Pilates Pose Classification Using MediaPipe and Convolutional Neural Networks with Transfer Learning

Kenneth Angelo Tanjaya, Mohammad Farid Naufal, Heru Arwoko
Tenik Informatika, Universitas Surabaya, Jalan Raya Kalirungkut, Surabaya 60293, Indonesia

ARTICLE INFO

Article history:

Received March 21, 2023
Revised April 13, 2023
Published April 17, 2023

Keywords:

Pilates;
Pose;
Classification;
Mediapipe;
Transfer learning;
CNN

ABSTRACT

A sedentary lifestyle can lead to heart disease, cancer, and type 2 diabetes. An anaerobic exercise called pilates can address these problems. Although pilates training can provide health benefits, the heavy load of pilates poses may cause severe muscle injury if not done properly. Surveys have found that many teenagers are unaware of the movements in pilates poses. Therefore, a system is needed to help users classify pilates poses accurately. MediaPipe is a system that accurately extracts the real time human body skeleton. Convolutional Neural Network (CNN) with transfer learning is an accurate method for image classification. There have been several studies investigated pilates poses classification. However, there is still no research applies the MediaPipe as a skeleton feature extractor and CNN with a transfer learning to classify pilates poses. In addition, previous research still does not implement the pilates poses classification in real-time. Based on this problem, this study creates a system using MediaPipe as a feature extractor and CNN with transfer learning as a real-time pilates poses classifier. This study utilized five types of Pilates poses: Warrior, Tree, Plank, Goddess, and Downward Dog.This system runs on a mobile device and gets information from a camera sensor. The results from MediaPipe then be classified by pre-trained CNN architectures with transfer learning: MobileNetV2, Xception, and ResNet50. The best model was obtained by MobileNetV2, which had an f1 score of 98%. Ten people who didn't know much about Pilates also tested the system. They all agreed that the app could accurately identify Pilates poses, make people more interested in Pilates, and help them learn more about Pilates.

Corresponding Author:

Mohammad Farid Naufal, Universitas Surabaya, Jl. Raya Kalirungkut, Surabaya 60293, Indonesia
Email: faridnaufal@staff.ubaya.ac.id

1. INTRODUCTION

Sedentary lifestyle is a habit of someone who lacks physical activity or exercise [1]. A sedentary lifestyle has been identified as one of the global health problems in the activities of adolescent life. About a third of teenagers have a physical lifestyle and 41.5% of teenagers spend time sitting more than or equal to four hours per day. If this lifestyle is continued without prevention, the risk of heart disease, cancer, and type 2 diabetes will increase. This can be avoided by starting to exercise such as fast running, HIIT (high-intensity interval training), push-ups, some muscle weight exercises, and pilates. By doing structured movements, exercise has several benefits, such as endurance, strength, balance, and the body's flexibility [2].

Pilates is an anaerobic sport that aims to increase muscle strength, endurance, flexibility, balance, and spinal stability (Shedden, 2006). Despite the health benefits, the heavy weight involved in these exercises can cause serious injury to muscles or ligaments. Many people do these exercises regularly, but they may do the incorrect poses. This could be due to a lack of formal training through classes or personal trainers. In addition it could be due to muscle fatigue or using too much weight.

For the Pilates exercise process to run smoothly, a pose classification system that can assist users in recognizing accurate Pilates poses needs to be designed [3]. Pose classification is a type of computer technology that uses visual techniques to estimate the location of a person or object. Pose classification is done by placing keypoints according to the location of coordinates on a person's body parts from camera. Pose classification aims to recognize the poses correctly.

An example of a system that can extract human body poses is MediaPipe. MediaPipe is an open-source framework designed explicitly for complex pipelines that utilize the GPU or CPU to run models such as face detection, hand detection, object detection, and finally, pose estimation [4]–[7] Convolutional Neural Network (CNN) is an algorithm used for image classification [8]. CNN uses a convolution layer to extract features, which can then be fed directly into the neural network model. Transfer learning is a CNN model created for one job and used as a basis for models on different tasks [9]. Transfer learning is a fast and accurate method for classifying images because it has a pre-trained model.

Human posture detection has advanced significantly during the last few decades. Due to current computers increased computing power and recent advances in deep learning, posture detection has evolved [10], [11]. There are several previous studies conducted similar research. Kishore *et al.* [12] examined the pose classification in five Yoga poses by comparing four types of pose estimation, namely EpipolarPose, OpenPose, and PoseNet. The results showed that the research has better accuracy than other methods, with an accuracy of each pose of 78.78% for the half-moon pose, 90.9% for the mountain pose, 85.75% for the triangular pose, 81.81% for the warrior pose, and 88.81 % for tree pose. The weakness of this research is still not implementing CNN transfer learning architecture. A method for correcting posture is suggested by Chen *et al.* [13] in their book YogaST. It employ kinect technology to divide positions into three groups: Tree Pose, Downward Dog, and Warrior Pose. However, this research uses the Kinect sensor, which is not a cheap device. Using high-level human skeletal data that has been preprocessed, Pismenskova *et al.* [14] provide neural networks model. The researchers identified 16 important spots on the human body that correspond to different traits. The input layer set as a whole, which consists of 36 neurons, had the best accuracy of 85%. However, conventional skeletonization procedures are far more precise and effective than the Deep learning-based techniques. A vision-based intelligent teaching system for yoga practitioners is suggested by Kale *et al.* [15]. The suggested strategy is taught using a variety of yoga streams produced by the eight yoga professionals shown in the films. To identify posture from the photos, Byeon *et al.* [16] provide an ensembled deep model based on several CNN models. 51 different domestic settings are used to educate the network for fundamental yoga postures. One potential limitation of this research could be the lack of diversity in the study sample. The posture database used in the experiments was constructed at the Electronics and Telecommunications Research Institute (ETRI), Korea, and only included images from 51 home environments. As such, the findings may not generalize to other populations or environments with different characteristics. Additionally, the study did not report any information on the age or health status of the individuals in the images, which could affect the performance of the posture recognition system. Furthermore, the study did not provide any information on the potential ethical considerations or limitations associated with using deep learning algorithms for posture recognition in home environments.

