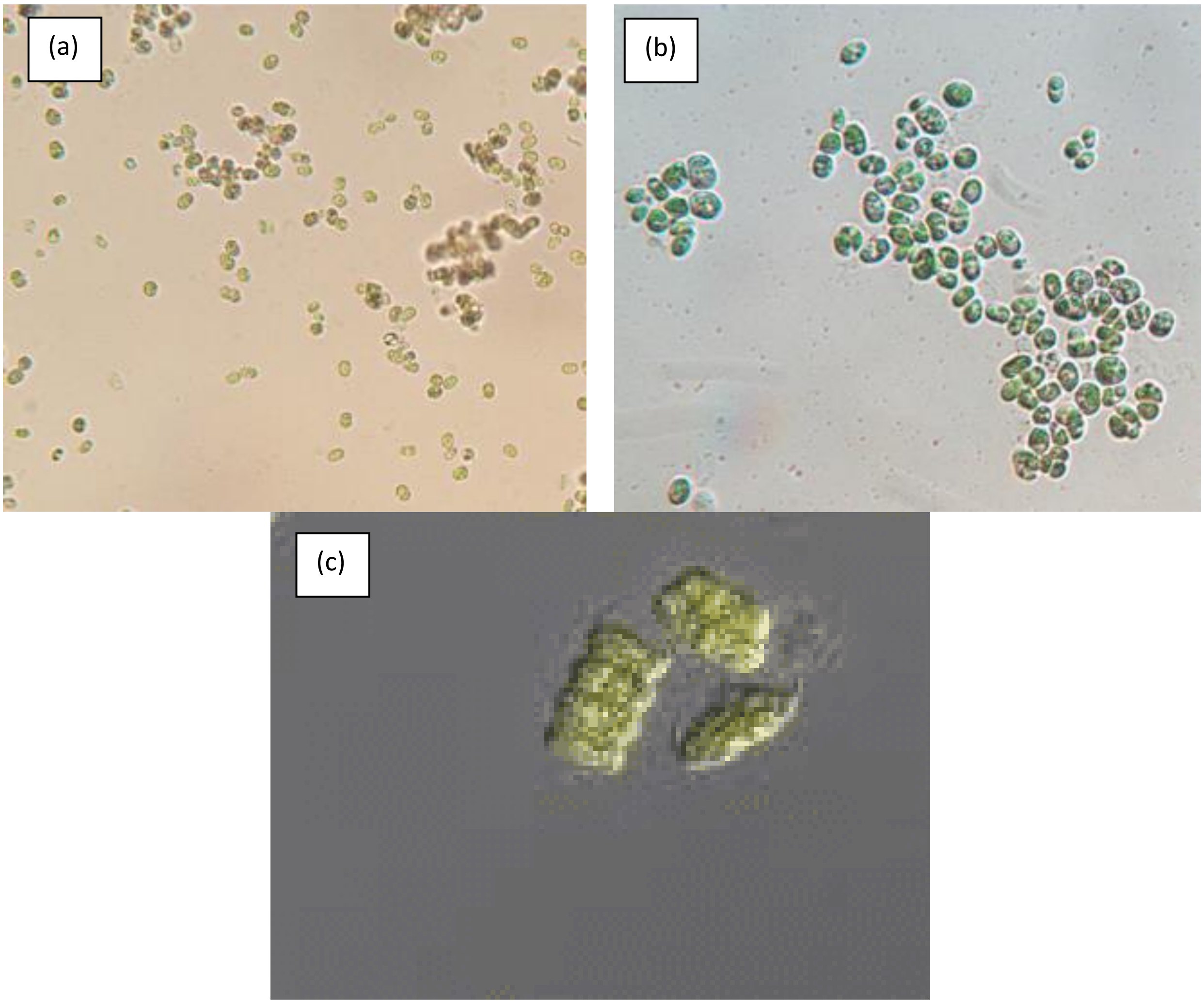


Fig. 1. Images of *Desmodesmus perforatus* at (a) 20X (b) 40X and (c) 100X magnification

2.2 Data Augmentation

The images are re-scale to 256 × 256 dimensions and are augmented before being fed to the convolutional layer. In addition, data augmentations are done to increase the number of datasets for the training and validation process of the architecture. In this research, the augmentation of the images includes random flipping, rotating, zooming, and contrast. Table 1 consists of the description of each augmentation technique used. The total dataset after augmentation is used for the model's training and validation at 80% and 20%, respectively.

