Paper-Human Action Behavior Recognition in Still Images with Proposed Frames Selection Using… Human Action Behavior Recognition in Still Images with Proposed Frames Selection Using…

Article *in* International Journal of Online and Biomedical Engineering (iJOE) · May 2023

DOI: 10.3991/ijoe.v19i06.38463

CITATION
1

READS
150

2 authors:

Mohammed Thamer
University of Technology, Iraq
1 PUBLICATION **1** CITATION
SEE PROFILE

Ayad Rodhan Abbas
University of Technology- Iraq
49 PUBLICATIONS **235** CITATIONS
SEE PROFILE

Human Action Behavior Recognition in Still Images with Proposed Frames Selection Using Transfer Learning

https://doi.org/10.3991/ijoe.v19i06.38463

Mohammed T. Abdulhadi(✉), Ayad R. Abbas
University of Technology, Baghdad, Iraq
mohammed.t.abdulhad@uotechnology.edu.iq

Abstract—One of the most difficult challenges is recognizing human actions., especially in still images where there isn't much movement. Therefore, Using the transfer learning strategy, we suggested a technique for identifying human action., which consists of training some of the layers of deep learning techniques while freezing others. Also presented a way for data split, which is to choose some frames because we are working on a large dataset such as UCF-101, and this method is summarized by discovering the features for each frame, then clustering the elements, and then choosing a percentage of each cluster for training and test data. We used three techniques. They are VGG16, Inception V3, and xception. The proposed models have been implemented on UCF-101 Dataset. Depending on three data split methods with the dataset, the random split method, and the proposed split method, the Inception V3 achieved the highest accuracy. In contrast, the VGG16 achieved the least accuracy, and the accuracy of the xception was close to that of the Inception V3. By comparing the size of the dataset, the proposed methods achieved good results: the VGG16 in the proposed split attained an accuracy of 92.5%, the Inception V3 in the proposed split attained an accuracy of 98.12%, and the xception in the proposed split attained an accuracy of 95.16%. The VGG16 network is simple, so the VGG16 is less accurate. While the network in Inception V3, xception, is more extensive and complex, the learning space is more significant, although the network size is more prominent in Inception V3, xception. We only trained some blocks in the top layer.

Keywords—Human action, transfer learning, deep learning, CNN, VGG16, Inception V3, xception, k-mean, keyframe

1 Introduction

One of the most significant issues with computer vision has always been recognizing human activities [2]. Video-based action recognition is a more established and well-researched field of study than still image-based action recognition. Due to the rise in the number of photos made public through social networks in recent years, this has attracted a lot of attention. Action detection in still images is still a complex topic

since motion cannot be readily approximated from a still image, and spatiotemporal information cannot be used to characterize the action. Recognizing actions in static photos is doable and helpful, even though it is more evident and straightforward in videos. A current field of study in computer vision and pattern recognition is action recognition in static images. Analyzing human behavior in the vast amount of photos on the Internet is a highly fascinating endeavor. The main characteristics of this issue are the lack of motion in still images and the inapplicability of existing action recognition in non-videos to the prediction of action in still images because videos contain spatiotemporal cues. Spatiotemporal characteristics are frequently used in videos to identify steps. However, static photographs have no motion cues and a lot of clutter. Image Annotation, Action/Behavior-Based Image Retrieval, Frame Reduction in Videos, and Human-Computer Interaction are applications of action recognition in still images [1], [3], [4].

A machine learning technique called transfer learning focuses on taking data from a domain comparable to the target domain to enhance learning capabilities or decrease the number of labeled samples needed in the target domain [6]. Fixed feature extraction, fine-tuning and layer freezing, and pre-trained models are the three basic CNN transfer learning scenarios [7]. The improved feature extraction scenario eliminates a pre-trained, fully connected final layer from the CNN model. In contrast, the input and feature extraction layers keep their weights and structures and are selected feature extractors. The pre-trained model is maintained in the fine-tuning and layer-freezing scenario by adjusting the importance of the pre-trained network. The CNN network's higher layers may or may not be the only levels that go through the fine-tuning procedure [6]. The transfer learning approach is a set of strategies that include transferring the knowledge that CNN models have learned [8]

These are our contributions to this paper, in summary:

- A new algorithm is suggested to divide the data using an intelligent mechanism that extracts the features from each frame, clusters the frames, and then chooses the keyframe from clustering.
- Three Deep Learning algorithms—Inception V3, xception, and VGG16—are applied for fine-tuning transfer learning.
- The level of human action recognition on a still image is addressed using a large-scale dataset, such as the UCF-101.

2 Related work

The recognition of action in still images has recently been a research subject. All of the studies related to the following use a still image. [1] suggests that the classifier is a support vector machine (SVM). A deep neural network, like a residual neural network, is used to extract features. Benchmark datasets like the Pascal VOC action and Stanford 40 datasets are used to test the proposed model. [20] In a unique way to recognize human actions, the pre-trained Convolutional Neural Network (CNN) model

is used as a feature extractor. Deep representations are then used by the Support Vector Machine (SVM) classifier to figure out what actions are happening. Knowledge from a large data set can help CNN recognize activities, even with just a tiny amount of training data. Data from Stanford 40 that is available to the public is used to look at the proposed strategy. [21] They use pre-trained CNNs and the transfer learning method to get around the lack of large labeled action recognition datasets. CNN's last layer also has class-specific data, so they use an attention method on its output feature maps to pull out more powerful and distinguishable features for putting people's behaviors into groups. Using the Stanford 40 dataset, they also test how well the eight CNNs that have already been trained on their framework work. [22] They suggest a multi-task learning approach to deal with the problem of irrelevant and misleading backgrounds when recognizing actions in still images. They want to focus the network's activations on people to suppress the activations of deceptive objects or backgrounds. They use a new human-mask loss to automatically direct the activations of the feature maps to the target human. They offer a deep learning method that can predict the human activity class and the heatmap of where people are simultaneously. The technique yields cutting-edge outcomes, scoring 94.06% on the Stanford40 dataset and 40.65% on the MPII dataset. [23] They try to figure out how to tell what a person is doing from a still image and use deep ensemble learning to automatically break down the body position and figure out what it means. First, the nonsequential convolutional neural network (NCNN) module is added to the top of the already trained model. This makes an NCNN-based model that goes from beginning to end. The NCNN is not set up in a particular order so that it can learn both spatial and channel-wise features at the same time. [24] They solve the problem of how to avoid being caught using still images from the web and social media. They use various image classification networks, such as VGG16, ResNet50, ResNetXt-50, and ViT.

