UNIVERSITI MALAYSIA PAHANG

ORIGINAL ARTICLE

Badminton Smashing Recognition through Video Performance by using Deep Learning

Yip Zi Ying[1], Ismail Mohd Khairuddin[1], Wan Hasbullah Mohd Isa[1], Anwar P.P. Abdul Majeed[1], Muhammad Amirul Abdullah and Mohd Azraai Mohd Razman[1]

[1]Faculty of Manufacturing and Mechatronic Engineering Technology, Universiti Malaysia Pahang, 26600 Pahang, Malaysia.

ABSTRACT – Nowadays, badminton become the hot trends sport in Malaysia due to the influence of Lee Zii Jia which is the Malaysian badminton player and he has been participate the men's single badminton in Tokyo 2020 Olympic Game at the Musashino Forest Sports Plaza in Tokyo. Due to this reason, sport analysis become one major contribution in analysing and improving the performance of athlete. Hence, this project constructs a badminton smashing recognition through video performance by using the deep learning. The main purpose of this project is to evaluate the performance of the models in classifying the types of smashing in badminton. The models will be trained using Deep Learning models of ResNet-18, GoogleNet and VGG-16 and the best precision of badminton smashing accuracy were compared. In this project, we found that ResNet-18 has the best performance of accuracy of 97.51% and 98.86% on both training and testing datasets respectively by using the software Jupyter. On other hand, GoogleNet has the highest accuracy of 83.04% and 97.20% on both training and testing datasets respectively by using hardware Jetson Nano.

INTRODUCTION

To increase the performance of the athlete, deep learning was used in sports as the Artificial Intelligence (AI) coach. In deep learning, Convolutional Neural Network (CNN) is the common approach that has been used for sports activity recognition[1]. With the help of using deep learning in sports activity recognition, it may help the coach improve overall athletic performance by providing instant video feedback on the court[[2].

There are various types of badminton smashing. Thus, badminton smashing recognition accuracy through video performance by using deep learning was proposed in this project. Therefore, this research was carried out to identify the classes of the badminton smashing that the data is collected from the mature badminton players who have at least 5 years' experience. Secondly, we will classify out the badminton smashing out of various type of badminton smashing. After calssify the types of smashing, we will evaluate the performance of the deep learning models. The deep learning models that are going to be used in this project are ResNet-18, GoogleNet and VGG-16. In this research, we are going to evaluate the performance of deep learning models on both software and hardware Jetson Nano.

The remaining part of the paper starts with the related work and the methodology where the systematic steps on developing the training tools on both software Jupyter and hardware Jetson Nano. Then, the results were laid out as well as associated discussion and lastly to conclude the paper.

RELATED WORK

Classifier of Deep Learning

There are three common types of classifier of deep learning which are Convolutional Neural Network (CNN), Artificial Neural Network (ANN) and Recurrent Neural Network (RNN). The CNN classifier consisting of two convolutional layers, two pooling layers, a fully connected layer, and a softmax layer can be used to divide the sports activities into table tennis, tennis, badminton, golf, batting baseball, shooting basketball, volleyball, dribbling basketball, running, and bicycling, respectively[3]. The convolutional layer is the core building block of a Convolutional Neural Network (CNN)[4]. The layer's parameters consist of kernels that have a small receptive field. The output of each convolutional layer is called a feature map[5].

 The ANN is a group of multiple perceptrons or neurons at each layer and it is also known as Feed-Forward Neural network. ANN is made of three layers which are input layers, output layers, and hidden layers. The nodes in the input layer must be connected to the nodes in the hidden layer, and each hidden layer node must be connected to the nodes in the output layer. The RNN is a type of artificial neural network which uses sequential data or time-series data[6].

Computer Vision

Besides that, computer vision also can be applied in this research. Computer vision enables the computer to see, observe and understand[7].With the help of the computer vision. It may help the coaches to analyze the weakness of the player and provide the training according to the weakness of the player. With the use of computer vision, we can do the e processing, human motion tracking, color tracking, object detection and others.

Object detection is a well-known computer technology related to computer vision and image processing that focuses on finding objects or instances of a particular class, such as humans, flowers, animals in digital images and videos [8]. We can use edge matching, divide and conquer search, grayscale matching, and gradient matching to put our object detection skills into practise.

Colour Tracking is also used for tracking the athletes' movement and it is considered the easiest way to do the tracking from one frame to another frame. The players are being tracked based on the colors of the jersey. The tracking system based on the color detection may help to track the position of the players in the next frame [9].

