Automated badminton smash recognition using convolutional neural network on the vision based data

N A Rahmad[1], M A As'ari[2], K Soeed[1] and I Zulkapri[1]
[1]School of Biomedical Engineering and Health Sciences, Faculty of Engineering, UniversitiTeknologi Malaysia, Johor Bahru, Malaysia
[2] Sport Innovation and Technology Center (SITC), Institute of Human Centered Engineering (IHCE), UniversitiTeknologi Malaysia, Johor Bahru, Malaysia

E-mail: amir-asari@biomedical.utm.my

Abstract. Sport performance analysis in sports practice cannot be separable. It is important to help coach analyse and improve the performance of their athletes through training or game session. Due to the advancement of technology nowadays, the notational analysis of the video content using various software packages has become possible. Unluckily, the coach needs to recognize the actions manually before doing further analysis. The purpose of this study is to formulate an automated system for badminton smash recognition on widely available broadcasted videos using pre-trained Convolutional Neural Network (CNN) method. Smash and other badminton actions such as clear, drop, lift and net from the video were used to formulate the CNN models. Therefore, two experiments were conducted in this study. The first experiment is the study on the performance between four different existing pre-trained models which is AlexNet, GoogleNet, Vgg-16 Net and Vgg-19 Net in recognizing five actions. The results show that the pre-trained AlexNet model has the highest performance accuracy and fastest training period among the other models. The second experiment is the study on the performance of two different pre-trained models which is AlexNet and GoogleNet to recognize smash and non-smash action only. The results show that the pre-trained GoogleNet model produces the best performance in recognizing smash action. In conclusion, pre-trained AlexNet model is suitable to be used to automatically recognize the five badminton actions while GoogleNet model is excellent at recognizing smash action from the broadcasted video for further notational analysis.

1. Introduction

In computer vision, deep learning is currently the most powerful machine learning used by many researchers for recognizing actions. There are various applications for action recognition such as video surveillance[1], sports performance analysis, rehabilitation and virtual reality [2].

Deep learning which a subtype of machine learning is a new trend in action recognition because its ability to provide results that are more precise compared to the handcrafted machine learning method. The advantage of using deep learning is the network learns and extracts the features automatically from the raw images without being manually extracted. Many works have been done by the previous researches in recognizing action using deep learning method such as CNN [3] and Long-Short Term Memory (LSTM) [4, 5].

In sports performance analysis using vision based approach, the coach needs to manually interpret the broadcast video and key in the athlete’s action into the software packages one by one before further analysis. To provide a more efficient and accurate analysis to coach, an automated action recognition from the broadcast video is crucial. Smash is among the actions performed by the player in a match or game. It is an offensive overhead shot with downward trajectory. It is crucial for coach to analyse the performance of smash action in order to improve the smash technique. Therefore, the aim of this study is to formulate an automated system for badminton smash and other badminton actions on widely available broadcasted videos using pre-trained Convolutional Neural Network (CNN) method. In this study, CNN for automatically recognizing smash and other badminton actions has been formulated from the broadcasted video of badminton match.

2. Material and methods

In this study, we are using video data inputs that have been extracted into a sequence of image frames. The experiment was fully conducted on our own constructed dataset consists of five different badminton actions – clear, drop, lift, net shot and smash. Figure 1 until Figure 5 shows the sample of our dataset used in this study. The purpose of constructing our own dataset is because there is no publicly available datasetAll image frames were systematically extracted from five broadcasted badminton match videos obtained from Badmintonworld.tv YouTube channel. Each video of resolution 720p was extracted using the VirtualDub software into image frames before annotating the frames into the corresponding actions. Table 1 shows the details of the distribution of our dataset. For the total number of data, 80% were used to train the deep learning networks and another 20% were used for testing the trained networks randomly to measure the network’s generalization ability.

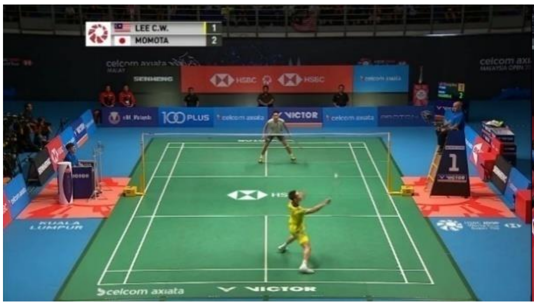


Figure 1. Clear action.