3 Methodology

3.1 UCF101 dataset

The UCF101 dataset, an extension of UCF50, consists of 13,320 video clips separated into 101 categories. These 101 categories can be attributed to 5 different types (Body motion, Human-human interactions, Human-object interactions, Playing musical instruments, and Sports). The total running time of these videos is around 27 hours. The videos all have a steady frame rate of 25 FPS and 320 x 240 quality, and they were all downloaded from YouTube [9] [10].

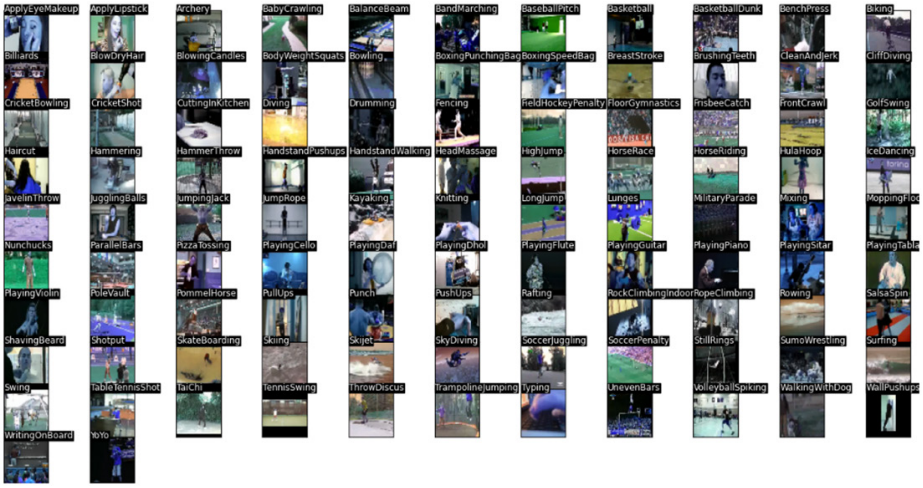

Fig. 1. UCF-101 classes

3.2 Transfer learning

Transfer learning is a technique for transmitting information from a pre-trained version to the target version using the pre-educated version's learned weights as the target version. The two models must behave similarly for this technique to produce relevant findings [15]. Large datasets, like ImageNet [11], train the version to generate these weights. The target version uses the previously acquired skills from the educated version. Less educational statistics are needed with transfer learning, and the result is a version that requires the least amount of time in school while still making remarkable strides. The approach the CNN version initially taught, utilizing significant statistical data, is how transfer learning mainly works. It is then much improved for training on a limited dataset during the second phase [12]. Many common transfer learning tendencies exist, including VGG, ResNet, Inception, and others. The pre-trained CNN version Inception V3 [13] [14].

3.3 The proposed approach

First, we convert the videos into a set of frames because our work is limited to still images, and then we change the size of the frames according to the CNN architecture (if we use VGG16, we change the size of the frames to 224 * 224 and if we use Inception V3 or xception we change the size of the frames to 299 * 299). The dataset that we used has a three-split, to which we added a random split by random selection (80% training, 20% test). We also proposed a methodology that selects keyframes. After the data was divided into two parts, depending on which of the split methods we used, we applied augmentation to the training part to increase the diversity of training part. The data augmentation used includes shear range by 0.2, horizontal flip, rotation range by 10, width shift range by 0.2 and height shift range by 0.2. Figure 2 shows the proposed approach.

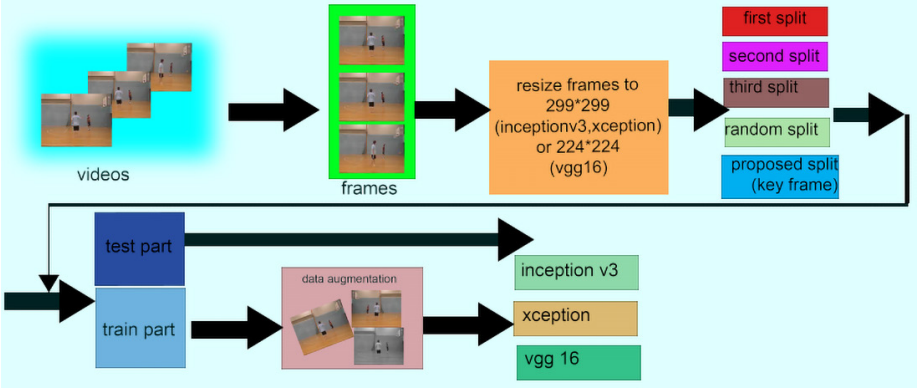

Fig. 2. The proposed approach

3.4 Proposed data split

The dataset we used has three different splits [9]. In addition, data augmentation was also used [16] [17] [18] [19], along with a random split that chose 80% for training and 20% for testing. We know that the video contains many frames, and some of these may be similar in their features, so we suggested a way to choose the keyframes. This method relies on reading the videos of each class and converting the videos into a total of the frame, and from this, extracting the features from this frame using Inception V3 by deleting the last softmax layer and then making a cluster for these features to divide the frame into two groups and take 64% of the training data from The cluster1, cluster2 and 16% of the test data were taken from the first cluster, second cluster (the training data did not intersect with the test data). The purpose of this method is to take various features and ignore 20% of the frames that are similar in characteristics. Figure 3 shows a flowchart of the proposed Data split.

Algorithm Pseudo-Code of proposed Data split (Keyframes)
Input: Ucf-101 Dataset, K=2 **Output:** training data, testing data **Foreach** class in the dataset Read all videos of class and Convert videos to frames **Foreach** frame in class Set model equal to inceptionv3 using ImageNet weights and deleting the last softmax layer Features(frame) Extraction using model **End Loop** Set the initial cluster center randomly Perform k-mean clustering on the frames of the class depending on K Set randomly 64% for training data from cluster1, cluster2 Set randomly16% for testing data from cluster1, cluster2 **End Loop**

