MEKATRONIKA
e-ISSN: 2637-0883
VOL. 2, ISSUE 2, 1 – 12
DOI: https://doi.org/10.15282/mekatronika.v2i2.6683

ORIGINAL ARTICLE

The Classification of Skateboarding Tricks : A Transfer Learning and Machine Learning Approach

Muhammad Nur Aiman Shapiee[1], Muhammad Ar Rahim Ibrahim[1], Muhammad Amirul Abdullah[1], Mohd Azraai Mohd Razman[1], Rabiu Muazu Musa[2], Noor Azuan Abu Osman[2] and Anwar P.P. Abdul Majeed[1,3*]

[1]Innovative Manufacturing, Mechatronics and Sports Laboratory, Faculty of Manufacturing and Mechatronic Engineering Technology, Universiti Malaysia Pahang, 26600 Pekan, Pahang Darul Makmur, Malaysia.
[2]Centre for Fundamental and Continuing Education, Universiti Malaysia Terengganu, Terengganu, Malaysia
[3]Centre for Software Development and Integrated Computing, Universiti Malaysia Pahang, 26600 Pekan, Pahang Darul Makmur, Malaysia

ABSTRACT – The skateboarding scene has arrived at new statures, particularly with its first appearance at the now delayed Tokyo Summer Olympic Games. Hence, attributable to the size of the game in such competitive games, appraisal approaches have progressively increased due consideration by pertinent partners, particularly with the enthusiasm of a more goal-based assessment. This study purposes for classifying skateboarding tricks, specifically Frontside 180, Kickflip, Ollie, Nollie Front Shove-it, and Pop Shove-it over the integration of image processing Transfer Learning (TL) to feature extraction enhanced with traditional Machine Learning (ML) classifier. A male skateboarder performed five tricks every sort of trick consistently, and the YI Action camera captured the movement by a range of 1.26 m. Then, the image dataset were features built and extracted by employing three TL models, and afterwards evaluated using k-Nearest Neighbor (k-NN) classifier. The perception via the initial experiments showed, the MobileNet, NASNetMobile, and NASNetLarge coupled with optimized k-NN classifiers attain a classification accuracy (CA) of 95%, 92% and 90%, respectively on the test dataset. Besides, the result evident from the robustness evaluation showed the MobileNet+k-NN pipeline is more robust as it could provide a decent average CA than other pipelines. It would be demonstrated that the suggested study could characterize the skateboard tricks sufficiently and could, over the long haul, uphold judges decision for giving progressively objective-based decisions.

ARTICLE HISTORY
Received: 20th May 2020
Revised: 7th Sept 2020
Accepted: 27th Oct 2020

KEYWORDS
Transfer learning
Machine learning
Classification
Skateboarding

INTRODUCTION

The total assets from the skateboarding business are roughly $USD 5 billion out of 2010, with 11.08 million dynamic skateboarders worldwide [1]. The 2021 Tokyo Summer Olympics Games would make its first appearance. Hence the skateboarding scene has arrived at new statures. Moreover, the expansion of the skateboarding scene because of several events been held. For example, X Games and Street League has brought about development in the skateboarding market scene. This development, to some degree, requests new inventive methodologies in assessing the sport that currently has serious rivalries, particularly along with the assessment of the tricks performed are customarily conceded by judges and abstract implies to be exact frequently obligated to predisposition and loose appraisal.

RELATED WORK

Owing to the advent of computational intelligence, deep learning, particularly Convolution Neural Networks (CNN) has gained due attention amongst the research community across different knowledge domains. For instance, CNN already engaged in human motion analysis through data extracted from a tri-axial accelerometer data five Nexus 6P Huawei smartphones [2]. The author instructed five subjects, graduate students, to record three (3) activities, namely staying still, walking and running. The sensor is placed in various positions, such as carried in a bag pack, carried by hand, and located in a pocket. As for the evaluation, the authors proposed one dimension CNN method that was different from the conventional Random Forest (RF) method. MATLAB implemented the RF and the 1D CNN implemented by TensorFlow then the precision and recall were calculated for the evaluation metric. The proposed 1D CNN for feature10 and feature20 approach 91.32% and 92.71% respectively outperform the RF method of 85.72% and 89.10% by feature10 and feature20.

CNN has also been used in an image classification of breast cancer histopathological [3]. The author was using microscopic biopsy images of malignant and benign breast tumours collected from the BreaKHis database, with a total of 7909 images. The author uses SCC-131AN Samsung digital color camera and Olympus BX-50 with a relay lens system microscope that employed magnifying factors of 400x, 200x, 100x, and 40x. The dataset has been distributed into 70:30 for training and testing, respectively, as a guarantee of the validation. The CNN architectures were employed

*CORRESPONDING AUTHOR | Anwar P.P. Abdul Majeed | amajeed@ump.edu.my

on an NVIDIA Tesla K40m GPU using the Caffe framework. As the results, accuracy at Patient Level: 40X, 100X, 200X and 400X at 96.7%, 88.48%,84.6% and 86.16% respectively plus accuracy at Image Level : 40X, 100X, 200X and 400X at 85.6%, 83.5%,82.7% and 80.70% respectively.

Transfer Learning (TL) employment has already been explored in the study of emotion recognition through the integration of image processing on small datasets [4]. The authors utilized two types of datasets, namely the EmotiW dataset and the FER-2013 dataset. Seven basic expressions, namely happy, surprised, neutral, disgusted, fear, sad, and angry, focused on classifying. The EmotiW images have been converted to grayscale, 256 x 256 resized, then normalized using min-max intensity normalization and intensity. Meanwhile, for the FER-2013 dataset, the size is 48 x 48 smaller than EmotiW dataset but showed a significant performance boost during pre-training. TL with deep CNN architecture (VGG-CNN-M-2048) was proposed as the architecture for the study. The hyperparameter has been set to; initial learning rate: 0.001, momentum: 0.9, and weight decay: 0.0005. The best overall accuracy comes with 48.5% of the validation set and 55.6% of the set against the challenge baseline 35.96% and 39.13%**.**

The employment of the TL paradigm was also adopted for X-ray baggage security imagery purposed on object classification [5]. The researchers pointed out two definite objective issues; firstly, two-category of firearm recognition issues (i.e. knife or no knife). Secondly, it was classified in several categories for X-ray object classification problems on a firearm, firearm-component, laptop, camera, knives, and ceramic knives that employ a similar concept. All images were obtained from AlexNet and GoogLeNet databases. The classification of the first target problem was done through Support Vector Machine (SVM) and RF models, whereby the RF and SVM models attained a true positive rate of 85.81% and 80.74%, respectively. Conversely, the authors utilized a fine-tuned dense network for the second set experiment on the extracted features from AlexNet and GoogLeNet to evaluate the problem [6,7]. The present study attained a mean average precision rate of 95.26% and 98.40%, respectively, clearly demonstrating the efficacy of CNN.