Human Pose Estimation (HPE) has gotten a lot of attention in recent years, with deep learning boosting its performance and providing exciting new applications, such as in sports and physical activity (SPE)[10]. Human Pose Estimation applications are still in their early phases. Users can benefit from accurate pose analysis in sports footage to help them improve their skills[11]. An AI coach system was proposed to provide personalized athletic training experiences for posture-wise sports activities [11].

EXPERIMENTAL SETUP

Data Collection

The data collection on badminton smashing was carried out at Dewan Serbaguna, University Malaysia Pahang (UMP). The types of smashing that need to be collected are backhand, forehand and jump smash. The data was collected from the players who have at least 5 years experience in playing badminton. Besides that, the video was captured using EKEN H9R Action Camera and Xiaomi phone camera in Full HD quality. Figure 1 (a) shows the backhand smash that conducted by a male badminton player who has 5 years experience in playing badminton. Figure 1 (b) shows the forehand smash that conducted by a male badminton player who has 10 years' experience in playing badminton. Figure 1 (c) shows the jump smash that conducted by a male volunteer badminton player who has at least 6 years' experience.
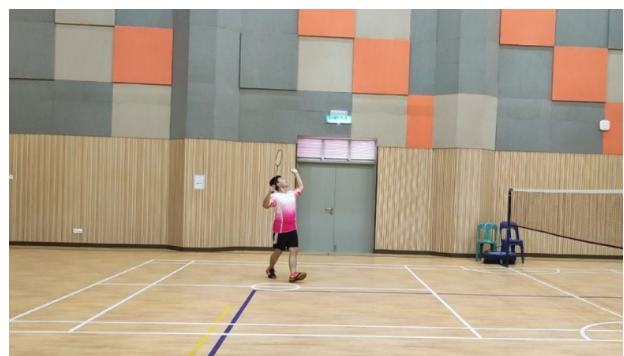
(a)   (b)

(c)

Figure 1. Smashing motion **(a)** Backhand smash **(b)** Forehand smash **(c)** Jump smash

Data Pre-processing

All of the images data were read before the dataset was split. All of the images were resized into the same shape and converted from RGB mode into grayscale. The conversion of RGB mode into grayscale may reduce the dimension of the images from 3 dimensional to 1 dimensional. The reduction in the dimension of the images can reduce the time for training the images. Besides that, it is also necessary to convert the image data into numpy array which is readable by the computer.

Train and Test Data Split

In deep learning, there are 8324 images were split into the 75% training dataset and 25% testing dataset. The training data is fed into the deep learning model to discover and learn the pattern while the testing data is used to let us know whether the model is working accurately.

Define models

In this project, we use three deep learning models which are ReNet-18, GoogleNet and VGG-16. An evaluation performance of the models will be done to determine which model has the highest accuracy in doing the classification of the smashing such as backhand, forehand and jump smash. The evaluation performance of the models will be done on both software Jupyter and hardware Jetson Nano.

Evaluate the performance of the models

a) Recall

Sensitivity, Probability of Detection, and True Positive Rate are other terms for recall. The percentage of positive predictions that are correct compared to the total number of positive examples. Below is the equation to get the value of recall through the four basic terms in confusion matrix which are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

b) Precisions

The precision is also called as positive predicted value. It is the ratio of correct positive predictions to the total predicted positives. Below is the equation for precision.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

c) F-measure

Precision and recall have a harmonic mean which is F-measure. It considers both false positives and false negatives. As a result, it works well with an unbalanced dataset. Aside from that, the F1 measure weights recall and precision equally. Below is the equation to get F-measure from four basic terms.

$$F - measure = \frac{2 * precision * recall}{precision + recall} \tag{3}$$

d) Accuracy

Accuracy is defined as the ratio of correctly predicted examples by the total examples. The higher the accuracy indicates that more accurate of the predictions in both positive and negative classes.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{4}$$

EXPERIMENTAL RESULTS

Evaluation Performance on Software Jupyter

Table 1 until Table 8 demonstrates the results obtained from comparing different models with repective performances which corresponds to the confusion matrices in Figure 2 to Figure 7.