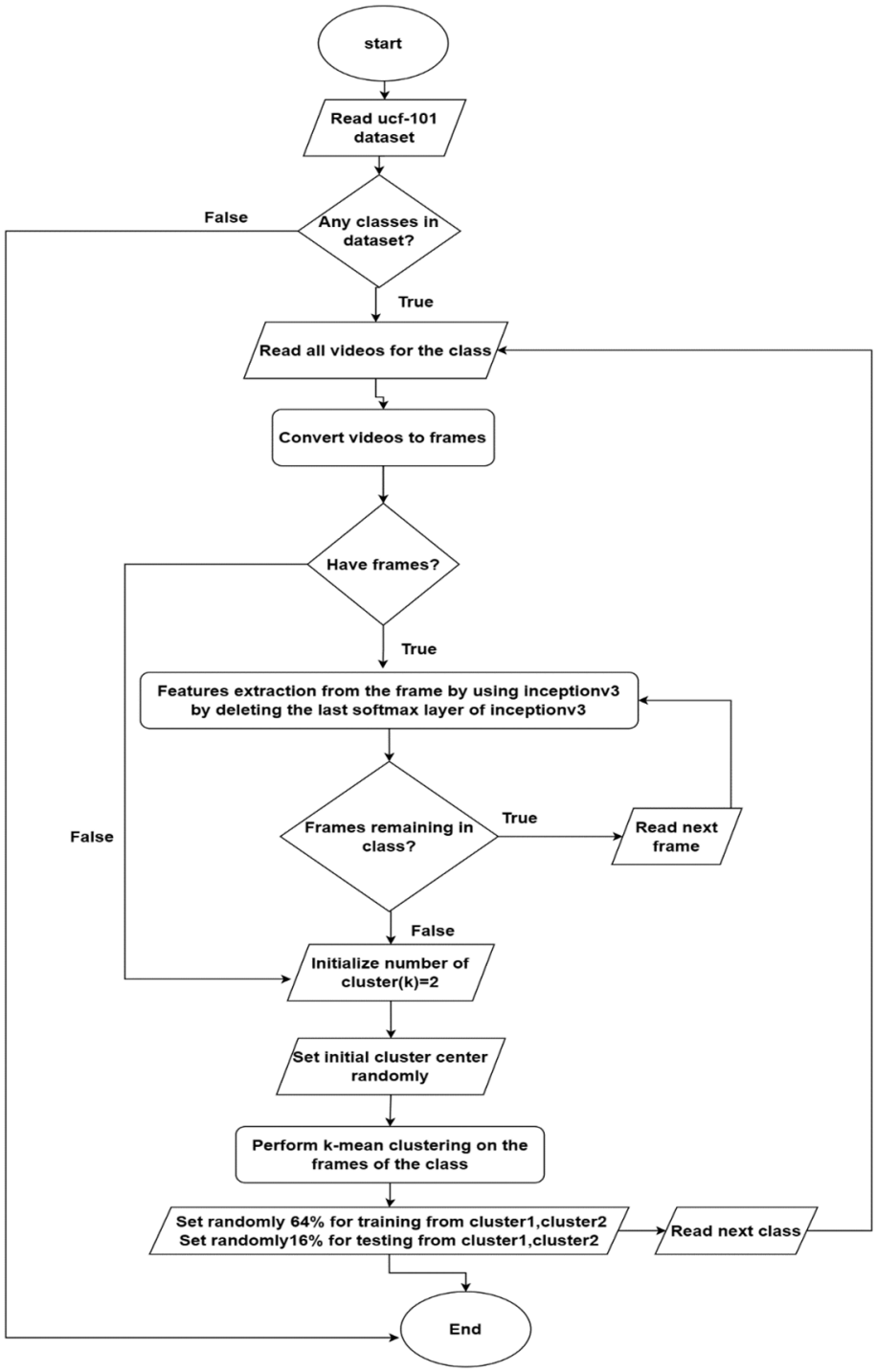

Fig. 3. Flowchart of proposed data split

3.5 VGG16 model

In the first model, applying VGG16 [27] layers with ImageNet weights, because VGG16 is a lightweight and straightforward model, we adopted it in our work. we use a dataset with 101 classes to test this model's accuracy. In this work, the shape is (224,224,3). In addition, A 1024-unit layer that is fully connected and has the Relu activation function and an output layer of size 101 to the number of dataset classes. This last layer uses the softmax activation function. First, freeze all VGG16 layers by making them non-trainable. While the other layers are frozen, train the new layers for a few epochs. This will allow the new FC layers to begin initializing with real "learned" values rather than purely random ones. Then compile the model using Adam Optimizer, and then do a training with ten epochs and choose one of the five splits for training and testing data. Then unfreeze the last three blocks (from conv3 to conv5) and make them trainable. For these changes to take effect, we must recompile the model with a low learning rate (0.001). We employ the SGD optimization algorithm used. Finally, we train the model again with 300 epochs. Figure 4 below shows the VGG16 model with new layers.

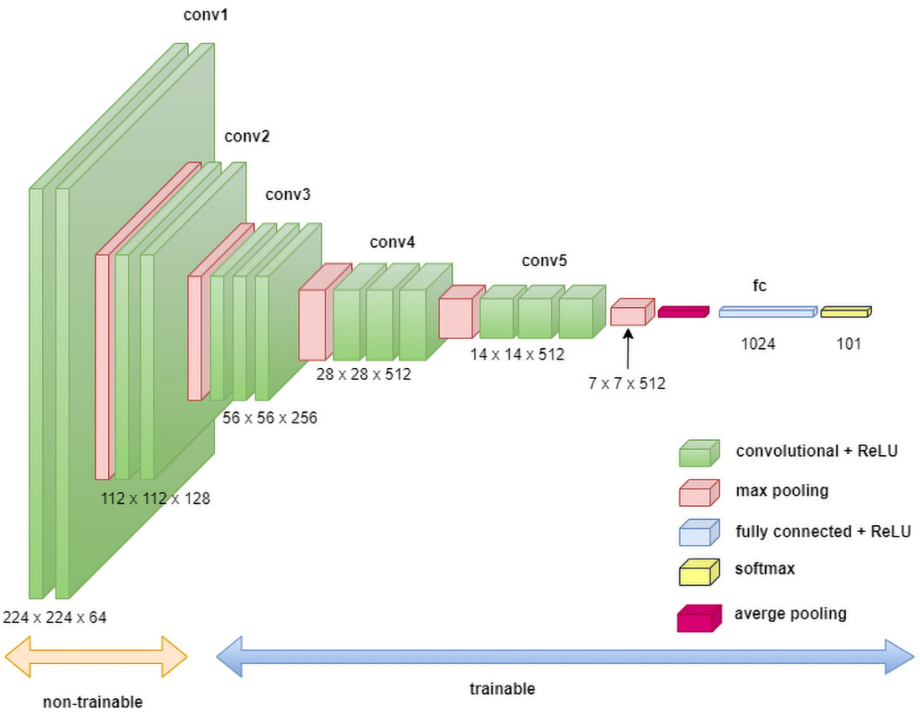

Fig. 4. VGG16 model with new layers

3.6 Inception V3 model

In this model, applying Inception V3 [25] with ImageNet weights, One of the most sophisticated computer vision models available right now is Inception V3. Szegedy et al. of Google Research created this. On the ImageNet dataset, it has been demonstrated

that the image recognition model Inception V3 can achieve higher than 78.1% accuracy. In this paper, the shape is (299,299,3). In addition, A 1024-unit layer that is fully connected and has the Relu activation function and an output layer of size 101 to the number of dataset classes. This last layer uses the softmax activation function. First, freeze all Inception V3 layers by making them non-trainable. Then compile the model using Adam Optimizer, and then do a training with ten epochs and choose one of the five splits for training and testing data. Then unfreeze the last two blocks (from block 10 to block 11) and make them trainable. For these changes to take effect, we must recompile the model with a low learning rate (0.001), then employ the SGD optimization algorithm used. Finally, we train the model again with 300 epochs. Figure 5 below shows the Inception V3 model with new layers.