The usage of sensors induced devices body-worn, object-based, and Inertial Measurement Unit (IMU) are worth noting among the researchers. It focuses on using CNN methods for human activity recognition (HAR), image classification, and object detection[8,9]. This strategy embraces a profound CNN to computerize new material by crude information sources systematically. The performances of CNN accuracy was monitored by tuning various important hyperparameters such as convolution layers and kernel sizes. Experimental results show that the CNN model accomplished critical accelerate in registering and choosing the last class and minor improvement in whole CA equalled with the benchmark classifiers, namely SVM, k-NN, Deep Belief Network, and Multilayer perceptron networks.

Although there are studies that have been carried out on the use of ML and TL on sports such as activity recognition, limited research has been carried out on skateboarding, especially image processing method and Inertial Measurement Unit (IMU) sensors were by [10–13] and [14,15] correspondingly. It is essential for an idea of TL, especially the utilization of Inception V3 as a feature extraction technique must be investigated concerning skateboarding [12]. The features extracted were used to assess a few models. It appeared through the examination that the CA of the *k*-NN and Logistic Regression (LR) models yielded commendable classification accuracies.

This paper evaluates one classical ML model, namely k-NN, to classify various skateboarding tricks, i.e. Nollie, Ollie, Frontside 180, Kickflip, and Pop Shove-it created on featured engineered through the three pre-prepared CNN models as a form of TL. Moreover, the readers should know that such a general evaluation has yet been reported in the literature. This result from the examination gives out helpful for further target grounded assessment for judges just as giving away to skateboarders to advance the performance gradually.

METHODOLOGY

Experimental setup

This study is divided into several phases, as illustrated in Figure 1. In the first phase, the type of tricks and stances of the skateboard parameter that stimulates are identified. The selection of tricks was investigated parallel to the skill of skateboarders and from the study in the literature. Methods on measuring these parameters also will be looked upon, and the experimental setup has to be aligned or tailored towards image processing techniques. The second phase of this study will analyze the performance of the skateboard movement using image processing methods. The movements and patterns of the skateboard are scrutinized through image processing based on the parameters found significant in identifying the skateboarding trick manoeuvres. The third phase of this study will focus on the documentation for the most appropriate TL approach to feature extraction and then optimize the hyperparameters of the selected ML model, i.e. *k*-NN, towards its ability to classify the tricks as mentioned earlier.

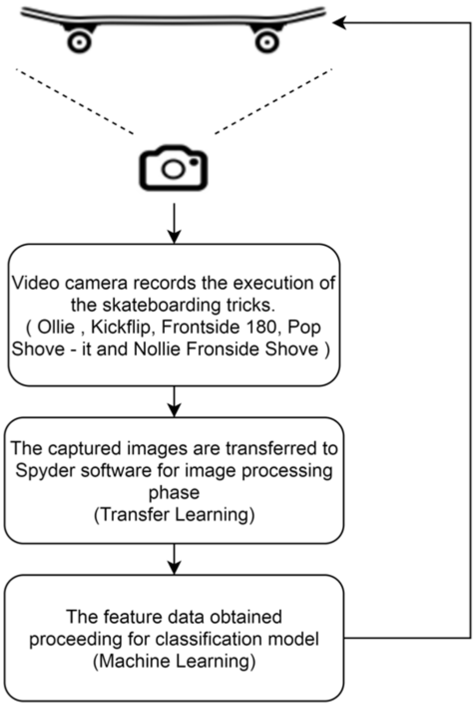


Figure 1. Image schenatic system

One male skateboarder from University Malaysia Pahang skateboarding club, who is 24 years old, 170 cm, 54 kg with almost five years experience, was recruited for the data collection. [14]. As appeared in Table 1, the skateboarder must perform five types of tricks and be repetitive five (5) times per trick, tallied if the trick was effectively executed. The axis of the turn is alluding to the goofy position way. The trick was picked dependent on the experience of the skateboarder. The experimental setup was conducted in a room with a white background (wall).

The Full HD 1080p 60 frames per second YI Action Camera has been employed to capture the execution of skateboarding tricks. The camera specification could be referred to in [12,13], while the research arrangement concerning the camera direction is depicted in Figure 2 [13]. The images caught deriving via research arrangement appear in Figure 3. The separation between the skateboarder and the camera was 1.26 m as it is deliberately positioned to guarantee that the whole perspective on the skateboarder is caught.

In this present investigation, the Keras and Tensorflow Python libraries evoke the pre-trained CNN models. The detailed mathematical treatment of the models is not detailed in this paper; however, the readers are urged to allude to the previous writing [16,17]. Open-source, Keras is an excellent neural network library as it provides a set of APIs to run different neural network backbones such as Tensorflow and Theano. Conversely, an open-source library for numeric computation and dataflow programming allows for the deep learning models known as Tensorflow [18]. It uses the data flow graph to represent complex mathematical operations on multidimensional data arrays.