When it comes to posture identification in the multidisciplinary field of Pilates, there has been limited research on posture classification. Previous studies have used keypoint identification techniques, but this study aims to address the limitations of existing models by creating a more accurate model for categorizing Pilates poses. Interestingly, there have been no previous attempts to classify Pilates poses using a CNN with transfer learning model from mobile devices camera. This study focuses on classifying five Pilates poses - Downward Dog, Tree, Goddess, Warrior, and Plank - using MediaPipe as a feature extractor and CNN with Transfer Learning as a classifier. The transfer learning architectures used are MobileNetV2, Xception, and ResNet50, and their performances are compared to help researchers choose the best architecture for classifying Pilates poses. The study seeks to answer the following questions: Can a CNN with transfer learning model be used to classify Pilates poses from mobile devices camera? Which transfer learning architecture (MobileNetV2, Xception, or ResNet50) performs the best for classifying Pilates poses? How does the performance of each architecture compare, and what is the best option for researchers seeking to classify Pilates poses?

2. METHODS

The methods used in the research are dataset collection, feature extraction, model training and testing, and model implementation. Fig. 1 shows the research methods used in this study.

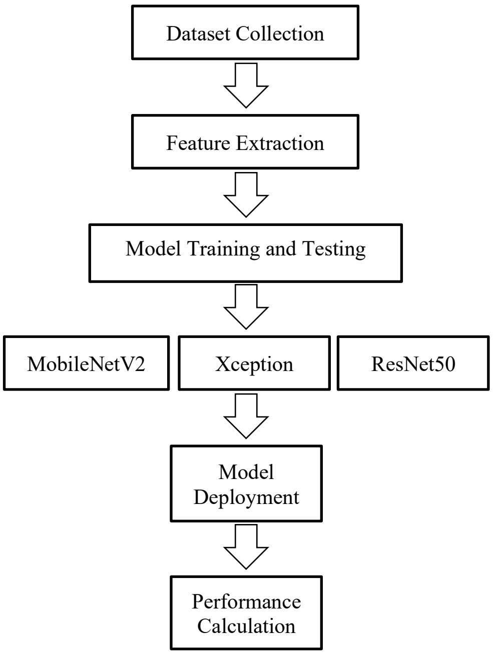


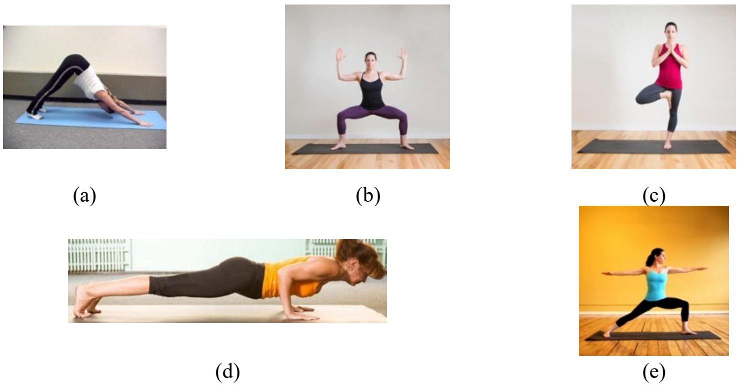
Fig. 1. Research methodology

2.1. Dataset Collection

In this study, the dataset was obtained from Kaggle, which is a combination of several accounts that provide pictures of Pilates poses. The dataset contains 1,253 images of poses that are classified into 5 categories: Warrior, Tree, Plank, Goddess, and Downward Dog. The distribution of images for each pose can be found in Table 1. Sample images of the Pilates poses can be viewed in Fig. 2. There was no preprocessing stage involved in this study as Mediapipe directly extracts the skeleton from the image.

Table 1. Number of images per pose

Pose	Total
Warrior	345
Tree	196
Plank	178
Goddess	210
Downward Dog	324


(a) (b) (c)

(d) (e)

Fig. 2. Downward dog (a), Goddess (b), Tree (c), Plank (d), and Warrior (e)

2.2. Feature Extraction

The feature of each pilates pose image is extracted using the MediaPipe. An image will be loaded first in RGB format. The process starts by using a pre-trained deep neural network model to detect key points or landmarks on the body, such as the nose, shoulders, elbows, wrists, hips, knees, and ankles.These key points are then used to construct a skeletal model that represents the posture and pose of the human body. MediaPipe's pose estimation pipeline is optimized for real-time performance and can operate on a variety of input sources, including webcams, videos, and image files. This makes it a useful tool for a range of applications, including fitness tracking, motion capture, and virtual reality.After all the pose images are extracted, the skeleton features are then used as training and testing data. Fig. 3 shows the results of the skeleton feature extraction using MediaPipe.

