Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.028003
Article

Tech Science Press

Optimal Deep Convolutional Neural Network with Pose Estimation for Human Activity Recognition

S. Nandagopal[1,*], G. Karthy[2], A. Sheryl Oliver[3] and M. Subha[4]

[1]Department of Computer Science and Engineering, Nandha College of Technology, Erode, 638052, Tamilnadu, India
[2]Department of Electronics and Communication Engineering, Kalasalingam Academy of Research and Education, Krishnankoil, 626126, Tamilnadu, India
[3]Department of Computer Science and Engineering, St. Joseph's College of Engineering, Chennai, 600119, Tamilnadu, India
[4]Department of Electronics and Communication Engineering, University College of Engineering Nagercoil, Nagercoil, 629004, Tamilnadu, India
*Corresponding Author: S. Nandagopal. Email: asnandu1977@gmail.com
Received: 30 January 2022; Accepted: 10 March 2022

Abstract: Human Action Recognition (HAR) and pose estimation from videos have gained significant attention among research communities due to its application in several areas namely intelligent surveillance, human robot interaction, robot vision, etc. Though considerable improvements have been made in recent days, design of an effective and accurate action recognition model is yet a difficult process owing to the existence of different obstacles such as variations in camera angle, occlusion, background, movement speed, and so on. From the literature, it is observed that hard to deal with the temporal dimension in the action recognition process. Convolutional neural network (CNN) models could be used widely to solve this. With this motivation, this study designs a novel key point extraction with deep convolutional neural networks based pose estimation (KPE-DCNN) model for activity recognition. The KPE-DCNN technique initially converts the input video into a sequence of frames followed by a three stage process namely key point extraction, hyperparameter tuning, and pose estimation. In the keypoint extraction process an OpenPose model is designed to compute the accurate keypoints in the human pose. Then, an optimal DCNN model is developed to classify the human activities label based on the extracted key points. For improving the training process of the DCNN technique, RMSProp optimizer is used to optimally adjust the hyperparameters such as learning rate, batch size, and epoch count. The experimental results tested using benchmark dataset like UCF sports dataset showed that KPE-DCNN technique is able to achieve good results compared with benchmark algorithms like CNN, DBN, SVM, STAL, T-CNN and so on.

Keywords: Human activity recognition; pose estimation; key point extraction; classification; deep learning; RMSProp

1 Introduction

Human activity recognition (HAR) has gained significant attention among the computer vision (CV) communities [1]. Automated HAR enables computer scientists and engineers to develop smart surveillance systems, more natural human computer interfaces, and semantically aware video indexes [2]. It finds applicability in several areas, in law enforcement, HAR analysis on millions of videos taken by camera positioned over the street can be used to rapidly detect criminal or suspicious behaviors. Having effective HAR from realtime video lets emergency workers quickly spot criminal attempts to loot shops to an optimum but other attention focuses on rescue missions or a smaller set of peoples waving on top of the roofs waiting that rescued [3]. Pose estimation and HAR have attracted various applications including human–computer interfaces and video-based recognition [4]. But interms of speed and accuracy, still various studies have been carried out. Usually, pose estimation and Activity recognition are separately handled. In spite of the fact that pose is highly associated with activity recognition, for the advantages of activity recognition, a technique of resolving problems simultaneously isn't studied actively. In spite of the explosion of video data, the capacity of automatically recognizing and understanding human activity is still relatively low [5]. This is mainly because of several problems inherent to the detection tasks, like difficulty of the visual stimuli interms of camera motion, larger variability in human implementation style, viewpoint changes, background clutter, and so on. Many existing works have presented deep architecture for recognizing activities [6]. The Deep Learning (DL) method was extensively employed in image classification, recognition, predictive, and evaluation analysis in CV system. It extracts data directly in the original information and considerably forms a feature expression [7]. Firstly, a novel data are pre-processed, then data feature is removed by backpropagation (BP) and hierarchical forward propagation. Every layer expression was abstracted thus the final expression could be better describe the input data [8]. For a DL approach, Deep Belief Network (DBN) model is advantage in good modeling and action detection abilities. It processes many input features and establishes the connection among adjacent times for extracting the action context data with no considering the action feature distribution. Thus, it is employed in activity recognition [9]. Carrying out Human Activity Recognition (HAR) system using DL method made learning by stimulating human neural network. But it is known that HAR system doesn't focus on the whole scene at the same time [10]. Rather, human sequentially focuses on distinct portions of the scene for extracting appropriate data. Many current CV methods don't utilize an attention method and aren't studied actively in several areas of video or image. This study designs a novel key point extraction with deep convolutional neural networks based pose estimation (KPE-DCNN) model for activity recognition. The KPE-DCNN technique aims to initially identify the keypoints in the input frame and then determine the final activity label. The KPE-DCNN technique follows a three stage process namely key point extraction, hyperparameter tuning, and pose estimation. In addition, a new approach called OpenPose model is designed to compute the accurate key points in the human pose. Moreover, RMSProp optimizer with optimal DCNN model is used to determine the final activity label based on the extracted key points where the RMSProp optimizer is used to optimally adjust the hyperparameters such as learning rate, batch size, and epoch count. To demonstrate the promising performance of the KPE-DCNN approach, a wide range of experimental analyses is implemented on the benchmark videos. The rest of the article Section 2 describes the literature study performed on the proposed research in the past two decades, Section 3 elaborates the proposed research and Section 4 discusses results and analysis section and Section 5 discusses the findings of the research.