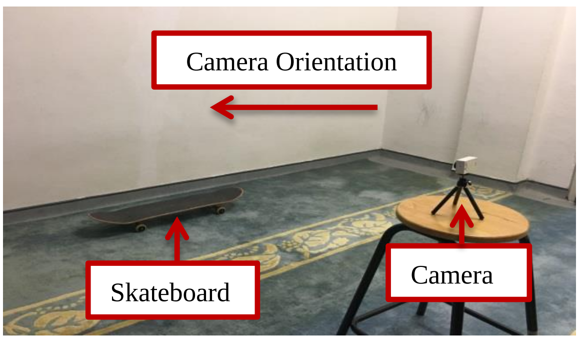
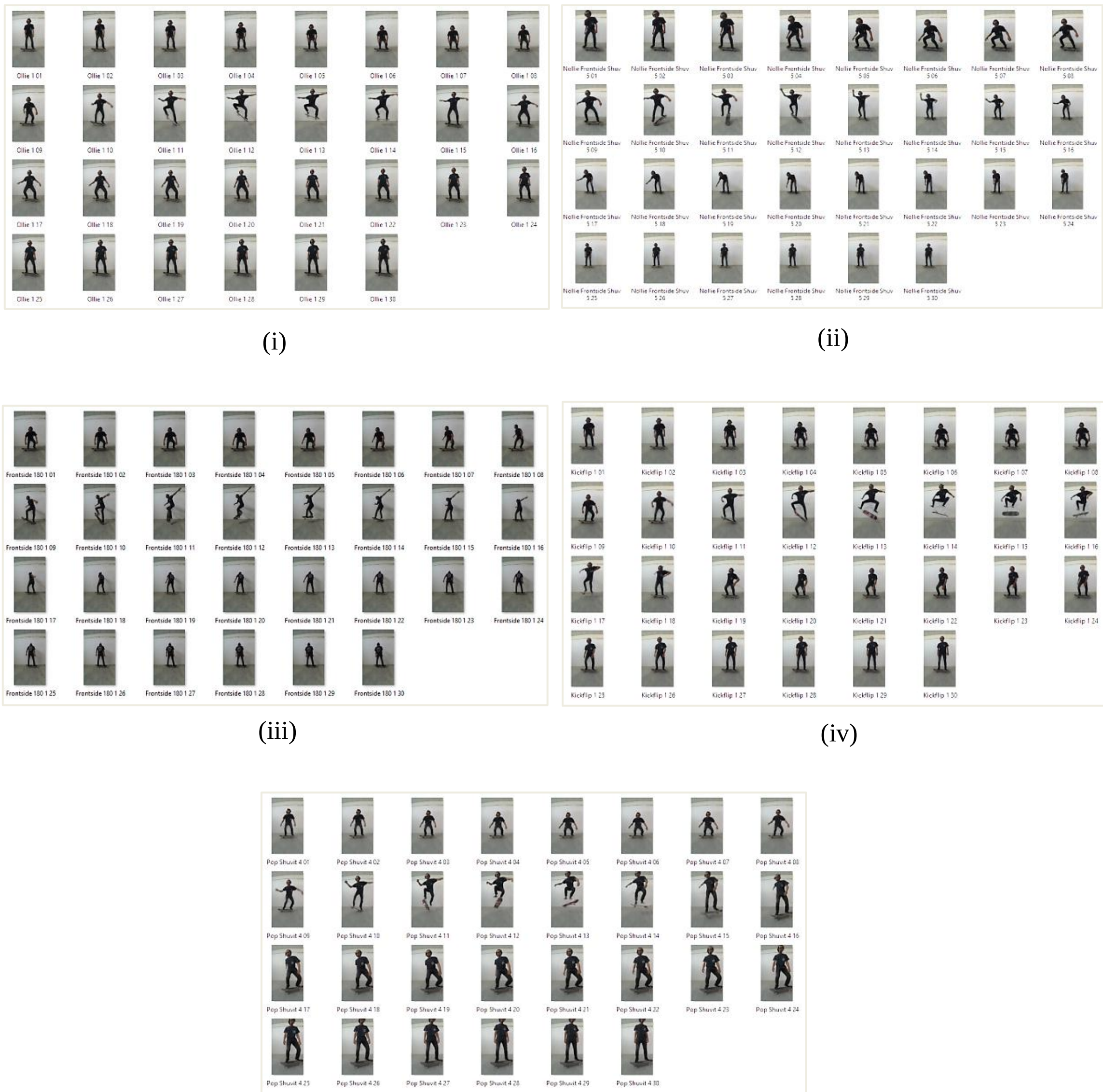


Figure 2. Experimental setup

Table 1. The tricks classes description

Tricks class	Description
Front-side 180, FS (0)	An Ollie but by a 180-degree turn. The board and the body jump and turn simultaneously in a frontward turn showing the front part of the body first.
Kickflip, K (1)	An Ollie and the rider hit their foot out and flips the board all way around laterally its long axis with their toes, allow the board to spin 360 degrees, and then catches it and lands.
Ollie, O (2)	A manoeuvre in skateboarding is when the skater hits the board's tail down at once, jumping to make the board pop into the air.
Pop Shove-it, PS (3)	An Ollie with the board turns 180 degrees (or more) without the board's tail hitting the ground under their feet.
Nollie Frontside Shove-it, NFS (4)	A manoeuvre in which the rider lifts the board into the air by pressing down on it with the front foot, raising the rear foot, and then raising the front foot. (mirror a frontside).



(i) (ii) (iii) (iv) (v)

Figure 3. Execution of the tricks; (i) O (ii) NFS (iii) FS (iv) K (v) PS

Data pre-processing

The dataset videos that have been collected have a variety in videos time length. By using the VLC media player 2.2.6, all the videos were trimmed from the start point of the execution until the landing of the tricks. The video was consequently utilized for the image extraction progression for a time between two to three seconds (the execution of the trick only). Then, the employed of video to jpg converter v.5.0.101 for extract the frames per frame images by a dimension of 1080 x 1920 pixels as in Figure 4 [19]. Moreover, in Figure 5, the images were be resized to 300 x 300 pixels using Caesium software as it a proper form for the input shape condition of the TL model. The frames of the images were being embarked on 30 frames for every video to synchronize the datasets. A sum of about 750 images was extracted out of 25 videos caught by the confirmed landed tricks. All experiment is conducted on Keras API with TensorFlow by architectures based on Python on a laptop, an Intel Core i5-3217U 1.8GHz CPU and 4 GB RAM.