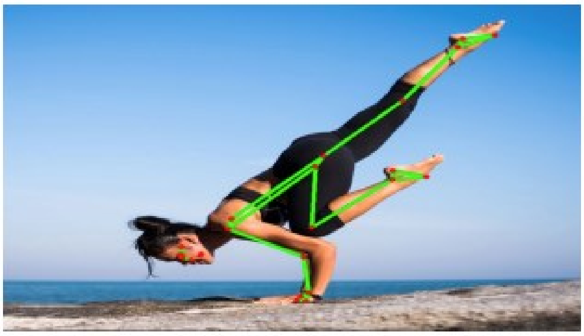


Fig. 3. The results of skeleton feature extraction using MediaPipe. Before extraction (a) and after extraction (b)

2.3. Model Training and Testing

Before training and testing the model, pose data will be split into two parts: testing and training data. The training data is used during the model training process, and the testing data is used to test the model. The two parts are divided into 80% for training data and 20% for testing data. The number of each data obtained after dividing is 1,002 images of poses for training data and 251 images of poses for testing data.

In this study, the pre-trained model used are MobileNetV2, Xception, and ResNet50. MobileNetV2 offers two additional features: linear bottlenecks and short links between bottlenecks [17]–[22]. There are inputs and outputs between models at the bottleneck. At the same time, the inner layer contains the model's ability to change the input from low-level concepts (pixels) to high-level descriptors (image categories). This method provides faster training and better accuracy. Fig. 4 shows the MobileNetV2 architecture. ResNet50V2 is an upgraded version of ResNet50V1. Changes are made to the formulation of link propagation between blocks in ResNet50V2. Using the ImageNet dataset, ResNet50V2 also performs well [23]–[28] Fig. 5 shows the ResNet50 architecture. Xception is an acronym for "extreme inception". Xception uses the same model parameters as InceptionV3 but excels on the ImageNet 17,000 class dataset [29]–[34]. ImageNet is a large image database based on the WordNet structure [35]–[40]. Fig. 6 shows the Xception architecture.

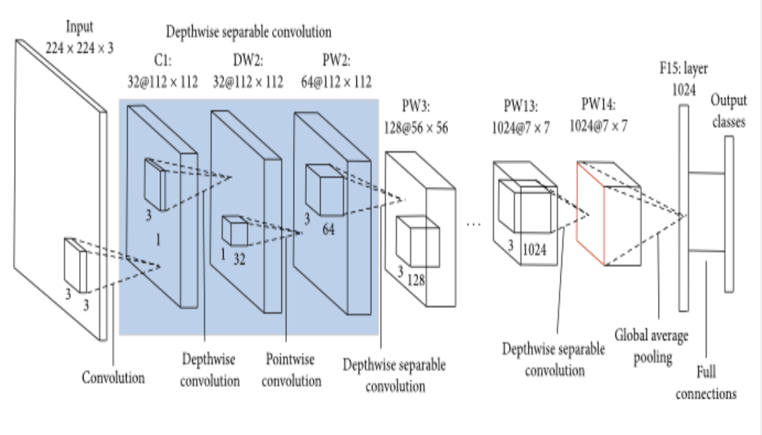


Fig. 4. MobileNetV2 architecture

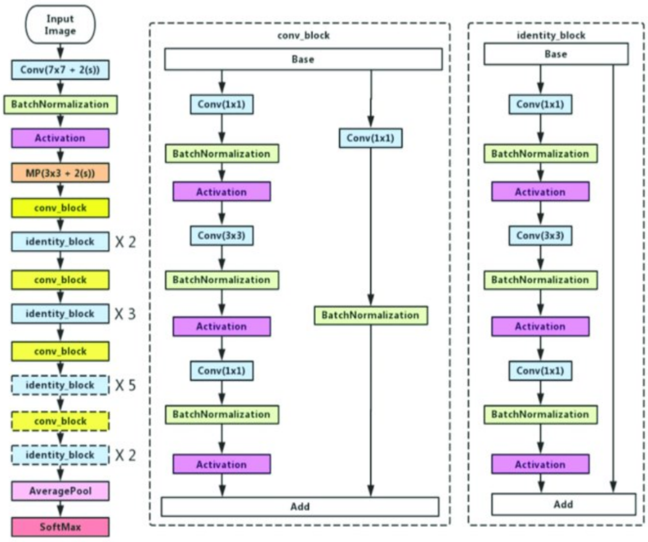


Fig. 1. ResNet50 architecture

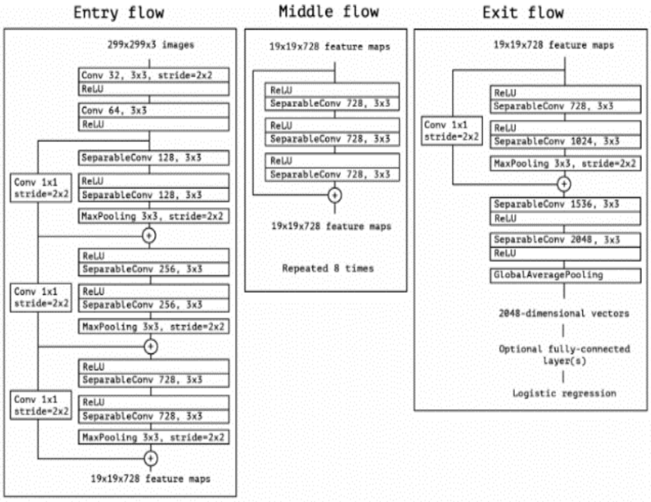


Fig. 6. Xception architecture

The MobileNetV2, Xception, and ResNet50 implementations use a 224×224 pixel image input with three channels. Several hyperparameters were adjusted, including 'include_top', which was set to false as the output layer was tailored according to the number of pose classes. The SGD (Stochastic Gradient Descent) optimizer was used in all three models, with a learning rate of 0.005. SGD was chosen for its efficiency, ease of implementation, and successful application to large-scale machine learning problems [37]-[40]. The loss function used was categorical cross-entropy, and the metric used was categorical accuracy. The model was trained for 50 epochs, as specified in Table 2.