Table 1
Data augmentation technique used for the microalgae images

Augmentation technique	Description
Random flipping	Flipped horizontally
Random rotation	Rotated at a random angle of 10°
Zoom	Zoom at a range of 20%
Contrast	Contrast edits in the range of 10%

Journal of Advanced Research in Micro and Nano Engineering
Volume 17, Issue 1 (2024) 96-104

2.3 Convolutional Neural Network

CNN, also known as convnets, is an artificial neural network for image recognition and processing comprised of the input, hidden, and output layers. The CNN used for image classification consists of two parts: they start with a series of pooling and convolution layers and end with a densely connected classifier [10]. The first part is called the convolutional base of the model. In the case of convnets, feature extraction consists of taking the convolutional base of a previously trained network, running the new data through it, and training a new classifier on top of the output. The input layers represent the datasets introduced to the architecture, whereas the convolutional, pooling and fully connected layers are hidden. The primary distinction between convolution and densely connected layers is that the former learns global patterns in the input feature space. Convolution layers, however, learn regional patterns [11].

Figure 2 shows the CNN architecture comprised of feature extraction (input, convolution, and pooling layer) and classification (fully connected and output). As it eliminates manual feature extraction, CNN is one of the most widely used deep neural networks [12]. For this research, the method includes using the same parameters setup of each API. Each architecture comprises 19 hidden layers, consisting of six convolutional, six normalizations, and six pooling layers. Besides, the network was trained using the open-source TensorFlow and Keras from the open library via Google Colaboratory notebook. The Google Colaboratory utilized NVIDIA K80 and GPU memory of 12GB.

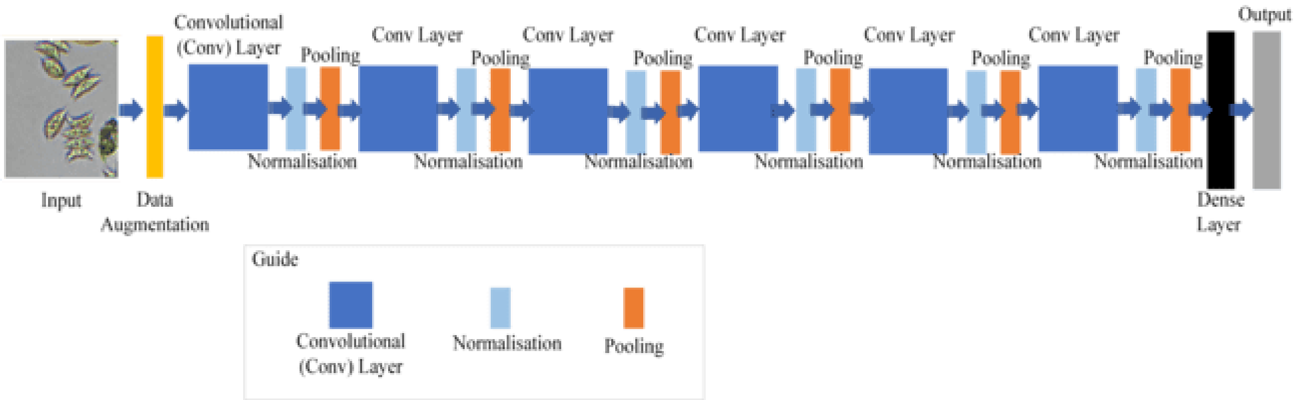


Fig. 2. CNN architecture used for this research

The same parameters are applied in this research to compare the functional and sequential architecture. Besides, the batch size is 64, running through 50 epochs. For both models, the activation functions applied are ReLU and softmax. The softmax is used at the dense layer. In addition, 0.2 dropout is implemented as it functions to shrink the squared norm of the weights and reduce the model's overfitting [13]. Also, the Adam optimizer was used with a learning rate of 0.001. Table 2 summarises the parameter used for both functional and sequential API architecture.

