Vol 5 No 2 (2023) *e*-ISSN: 2682-8383

Journal of Engineering Technology and Applied Physics

Integrated Circuit Packaging Recognition with Tilt Auto Adjustment using Deep Learning Approach

Siu Hong Loh[1,*], Peh Chiong Teh[1], Yong Kang Lee[1], Jia Jia Sim[2] and Kim Ho Yeap[1]
[1]*Faculty of Engineering and Green Technology, Universiti Tunku Abdul Rahman, 31900 Kampar, Malaysia.*
[2]*Faculty of Business and Finance, Universiti Tunku Abdul Rahman, 31900 Kampar, Malaysia.*
[*]*Corresponding author:* lohsh@utar.edu.my, *ORCiD:* 0000-0002-4192-1477
https://doi.org/10.33093/jetap.2023.5.2.9
Manuscript Received: 13 April 2023, Accepted: 6 May 2023, Published: 15 September 2023

***Abstract* — A deep-learning-based approach for recognizing integrated circuit (IC) packaging type is presented in this paper. The objective of this work is to design a deep-learning method that can recognize multiple types of packaging per detection, performing counting operations, and calculating the centre location of an IC with its tilting angle. The transfer learning from model You-Only-Look-Once (YOLO) v5 was chosen because it has been trained with the coco dataset and has a more reliable feature extraction system than the other models. In order to extract data from images, OpenCV was used, which allows the deep learning model to perform more efficient analysis of the input data. Apart from that, the principal component analysis (PCA) was used to estimate the angle of the IC in order to determine the rotation of each IC for the purpose of tilting adjustment. The developed model has an average confidence score of 85% and is capable of operating in a variety of conditions, as demonstrated by ANOVA analysis.**

Keywords— Artificial intelligence, Deep learning, Integrated circuit packaging type, Image recognition, ANOVA analysis

I. INTRODUCTION

Artificial intelligence (AI) is defined scientifically as the study and creation of a branch of intelligent machines that are created to comprehend with the environment logically and perform intelligent actions towards the stimulations [1]. Developing a computerised system which is able to complete tasks normally require human intellect is referred to as AI development [2]. This includes visual observation, speech recognition and translation across various language. An essential part of AI is its ability to make recognition algorithms work more efficiently [3]. AI could be used to identify products and perform calculations in a very short time in comparison to human effort. Image recognition works by gathering enough organized data that is classified with images separating all the unique characteristics of the detecting object [4]. There are two major approaches of training AI for image recognition. Firstly, training models can be done from scratch by collecting all the image data and classifying them while training. The other method is using the trained deep learning model. These pre-trained models are helpful in object recognition.

Introduced by Hinton *et al.* [5], deep learning (DL) is a technique that replicates how human learn and analyse data. DL is particularly useful for information scientists tasked with gathering, processing, and analysing massive amount of information. DL works at a higher level of abstraction and utilizes many representation layers. Some of the great examples of DL technology are segmenting analysis, speech interpreting and classification of images [6]. For the old machine learning method, the feature extraction process requires a significant amount of time for training. The success rate on the classification of the image depends on the ability of the programmer to accurately state the properties of the object of interest. With DL technology, the program is capable of building the classification by itself without supervision, resulting in a more accurate and faster outcome than a standard machine learning approach.

There is a large quantity of information applied to train a model in DL. Hence, a technique known as the transfer learning can be adopted to improve efficiency when training new models. Usually, transfer learning will be applied if the new data sets have similarities to the pre-trained model [7]. Training large data sets from scratch might take up

Published by MMU PRESS. URL: https://journals.mmupress.com/index.php/jetap/index

significant amount of time. Hence, training a new model on the pre-trained model can pace up the learning process of the model. One of the popular object detection models recently is YOLO (You Only Look Once), first introduced by Redmon *et al.* in 2015 [8]. Using neural networks technique, YOLO is famous for its speed and accuracy. In this work, we adopted transfer learning from model YOLO version 5 (YOLOv5) to identify integrated circuit (IC) package type. Additionally, the angle of tilt of each IC in the image could be calculated.