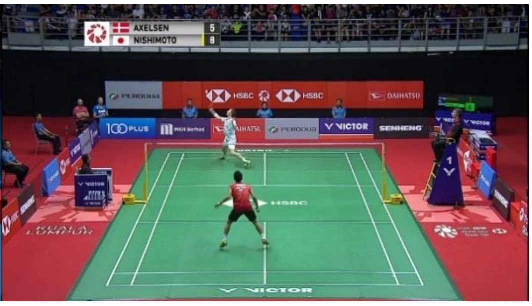


Figure 2. Drop action.

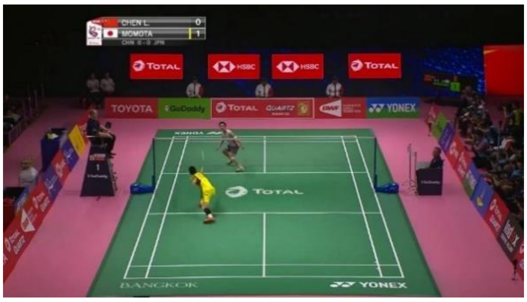


Figure 3. Lift action.

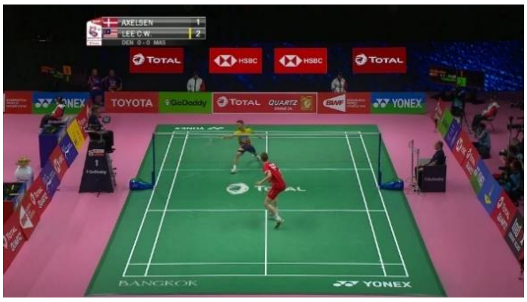


Figure 4. Net shot action.

Figure 5. Smash action.

Table 1. The distribution of dataset for the first experiment.

Action	Number of total samples	Number of training samples	Number of testing samples
Clear	200	160	40
Drop	100	80	20
Lift	398	318	80
Net shot	270	216	54
Smash	528	422	106
Total	**1496**	**1196**	**300**

The procedure for both experiments is basically the same. At the very beginning, the input data which is the image frames were fed into each of pre-trained CNN model. Each model was trained and fine-tuned with our input data using the MATLAB 2018b software. Model training is a process where a network recognizes and learns the pattern in the data. Table 2 shows the training options that we used to train each model. Other parameters not mentioned below are set in default values and remain constant for all models. The definition of each training option term is as follows.

- Training optimizer: solver to train the network that will optimize the network.
- Mini-batch size: subset's size of the training set that is used to evaluate the gradient of the loss function and update the network's weight.
- Maximum epochs: maximum full pass number of the training algorithm over the entire training set.
- Execution environment: hardware resource for training the network.
- Initial learning rate: learning rate used for the training of the network.

Table 2. Training options.

Training options	AlexNet	GoogleNet	Vgg-16 Net	Vgg-19 Net
Training optimizer	Sgdm	Sgdm	Sgdm	Sgdm
Mini-batch size	5	5	1	1
Maximum epochs	10	10	10	10
Execution environment	GPU	GPU	CPU	CPU
Initial learning rate	0.0001	0.0001	0.0001	0.0001

Secondly, each deep network automatically extracted the visual features of each image frame from the low-level features such as colors and edges until high-level features which are the representation of the action. The advantage of using deep learning in the recognition task is that manual feature extraction is not needed. Therefore, input data can be directly fed into the network for automatic feature extraction. After the model learned the features automatically during thetraining process, it classified the testing input data into five actions. Classification in deep learning is a process of identifying the class of each image based on the pattern learned earlier. Lastly, the performance of each model was evaluated in term of the performance accuracy and visualized using the confusion matrix. In deep learning, most researchers are focusing on quantitative evaluationwhich is the performance accuracy of the network. Therefore, the purpose of the evaluation is to evaluate the performance of the deep learning model in doing an automated recognition task. Figure 6 illustrates the methodology of the experimental work.