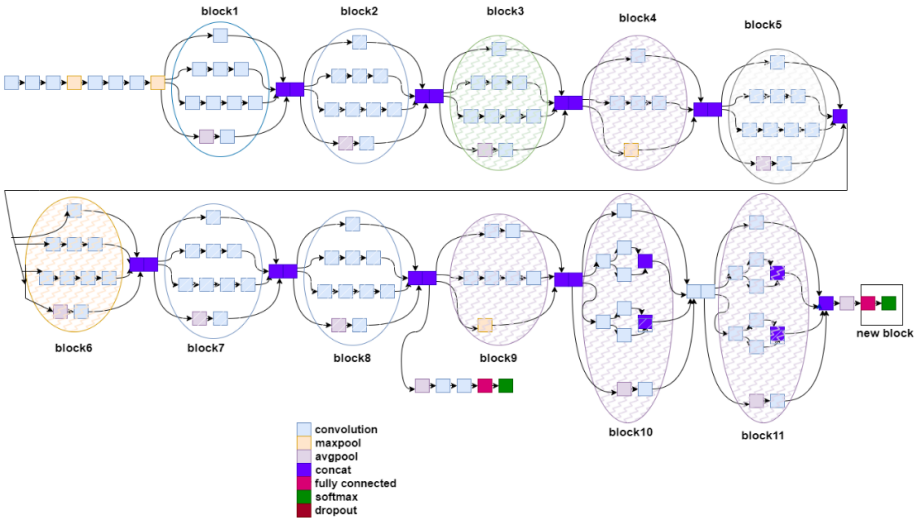

Fig. 5. Inception V3 model with new layers

3.7 Xception model

In this model, applying xception [26] with ImageNet weights, We demonstrate that this architecture, called Xception, greatly outperforms Inception V3 on a bigger image classification dataset that consists of 350 million images and 17,000 classes, while somewhat outperforming Inception V3 on the ImageNet dataset (for which Inception V3 was built). The performance improvements come from more effective use of model parameters rather than additional capacity because the Xception architecture uses the same number of parameters as Inception V3 does [26]. In this paper, the shape is (299,299,3). In addition, A 1024-unit layer that is fully connected and has the Relu activation function and an output layer of size 101 to the number of dataset classes. This last layer uses the softmax activation function. First, we freeze all xception layers by making them non-trainable. Then compile the model using Adam Optimizer, and then do a training with ten epochs and choose one of the five splits for training and testing data. Then unfreeze the last two blocks (from block 11 to block 12) and make

them trainable. For these changes to take effect, we must recompile the model with a low learning rate (0.001), and then employ the SGD optimization algorithm used. Finally, we train the model again with 300 epochs. Figure 6 below shows the xception model with new layers.

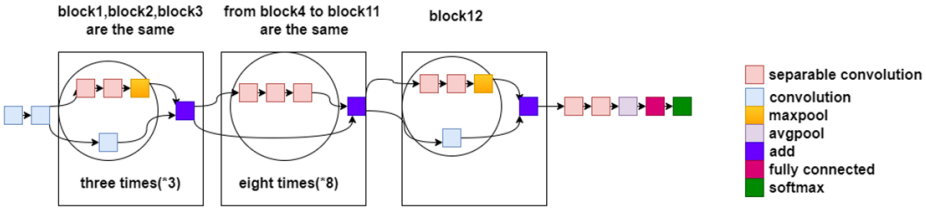

Fig. 6. Xception model with new layers

4 Results

Measurements of the training accuracy, training top k categorical accuracy, and testing top k categorical accuracy (k = 5) have been used to assess the proposed system. These are the results of the three models' evaluations. We point out that the models that were used in this paper on split1, split2, and split3 suffered from the problem of overfitting, and the results were somewhat fragile because these divisions were divided at the video level and did not take into account still images, so we suggested random split, proposed split Which did not suffer from the problem of overfitting and achieved excellent results The VGG16 model with split1 has a training accuracy value of 91.34%, a testing accuracy value of 58.125%, a training top k categorical accuracy value of 98.44%, and a testing top k categorical accuracy value of 78.28%. Figure 7 shows the accuracy of the VGG16 model with split1 and their loss and shows that the model suffers from the overfitting problem.

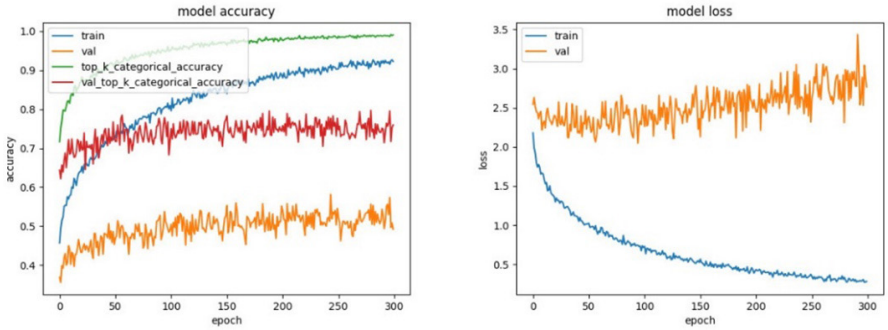

Fig. 7. Show accuracy with their loss using VGG16 (split1)

In the VGG16 model with split2, the training accuracy is 91%. However, the testing accuracy is 57.344%, and the training top k categorical accuracy is 98.69%. However, testing top k categorical accuracy 77.19%, Figure 8 shows the accuracy of the VGG16

model with split2 and their loss and shows that the model suffers from an overfitting problem.

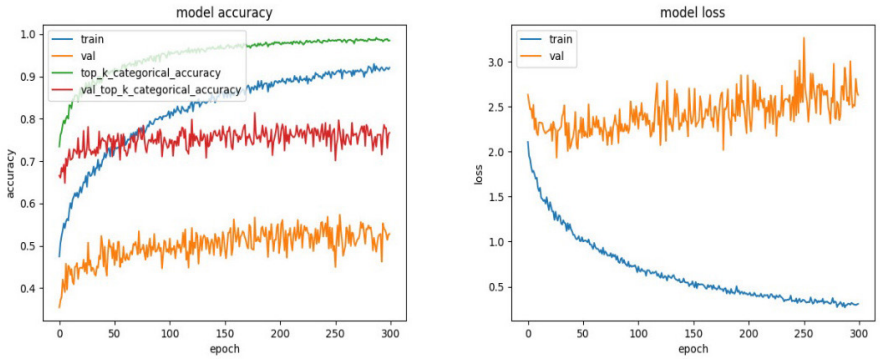

Fig. 8. Show accuracy with their loss using VGG16 (split2)

The VGG16 model with split3 has a training accuracy value of 90.34%, a testing accuracy value of 55.31%, and a training top k categorical accuracy of 98.53%. However, testing top k categorical accuracy of 79.53%. Figure 9 shows the accuracy of the VGG16 model with split3 and their loss and shows that the model suffers from an overfitting problem.