Table 1. Recall, precision, F-measure and accuracy of ResNet-18, GoogleNet and VGG-16 for training dataset

Models		Recall	Precision	F-measure	Accuracy
ResNet-18	Backhand	0.998821	0.954904	0.976369	0.986863
	Forehand	0.973311	0.997436	0.985226	0.988786
	Jump	0.986965	0.996241	0.991581	0.994233
GoogleNet	Backhand	0.997052	0.998229	0.99764	0.998718
	Forehand	0.989575	0.99664	0.993095	0.994713
	Jump	0.997207	0.988463	0.992816	0.995034
VGG-16	Backhand	0.985259	0.994643	0.989929	0.994553
	Forehand	0.940075	0.988621	0.963737	0.972765
	Jump	0.9986	0.939833	0.968326	0.977571

Table 2. Recall, precision, F-measure and accuracy of ResNet-18, GoogleNet and VGG-16 for testing dataset

Models		Recall	Precision	F-measure	Accuracy
ResNet-18	Backhand	0.996466	0.893819	0.942356	0.966859
	Forehand	0.935000	0.996005	0.964539	0.973583
	Jump	0.977654	1.000000	0.988701	0.992315
GoogleNet	Backhand	0.985866	0.989362	0.987611	0.993276
	Forehand	0.990000	0.991239	0.990619	0.992795
	Jump	0.984637	0.980529	0.982578	0.987992
VGG-16	Backhand	0.982301	0.996409	0.989305	0.994236
	Forehand	0.932584	0.986790	0.958922	0.96926
	Jump	0.998603	0.930990	0.963612	0.974063