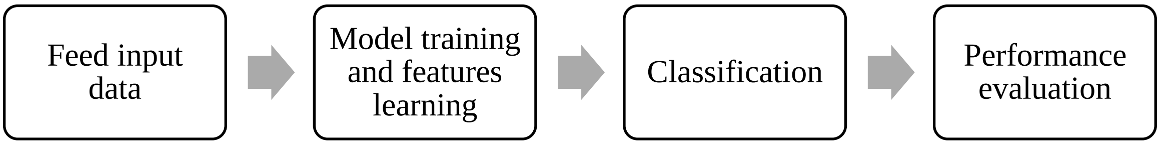


Figure 6. The block diagram of methodology.

For the second experiment, we repeat the same experimental setup on AlexNet and GoogleNet model only. This is because both models show impressive results in the first experiment. Therefore, the second experiment is done for further investigation in their performance. 1495 image frames were used in the second study annotated into two classes - smash and non-smash. Table 3 shows the details of the distribution of our dataset for the second experiment.

Table 3. The distribution of dataset for the second experiment.

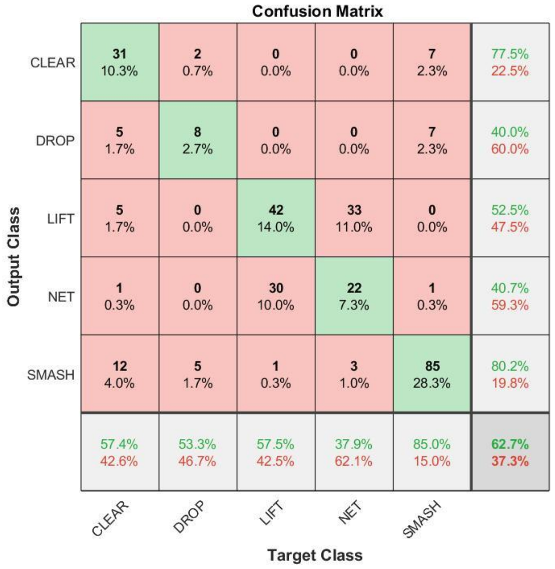
Action	Number of total samples	Number of training samples	Number of testing samples
Smash	528	422	106
Others	967	774	193
Total	1495	1196	299

3. Main results

The performance of each pre-trained model in recognizing different badminton actions was evaluated in term of the percentage of accuracy and visualized using the confusion matrix. Rows represent predicted or output classes while the columns represent actual target classes. All the correct predicted classes are in the diagonal while false classes are in the non-diagonal. For the first experiment, Table 4 shows the accuracy and the training period of four pre-trained models. Figure 7 until Figure 10 shows the confusion matrix of each model.

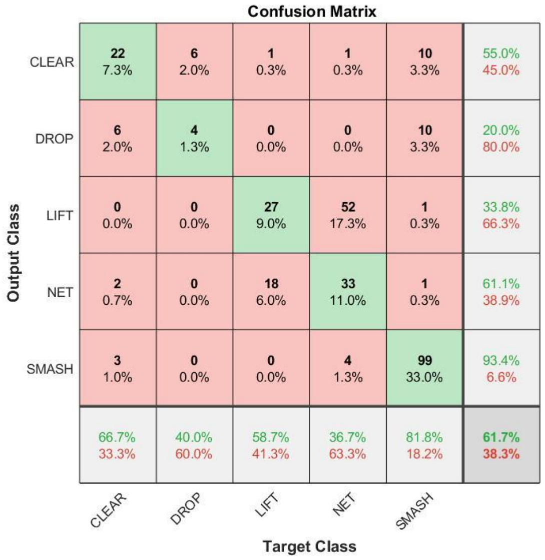
Table 4. The accuracy and training time for each pre-trained CNN model.