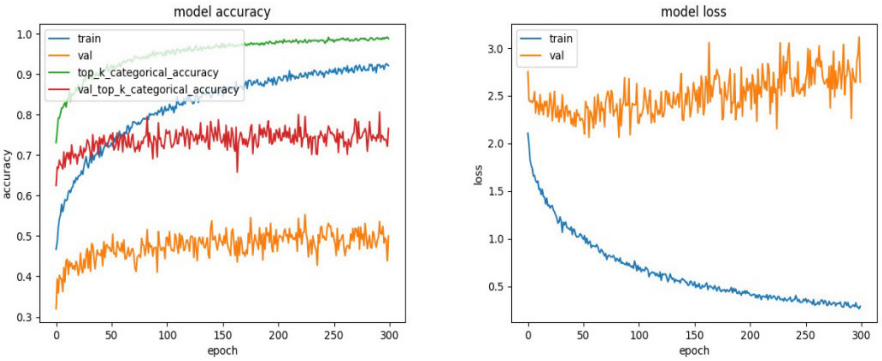

Fig. 9. Show accuracy with their loss using VGG16 (split3)

In the VGG16 model with random split, the training accuracy is 89.25%. However, testing accuracy is 87.19%, and training top k categorical accuracy is 98.25%. However, testing top k categorical accuracy is 96.72%. Figure 10 shows the accuracy of the VGG16 model with the random split and their loss.

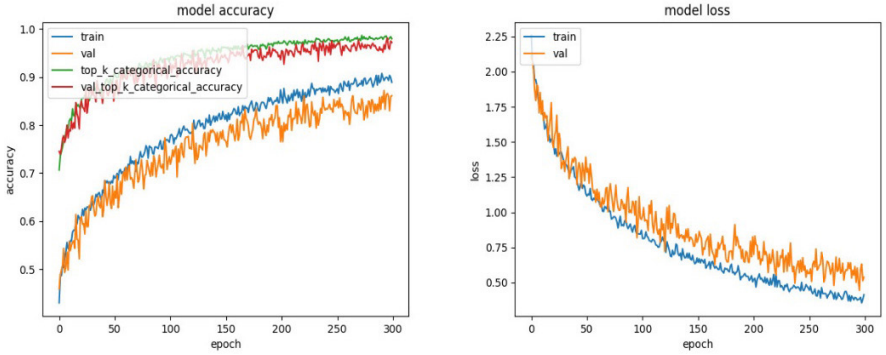

Fig. 10. Show accuracy with their loss using VGG16 (random split)

In the VGG16 model with the proposed split, the training accuracy is 87.63%. However, testing accuracy is 92.5%, and training top k categorical accuracy is 97.95%. However, testing top k categorical accuracy is 98.28%. Figure 11 shows the accuracy of the VGG16 model with the random split and their loss.

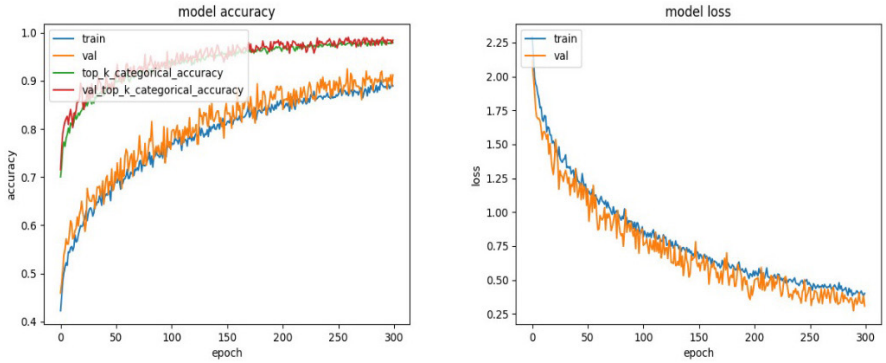

Fig. 11. Show accuracy with their loss using VGG16 (proposed split)

In the Inception V3 model with split1, the training accuracy is 96.38%. However, testing accuracy is 75.78%, and the training top k categorical accuracy is 99.72%. However, testing top k categorical accuracy is 90.78%. Figure 12 shows the accuracy of the Inception V3 model with split1 and their loss and shows that the model suffers from an overfitting problem.

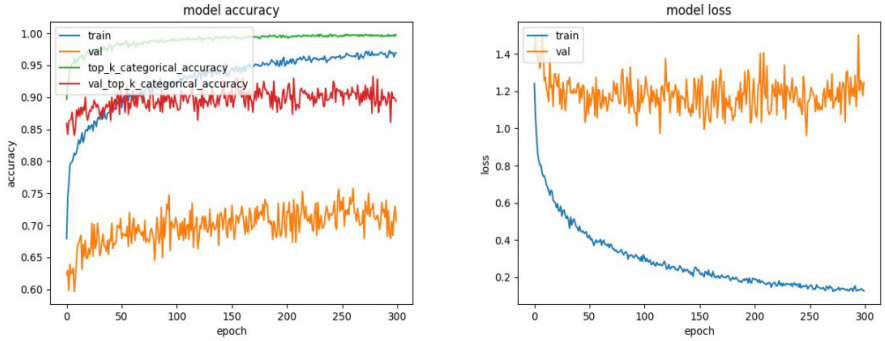

Fig. 12. Show accuracy with their loss using Inception V3 (split1)

In the Inception V3 model with split2, the training accuracy is 96.5%. However, testing accuracy is 75.81%, and training top k categorical accuracy is 99.62%. However, testing top k categorical accuracy is 88.59%. Figure 13 shows the accuracy of the Inception V3 model with split2 and their loss and shows that the model suffers from an overfitting problem.