Table 2
Functional API and Sequential API architecture parameter setup

Parameter Setup	
Number of Epochs	50
Batch Size	64
Strides	(1,1)
Padding	'same'
Number of classes	5
Activation function	ReLU, softmax
Dropout	0.2
Optimizer	"Adam"
Learning rate	0.001
Data augmentation	Yes

2.4 CNN Functional API and Sequential API

Two APIs require consideration upon implementing the CNN architecture: functional API and sequential API. Figure 3 shows the visualization of the sequential API, whereas Figure 4 shows the sequential API allows the creation of a model by stacking layers. Sequential API is most used compared to functional API as it is easy and efficient. However, it does not offer as much flexibility as the functional API. Based on Figure 3, the functional API, on the other hand, allows the layers to connect more than just the linear layers but also the branched layers and enable the user to connect and stack the layers to different layers. Also, most complex networks implemented the use of functional API, such as the Residual Network.

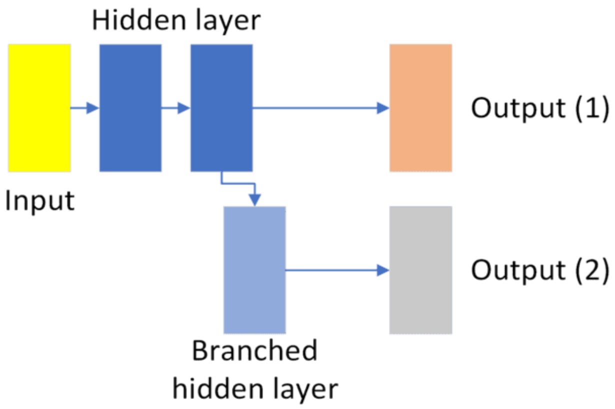


Fig. 3. CNN architecture for functional API

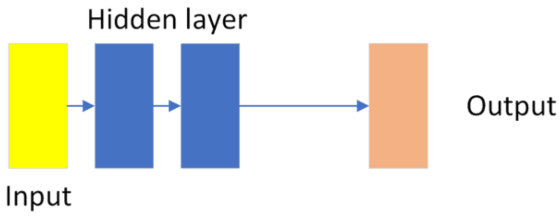


Fig. 4. CNN architecture for sequential API

Choosing which API to implement before running the architecture is essential, as the model would not be able to train appropriately without a proper API. However, little research has shown the difference in choosing a specific API. Therefore, by using two different APIs on the same set of CNN architecture and dataset, a comparison will be made to see the difference and the significance of choosing the correct API.

3. Results

3.1 Data Analysis

All the 1000 images self-collected using the light microscope were augmented before being used as input of the CNN architecture. Data augmentation was done to improve the number of data by transforming the data [14]. In addition, data augmentation was essential to reduce overfitting in models and contribute to higher model accuracy [15]. Figure 5 depicts examples of images after random flipping, rotating, zooming, and contrast for the microalgae images. All the images produced were added to the dataset and used to train and validate the model's accuracy.