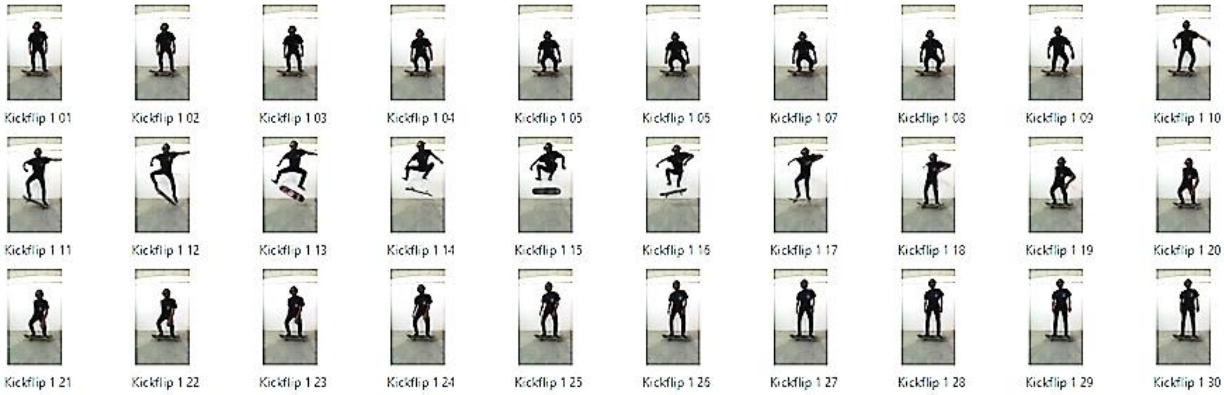


Figure 4. The original size 1080 x 1920 pixels

Figure 5. The resized images 300 x 300 pixels

Data augmentation

The images related study needs large datasets to get accurate performance, plus access to the big data source, especially for skateboarding images/videos, was limited. Moreover, overfitting on image data is a common fault that easily been reduce by artificially enlarge the dataset using the data augmentation technique [6]. Data augmentation is a technique that extends the diversity of data without actually collecting new data that save time and expense. The horizontal flip (Figure 6), positive 90º rotate (Figure 7), and negative 90º rotate (Figure 8) are common types of data augmentation techniques used to train large neural networks [20]. From the initial 750 images (used for model development), approximately 2250 new images (used for independent testing) were generated via the augmentation described above techniques. Therefore, a total of 3000 images are used in this investigation.

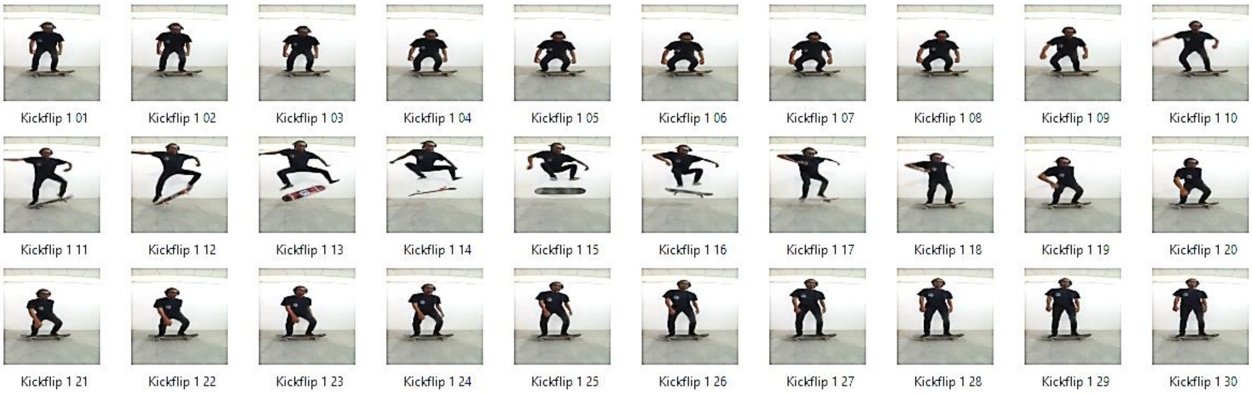


Figure 6. The horizontal flip images

Figure 7. The positive 90º rotate images

Figure 8. The negative 90º rotate images

Transfer learning

TL in deep CNNs is a state-of-the-art employment strategy apart from being computationally inexpensive, which is valuable in preparing models from a restricted dataset that would dodge the idea of overfitting from a roughen model [21,22]. Several common family models are available, namely AlexNet, GoogLeNet, VGG, DenseNet, and Inception via [6,7,23,24] and [25]. In a total of three TL models would be proposed in this study as preferred in Table 2. The CA obtained from each of these models will be study and compared.

The top-5 and top-1 accuracy talk about the model's performance on the ImageNet dataset that has been validated (Krizhevsky & Hinton, 2012). Top-5 accuracy implies that any of your models that offer the five most possible responses are precisely the predicted answer. Top-1 accuracy is the prediction accuracy that the model answer purposely (the one with the highest probability) would be predicted. Depth refers to the topological depth of the network. The default input size was the size of the input shape of each model, and each of it has to be (* x * x 3) as RGB ought to be three information sources channels. The flatten size as the last fully connected layers, a combination of these features, creates a complete CNN architecture model.

Table 2. Individual of transfer learning models

Model	Top-5 Accuracy	Top-1 Accuracy	Input Size	Flatten Size
MobileNet	0.895	0.704	224x224	7x7x1024
NASNetMobile	0.919	0.744	224x224	7x7x1056
NASNetLarge	0.960	0.825	331x331	11x11x4032

Classification

In this paper, the research evaluates ML technique for classified skateboarding tricks, specifically *k*-NN. The detailed articulations for the models are hardly expounded in this paper. In any case, the readers are urged to allude to the previous writing [26,27]. The *k*-NN model employed the distance between the marked info spaces of neighbors. The Euclidean distance approach is usually applied to ascertain the distance among all neighbors and demonstrated successfully in classifying skateboarding tricks images on the same classifier evaluated in [12]. The main distance functions of *k*-NN:

$$\rho(x, x') = \| x - x' \| = \sqrt{\sum_{i=1}^{d} (x_i - x'_i)^2} \quad (1)$$

Where ρ can be denoted as the Euclidean distance between the point of xi of new data and x′i of the training data, the notation of d is the dimension of the real number set. As for hyperparameter tuning, the number of neighbors, k, and other distance metrics will be evaluated to gauge the minimum cost function of the objective model. The evaluation of hyperparameter by tuning the number of neighbor and distance metrics will provide the variables' optimum value, hence fitting into the model to resolute into the final accuracy rate compared.

In the present study, the four distance metrics employed are Minkowski, Euclidean, Cosine, and Manhattan, with the number of neighbors *k*, ranging from 20 to 30 are explored. The best hyperparameter is ascertained via the Grid search technique. The classifiers' fulfilment on the dataset was shown through progressive classification assessed depending on the CA and confusion matrix.