Pre-trained model	Accuracy (%)	Time elapsed (hh:mm:ss)
AlexNet	62.7	00:23:07
GoogleNet	61.7	00:30:58
Vgg-16 Net	35.3	09:14:46
Vgg-19 Net	35.3	11:03:25



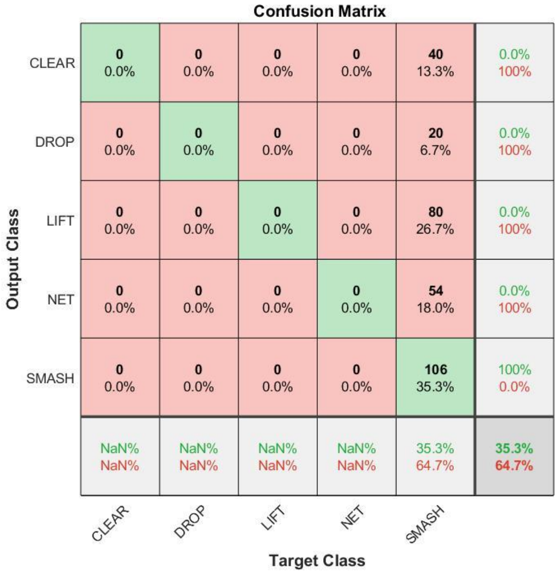
	CLEAR	DROP	LIFT	NET	SMASH	
CLEAR	31 10.3%	2 0.7%	0 0.0%	0 0.0%	7 2.3%	77.5% 22.5%
DROP	5 1.7%	8 2.7%	0 0.0%	0 0.0%	7 2.3%	40.0% 60.0%
LIFT	5 1.7%	0 0.0%	42 14.0%	33 11.0%	0 0.0%	52.5% 47.5%
NET	1 0.3%	0 0.0%	30 10.0%	22 7.3%	1 0.3%	40.7% 59.3%
SMASH	12 4.0%	5 1.7%	1 0.3%	3 1.0%	85 28.3%	80.2% 19.8%
	57.4% 42.6%	53.3% 46.7%	57.5% 42.5%	37.9% 62.1%	85.0% 15.0%	62.7% 37.3%

Figure 7. Confusion matrix AlexNet.



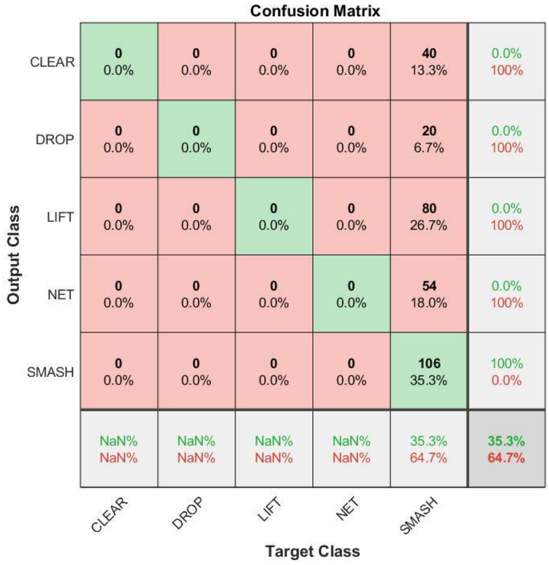
	CLEAR	DROP	LIFT	NET	SMASH	
CLEAR	22 7.3%	6 2.0%	1 0.3%	1 0.3%	10 3.3%	55.0% 45.0%
DROP	6 2.0%	4 1.3%	0 0.0%	0 0.0%	10 3.3%	20.0% 80.0%
LIFT	0 0.0%	0 0.0%	27 9.0%	52 17.3%	1 0.3%	33.8% 66.3%
NET	2 0.7%	0 0.0%	18 6.0%	33 11.0%	1 0.3%	61.1% 38.9%
SMASH	3 1.0%	0 0.0%	0 0.0%	4 1.3%	99 33.0%	93.4% 6.6%
	66.7% 33.3%	40.0% 60.0%	58.7% 41.3%	36.7% 63.3%	81.8% 18.2%	61.7% 38.3%

Figure 8. Confusion matrix GoogleNet.



	CLEAR	DROP	LIFT	NET	SMASH	
CLEAR	0 0.0%	0 0.0%	0 0.0%	0 0.0%	40 13.3%	0.0% 100%
DROP	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 6.7%	0.0% 100%
LIFT	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80 26.7%	0.0% 100%
NET	0 0.0%	0 0.0%	0 0.0%	0 0.0%	54 18.0%	0.0% 100%
SMASH	0 0.0%	0 0.0%	0 0.0%	0 0.0%	106 35.3%	100% 0.0%
	NaN% NaN%	NaN% NaN%	NaN% NaN%	NaN% NaN%	35.3% 64.7%	35.3% 64.7%

Figure 9. Confusion matrix Vgg-16 Net.