Table 3. Confusion matrix of ResNet-18 for training dataset

	Backhand	Forehand	Jump
Backhand	1694	2	0
Forehand	56	2334	8
Jump	24	4	2120

Table 4. Confusion matrix of GoogleNet for training dataset

	Backhand	Forehand	Jump
Backhand	1691	3	2
Forehand	2	2373	23
Jump	1	5	2142

Table 5. Confusion matrix of VGG-16 for training dataset

	Backhand	Forehand	Jump
Backhand	1671	23	2
Forehand	9	2264	135
Jump	0	3	2145

Table 6. Confusion matrix of ResNet-18 for testing dataset

	Backhand	Forehand	Jump
Backhand	1694	2	0
Forehand	56	2334	8
Jump	24	4	2120

Table 7. Confusion matrix of GoogleNet for testing dataset

	Backhand	Forehand	Jump
Backhand	558	0	8
Forehand	2	792	6
Jump	4	7	705

Table 8. Confusion matrix of VGG-16 for testing dataset

	Backhand	Forehand	Jump
Backhand	556	9	1
Forehand	2	747	52
Jump	0	1	715
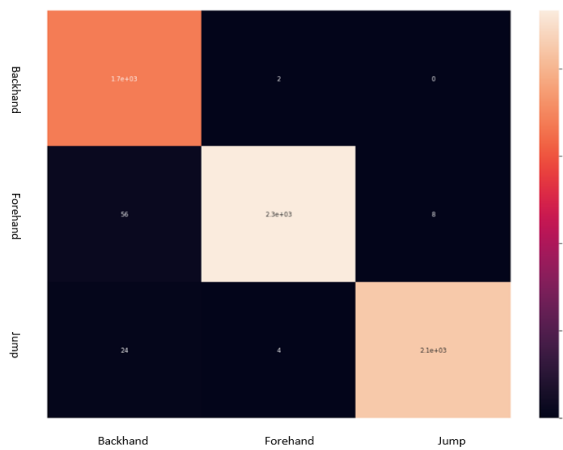
Figure 2. Confusion matrix of ResNet-18 for training dataset
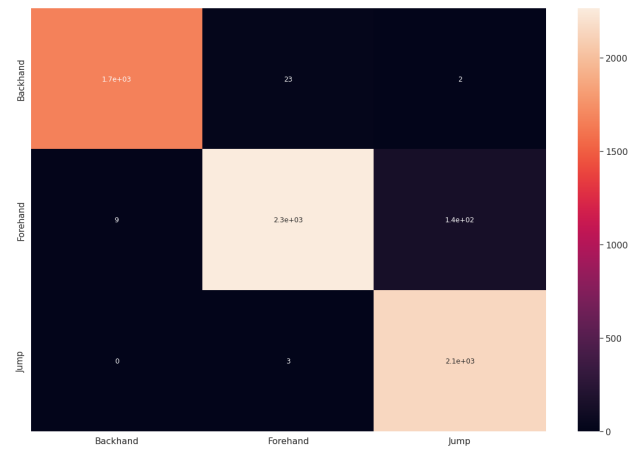
Figure 3. Confusion matrix of VGG-16 for training dataset
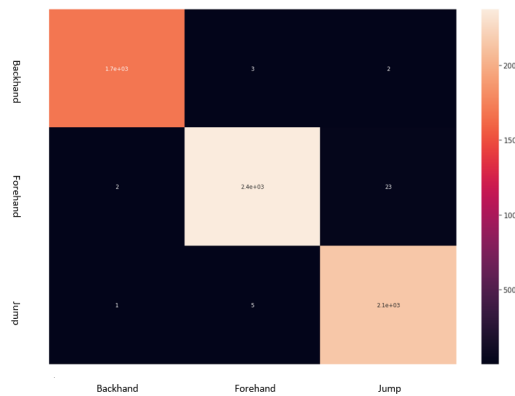
Figure 4. Confusion matrix of GoogleNet for training dataset
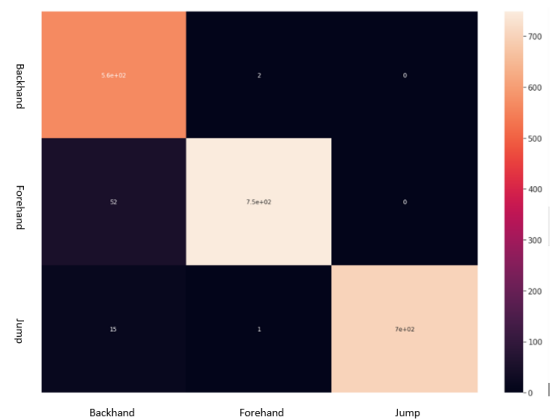
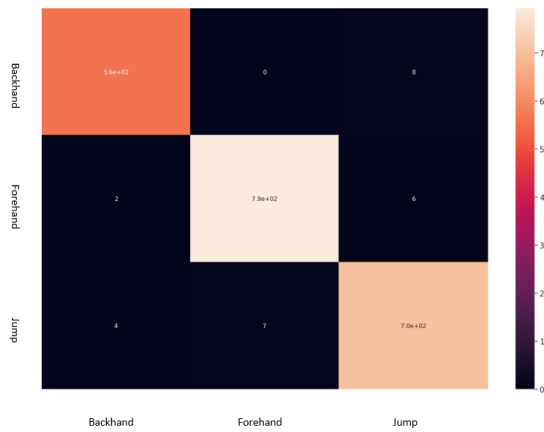**Figure 5.** Confusion matrix of ResNet-18 for testing dataset

Figure 6. Confusion matrix of GoogleNet for testing dataset
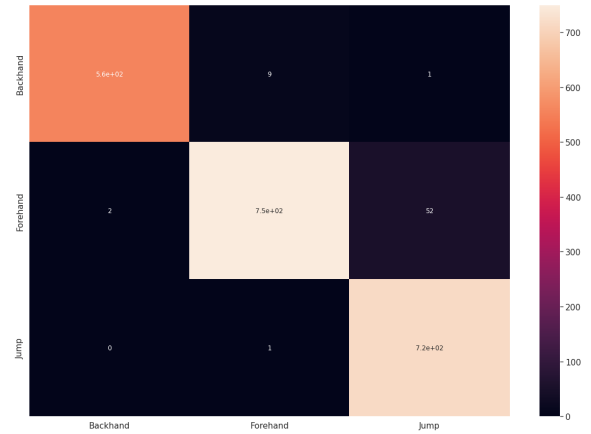

Figure 7. Confusion matrix of VGG-16 for testing dataset
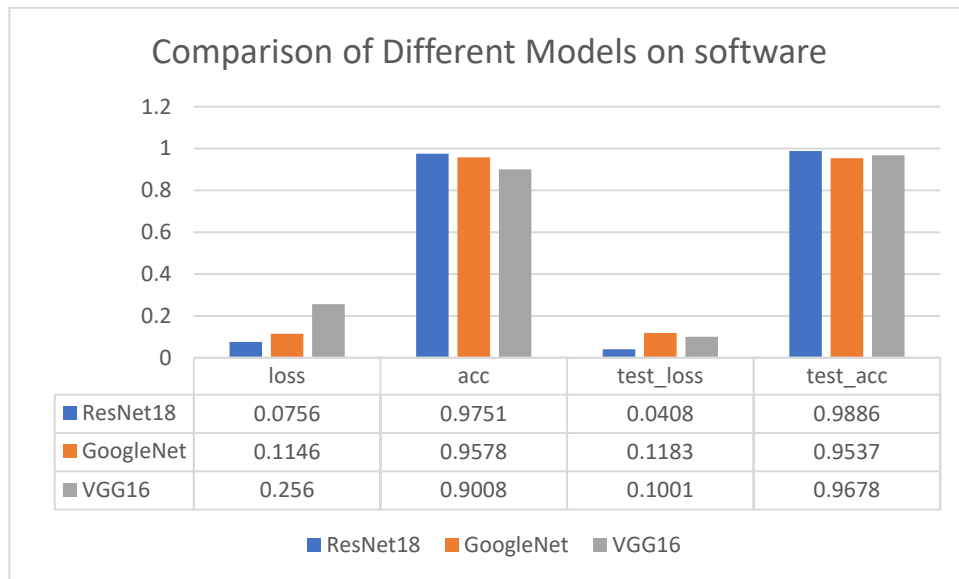