There were previous works developed to recognize IC package type. Blaes and Young presented an image processing system that capable of detecting ICs on printed circuit board (PCB) and reading part labels using optical character recognition (OCR) [9]. However, no AI algorithm was used in this study. Using DL neural network technique, Voon et al. proposed a system that is able to detect ICs and recognize text label printed on IC packages [10]. Reza et al. addressed two specific problems in their research work: electronic component detection and verification. They introduced a technique to locate ICs on PCBs using deep-learning approach with convolutional neural networks (CNN) [11]. In another study, Reza and Crandall successfully demonstrated that in their developed algorithm, an IC image could be compared with a known reference image to verify that it is the same device [12].

This work provides several contributions as compared to previous studies. Firstly, our developed deep-learning model can detect up to five main classes of IC packages and accurately measure the angle of rotation simultaneously. Secondly, we further conduct statistical analysis to evaluate the performance of the detection and counting process of the IC packages. Hence, this provides some insights on the factors that affect the detection process. To the best of our knowledge, our work here is among the first to conduct statistical analysis on the factors affecting detection process of IC package types.

II. METHODOLOGY

This research work is composed of three major components. Using Google Colab, the first part of the project involves the AI training process, which include data preparation, model augmentation, and training of the AI model. Eventually, the model with the highest success rate will be passed to the Python environment for AI detection. In this project, a total sample of dataset consisted of 2957 images. Figure 1 shows the flowchart of the overall project design and implementation.

The first part of the project (contents inside the yellow box in Fig. 1) is further illustrated in Fig. 2. To increase the dataset for the image, Keras Library provide series of function which name *ImageDataGenerator* in which it can perform multiple transformation to process the images using the function. Multiple augmentation is set to increase the dataset. The augmentation performed include rotation, height shift, width shift, flipping, brightness and channel shift. This step is to allow the AI to have a better learning process and prevent overfitting to occur. To begin with, the path for the input image is declared (Fig. 3(a)). Next, a code segment is executed to select the augmented method (Fig. 3(b)), followed by performing augmentation random and multiple times (Fig. 3(c)). After that, the next step is to setup repository for the YOLO environment (Fig. 3(d)). Lastly, AI training is initiated (Fig. 3(e)).

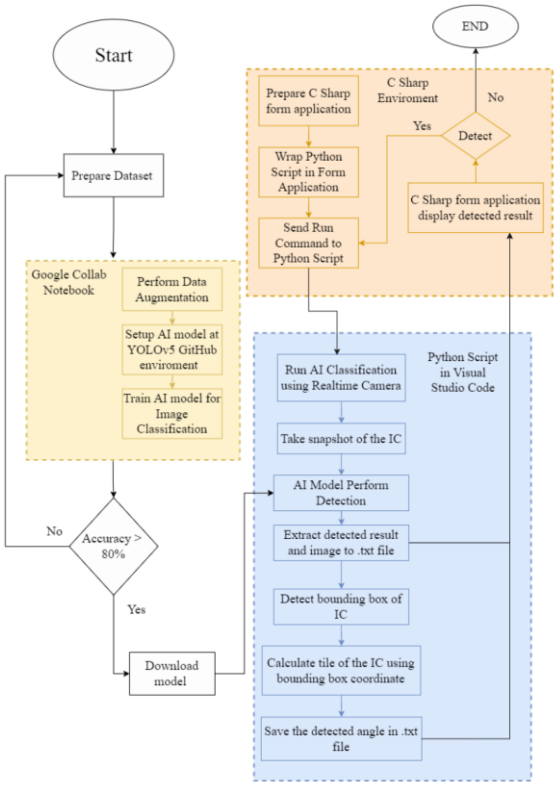


Fig. 1. Flowchart of the overall project design and implementation.

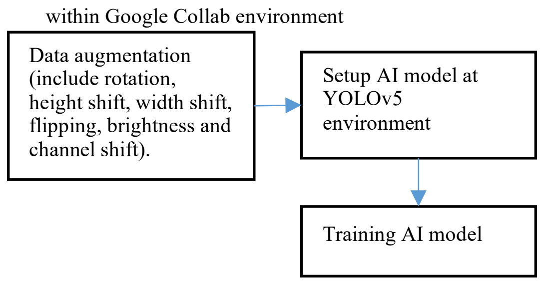


Fig. 2. Flowchart of the AI training process.