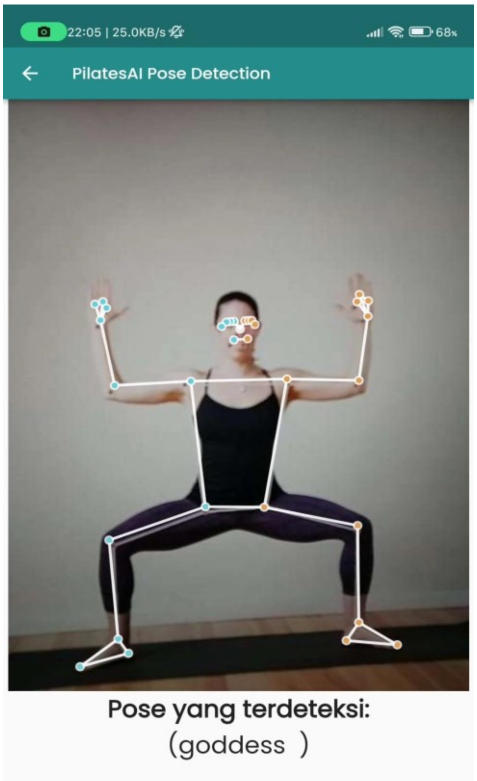
During the testing process, each model was evaluated on the testing data and generated a confusion matrix with class 0 for the downward dog pose, class 1 for the goddess pose, class 2 for the plank pose, class 3 for the tree pose, and class 4 for the warrior pose. In addition to the confusion matrix, the model's accuracy, precision, recall, and f1 score were compared.

Table 2. Hyperparameter of Transfer Learning Model

Hyperparameter	Value
Input Value	(224, 224, 3)
Include Top	False
Optimizer	SGD
Learning Rate	0.005
Epoch	50
Dense Layer	1 dense layer with 128 neurons

2.4. Model Deployment

Implementation of a pose classification system using the Python programming languages and Flutter framework. The model results obtained from the training and testing methods will be converted into a TensorflowLite model. When the pose classification is accessed, the camera will be launched and the TensorflowLite model will be loaded. After the camera is turned on, the results of the camera frame appear on the user's screen while sending the camera frame to MediaPipe. MediaPipe then extracts the skeleton keypoints for each body part. The results of these keypoints will be received by the application and will be painted onto the image from the camera. Then the image is retrieved by a function and converted into a list containing RGB bytes. Then the images are normalized so that they have a pixel range of 0 to 1. The list will be entered into the TensorFlow model so that the system will receive input and classify images according to the label. Finally, the classification results will be received by the user. Fig. 7 shows the results of implementing pose classification.



Pose yang terdeteksi:
(goddess)

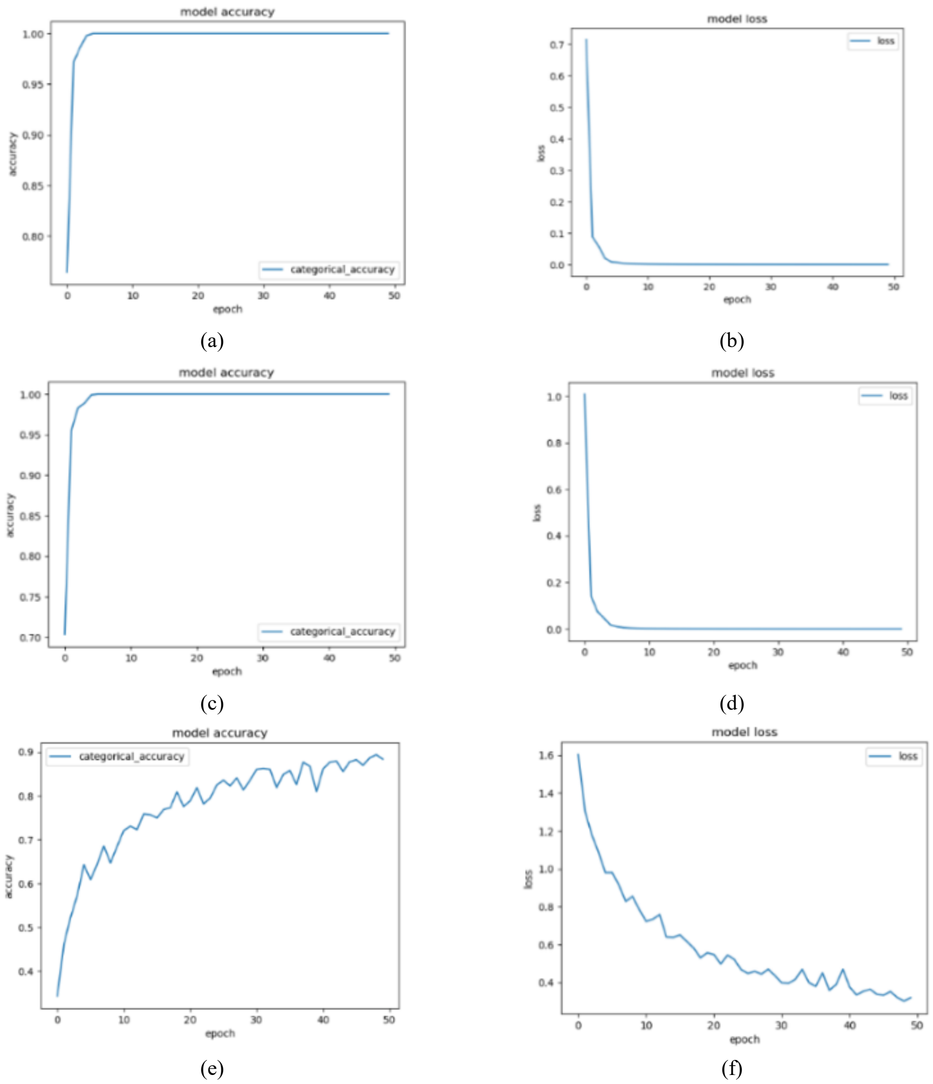
Fig. 7. Implementation of pose classification in android device