Figure 8. Comparison of models ResNet-18, GoogleNet and VGG-16 on software

We may deduce from Table 1 that the backhand class has the maximum recall value of 99.88%, while the forehand class has the lowest recall value of 97.33 %. Aside from that, the class with the greatest precision value, 99.74 %, is forehand smashing. The highest precision number implies that out of all positive images predicted, forehand smashing has the largest percentage of actually positive images. Furthermore, jump smashing has the greatest F-measure score, at 99.16 %. Jump smashing is the class with the highest accuracy in the training dataset when employing model ResNet-18, with a classification accuracy of 99.42 %.

According to Table 2, we can know that the backhand smashing has the highest recall of 99.65% and the lowest precision of 89.38 %. On another hand, jump smash has the maximum precision rating of 100%. Jump smash is the class with the highest accuracy in the testing dataset, with a value of 99.23% while backhand smash has the lowest accuracy which is 96.69% in classifying jump smash from other types of smashing skills.

From Figure 8, it shows that, ResNet-18 has the highest accuracy on both training and testing datasets which are 97.51% and 98.86% respectively. ResNet-18 has the best accuracy performance in recognizing the badminton smashing. Besides that, the lowest value loss of ResNet-18 on both training and testing datasets indicates that it makes the least when making prediction using ResNet-18.

The classes that have the highest accuracy for GoogleNet and VGG-16 in training datasets are the backhand smash which have a value of 99.87% and 99.46% respectively. For the testing datasets, the classes that have the highest accuracy for GoogleNet and VGG-16 are backhand smash which have a value of 99.32% and 99.42% respectively.

In short, ResNet-18 has the highest accuracy on both training and testing datasets which are 97.51% and 98.86% respectively which show the best performance in classifying the types of smashing. Jump smash is the class that has the highest accuracy in both training and testing datasets which are 99.42% and 99.23% respectively by using model ResNet-18.

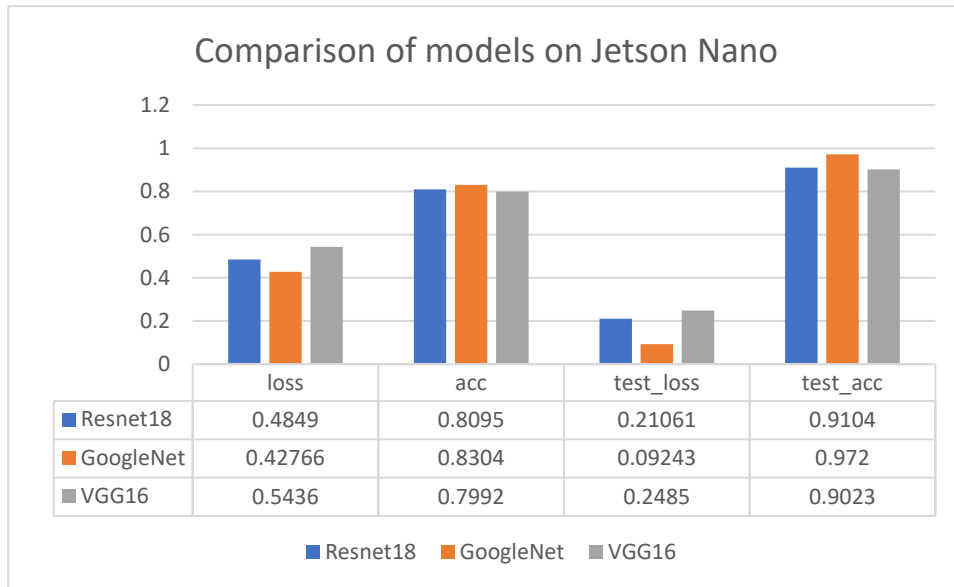**Evaluation performance on hardware Jetson Nano**