The Python environment and the C Sharp form application are the next two parts, with the Python environment performing the majority of the functions, such as IC detection, IC counting, centre detection, and angle calculation. On the other hand, the C Sharp form application performs the remaining functions. In the first instance, all of the output results will be displayed on the C Sharp form application's screen. The C Sharp form application then functions as a user interface application that incorporates all of the processing functions into a single package. The primary function is to select the camera input type and to run detection on the integrated circuit. Both Python and C Sharp will communicate with one another through the use of a text file, in which they will both check for the run and complete commands in the text file simultaneously.

```
from keras.preprocessing.image import ImageDataGenerator
from skimage import io

DIR = '/content/drive/MyDrive/YoloV5'
SAVE = '/content/drive/MyDrive/Augmented'
```

(a)

```
[ ] datagen = ImageDataGenerator(
        rotation_range=45,      #Random rotation between 0 and 45
        width_shift_range=0.2,   #% shift
        height_shift_range=0.2,
        zoom_range=0.5,
        brightness_range=[0.5,2],
        horizontal_flip=True,
        fill_mode='reflect')    #Also try nearest, constant, reflect, wrap
```

(b)

Perform augmentation

```
i = 0
for batch in datagen.flow_from_directory(directory= DIR,
                                          batch_size=16,
                                          target_size=(256, 256),
                                          color_mode="rgb",
                                          save_to_dir=SAVE,
                                          save_prefix='aug',
                                          save_format='jpg'):
    i += 1
    if i > 100:
        break
```

(c)

Setup

Clone repo, install dependencies and check PyTorch and GPU.

```
!git clone https://github.com/ultralytics/yolov5  # clone repo
%cd yolov5
%pip install -qr requirements.txt  # install dependencies

import torch
from IPython.display import Image, clear_output  # to display images
```

(d)

```
# Train YOLOv5s on COCO128 for 3 epochs
!python train.py --img 640 --batch 16 --epochs 15 --data custom_data.yaml --weights yolov5s.pt --cache
```

(e)

Fig. 3. Code fragments to setup AI model at YOLOv5 environment: (a) Declare path for input and saving directory, (b) Code segment for selecting augmentation method, (c) Code segment to run the augmented image, (d) Code segment for setting up YOLOv5 environment and (e) Code segment for AI training.

The Python script will be able to start the IC detection when the Run command is issued from the C Sharp form application. The C Sharp form application will capture and save input images in a shared directory, which will be accessible to all users. The Python script will then read the input image from the image captured by the C Sharp form application in the shared directory, and the detected image with bounding box and border detection will be placed in another shared directory so that the C Sharp form application can display the results. Furthermore, the detected information will be saved in a text file, and the C Sharp application will read the output information from the txt tile, followed by a complete command from the Python script, which will cause the C Sharp form application to display the result to the user, as shown in Fig. 4. After the bounding box plotting, the angle of rotation will be estimated using principal component analysis (PCA), a function available in the OpenCV library.

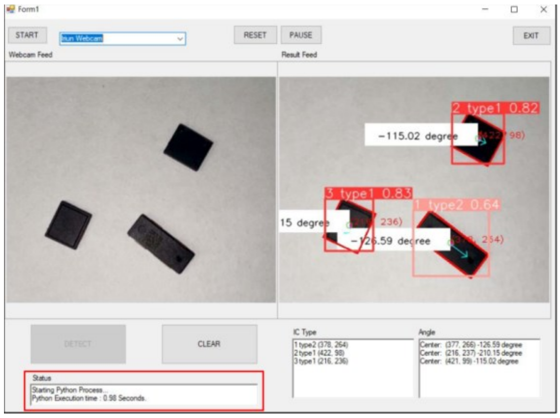


Fig. 4. The graphic user interface showing various output information.

III. RESULTS AND DISCUSSIONS

Firstly, several deep learning models were trained from the YOLOv5 environment for the implementation steps. This includes a batch number from 4, 8, 16, and 20. Then, analysis based on the model's accuracy was evaluated and the best model for detection was determined. Next, the evaluation of the detection accuracy and the centre detection of the model with the counting function were analysed. The benchmark for the confidence score is 80%. Finally, with the success of the model creation, the implementation of the whole system was analysed using ANOVA analysis.