3. RESULTS AND DISCUSSION

The training process with the MobileNetV2 architecture runs for 50 epochs. The computer specifications used for the model training process can be seen in Table 3. The model managed to get the lowest loss in the 50th iteration with a training accuracy of 99.80%. The training process runs for an average of 19 seconds in each epoch. The training process with the Xception architecture runs for 50 epochs. The model managed to get the lowest loss in the 50th epoch with a model training accuracy of 98.80%. The model training process runs for an average of 3 minutes in each iteration. The training process with the ResNet50 architecture runs for 50 epochs. The model managed to get the lowest loss in the 50th iteration with a model training accuracy of 46.68%. The model training process takes an average of 2.5 minutes in each epoch. Graphs of accuracy and loss of the MobileNetV2, Inception, and ResNet50 architectural models can be seen in Fig. 8.

Table 2. The computer specifications used for the model training process

Parameter	Specification
CPU	Intel(R) Xeon(R) CPU @ 2.30GHz
RAM	13.0 GB
Space of Disk	107 GB
GPU Model Name	NVIDIA Tesla K80 (12 GB VRAM)



(a) (b)

(c) (d)

(e) (f)

Fig. 8. Accuracy and loss graph of MobileNetV2 (a)(b), Xception (c)(d), and ResNet50 (e)(f)

After model training, the evaluation results of the three pre-trained models were combined for comparison. Based on the results of the three models, MobileNetV2 and Xception have the best scores with an average accuracy score of 94%, an average precision of 94%, an average of 94% recall, and an average F1 score of 94%. Fig. 9 shows the confusion matrix table for each model. Table 4 shows the detailed evaluation results of the three CNN transfer learning architectures.

In general, MobileNetV2 and Xception outperform ResNet50 in pose classification using skeleton data for several reasons: Both MobileNetV2 and Xception use depthwise separable convolutions, which can be more effective for learning spatial features in sequential data, such as joint positions. These convolutions can capture correlations between features more effectively, which can improve the accuracy of the model; Xception uses a more complex block structure than ResNet50, which can improve its ability to learn complex features from sequential data. This is particularly important for pose classification, where the relationships between joint positions are complex and difficult to model.

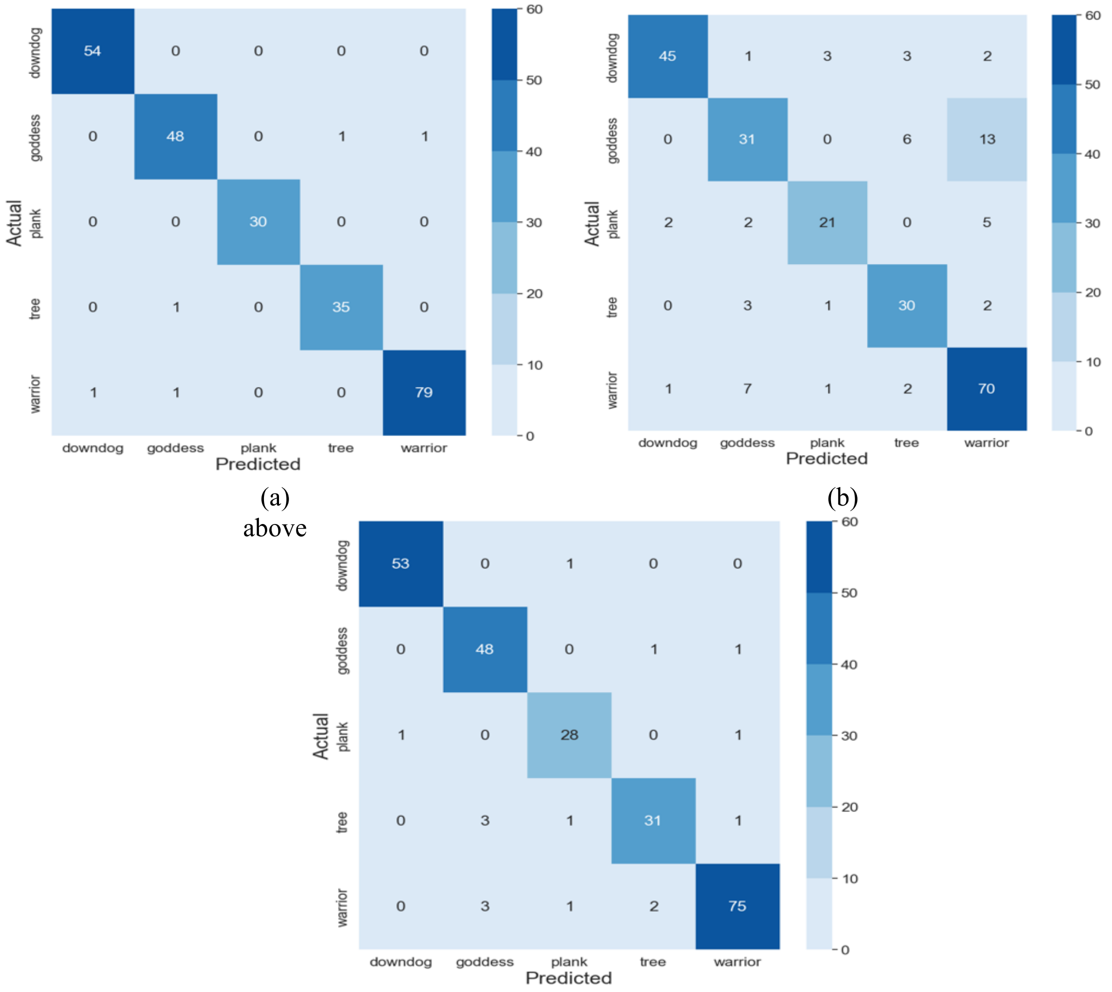
Table 4. Evaluation results of the three CNN transfer learning architectures