Proposed Architecture

In the present study, pre-trained CNN models were utilized to extract the features of the tricks before it was fed to the classifier [28]. The flow of the proposed methodology does show in Figure 9. The methodology is divided into three main phases, namely the TL model where feature extraction arises, the Classification Model for the classified trick, and Model Evaluation in which the CA of the models was evaluated, and the best pipeline is identified.

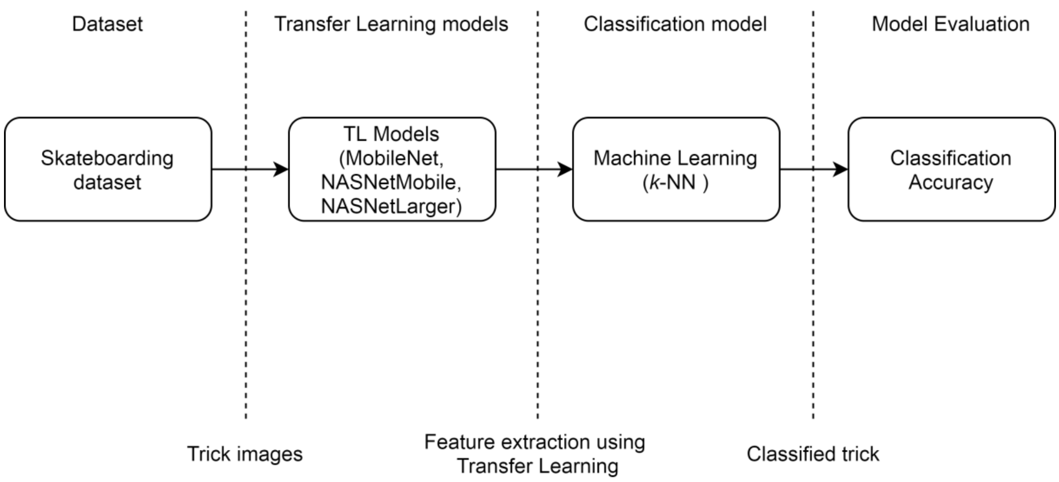


Figure 9. Transfer learning and classification model proposed architecture

EXPERIMENTAL RESULTS

Classification of TL models coupled with *k*-NN classifier

The performance of the trick was recorded as all out of 40 tricks occasions were completed by the skateboarder. From the 40 tricks, just 25 tricks stood discovered to be complete landing. A sum of around 750 images was extracted (in this the images, i.e., 149 for FrontSide180, 150 for Kickflip, 127 for Nollie Front Shove-it, 146 for Ollie and 150 for Pop Shove-it) obtained and then would proceed through a train, validation, and test split of 60:20:20 ratio with training, validation and testing respectively. Then, the data would be extracted and engineered using the TL models coupled with the *k*-NN classifier.

Table 3 lists the CA performance of the TL+ *k*-NN fusion pipelines, respectively. The best TL pipeline was the MobileNet with an optimized *k*-NN model that employs the Manhattan distance metrics with 20 number neighbors. The evaluated models are depicted in Figure 10. The MobileNet model achieved a CA of 99.00% on training, 97.00% on validation, and 95% on testing, respectively. Moreover, the NASNetMobile model achieved CA on training for 98.00%, validation for 94.00%, and testing for 92.00%, correspondingly. Lastly, NASNetLarge model CA for training, validation, and testing attained 95.00%, 92.00%, and 90.00%, respectively.

The performance can be increased by performing hyperparameter optimizing or fine-tuning strategies on the proposed model. Plus, increase the presence of the dataset towards the data collection phase, or simply implant data augmentation methods as it saves time and expense.

Table 3. The CA of the TL + *k*-NN fusion pipelines

	CA		
		Phase	
Model + *k*-NN	Training	Validation	Testing
MobileNet	99%	97%	95%
NASNetMobile	98%	94%	92%
NASNetLarge	95%	92%	90%

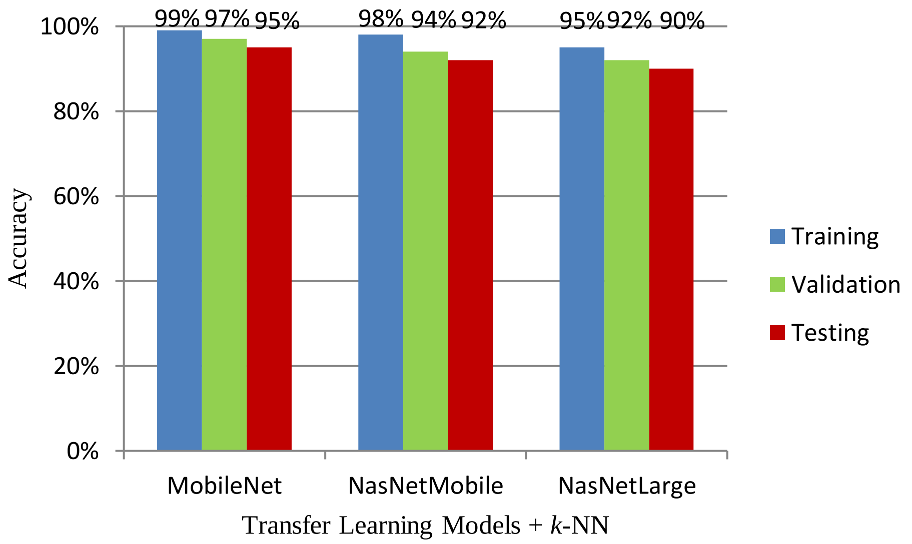
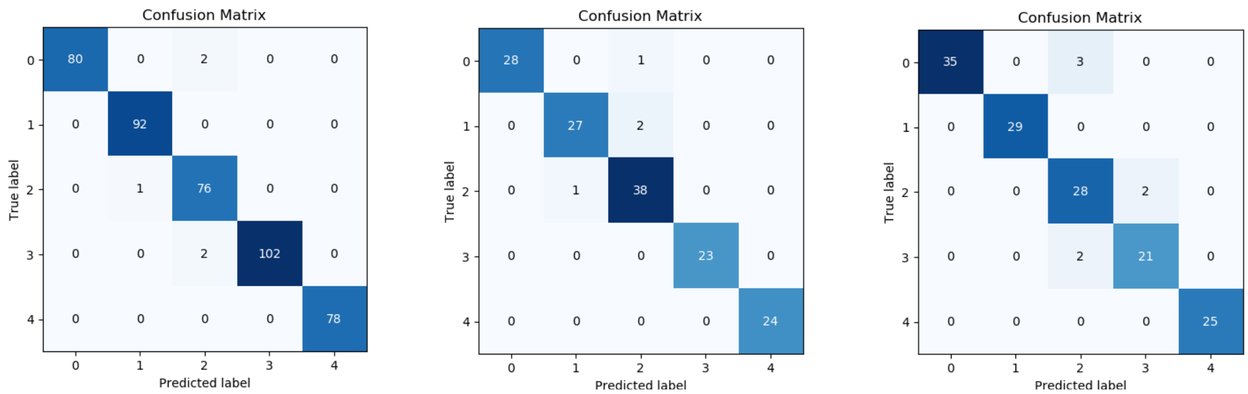