A. YOLOv5 Training Model

During the model training process, we varied the number of epochs and batch size to observe the optimum combination. We observed that the best performing model was the retrained YOLOv5 model with the batch size of 20 and the epochs number of 50. In addition, we used the train-test ratio of 80:20, adopted from the well-known Pareto principle [13]. In other words, about 600 images were used for testing (from a total of 2957 images). The experimental results showed that the model achieved an average prediction score of 80%. Figure 5 shows the prediction result for different classes. There were five main classes for the IC package product which included DIP 6 (Double In-line Package 6-pin), DIP 14 (Double In-line Package 14-pin), QFP (Quad Flat Package), SOP (Small Outline Package) and TO-220 (Transistor Outline). Almost all ICs were labelled correctly with high accuracy.

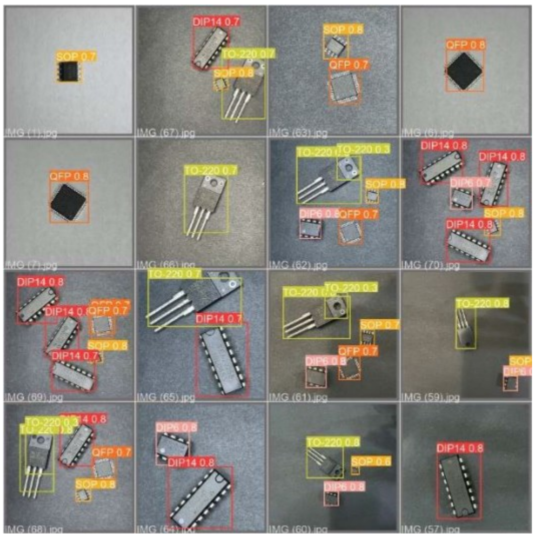


Fig. 5. Prediction results for multiple classes.

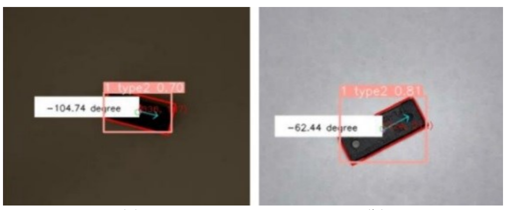
B. *ANOVA Analysis*

The performance of the detection and counting process of the IC were affected by several factors. We conducted one-way Analysis of Variance (ANOVA) [14], a statistical technique to obtain the optimal condition for the detection process. There were previous studies which have adopted ANOVA analysis to observe differences between groups of data [15–17]. For our study here, two factors were considered and evaluated in this analysis, namely: lighting conditions and multiple IC detection.

Firstly, the brightness level was adjusted to study its effect on the detection process. The focus distance was held constant. Each condition was tested 20 times and the results are shown in Table I. The twenty trials were conducted by manually capturing the IC with the camera of a smartphone. The room light was adjusted to control the brightness level, as shown in Fig. 6.

The summary of the ANOVA analysis is shown in Table II. The F ratio value and the p-value are 96.624 and 0.000, respectively, indicating that there is a statistically significant difference between groups with varied lighting conditions. Therefore, the brightness level is one of the factors affecting the detection process, and the detection algorithm will perform better at a normal brightness level.

Secondly, an experiment was conducted to study the ability of the system to detect multiple IC. The brightness level was fixed at normal brightness level. Three sets of conditions were considered, and each condition was tested 20 times and the results are shown in Table III.



(a) (b)

Fig. 6. Brightness level of the IC: (a) low and (b) normal.

Table I: Detection status and confidence score for each trial under different lighting conditions.

No. of Trials	**Low Brightness**		**Normal Brightness**	
	Status	***Confidence score***	***Status***	***Confidence score***
1	Success	70	Success	88
2	Success	81	Success	84
3	Success	70	Success	87
4	Success	74	Success	81
5	Success	75	Success	87
6	Success	77	Success	83
7	Success	70	Success	80
8	Success	78	Success	80
9	Success	77	Success	83
10	Success	76	Success	81
11	Success	79	Success	85
12	Success	79	Success	85
13	Success	76	Success	86
14	Success	77	Success	87
15	Success	75	Success	87
16	Success	76	Success	82
17	Success	74	Success	88
18	Success	71	Success	87
19	Success	71	Success	87
20	Success	77	Success	83
	SR = 100%	Average = 75.15	SR = 100%	Average = 84.55

Note: SR = success rate;

Table II: Summary of ANOVA analysis for detection process under different lighting conditions.