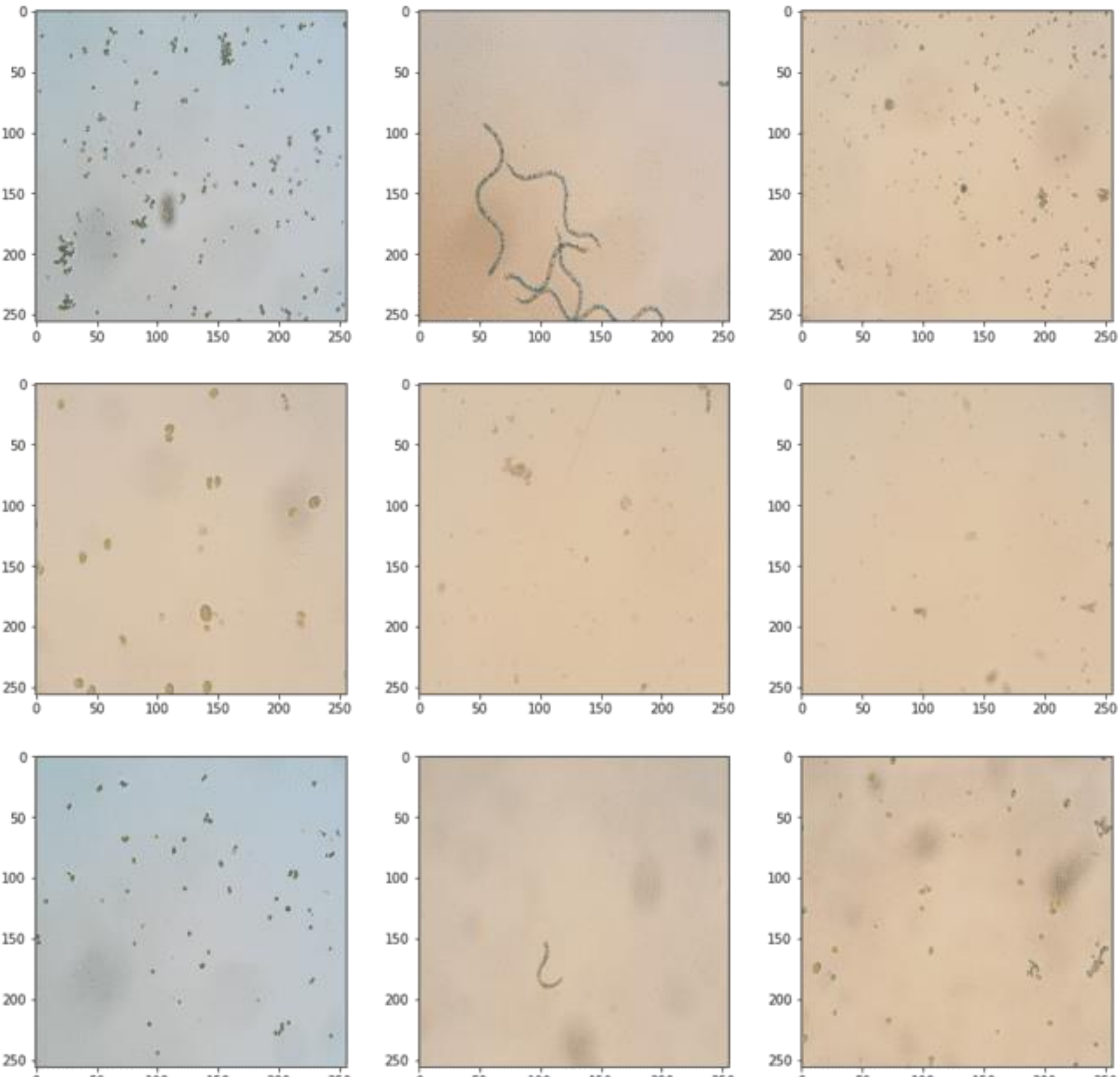


Fig. 5. Data augmentation for microalgae images

3.2 Training and Validation

Using the collected dataset and implementing data augmentation, the images are input for the CNN model for both the functional and sequential API. By the end of 50 epochs, two sets of graphs for each model were presented. Figure 6 and 7 represents the graph of the functional API model, the accuracy, and the loss, respectively. For five classes comprised of *Acutodesmus obliquus*, *Monoraphidium* sp., *Spirullina* sp., *Tetradesmus deserticola*, and *Desmodesmus perforates*, an accuracy of 0.84 was achieved with a model loss of 3.77.

Figure 6 shows that the model's training accuracy learning curve has reached a linear trend, whereas the validation accuracy is still increasing. The increasing trend is because the learning process of the model is still ongoing [3]. Therefore, the validation results have not yet reached an endpoint, as the graph shows a significant gap between the validation and training accuracy [16]. Whereas Figure 7 shows the loss score of the functional model. The learning curve for both the training and validation loss score shows a consistent declination, proving that the model is under-fitted. An under-fitted model could be summarised when the learning curve for the training loss shows a continuous declination until the end of the training [11].