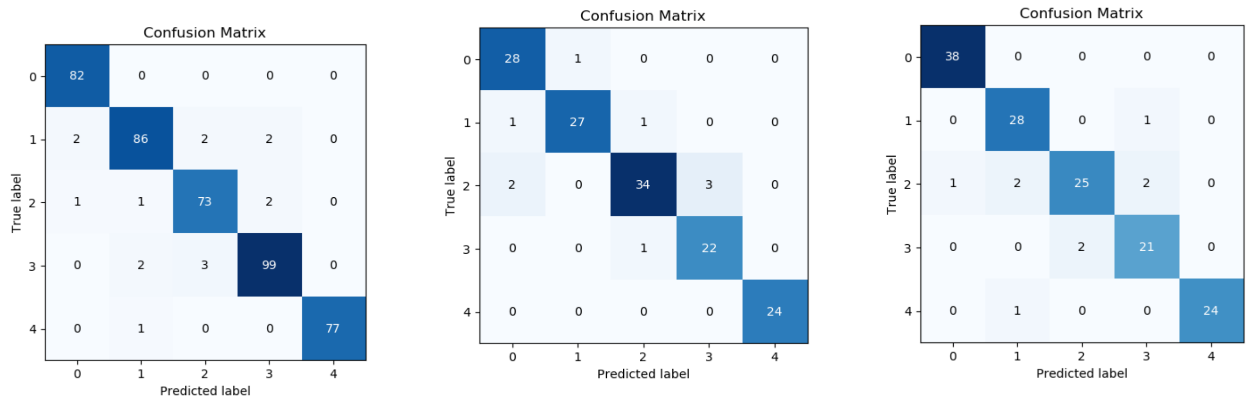
Figure 10. The performance for the fusion pipelines

The additional evaluation might be through on the confusion matrix for VGG19, DenseNet201, plus ResNet152 models (Figure 11, Figure 12 and Figure 13, correspondingly). It could be seen from Figure 11 (i) and (iii), the misclassification ascends by the FS and PS images trick that misclassified for O then O misclassified as K for (i) and O missclassified as PS for (iii). Meanwhile, the misclassification for (ii) was arising from FS and K that misclassified for O then O misclassified for PS. Furthermore, evident from Figure 12 (i) and (iii) both have misclassification over all the classes except for the FS. For (ii) the misclassification transpired over all classes except for the NFS. Moreover, from the confusion matrix in Figure 13 for the (i), (ii), and (iii) we can observe that the misclassification transpired across all classes except for the NFS classes. The proposed procedure could classify skateboarding tricks, especially by melding MobileNet with a *k*-NN classifier.



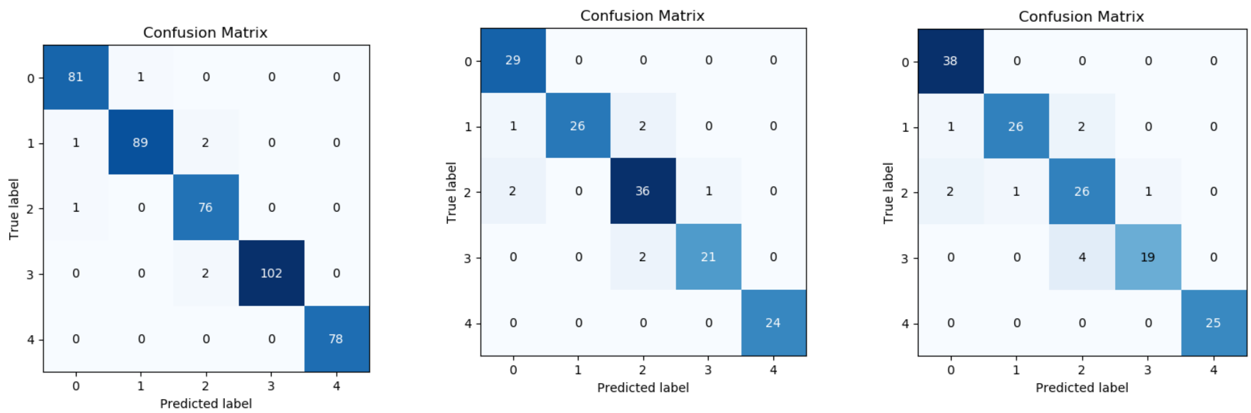
(i) (ii) (iii)

Figure 11. The confusion matirx for (i) training (ii) validation (iii) testing of MobileNet *k*-NN model



(i) (ii) (iii)

Figure 12. The confusion matirx for (i) training (ii) validation (iii) testing of NASNetMobile *k*-NN model



(i) (ii) (iii)

Figure 13. The confusion matirx for (i) training (ii) validation (iii) testing of NASNetLarge *k*-NN model

Robustness evaluation

In section 2.3, a total of 2250 new images data were developed using data augmentation methods. Using this augmented data evaluates the robustness of the developed architectures via this form of independent testing. Hence, the model's effectiveness in classifying the images under a variety of conditions could be ascertained.

Firstly, the horizontal flip method, as in Figure 14 could be observed. The CA for MobileNet model achieved a 95%, while the NASNetMobile model achieved 98%. The NASNetLarge model attained 97% of CA. Next, in Figure 15, a positive 90^{o} rotate technique was noticed. The MobileNet model achieved 96% of CA. Meanwhile, the NASNetMobile and NASNetLarge models succeeded a CA of 94% and 92%, correspondingly. Furthermore, negative 90 ^{o} rotate techniques as in Figure 16. The CA has mirrored the previous positive 90-degree method results, as MobileNet achieved 96% and both NASNetMobile and NASNetLarge reached 94% and 92% of CA, respectively. It is evident from the robustness evaluation that the MobileNet pipeline was more robust as it could well classify the tricks under a variety of conditions.