Source	**DF**	**SS**	**MS**	**F**	**p-value**
Between Groups	1	883.600	883.600	96.624	0.000
Within Groups	38	347.500	9.145		
Total	39	1231.100			

Note: DF = degrees of freedom; SS = sum of squares; MS = mean square

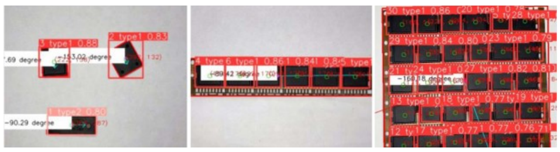
Table III: Detection status and confidence score for each trial with multiple IC detection.

No. of Trials	**Multiple IC**		**Single roll IC on lead frame**		**Multiple roll IC on lead frame**	
	Status	***CS***	***Status***	***CS***	***Status***	***CS***
1	S	87	S	86	S	81
2	S	80	S	81	S	80
3	S	82	S	85	S	82
4	S	87	S	86	S	88
5	S	82	S	85	S	89
6	S	85	S	77	S	82
7	S	83	S	82	S	81

8	S	81	S	86	S	88
9	S	87	S	81	S	84
10	S	82	S	87	S	82
11	S	80	S	89	S	89
12	S	84	S	81	S	87
13	S	87	S	82	F	83
14	S	85	S	82	S	85
15	S	82	S	81	F	81
16	S	80	S	81	F	84
17	S	85	S	86	S	84
18	S	86	S	80	F	89
19	S	81	S	88	S	89
20	S	83	S	83	S	84
	SR = 100%	A = 83.45	SR = 100%	A = 83.45	SR = 80%	A = 84.6

Note: CS = Confidence score; S = success; F = fail; SR = success rate; A = average

It could be observed from Table III that trial 13, 15, 16 and 18, under "Multiple roll IC" column did not detect and count successfully the actual number of IC in the images. This indicates that the developed system might have an accuracy problem when the number of ICs in the image exceeds certain numbers (above 30). A sample of images for multiple IC detection is depicted in Fig. 7.



(a) (b) (c)

Fig. 7. Multiple IC detection: (a) Multiple IC, (b) Single roll IC on lead frame and (c) Multiple roll IC on lead frame.

From Table IV, the F ratio value and the p-value are 1.012 and 0.370, respectively, implying a low variation of mean values among groups with different IC count. Therefore, it could be concluded that the detection of multiple IC in an image does not affect the accuracy of the counting process as long as the number does not exceed 30.

Table IV: Summary of ANOVA analysis for detection process with multiple IC.

Source	DF	SS	MS	F	p-value
Between Groups	2	17.633	8.817	1.012	0.370
Within Groups	57	496.700	8.714		
Total	59	514.333			

Note: DF = degrees of freedom; SS = sum of squares; MS = mean square

Figure 6 illustrates the box plot of the two factors, namely the lighting conditions and multiple IC detection that might affect the detection and counting results of the IC package. Table V shows the correspond summary statistic of the box plot in Fig. 8. From the box plot, the first (lighting conditions) group has larger dispersion (taller boxes) as compared to the second (multiple IC detection) group. Large data dispersion here indicates that there was a significant variation in the confidence score of the detection process. Small data dispersion is preferred as the confidence score for each trial under different conditions (different IC count) is rather similar. Therefore, it can be concluded that the brightness level affects the detection process significantly. In contrast, the detection of multiple IC in an image does not affect the accuracy of the counting process as long as the number does not exceed 30.

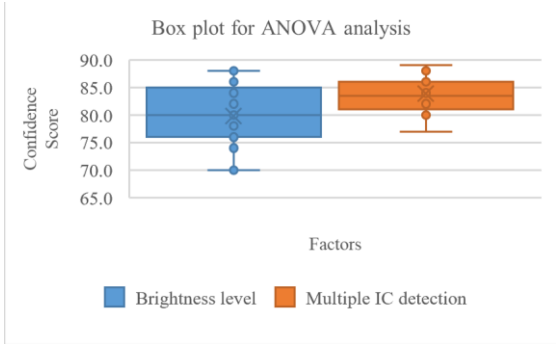


Fig. 8. Box plot for ANOVA analysis.

Table V: Summary statistics of box plot.