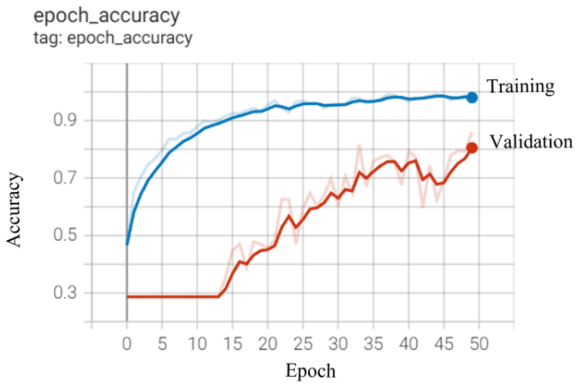


Fig. 6. Graph of functional model accuracy against the number of epochs

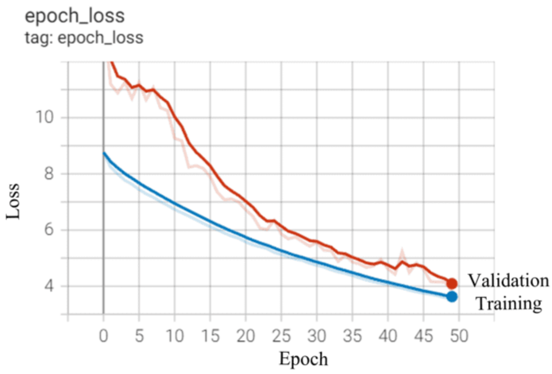


Fig. 7. Graph of functional model loss against the number of epochs

On the other hand, the sequential API architecture accuracy graph (Figure 8) learning curve shows a vast difference between the training and validation. Similarly, Figure 6 depicts the increasing trend of the validation accuracy, for Figure 8 shows the model is still learning. For both models, the accuracy can still be improved by increasing the number of epochs [9] to provide a longer learning time for the model and enhance the parameters. Sequential API architecture managed to get an accuracy of 0.89. Compared to functional API architecture, the accuracy of the sequential API is higher, but the difference is insignificant (0.05).

Besides, after 50 epochs, the sequential API model scored a loss of 0.32. Therefore, in Figure 9, there is a considerable gap between the training and validation loss; a more substantial loss for validation demonstrates that the model is under-fitted [17]. The result indicates that the model could learn further as the process was prematurely halted, such as the model's accuracy; increasing the number of epochs could help improve the learning time and further reduce the model’s loss [16]. There is a vast difference between the functional and sequential API loss of 3.77 and 0.32, respectively. Therefore, the gap in both models’ loss is significant, with a difference of 3.45. Besides, it stated that the best loss is the one scored nearest to 0.0 [11]; minimizing the loss score would be best for the model.

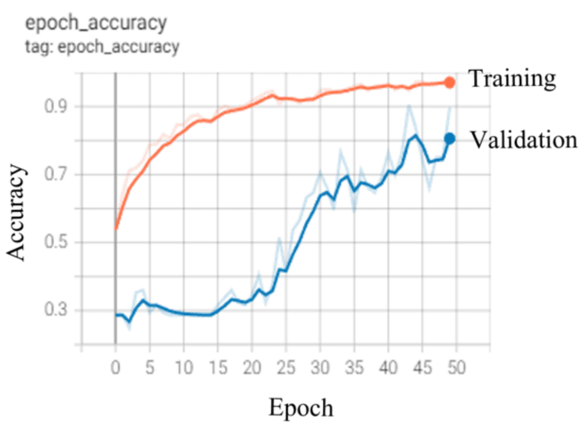


Fig. 8. Graph of sequential model accuracy against the number of epochs

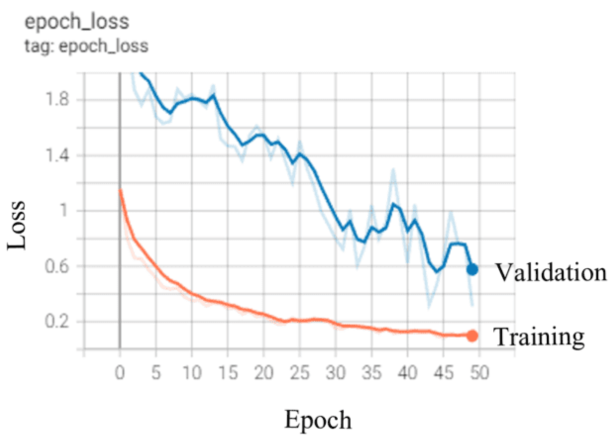


Fig. 9. Graph of sequential model loss against the number of epochs

Table 3 summarises the functional and sequential API results for both the accuracy and loss of the model. Based on the table, the sequential API performs better with an accuracy of 0.89. 0.05 higher than the 0.84 accuracies from the functional API model. Meanwhile, the sequential API's loss score is smaller than the functional one, with 0.32 and 3.77 loss scores, respectively.

Table 3
Summarisation of both Sequential and Functional API architecture's result

Architecture's API	Accuracy	Loss
Functional	0.84	3.77
Sequential	0.89	0.32

4. Conclusions

The comparison is made to assess how APIs affect architecture development and to determine which works best for a linear conventional CNN architecture. The CNN model for both APIs used in this research comprises 19 hidden layers, consisting of six convolutional, six normalizations, and six pooling layers. The parameters for both models are constant, resulting in an accuracy of 0.85 and 0.89 for functional and sequential, respectively. Meanwhile, the loss score for functional is 3.77, while sequential scored 0.32. The results signify that using sequential is best for a simple and linear architecture. Hence, choosing the sequential API beforehand would be better as it produces a model that balances the accuracy parallel to the loss. On the other hand, despite the flexibility of the functional API, its application for a simple linear architecture may cause underfit as the high accuracy does not balance the enormous loss. Therefore, the role of API before training a model plays a significant role in achieving a better result.