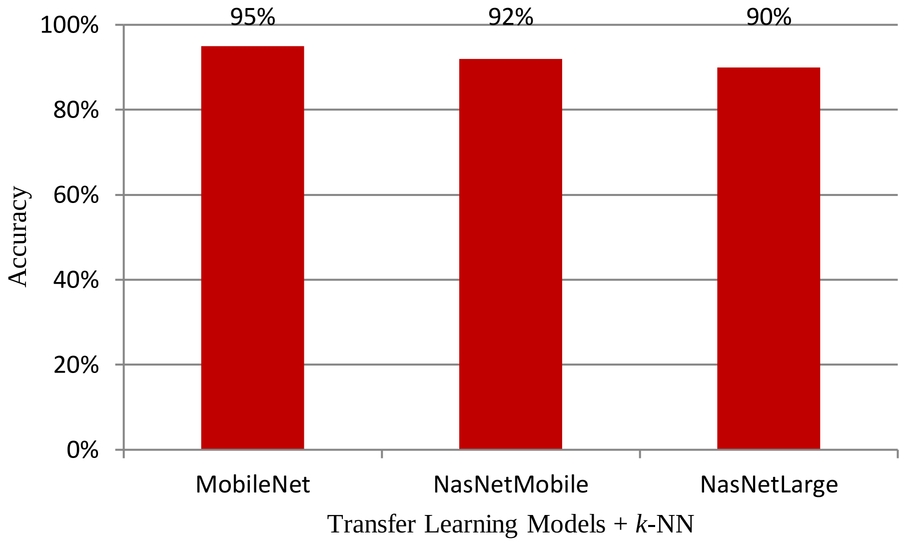


Figure 14. The classification accuracy of model robustness by horizontal flip technique

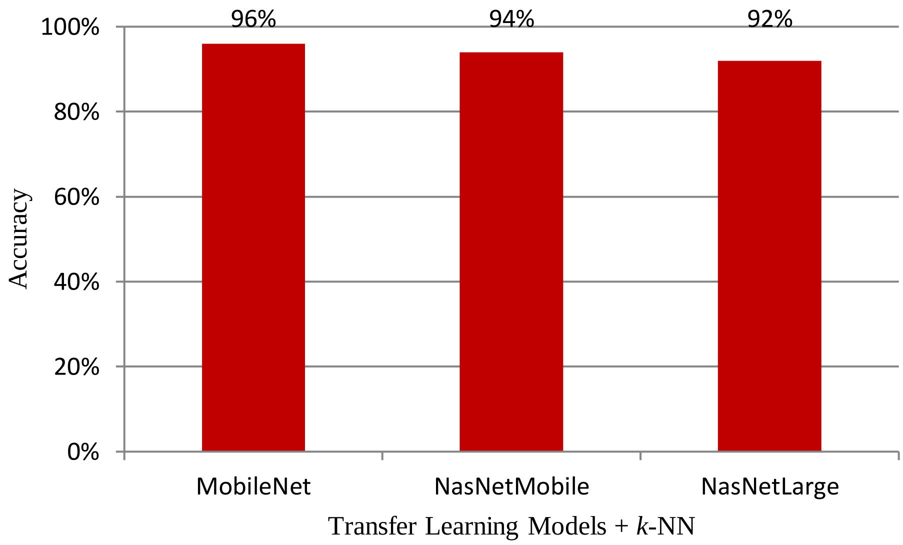


Figure 15. The classification accuracy of model robustness by positive 90º rotate technique

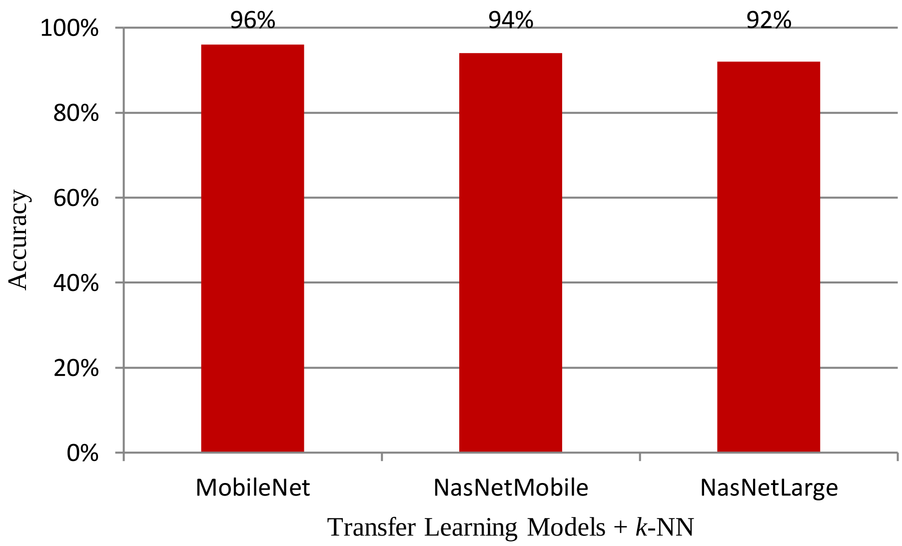


Figure 16. The classification accuracy of model robustness by negative 90º rotate technique

CONCLUSION

In conclusion from the current experimental suggests that the offline skateboarding tricks classification utilizing image processing does show significant positive results. It this been proved through the study that the employment of TL could extract relevant features that eventually provide rational CA of the assessed skateboarding tricks [28]. It has shown that the MobileNet model achieved a decent CA of 95%. Moreover, it's been approved the robustness of the MobileNet enhanced *k*-NN pipeline was capable and effective in classifying skateboarding images. Hence, the purposes of the data augmentation technique are proved as it could increase the diversity of data without actually collecting new data that save time and expense. It is worth noting that the current study would be additionally examined by evaluating different ML models, namely, SVM and RF, to enhance the hyperparameters of various ML models. The current research findings recommend the suitability of the present model for giving an objective-based decision on skateboarding tricks rather than ordinary personal methods that are now being practiced in this skateboarding competition.

ACKNOWLEDGEMENT

The authors should acknowledge UMP for supporting this study under an internal grant (FRGS/1/2019/TK03/UMP/02/6 & RDU1901115).