Figure 9. Comparison of models on Jetson Nano

GoogleNet has the highest accuracy on both training and testing datasets which are 83.04% and 97.20% respectively as in Figure 9. GoogleNet has the best accuracy performance in recognizing the badminton smashing by using hardware Jetson Nano. Besides that, the lowest value loss of GoogleNet on both training and testing datasets are 0.4277 and 0.0924 respectively indicating it makes the least when making predictions.
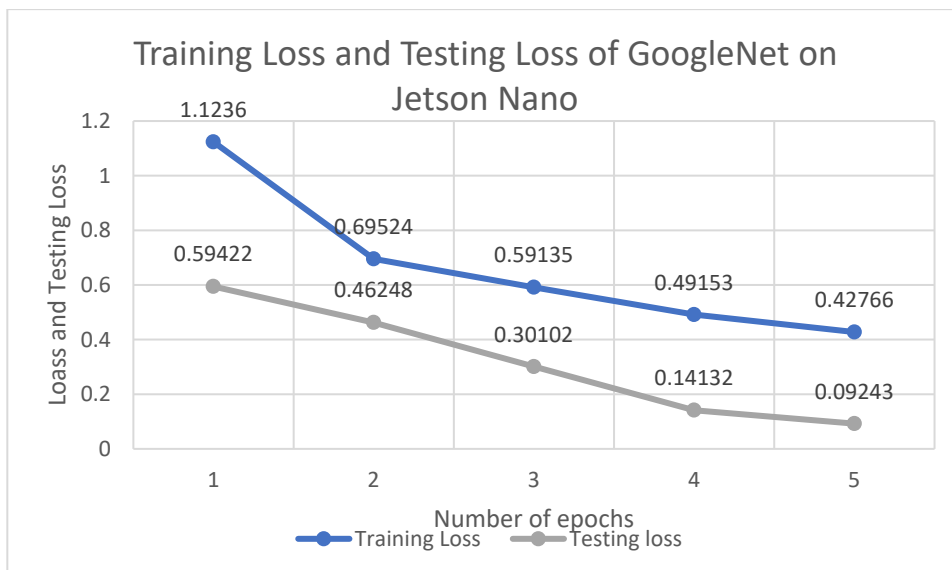

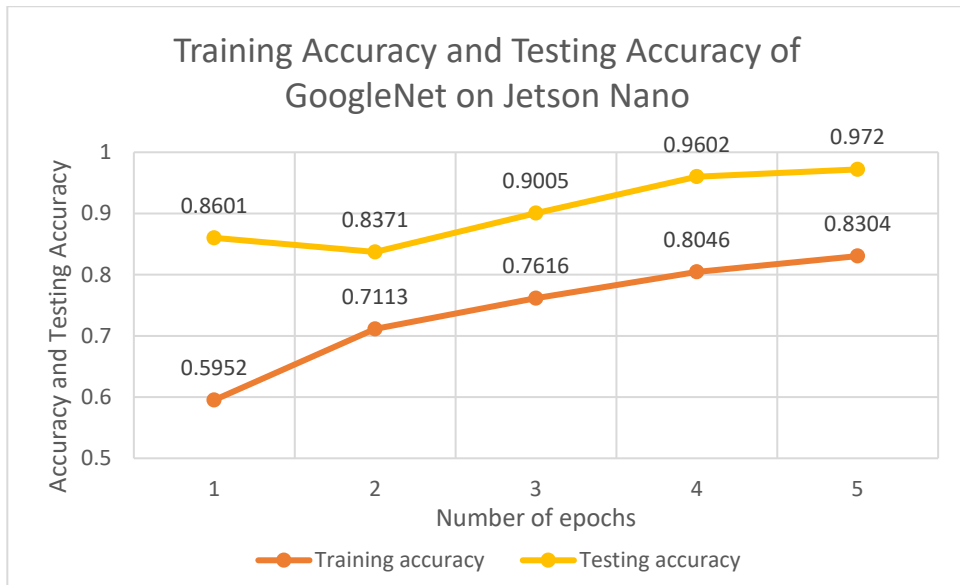**Figure 10.** Training loss and testing loss of GoogleNet on Jetson Nano

Figure 11. Training accuracy and testing accuracy of GoogleNet on Jetson Nano

From Figure 10 and 11 it shows that the training loss is always higher than the testing loss, and the value for both the training and testing datasets decreases as the number of epochs increases, as shown in Figure 9. The training loss is 0.4277 , while the testing loss of GoogleNet is 0.0924 at last epoch.