2 Literature Review

This section offers a comprehensive review of recently developed HAR and poses estimation approaches. Li and Chuah [11] presented an efficient and robust HAR system named ReHAR that is

utilized for managing group and single activity predictions. Firstly, create optical flow images for all the video frames. Next, video frames and respective optical flow image is fed to Single Frame Representation method to create representation. At last, Long Short Term Memory (LSTM) is utilized for predicting the final activity-based representation. The entire method is trained end-to-end for allowing effective representation to be created for the last action recognitions. In Guo and Wang [12], DBN is enhanced, also a human sport behaviour detection model based on specific spatio-temporal features are presented for obtaining, recognizing, and analyzing human sport behaviors from huge video data. The generated method is simulated on University of Central Florida (UCF) and Royal Institute of Technology (KTH) data sets, which provide an experiment for succeeding body detection and sports development in China. Nadeem et al. [13] present a unified architecture which explores multi-dimensional features using a fusion of body parts and quadratic discriminative analysis that employs this feature for marker less human pose estimation. In multi-level features are removed as displacement parameters to function as spatio-temporal property. This property represents the corresponding position of the body part regarding time.

Agahian et al. [14] introduced an architecture for recognizing human activities with three-dimensional skeleton data. The major component of architecture is encoding and pose representation. Assume that human action is characterized by spatiotemporal poses, it can bedetermine a pose descriptors consist of three components. Initially, it comprises standardized co-ordinates of raw skeleton joints data. Next, comprise the temporal displacement data related to a predetermined temporal offset and lastly keep the displacement data relevant to the preceding timestamp in the temporal resolution. Kim and Lee [15] projected a HAR that considered activity recognition, visual attention, and pose estimation. By using the visual attention model, weight is added to the essential part for enabling attention estimation. The visual attention model employs a soft visual attention method for enabling estimation without improving the number of estimations. Pham et al. [16] introduced a DL based multi-task architecture for joint three-dimensional HAR and human pose estimation from Red Green Blue (RGB) sensor using camera. The method consists of two phases. Initially, a real-time two pose detector is run to define the accurate pixel position of keypoint of human body. Then, a two-stream Deep Neural Network (DNN) is trained and designed for mapping two dimensional keypoint into three-dimensional poses. During the next phase, the Effective Neural Architecture Search (ENAS) approach was positioned for finding an optimum network structure. Luo et al. [17], developed a method to automatically estimate the pose of distinct construction equipment's in video taken on construction sites with CV and DL methods. Initially, keypoint of equipment are determined where the image is gathered from the surveillance camera is marked up for generating the ground truth label. Next, the architecture of three kinds of DL networks that are Stacked Hourglass Network (HG), Cascaded Pyramid Network (CPN), and an ensemble model (HG-CPN) integrate Cascaded Pyramid and Stacked Hourglass Networks are trained and constructed under the trained environments. Though several works are available in the literature, it is still needed to improve the overall recognition performance. Besides, only few works have focused on the hyperparameter tuning process which if concentrated in this work.

3 The Proposed Model

In this study, a new KPE-DCNN model has been developed for activity recognition. Initially, the RGB input videos are converted to sequence of frames using the 24frames/s. The KPE-DCNN technique encompasses three major processes as OpenPose based key point extraction, DCNN based activity classification, and RMSProp based Optimization. The application of the RMSProp model helps to properly tune the hyperparameters of the DCNN model and it helps to considerably boost the detection performance. Fig. 1 illustrates the overall process of KPE-DCNN technique.