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score**
MobileNetV2	98	98	98	98
Xception	94	94	94	94
ResNet50	78	79	78	78

Validation is done to ensure that the system is comparable to the purpose. The validation form is a questionnaire with 10 respondents. The respondents are students who are not familiar with Pilates poses and are aged between 18-22 years old, with the aim of assessing whether the system created can attract them to learn Pilates or not. Validation is done by filling out the questionnaire after the user has used the application. Based on the results, the majority of respondents strongly agreed that the application could accurately detect pilates poses, make it easier for users to get pose information, introduce users to Pilates poses, increase interest in exercising pilates poses. Table 5 shows the results of the questionnaire conducted on the respondents. Some users also provide criticism and suggestions such as the detected pose is accurate, but the camera runs slowly.

Table 5. The results of the questionnaire conducted on the respondents

Question	Strongly disagree	Disagree	neutral	Agree	Strongly agree
"The application interface makes it easier for users to obtain information about Pilates poses and detect Pilates poses	0	0	0	1	9
"The application can accurately detect the user's Pilates pose."	0	0	0	4	6
"The application can predict the user's pose quickly."	0	0	1	7	2
"The application can introduce users to Pilates poses."	0	0	0	0	10
"The application can increase the interest of users in learning other Pilates poses."	0	0	0	1	9
"The application can increase the user's interest in exercising."	0	0	0	1	9



	downdog	goddess	plank	tree	warrior
downdog	54	0	0	0	0
goddess	0	48	0	1	1
plank	0	0	30	0	0
tree	0	1	0	35	0
warrior	1	1	0	0	79

(a)
above

	downdog	goddess	plank	tree	warrior
downdog	45	1	3	3	2
goddess	0	31	0	6	13
plank	2	2	21	0	5
tree	0	3	1	30	2
warrior	1	7	1	2	70

(b)

	downdog	goddess	plank	tree	warrior
downdog	53	0	1	0	0
goddess	0	48	0	1	1
plank	1	0	28	0	1
tree	0	3	1	31	1
warrior	0	3	1	2	75

(c)

Fig. 9. Confusion matrix of MobileNektV2 (a), Xception (b), dan ResNet50 (c)

4. CONCLUSION

This study developed a real-time Pilates pose classifier using MediaPipe as a feature extractor and CNN with transfer learning. The transfer learning models used in this study were MobileNetV2, Xception, and ResNet50. The study trained the system using five types of Pilates poses, which were Warrior, Tree, Plank, Goddess, and Downward Dog. The feature extraction process using MediaPipe framework has been running smoothly and can extract the keypoints of the skeleton of each body part. Classification results using the transfer learning architecture MobileNetV2 and Xception yielded the best values in the form of an accuracy score of 94%, a recall of 94%, a precision of 94%, and an f1 score of 94%. Models can be converted for use in the Flutter app and can recognize pilates poses precisely and in real time. The application runs smoothly, is easy to use, and is useful in increasing the interest of users in doing Pilates. This has been tested on 10 respondents who were not familiar with Pilates, and they were attracted to it. Suggestions obtained during the CNN implementation process for pilates poses classification are adding pose dataset so that the model accuracy is higher. In addition, using other pre-trained CNN models might increase the accuracy of Pilates pose classification or speeding up the frame capture time from mobile device camera. The survey conducted on the respondents is still limited. Therefore, in the future, it can be tested on respondents with different backgrounds to avoid potential biases in the survey.

REFERENCES

[1] D. S. Kehler and O. Theou, “The impact of physical activity and sedentary behaviors on frailty levels,” *Mech. Ageing Dev.*, vol. 180, pp. 29–41, 2019, https://doi.org/10.1016/j.mad.2019.03.004.

[2] A. Batrakoulis, “Psychophysiological Adaptations to Pilates Training in Overweight and Obese Individuals: A Topical Review,” *Diseases*, vol. 10, no. 4, p. 71, 2022, https://doi.org/10.3390/diseases10040071.

[3] J. Suto, S. Oniga, C. Lung, and I. Orha, “Comparison of offline and real-time human activity recognition results using machine learning techniques,” *Neural Comput. Appl.*, vol. 32, no. 20, pp. 15673–15686, 2020, https://doi.org/10.1007/s00521-018-3437-x.

[4] I. A. Putra, D. Nurhayati, and D. Eridani, “Human Action Recognition (HAR) Classification Using MediaPipe and Long Short-Term Memory (LSTM),” *Teknik*, vol. 43, no. 2, pp. 190–201, 2022, https://doi.org/10.14710/teknik.v43i2.46439.

[5] A. Latreche, R. Kelaiaia, A. Chemori, and A. Kerboua, “Reliability and validity analysis of MediaPipe-based measurement system for some human rehabilitation motions,” *Measurement*, vol. 214, p. 112826, 2023, https://doi.org/10.1016/j.measurement.2023.112826.