Summary Stats	Brightness level	Multiple IC detection
Minimum	70	77
1st Quartile: Q_1	76	81
Median	80	83.5
3rd Quartile: Q_3	85	86
Maximum	88	89
IQR	9	5
Mean	80	84

IV. CONCLUSION

This work demonstrates how to effectively integrate deep-learning-based object detection with the tilt auto adjustment feature in Python and C Sharp form applications. The YOLOv5 repository was used in the effective creation of the AI model. It was created using transfer learning and was capable of detecting different kind of IC packages in the same image, simultaneously computing the angle of each IC. With the maximum number of 30 ICs per detection in an image, the developed model can achieve a success rate of 100%, and the average confidence score of 85%. Lastly, the results from ANOVA analysis revealed that the brightness level could affect the detection process significantly.

REFERENCES

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Fourth Ed., Pearson, 2021.

[2] E. Burns, "What is artificial intelligence (AI)? - AI definition and how it works," *Enterprise AI.* https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence (Accessed on 11 April 2023).

[3] I. A. Joiner, "Artificial Intelligence," *Emerging Library Technologies*, Elsevier, pp. 1–22, 2018.

[4] E. Lisowski, "Using Artificial Intelligence (AI) for Image Recognition," *Addepto*, May 2021. https://addepto.com/blog/using-artificial-intelligence-ai-for-image-recognition/ Aaccessed on 11 April, 2023).

[5] G. E. Hinton, S. Osindero and Y. W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[6] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 6, p. 420, 2021.

[7] M. Hussain, J. J. Bird and D. R. Faria, "A Study on CNN Transfer Learning for Image Classification," *in UK Workshop Computat. Intellig.*, *Adv. Computat. Intellig. Sys.*, Springer, vol. 840, pp. 191–202, 2019.

[8] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Comp. Vision and Pattern Recogn.*, 2015, doi: 10.48550/ARXIV.1506.02640.

[9] P. Blaes and C. Young, "Mobile IC Package Recognition," Stanford University, 2008. [Online Available: https://stacks.stanford.edu/file/druid:yj296hj2790/Young_Blaes_Mobile_IC_Package_Recognition_and_Information_Retrieval.pdf]

[10] Y. N. Voon, K. M. Ang, Y. H. Chong, W. H. Lim and S. S. Tiang, "Computer-Vision-Based Integrated Circuit Recognition Using Deep Learning," in *Proc. 6th Int. Conf. Electrical, Control and Comput. Eng.*, *Lecture Notes in Electrical Eng.*, Springer, vol. 842, pp. 913–925, 2022.

[11] M. A. Reza, Z. Chen and D. J. Crandall, "Deep Neural Network–Based Detection and Verification of Microelectronic Images," *J. Hardw. Sys. Secur.*, vol. 4, no. 1, pp. 44–54, 2020.

[12] M. A. Reza and D. J. Crandall, "IC-ChipNet: Deep Embedding Learning for Fine-grained Retrieval, Recognition, and Verification of Microelectronic Images," *in 2020 IEEE Appl. Image. Pattern Recogn. Workshop*, Washington DC, pp. 1–10, 2020.

[13] V. R. Joseph, "Optimal Ratio for Data Splitting," *Statistical Analysis*, vol. 15, no. 4, pp. 531–538, 2022.

[14] R. M. Heiberger and E. Neuwirth, "One-Way ANOVA," *R Through Excel*, Springer, pp. 165–191, 2009.

[15] S. H. Loh, P. C. Teh, J. J. Sim, C. K. Tai, K. H. Yeap, Y. J. Lee and A. U. Mazlan, "Decoding Dot Peen Data Matrix Code with Deep Learning Capability for Product Traceability," *Appl. Model. and Simul.*, vol. 7, pp. 38–48, 2023.

[16] D. Mahmood, H. Nisar, V. V. Yap and C.-Y. Tsai, "The Effect of Music Listening on EEG Functional Connectivity of Brain: A Short-Duration and Long-Duration Study," *Mathematics*, vol. 10, no. 3, 349, pp. 1-19, 2022.

[17] R. Nawaz, H. Nisar, V. V. Yap and C. Y. Tsai, "The Effect of Alpha Neurofeedback Training on Cognitive Performance in Healthy Adults," *Mathematics*, vol. 10, no. 7, 1095, pp. 1-16, 2022.