	CLEAR	DROP	LIFT	NET	SMASH	
CLEAR	0 0.0%	0 0.0%	0 0.0%	0 0.0%	40 13.3%	0.0% 100%
DROP	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 6.7%	0.0% 100%
LIFT	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80 26.7%	0.0% 100%
NET	0 0.0%	0 0.0%	0 0.0%	0 0.0%	54 18.0%	0.0% 100%
SMASH	0 0.0%	0 0.0%	0 0.0%	0 0.0%	106 35.3%	100% 0.0%
	NaN% NaN%	NaN% NaN%	NaN% NaN%	NaN% NaN%	35.3% 64.7%	35.3% 64.7%

Figure 10. Confusion matrix Vgg-19 Net.

For the second experiment, Table 5 shows the accuracy and the training period of two pre-trained models. Figure 11 and Figure 12 shows the confusion matrix of each model.

Table 5. The accuracy and training time for each pre-trained CNN model.

Pre-trained model	Accuracy (%)	Time elapsed (hh:mm:ss)
AlexNet	86.0	00:23:00
GoogleNet	90.3	00:30:45

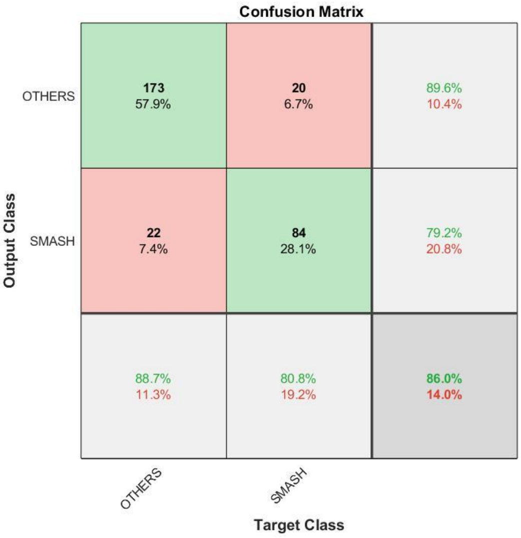


Figure 11. Confusion matrix AlexNet.

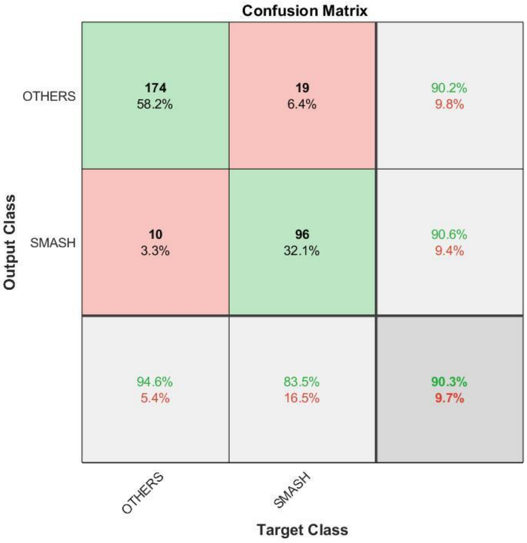


Figure 12. Confusion matrix GoogleNet.

4. Discussion

From the first experiment, the results confirm the usefulness of the AlexNet model as the most suitable pre-trained model for the recognition task of badminton actions compared to the other model with the highest accuracy of 62.7% and faster training period, which is 23 minutes and 7 seconds. However, the deeper network than AlexNet, which is GoogleNet has slightly lower accuracy which is 61.7% and longer training period which is 30 minutes and 58 seconds. It is acceptable that GoogleNet takes a longer training period as the network is deeper. A study by Saufi et al. [6] and Muhammad et al. [7] reported that the GoogleNet accuracy is higher than AlexNet as the deeper network produces a better accuracy whereas our study found otherwise. There is a possibility that the GoogleNet might be under fitted since there is not enough data to train the deeper GoogleNet network because work in [7] that train GoogleNet with 4900 images achieved 100% accuracy. Under fit usually occurs when the network cannot learns the trend of the data due to the lack number of data. Therefore, it could not able to classify the data properly and produces low performance accuracy. As shown in each confusion matrix, drop action has the lowest contribution to the network's performance accuracy because it has a smaller number of data compared to the other actions. Hence, the network cannot learns drop action properly and falsely classify it as other actions. We believe that to overcome the problem, we should increase the number of drop data as it only contributes the smallest percent in accuracy as shown in each confusion matrix because of its smallest number of data.