Acknowledgement
The research was supported by the Ministry of Higher Education (MoHE) of Malaysia through Fundamental Research Grant Scheme (FRGS/1/2020/TK0/UTM/02/88).

References
[1] Hamid, Zakri Abdul. "The power of algae," 2018.
[2] Guiry, Michael D. "How many species of algae are there?." *Journal of phycology* 48, no. 5 (2012): 1057-1063. https://doi.org/10.1111/j.1529-8817.2012.01222.x

[3] Khan, Riaz Ullah, Xiaosong Zhang, Mamoun Alazab, and Rajesh Kumar. "An improved convolutional neural network model for intrusion detection in networks." In *2019 Cybersecurity and cyberforensics conference (CCC)*, pp. 74-77. IEEE, 2019. https://doi.org/10.1109/CCC.2019.000-6
[4] Das, Probir, Siti Sarah Aziz, and Jeffrey Philip Obbard. "Two phase microalgae growth in the open system for enhanced lipid productivity." *Renewable Energy* 36, no. 9 (2011): 2524-2528. https://doi.org/10.1016/j.renene.2011.02.002
[5] Kong, Yu Man, Joon Hin Lee, Kiat Moon Lee, and Wah Yen Tey. "Techniques of improving microalgae in biomass clean energy: A short review." *Progress in Energy and Environment* (2019): 6-20.
[6] Alam, Firoz, Abhijit Date, Roesfiansjah Rasjidin, Saleh Mobin, Hazim Moria, and Abdul Baqui. "Biofuel from algae-Is it a viable alternative?." *Procedia Engineering* 49 (2012): 221-227. https://doi.org/10.1016/j.proeng.2012.10.131
[7] Tan Cheng Li. "Toxic tides: Risks from harmful microalgae," 2014.
[8] Andersen, Robert A., ed. *Algal culturing techniques*. Elsevier, 2005.
[9] Correa, Iago, Paulo Drews, Silvia Botelho, Marcio Silva de Souza, and Virginia Maria Tavano. "Deep learning for microalgae classification." In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 20-25. IEEE, 2017. https://doi.org/10.1109/ICMLA.2017.0-183
[10] Yadav, D. P., A. S. Jalal, Deviram Garlapati, Kaizar Hossain, Ayush Goyal, and Gaurav Pant. "Deep learning-based ResNeXt model in phycological studies for future." *Algal Research* 50 (2020): 102018. https://doi.org/10.1016/j.algal.2020.102018
[11] Chollet, Francois. *Deep learning with Python*. Simon and Schuster, 2021.
[12] Ongsulee, Pariwat. "Artificial intelligence, machine learning and deep learning." In *2017 15th international conference on ICT and knowledge engineering (ICT&KE)*, pp. 1-6. IEEE, 2017. https://doi.org/10.1109/ICTKE.2017.8259629
[13] Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15, no. 1 (2014): 1929-1958.
[14] Shijie, Jia, Wang Ping, Jia Peiyi, and Hu Siping. "Research on data augmentation for image classification based on convolution neural networks." In *2017 Chinese automation congress (CAC)*, pp. 4165-4170. IEEE, 2017. https://doi.org/10.1109/CAC.2017.8243510
[15] Perez, Luis, and Jason Wang. "The effectiveness of data augmentation in image classification using deep learning." *arXiv preprint arXiv:1712.04621* (2017).
[16] Hang, Jie, Dexiang Zhang, Peng Chen, Jun Zhang, and Bing Wang. "Classification of plant leaf diseases based on improved convolutional neural network." *Sensors* 19, no. 19 (2019): 4161. https://doi.org/10.3390/s19194161
[17] Emmert-Streib, Frank, and Matthias Dehmer. "Evaluation of regression models: Model assessment, model selection and generalization error." *Machine learning and knowledge extraction* 1, no. 1 (2019): 521-551. https://doi.org/10.3390/make1010032