[6] B. Sundar and T. Bagyammal, “American Sign Language Recognition for Alphabets Using MediaPipe and LSTM,” *Procedia Comput. Sci.*, vol. 215, pp. 642–651, 2022, https://doi.org/10.1016/j.procs.2022.12.066.

[7] J. Bora, S. Dehingia, A. Boruah, A. A. Chetia, and D. Gogoi, “Real-time Assamese Sign Language Recognition using MediaPipe and Deep Learning,” *Procedia Comput. Sci.*, vol. 218, pp. 1384–1393, 2023, https://doi.org/10.1016/j.procs.2023.01.117.

[8] Y. Pang and Y. Wang, “Water Spatial Distribution in Polymer Electrolyte Membrane Fuel Cell: Convolutional Neural Network Analysis of Neutron Radiography,” *Energy AI*, p. 100265, 2023, https://doi.org/10.1016/j.egyai.2023.100265.

[9] N. M. Notarangelo, K. Hirano, R. Albano, and A. Sole, “Transfer learning with convolutional neural networks for rainfall detection in single images,” *Water (Switzerland)*, vol. 13, no. 5, pp. 1–17, 2021, https://doi.org/10.3390/w13050588.

[10] S. Malek and S. Rossi, “Head pose estimation using facial-landmarks classification for children rehabilitation games,” *Pattern Recognit. Lett.*, vol. 152, pp. 406–412, 2021, https://doi.org/10.1016/j.patrec.2021.11.002.

[11] A. Lamas *et al.*, “Human pose estimation for mitigating false negatives in weapon detection in video-surveillance,” *Neurocomputing*, vol. 489, pp. 488–503, 2022, https://doi.org/10.1016/j.neucom.2021.12.059.

[12] N. K. Kishore, D. M., Bindu, S., & Manjunath, “Estimation of yoga postures using machine learning techniques.,” *Int. J. Yoga*, vol. 15, no. 2, pp. 137–144, 2022, https://doi.org/10.4103/ijoy.ijoy_137_22.

[13] H. T. Chen, Y. Z. He, and C. C. Hsu, “Computer-assisted yoga training system,” *Multimed. Tools Appl.*, vol. 77, no. 18, pp. 23969–23991, 2018, https://doi.org/10.1007/s11042-018-5721-2.

[14] M. Pismenskova, O. Balabaeva, V. Voronin, and V. Fedosov, “Classification of a two-dimensional pose using a human skeleton,” *MATEC Web Conf.*, vol. 132, p. 05016, 2017, https://doi.org/10.1051/matecconf/201713205016.

[15] G. Kale, V. Patil, and M. Munot, “A novel and intelligent vision-based tutor for Yogāsana: e-YogaGuru,” *Mach. Vis. Appl.*, vol. 32, no. 1, pp. 1–17, 2021, https://doi.org/10.1007/s00138-020-01141-x.

[16] Y. H. Byeon, J. Y. Lee, D. H. Kim, and K. C. Kwak, “Posture Recognition Using Ensemble Deep Models under Various Home Environments,” *Appl. Sci. 2020,* vol. 10, no. 4, p. 1287, 2020, https://doi.org/10.3390/app10041287.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, https://doi.org/10.1109/CVPR.2018.00474.

[18] L. Zhang, J. Wang, B. Li, Y. Liu, H. Zhang, and Q. Duan, “A MobileNetV2-SENet-based method for identifying fish school feeding behavior,” *Aquac. Eng.*, vol. 99, p. 102288, 2022, https://doi.org/10.1016/j.aquaeng.2022.102288.

[19] A. Michele, V. Colin, and D. D. Santika, “Mobilenet convolutional neural networks and support vector machines for palmprint recognition,” *Procedia Comput. Sci.*, vol. 157, pp. 110–117, 2019, https://doi.org/10.1016/j.procs.2019.08.147.

[20] D. Sutaji and O. Yıldız, “LEMOXINET: Lite ensemble MobileNetV2 and Xception models to predict plant disease,” *Ecol. Inform.*, vol. 70, p. 101698, 2022, https://doi.org/10.1016/j.ecoinf.2022.101698.

[21] J. Zhang, J. Jing, P. Lu, and S. Song, “Improved MobileNetV2-SSDLite for automatic fabric defect detection system based on cloud-edge computing,” *Measurement*, vol. 201, p. 111665, 2022, https://doi.org/10.1016/j.measurement.2022.111665.

[22] R. Saini, P. Semwal, and T. H. Jaware, “Brain Tumor Classification Using VGG-16 and MobileNetV2 Deep Learning Techniques on Magnetic Resonance Images (MRI),” *In Soft Computing and Its Engineering Applications: 4th International Conference, icSoftComp 2022*, pp. 300–313, 2023, https://doi.org/10.1007/978-3-031-27609-5_24..

[23] M. Rahimzadeh and A. Attar, “A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2,” *Informatics Med. Unlocked*, vol. 19, p. 100360, 2020, https://doi.org/10.1016/j.imu.2020.100360.

[24] R. K. *Et al.*, “A Comparative Analysis of Variant Deep Learning Models for COVID-19 Protective Face Mask Detection,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 6, pp. 2841–2848, 2021, https://doi.org/10.17762/turcomat.v12i6.5791

[25] M. B. Hossain, S. M. H. S. Iqbal, M. M. Islam, M. N. Akhtar, and I. H. Sarker, “Transfer learning with fine-tuned deep CNN ResNet50 model for classifying COVID-19 from chest X-ray images,” *Informatics Med. Unlocked*, vol. 30, p. 100916, 2022, https://doi.org/10.1016/j.imu.2022.100916.