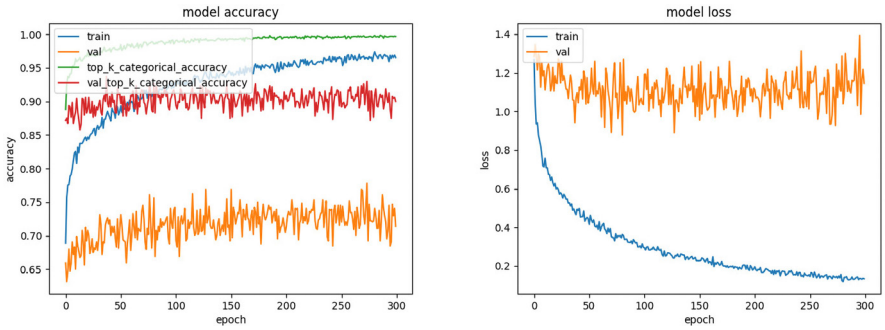

Fig. 13. Show accuracy with their loss using Inception V3 (split2)

In the Inception V3 model with split3, the training accuracy is 96.69%, however testing accuracy is 76.41%, and the training top k categorical accuracy is 99.69%. However, testing top k categorical accuracy of 90.78%, Figure 14 shows the accuracy of the Inception V3 model with split3 and their loss and shows that the model suffers from an overfitting problem.

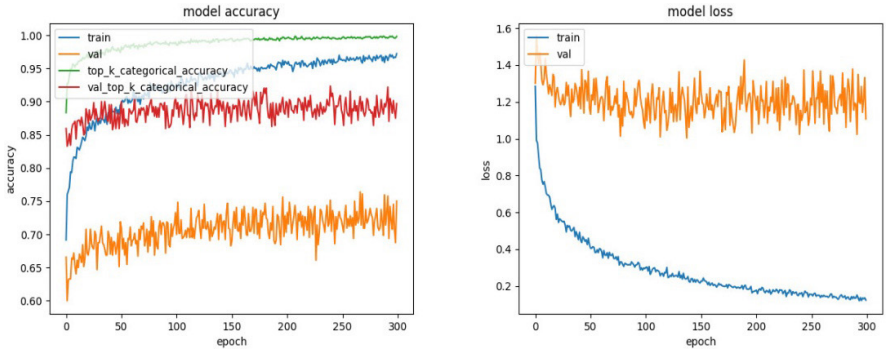

Fig. 14. Show accuracy with their loss using Inception V3 (split3)

In the Inception V3 model with random split, the training accuracy value of 96.13%, the testing accuracy value of 95.94%, and the training top k categorical accuracy is 99.66%; however, the testing top k categorical accuracy is 90.84%. Figure 15 shows the accuracy of the Inception V3 model with random split and their loss.

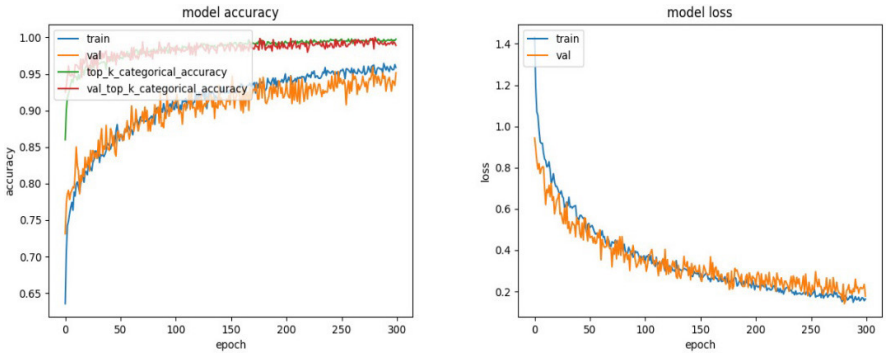

Fig. 15. Show accuracy with their loss using Inception V3 (random split)

In the Inception V3 model with the proposed split, the training accuracy is 95.72%. However, testing accuracy is 98.12%, and training top k categorical accuracy is 99.47%; however, testing top k categorical accuracy is 99.69%. Figure 16 shows the accuracy of the Inception V3 model with the proposed split and their loss.

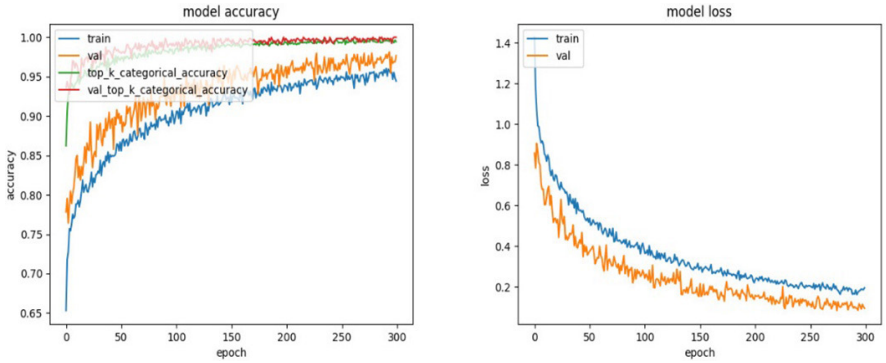

Fig. 16. Show accuracy with their loss using Inception V3 (proposed split)

In the xception model with split1, the training accuracy is 92.19%. However, testing accuracy is 73.75%, and training top k categorical accuracy is 98.75%; however, testing top k categorical accuracy is 89.38%. Figure 17 shows the accuracy of the xception model with split1 and their loss and shows that the model suffers from an overfitting problem.

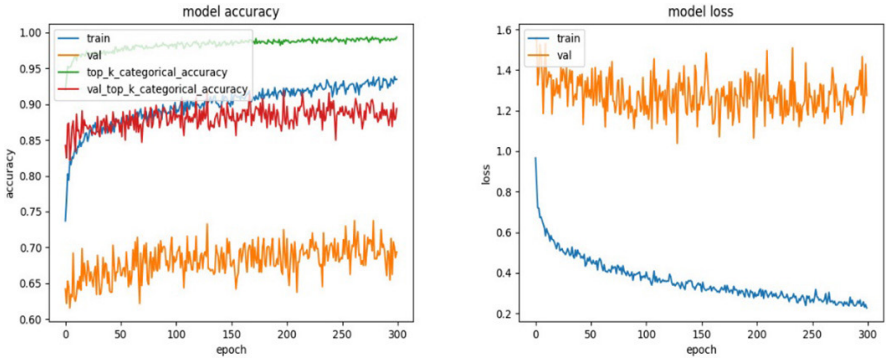

Fig. 17. Show accuracy with their loss using xception (split1)

In the xception model with split2, the training accuracy is 91.66%. However, testing accuracy is 73.12%, and training top k categorical accuracy is 98.66%; however, testing top k categorical accuracy is 89.53%. Figure 18 shows the accuracy of the xception model with split2 and their loss and shows that the model suffers from an overfitting problem.