The testing accuracy is always higher than the training accuracy, as seen in Figure 10. As the number of epochs increases, the accuracy of both the training and testing datasets improves. The highest training accuracy for GoogleNet can be attained at 83.04 %, while the highest testing accuracy for GoogleNet can be obtained at 97.20 %.

Comparison performance of models on software and Jetson Nano
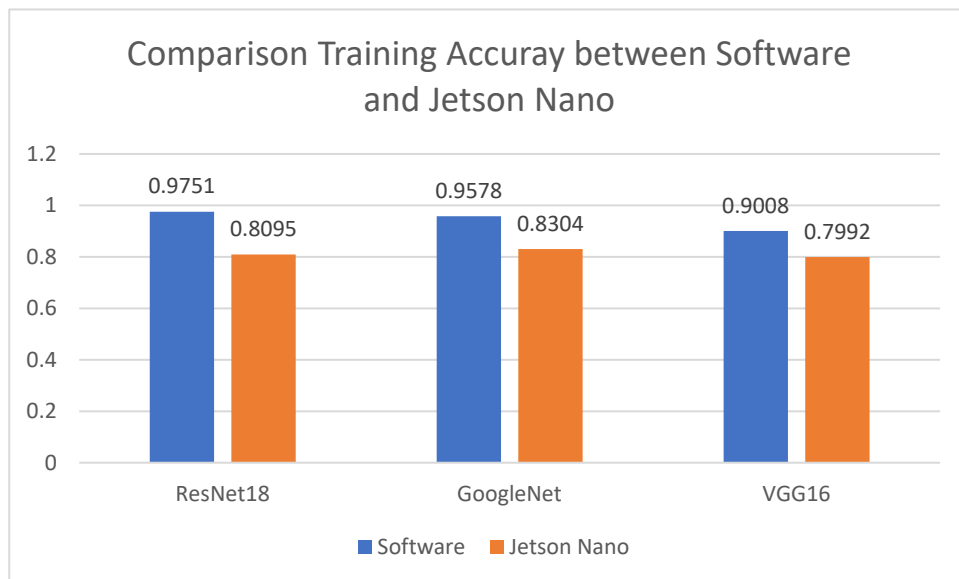
a) Training accuracy

Figure 12. Comparison training accuracy between software and Jetson Nano

The Figure 12 illustrates that the training accuracy on software is always higher than hardware Jetson Nano. ResNet-18 has the highest training accuracy of 0.9751 on software while VGG-16 has the lowest training accuracy of 0.256 on the software. The ResNet-18 has a value of training accuracy of 0.9751 on software which is higher than training accuracy

on Jetson Nano about 0.1656. On another hand, VGG-16 has the lowest training accuracy of 0.4277 on Jetson which is lower than the training accuracy on software about 0.1016.
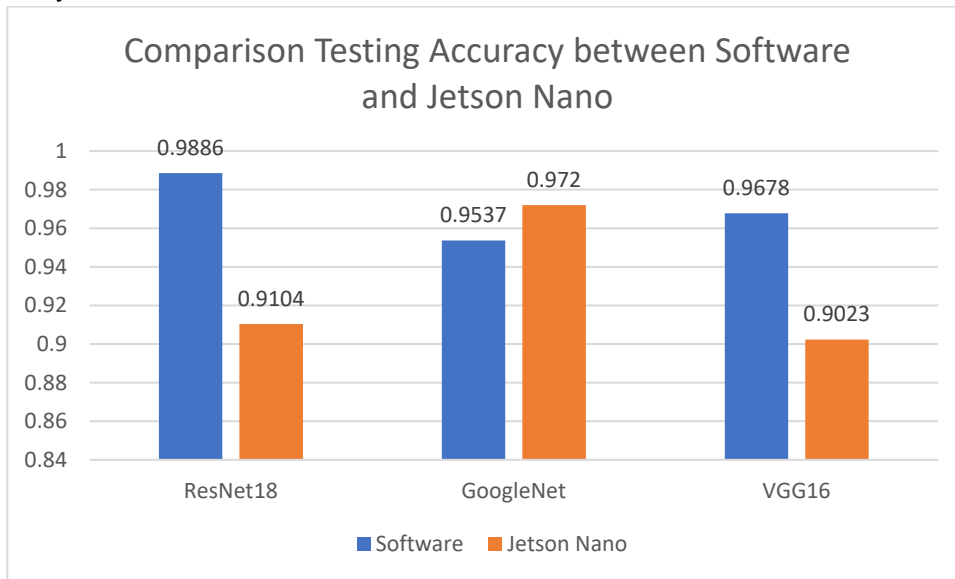
b) Testing accuracy