[26] M. R. Ibrahim, J. Haworth, and T. Cheng, “Weathernet: Recognising weather and visual conditions from street-level images using deep residual learning,” *ISPRS Int. J. Geo-Information*, vol. 8, no. 12, 2019, https://doi.org/10.3390/ijgi8120549.

[27] M. Chhabra and R. Kumar, “A Smart Healthcare System based on Concatenation of ResNet50V2 and Xception Model for Detecting Pneumonia from Medical Images,” *2022 Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. COM-IT-CON 2022*, pp. 161–167, 2022, https://doi.org/10.1109/COM-IT-CON54601.2022.9850612.

[28] A. Septiarini, H. Hamdani, E. Junirianto, M. S. S. Thayf, G. Triyono, and Henderi, “Oil Palm Leaf Disease Detection on Natural Background Using Convolutional Neural Networks,” *Proceeding - IEEE Int. Conf. Commun. Networks Satell. COMNETSAT 2022*, pp. 388–392, 2022, https://doi.org/10.1109/COMNETSAT56033.2022.9994555.

[29] F. Chollet, “XCeption: Deep Learning with Depthwise Separable Convolutions,” *Comput. Vis. Found.*, pp. 1251-1258, 2016, https://doi.org/10.1109/CVPR.2017.195.

[30] K. Shaheed *et al.*, “DS-CNN: A pre-trained Xception model based on depth-wise separable convolutional neural network for finger vein recognition,” *Expert Syst. Appl.*, vol. 191, p. 116288, 2022, https://doi.org/10.1016/j.eswa.2021.116288.

[31] S. Sharma and S. Kumar, “The Xception model: A potential feature extractor in breast cancer histology images classification,” *ICT Express*, vol. 8, no. 1, pp. 101–108, 2022, https://doi.org/10.1016/j.icte.2021.11.010.

[32] S. Ganguly, A. Ganguly, S. Mohiuddin, S. Malakar, and R. Sarkar, “ViXNet: Vision Transformer with Xception Network for deepfakes based video and image forgery detection,” *Expert Syst. Appl.*, vol. 210, p. 118423, 2022, https://doi.org/10.1016/j.eswa.2022.118423.

[33] A. Panthakkan, S. M. Anzar, S. Jamal, and W. Mansoor, “Concatenated Xception-ResNet50 — A novel hybrid approach for accurate skin cancer prediction,” *Comput. Biol. Med.*, vol. 150, p. 106170, 2022, https://doi.org/10.1016/j.compbiomed.2022.106170.

[34] M. Liao, Y. Q. Zhao, X. H. Wang, and P. S. Dai, “Retinal vessel enhancement based on multi-scale top-hat transformation and histogram fitting stretching,” *Opt. Laser Technol.*, vol. 58, pp. 56–62, 2014, https://doi.org/10.1016/j.optlastec.2013.10.018.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” *In 2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2010, https://doi.org/10.1109/CVPR.2009.5206848.

[36] M. Pintor *et al.*, “ImageNet-Patch: A dataset for benchmarking machine learning robustness against adversarial patches,” *Pattern Recognit.*, vol. 134, p. 109064, 2023, https://doi.org/10.1016/j.patcog.2022.109064.

[37] X. Li, M. Cen, J. Xu, H. Zhang, and X. S. Xu, “Improving feature extraction from histopathological images through a fine-tuning ImageNet model,” *J. Pathol. Inform.*, vol. 13, p. 100115, 2022, https://doi.org/10.1016/j.jpi.2022.100115.

[38] R. Fabricius and O. Šuch, “Detection of vowel segments in noise with ImageNet neural network architectures,” *Transp. Res. Procedia*, vol. 55, pp. 1289–1295, 2021, https://doi.org/10.1016/j.trpro.2021.07.112.

[39] M. A. Morid, A. Borjali, and G. Del Fiol, “A scoping review of transfer learning research on medical image analysis using ImageNet,” *Comput. Biol. Med.*, vol. 128, p. 104115, 2021, https://doi.org/10.1016/j.compbiomed.2020.104115.

[40] E. A. Smirnov, D. M. Timoshenko, and S. N. Andrianov, “Comparison of Regularization Methods for ImageNet Classification with Deep Convolutional Neural Networks,” *AASRI Procedia*, vol. 6, pp. 89–94, 2014, https://doi.org/10.1016/j.aasri.2014.05.013.

BIOGRAPHY OF AUTHORS

Kenneth Angelo Tanjaya, is currently pursuing his studies in the Informatic Engineering program at the University of Surabaya, where he enrolled in 2019. As a dedicated researcher, Kenneth is passionate about exploring the latest advancements in computer vision, deep learning, and image classification. He has been involved in several research projects related to these topics, demonstrating his ability to apply theoretical knowledge to practical problems. He is also an active member of his university's computer science community, participating in various extracurricular activities and events.

Mohammad Farid Naufal received his Bachelor's degree in Informatic Engineering from Institut Teknologi Sepuluh Nopember in 2010, where he gained a strong foundation in computer science and information technology. He continued his studies at the same university and obtained a Master's degree in Informatic Engineering in 2016. He is currently a full-time lecturer and researcher at Universitas Surabaya, where he continues to inspire and educate the next generation of informatics professionals. His research interests include artificial intelligence, machine learning, and computer vision. Email: faridnaufal@staff.ubaya.ac.id

Heru Arwoko, earned his Bachelor of Science in Physics in 1990 and a Master of Electrical Engineering degree in 2010 from Institut Teknologi Sepuluh Nopember. He is currently serving as a full-time lecturer in the Department of Informatics Engineering at the Universitas Surabaya. Heru has made significant contributions to the field of computer science. His research interests include computer vision, image classification, and physically based modeling. Email: heru_a@staff.ubaya.ac.id