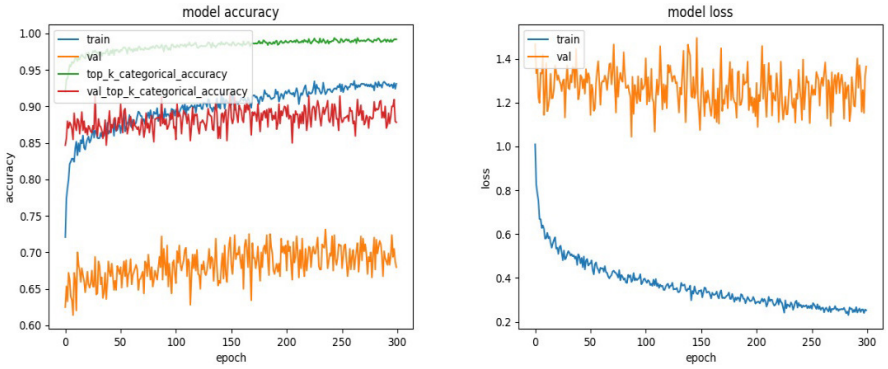

Fig. 18. Show accuracy with their loss using xception (split2)

In the xception model with split3, the training accuracy is 92.66%. However, testing accuracy is 74.84%, and training top k categorical accuracy is 99%; however, testing top k categorical accuracy is 90%. Figure 19 shows the accuracy of the xception model with split3 and their loss and shows that the model suffers from an overfitting problem.

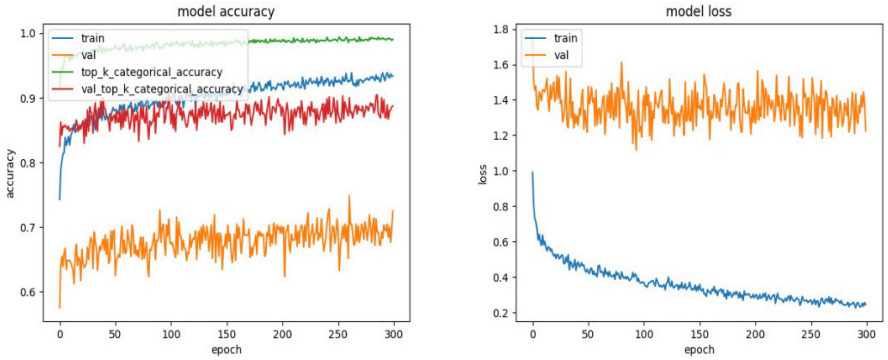

Fig. 19. Show accuracy with their loss using xception (split3)

The xception model with random split has a training accuracy value of 90.69%. However, the testing accuracy value of 92.5%, and the training top k categorical accuracy of 98.28%. However, testing top k categorical accuracy of 98.91%, Figure 20 shows the accuracy of the xception model with random split and their loss.

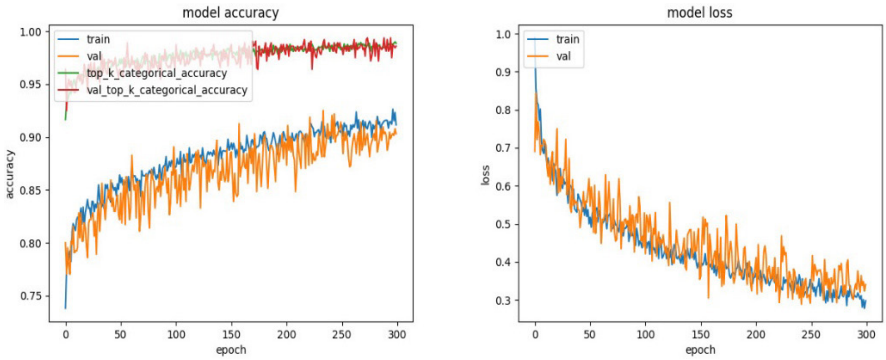

Fig. 20. Show accuracy with their loss using xception (random split)

In the xception model with the proposed split, the training accuracy is 91.12%. However, testing accuracy is 95.16%, and training top k categorical accuracy is 98.72%; however, testing top k categorical accuracy is 99.53%. Figure 21 shows the accuracy of the xception model with the proposed split and their loss.

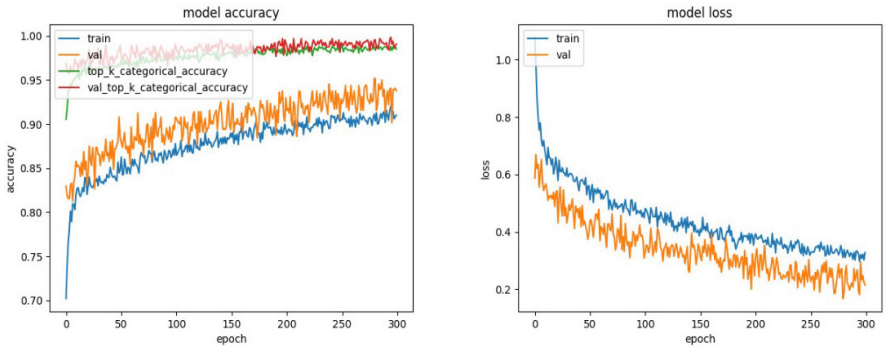

Fig. 21. Show accuracy with their loss using xception (proposed split)

The Inception V3 model achieved the highest accuracy in all data splits, while the xception model came after it, and VGG16 achieved the lowest accuracy. Figure 22 shows an accuracy comparison between different models.

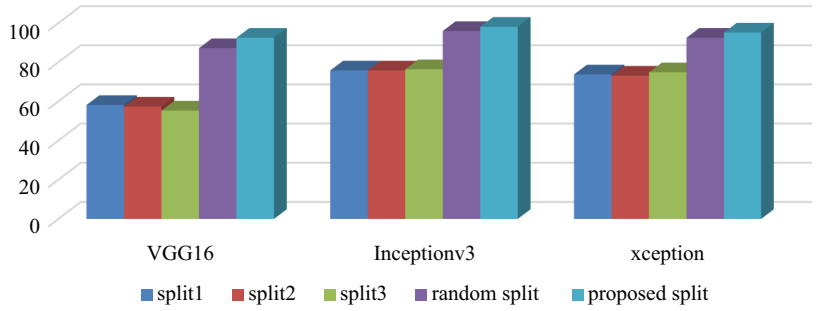

Fig. 22. Show accuracy comparison between different models

5 Conclusion

In this proposed system, we used the transfer learning approach to distinguish human action, we used ucf-101 as the dataset, which is distinguished by its large number of classes, reaches 101 classes, and this represents a great challenge to detect human action based on the still image, and because of the similarity between many In the data frame, we suggested a smart way to split data, and it achieved the highest results. We also used three deep learning techniques, VGG16, which achieved the least accuracy due to its small network, and Inception V3 achieved the highest accuracy. And xception achieved less accuracy than Inception V3. But it's pretty close. The VGG16 in the proposed split attained an accuracy of 92.5%, the Inception V3 in the proposed split attained an accuracy of 98.12%, and the xception in the proposed split attained an accuracy of 95.16%. We want to adapt the proposed split method to human action recognition at the video level by selecting the keyframe at the video level.