REFERENCES

[1] N.K. Corrêa, J. César, M. De Lima, T. Russomano, M. Araujo, Development of a skateboarding trick classifier using accelerometry and machine learning, 33 (2017) 362–369. https://doi.org/10.1590/2446-4740.04717.
[2] S. Lee, Human Activity Recognition From Accelerometer Data Using Convolutional Neural Network, 2017 IEEE Int. Conf. Big Data Smart Comput. (2017) 131–134. https://doi.org/10.1109/BIGCOMP.2017.7881728.
[3] F.A. Spanhol, P.R. Cavalin, L.S. Oliveira, C. Petitjean, L. Heutte, Deep features for breast cancer histopathological image classification, 2017 IEEE Int. Conf. Syst. Man, Cybern. SMC 2017. 2017–Janua (2017) 1868–1873. https://doi.org/10.1109/SMC.2017.8122889.
[4] H. Ng, V.D. Nguyen, V. Vonikakis, S. Winkler, Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning, (2015) 443–449.
[5] M.E. Kundegorski, M. Devereux, T.P. Breckon, Transfer Learning Using Convolutional Neural Networks for Object Classification Within X-Ray Baggage Security Imagery, (2016). https://doi.org/10.1109/ICIP.2016.7532519.
[6] A. Krizhevsky, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, (2012) 1–9.
[7] C. Szegedy, V. Vanhoucke, J. Shlens, Rethinking the Inception Architecture for Computer Vision, 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (2014) 2818–2826. https://doi.org/10.1109/CVPR.2016.308.
[8] J.B. Yang, M.N. Nguyen, P.P. San, X.L. Li, S. Krishnaswamy, Deep Convolutional Neural Networks On Multichannel Time Series For Human Activity Recognition, (2015) 3995–4001.
[9] T. Zebin, Human Activity Recognition with Inertial Sensors using a Deep Learning Approach, 2016 IEEE SENSORS. (2016) 1–3. https://doi.org/10.1109/ICSENS.2016.7808590.
[10] B.H. Groh, T. Kautz, D. Schuldhaus, IMU-based Trick Classification in Skateboarding, KDD Work. Large-Scale Sport. Anal. (2015).
[11] B.H. Groh, M. Fleckenstein, T. Kautz, B.M. Eskofier, classification and visualization of skateboard tricks using wearable sensors, Pervasive Mob. Comput. 40 (2017) 42–55. https://doi.org/10.1016/j.pmcj.2017.05.007.
[12] M.N.A. Shapiee, M.A.R. Ibrahim, M.A. Mohd Razman, M.A. Abdullah, R.M. Musa, M.H.A. Hassan, A.P.P. Abdul Majeed, The Classification of Skateboarding Trick Manoeuvres Through the Integration of Image Processing Techniques and Machine Learning, in: Lect. Notes Electr. Eng., Springer, 2020: pp. 347–356. https://doi.org/10.1007/978-981-15-2317-5_29.
[13] M.N.A. Shapiee, M.A.R. Ibrahim, M.A.M. Razman, M.A. Abdullah, R.M. Musa, A.P.P. Abdul Majeed, The Classification of Skateboarding Tricks by Means of the Integration of Transfer Learning and Machine Learning Models, Lect. Notes Electr. Eng. 678 (2020) 219–226. https://doi.org/10.1007/978-981-15-6025-5_20.
[14] M.A. Abdullah, M.A.R. Ibrahim, M.N.A. Bin Shapiee, M.A.M. Razman, R.M. Musa, A.P.P.A. Majeed, The Classification of Skateboarding Trick Manoeuvres Through the Integration of IMU and Machine Learning, in: Symp. Intell. Manuf. Mechatronics, Springer, 2019: pp. 67–74.
[15] M.A.R. Ibrahim, M.A. Abdullah, Nur, M.N.A. Shapiee, M.A. Mohd Razman, R. uazu Musa, M.A. Zakaria, Nn.A. Abu Osman, A.P.P.A. Majeed, The Classi fi cation of Skateboarding Trick Manoeuvres : A K -Nearest Neighbour Approach, Proc. 2019 Movement, Heal. Exerc. Int. Sport. Sci. Conf. (2020) 341–347.
[16] J. Koushik, Understanding Convolutional Neural Networks, 29th Conf. Neural Inf. Process. Syst. (NIPS 2016), Barcelona, Spain. (2016) 1–6.
[17] W. Rawat, Deep Convolutional Neural Networks for Image Classification : A Comprehensive Review, 2449 (2017) 2352–2449. https://doi.org/10.1162/NECO.
[18] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, G. Brain, I. Osdi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow : A System for Large-Scale Machine Learning, (2016).
[19] Y. Su, T. Chiu, C. Yeh, H. Huang, W.H. Hsu, Transfer Learning for Video Recognition with Scarce Training Data for Deep Convolutional Neural Network, (2015) 1–12.

[20] B. Zhang, L. Wang, Z. Wang, Y. Qiao, H. Wang, Real-time Action Recognition with Enhanced Motion Vector CNNs, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2016) 2718–2726.

[21] K.H. Jin, M.T. Mccann, E. Froustey, M. Unser, Deep Convolutional Neural Network for Inverse Problems in Imaging, 26 (2017) 4509–4522.

[22] S.S.S. Kruthiventi, K. Ayush, R.V. Babu, S. Member, DeepFix : A Fully Convolutional Neural Network for predicting Human Eye Fixations, (2015) 1–11.

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, 2016 IEEE Conf. Comput. Vis. Pattern Recognit. 45 (2016) 1951–1954. https://doi.org/10.1002/chin.200650130.

[24] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. (2015) 1–14.

[25] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. (2017) 2261–2269.

[26] S.A. Dudani, DISTANCE-WEIGHTED k-NEAREST NEIGHBOR RULES., IEEE Trans. Syst. Man Cybern. SMC-8 (1978) 311–313. https://doi.org/10.1109/TSMC.1978.4309958.

[27] Z. Taha, A. Razman, A.P.P.A. Majeed, The classification of hunger behaviour of Lates Calcarifer through the integration of image processing technique and k -Nearest Neighbour learning algorithm, (2018). https://doi.org/10.1088/1757-899X/342/1/012017.

[28] A.B. Sargano, X. Wang, P. Angelov, Z. Habib, Human Action Recognition using Transfer Learning with Deep Representations, (2017).