Contrary to expectations, we did not find a significant difference in performance accuracy between Vgg-16 Net and Vgg-19 Net even though the deeper Vgg-19 Net model should produce better accuracy. Both models have a longer training period, which is 9 hours 14 minutes 46 seconds and 11 hours 3 minutes 25 seconds respectively because they have been trained using the CPU. The accuracy for both Vgg-19 Net and Vgg-16 Net is the worst which is 35.3% that indicates the models cannot be used to recognize badminton actions.

We are aware that our study may have two limitations. The first is lack of dataset and the second is GPU is not enough memory to train dense Vgg-16 and Vgg-19 Nets, therefore, both of them were trained using the CPU. It is plausible that a number of limitations might have influenced the results obtained.

From each confusion matrix in the first experiment, we can see that smash action has the highest number of actions that correctly detected by the network. Therefore, we repeat the experiment to study the smash detection using AlexNet and GoogleNet model. Surprisingly, there is a huge improvement in the performance accuracy for this second experiment. The highest accuracy is 90.3% by GoogleNet. The performance accuracy of AlexNet model also improved from 62.7% in the first experiment to 86.0% in the second experiment.

The results from the first experiment are poor compared to the second experiment. We believe that the results are poor because these experiments were conducted on a frame-based dataset which only depends on spatial information and lack of temporal information. However, even the results are poor, the proposed approach is excellent in detecting the smash action.

5. Conclusion

The results in this study indicate that the GoogleNet pre-trained CNN model is the most suitable model that can be used among others to automatically recognize the smash action and provide the coach with the information for further notational analysis. The study shows that an automated action recognition system can be developed by using a deep learning approach. With the limited amount of dataset, pre-trained GoogleNet model can be used to automatically recognize the smash action from the broadcasted videos obtained online. As a summary, the automated action recognition using deep learning method of the vision based data could benefit the coach, players and the sports institution itself. However, our work clearly has some limitations. Despite this, we believe that our study could be a starting point towards enhancing the deep learning approach in sports practice. the research into improving the results is already in progress.

Acknowledgments

The authors would like to express their gratitude to UniversitiTeknologi Malaysia (UTM) and the Minister of Education (MOE), Malaysia for supporting this research work under Zamalah UTM and FRGS Research Grant No. R.J130000.7851.5F108.

References

[1] Kushwaha AKS and Srivastava R Multiview human activity recognition system based on spatiotemporal template for video surveillance system2015 *Journal of Electronic Imaging* **24**18.

[2] Li NF and Xiao Z G A fire drill training system based on vr and kinect somatosensory technologies 2018 *International Journal of Online Engineering***14** 163-76.

[3] Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R and Fei-Fei L 2014 *2014 IEEE Conf. on Computer Vision and Pattern Recognition* (Columbus:IEEE) pp 1725-32.

[4] Ma Z Cand Sun Z X Time-varying lstmnetworks for action recognition 2018 *Multimedia Tools and Applications***77**32275-85.

[5] Ullah A, Ahmad J, Muhammad K, Sajjad M and Baik S W Action recognition in video sequences using deep bi-directional lstm with cnn features 2018 *IEEE Access***6** 1155-66.

[6] Saufi MM, Zamanhuri M A, Mohammad N and Ibrahim ZDeep Learning for Roman Handwritten Character Recognition 2018 *Indonesian Journal of ElectricalEngineering and Computer Science***12** 455-60.

[7] Muhammad N A, Ab Nasir A, Ibrahim Z and Sabri N Evaluation of cnn, alexnet and googlenet

for fruit recognition2018*Indonesian Journal of Electrical Engineering and Computer Science***12**468-75.