6 Acknowledgment

The University of Technology, Baghdad, Iraq, has consistently supported the authors' work on this project, which they would like to acknowledge.

7 References

[1] S. R. Sreela and S. M. Idicula, "Action recognition in still images using residual neural network features," *Procedia Comput. Sci.*, vol. 143, pp. 563–569, 2018. https://doi.org/10.1016/j.procs.2018.10.432

[2] M. Abduljabbar Ali, A. Jaafar Hussain, and A. T. Sadiq, "Deep learning algorithms for human fighting action recognition," *Int. J. Onl. Eng.*, vol. 18, no. 02, pp. 71–87, 2022. https://doi.org/10.3991/ijoe.v18i02.28019

[3] W. Yang, T. Lyons, H. Ni, C. Schmid, and L. Jin, "Developing the path signature methodology and its application to landmark- based human action recognition," in *Stochastic Analysis, Filtering, and Stochastic Optimization*, Cham: Springer International Publishing, 2022, pp. 431–464. https://doi.org/10.1007/978-3-030-98519-6_18

[4] F. Li, K. Shirahama, M. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors (Basel)*, vol. 18, no. 3, p. 679, 2018. https://doi.org/10.3390/s18020679

[5] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: Recent updates, datasets, challenges, and applications," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 2259–2322, 2021. https://doi.org/10.1007/s10462-020-09904-8

[6] Y. Abdulazeem, H. M. Balaha, W. M. Bahgat, and M. Badawy, "Human action recognition based on transfer learning approach," *IEEE Access*, vol. 9, pp. 82058–82069, 2021. https://doi.org/10.1109/ACCESS.2021.3086668

[7] P. Goel and A. Ganatra, "A survey on deep transfer learning for convolution neural networks," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 06, pp. 8399–8410, 2020. http://sersc.org/journals/index.php/IJAST/article/view/25284

[8] M. Aatila, M. Lachgar, H. Hrimech, and A. Kartit, "Transfer learning in keratoconus classification," *Int. J. Onl. Eng.*, vol. 18, no. 15, pp. 43–58, 2022. https://doi.org/10.3991/ijoe.v18i15.33689

[9] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012. https://doi.org/10.48550/arXiv.1212.0402

[10] Y. Peng, Y. Zhao, and J. Zhang, "Two-stream collaborative learning with spatial-temporal attention for video classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 773–786, 2019. https://doi.org/10.1109/TCSVT.2018.2808685

[11] L. Alzubaidi *et al.*, "Novel transfer learning approach for medical imaging with limited labeled data," *Cancers (Basel)*, vol. 13, no. 7, p. 1590, 2021. https://doi.org/10.3390/cancers13071590

[12] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, 2021. https://doi.org/10.1186/s40537-021-00444-8

[13] M. Madanan and B. T. Sayed, "Designing a deep learning hybrid using CNN and Inception V3 transfer learning to detect the aggression level of deep obsessive compulsive disorder in children," *Int. J. Biol. Biomed. Eng.*, vol. 16, pp. 207–220, 2022. https://doi.org/10.46300/91011.2022.16.27

[14] M. Yani, S. S. M. T. Budhi Irawan, and S. T. M. T. Casi Setiningsih, "Application of transfer learning using convolutional neural network method for early detection of Terry's nail," *J. Phys. Conf. Ser.*, vol. 1201, no. 1, p. 012052, 2019. https://doi.org/10.1088/1742-6596/1201/1/012052

[15] L. R. Ali, S. A. Jebur, M. M. Jahefer, and B. N. Shaker, "Employing transfer learning for diagnosing COVID-19 disease," *Int. J. Onl. Eng.*, vol. 18, no. 15, pp. 31–42, 2022. https://doi.org/10.3991/ijoe.v18i15.35761

[16] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, "Basics of supervised deep learning," in *Studies in Big Data*, Singapore: Springer Singapore, 2020, pp. 13–29. https://doi.org/10.1007/978-981-13-6794-6_2

[17] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, 2019. https://doi.org/10.1186/s40537-019-0197-0

[18] B. Jabir, N. Falih, and K. Rahmani, "Accuracy and efficiency comparison of object detection open-source models," *Int. J. Onl. Eng.*, vol. 17, no. 05, p. 165, 2021. https://doi.org/10.3991/ijoe.v17i05.21833

[19] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," 2022. https://doi.org/10.48550/arXiv.2204.08610

[20] A. R. Siyal *et al.*, "Still image-based human activity recognition with deep representations and residual learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 5, 2020. https://doi.org/10.14569/IJACSA.2020.0110561

[21] S. Mohammadi, S. G. Majelan, and S. B. Shokouhi, "Ensembles of deep neural networks for action recognition in still images," in *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2019. https://doi.org/10.1109/ICCKE48569.2019.8965014

[22] L. Liu, R. T. Tan, and S. You, "Loss guided activation for action recognition in still images," in *Computer Vision – ACCV 2018*, Cham: Springer International Publishing, 2019, pp. 152–167. https://doi.org/10.1007/978-3-030-20873-8_10

[23] X. Yu *et al.*, "Deep ensemble learning for human action recognition in still images," *Complexity*, vol. 2020, pp. 1–23, 2020. https://doi.org/10.1155/2020/9428612

[24] S. Akti, F. Ofli, M. Imran, and H. K. Ekenel, "Fight detection from still images in the wild," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2022. https://doi.org/10.1109/WACVW54805.2022.00061

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. https://doi.org/10.1109/CVPR.2016.308

[26] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. https://doi.org/10.1109/CVPR.2017.195

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. https://doi.org/10.48550/arXiv.1409.1556

8 Authors

Mohammed T. Abdulhadi obtained a bachelor's degree from the University of Technology, Iraq, in 2012. He works in the Department of Computer Science at the University of Technology. He is a graduate student at the University of Technology.

Ayad R. Abbas have a PH.D., in Artificial Intelligent, from Wuhan University, School of Computer Science, China, 2009, M.SC., in Computer Science, from University of Technology, Computer Science Department, Iraq, 2005, B.Sc., in Computer Science, from University of Technology, Computer Science Department, Iraq, 2003, and B.Sc., in Chemical Engineering, from University of Baghdad, in Chemical Engineering Department, Iraq, 1999.

Article submitted 2023-01-30. Resubmitted 2023-02-26. Final acceptance 2023-03-02. Final version published as submitted by the authors.