Figure 13. Comparison testing accuracy between software and Jetson Nano

Figure 13 illustrates the comparison of testing accuracy between software and Jetson Nano by using different models. ResNet-18 shows the biggest difference of testing accuracy between software and Jetson Nano. The testing accuracy of ResNet-18 on software has a value of 0.9886 which is higher than the testing accuracy on Jetson Nano about 0.0782. On other hand, GoogleNet shows the smallest difference of testing accuracy between software and Jetson Nano. The testing accuracy of GoogleNet on Jetson Nano has a value of 0.9720 which is higher than the testing accuracy on software about 0.0183.

ResNet-18 has the highest testing accuracy of 0.9886 on software while GoogleNet has the lowest testing accuracy of 0.9537 on the software. Besides that, GoogleNet has the highest testing accuracy of 0.9720 on Jetson Nano while VGG-16 has the lowest testing accuracy of 0.9023 on Jetson Nano.

CONCLUSION

In this project, we establish in collecting the data of forehand smash, backhand smash, and jump smash from the mature badminton players who have at least 5 years of experience. Besides that, it could be concluded based on the results on software Jupyter and the hardware Jetson Nano. This project demonstrates how well the models RestNet-18, GoogleNet and VGG-16 in classifying the backhand, forehand, and jump smash.The deepp learning technique then will be applied to software and hardware to find the classification accuracy and will compare with multiple models such as ResNet-18, GoogleNe,t and VGG-16 that are suitable for badminton smashing recognition. By using the software Jupyter, the best model that we can obtain in this project is ResNet-18 which has an accuracy of 97.51% and 98.86% in training and testing datasets respectively. On another hand, the best model that we can obtain by using the hardware Jetson Nano is the GoogleNet which has an accuracy of 83.04% and 97.20% in both training and testing datasets respectively.

ACKNOWLEDGEMENT

REFERENCES

[1]   K. Rangasamy, M. A. As'ari, N. A. Rahmad, and N. F. Ghazali, "Hockey activity recognition using pre-trained deep learning model," *ICT Express*, vol. 6, no. 3, 2020, doi: 10.1016/j.icte.2020.04.013.

[2]   T. Kautz, B. H. Groh, J. Hannink, U. Jensen, H. Strubberg, and B. M. Eskofier, "Activity recognition in beach volleyball using a Deep Convolutional Neural Network: Leveraging the potential of Deep Learning in sports," *Data Mining and Knowledge Discovery*, vol. 31, no. 6, 2017, doi: 10.1007/s10618-017-0495-0.

[3]   Y. L. Hsu, H. C. Chang, and Y. J. Chiu, "Wearable Sport Activity Classification Based on Deep Convolutional Neural Network," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2955545.

[4]   Z. Geradts, N. Filius, and A. Ruifrok, "Interpol review of imaging and video 2016–2019," *Forensic Science International: Synergy*, vol. 2. 2020. doi: 10.1016/j.fsisyn.2020.01.017.

[5]   T. Charnock and A. Moss, "Deep Recurrent Neural Networks for Supernovae Classification," *The Astrophysical Journal*, vol. 837, no. 2, 2017, doi: 10.3847/2041-8213/aa603d.

[6]   S. Gollapudi, *Learn Computer Vision Using OpenCV*. 2019. doi: 10.1007/978-1-4842-4261-2.

[7]   B. Gupta, A. Chaube, A. Negi, and U. Goel, "Study on Object Detection using Open CV - Python," *International Journal of Computer Applications*, vol. 162, no. 8, 2017, doi: 10.5120/ijca2017913391.

[8]   R. Jin and J. Wang, "A vision tracking system via color detection," in *IEEE International Conference on Control and Automation, ICCA*, 2016, vol. 2016-July. doi: 10.1109/ICCA.2016.7505387.

[9]   A. Badiola-Bengoa and A. Mendez-Zorrilla, "A systematic review of the application of camera-based human pose estimation in the field of sport and physical exercise," *Sensors*, vol. 21, no. 18. 2021. doi: 10.3390/s21185996.

[10]   J. Wang, K. Qiu, H. Peng, J. Fu, and J. Zhu, "AI Coach: Deep Human Pose Estimation and Analysis for Personalized Athletic Training Assistance," 2019. doi: 10.1145/3343031.3350910.

[11]   M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, "Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2994222.