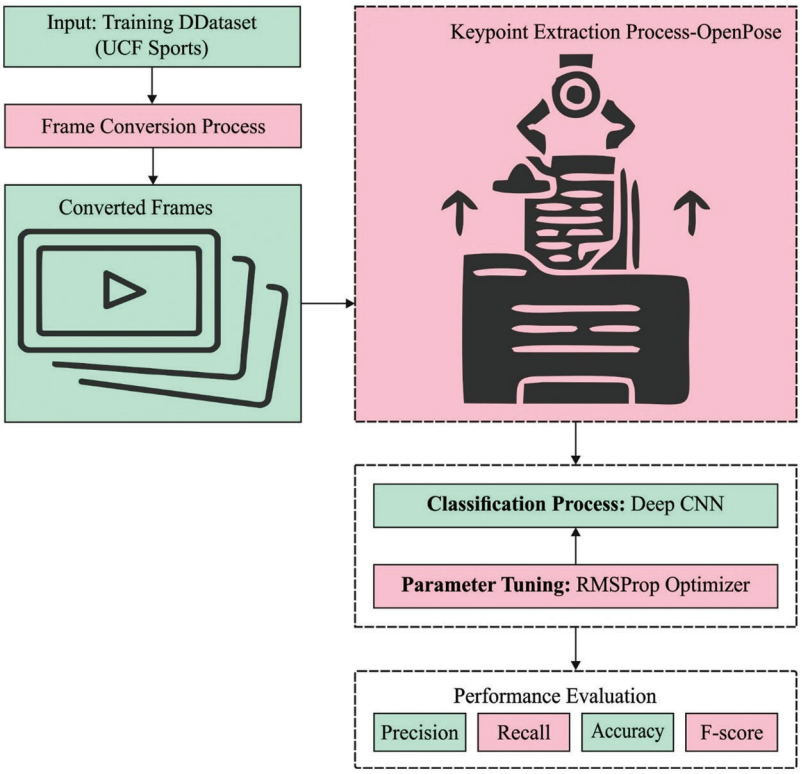

Figure 1: Overall process of kpe-DCNN technique

3.1 OpenPose Based Key Point Extraction

An increasing number of machine learning and computer vision application requires two-dimensional human pose estimation as an input for the system. The presented two-dimensional body pose estimation library includes Alpha-Pose [18] or Mask R- Convolutional Neural Network (CNN) [19], need their user to display to visualize the results, implement most of the pipeline, an output file generation with the result (for example, XML or JSON files), their own frame readers (for example, camera, video, or images streaming), and so on [20]. Furthermore, current facial and body keypoint detector isn't integrated, which requires distinct library for all the purposes. OpenPose resolves each of these problems. It could run-on various platforms, containing Windows, Ubuntu, embedded systems (for example, NvidiaTegra TX2), and Mac OSX. Also, it has supported distinct hardware, like OpenCL GPUs, CPUonly devices, and CUDA GPUs. The user may choose an input between video, images, Internet Protocol (IP) camera, and webcam streaming. Also, they enable or disable all the detectors (hand, body, foot, and face), skip frames for fast processing, select either to show the result or save them on disk, control how much GPU to use, enable pixel coordinate normalization and so on. The core block is an integrated body+footkeypoint detectors. It is simultaneously use the original body-only model trained on MPII and COCO data sets. According to the output of facial body detector, bounding box proposal could be roughly evaluated from some body parts, especially neck, ears, eyes, and nose. The proposed model has extracted 36 key points in OpenPose (i.e., 2 points for each part). Besides, cosine similarity searching technique is applied in this study. Likewise, the hand bounding box proposal is developed with the arm keypoint. These methods inherit the problem of top-down models. Though the facial keypoint detectors have been trained in similar way as that of hand keypoint detectors. Also, the libraries include three dimensional keypoint pose detections, by implementing three-dimensional triangulation with nonlinear Levenberg-Marquardt refinements over the

result of synchronized camera view. The inference time of OpenPose outperforms advanced methodologies when preserving higher-quality outcomes.

3.2 Activity Recognition Using DCNN Model

Once the keypoints are extracted, they are converted into .excel file. Totally, a set of 18 keypoints are extracted from each frame, consisting of X and Y points. These key points are fed as input into the DCNN model [21] to determine the activity label for the input video frame. With the help of CNN, we could learn rich features from the key point level and the character level, respectively. The upper components consist of two hidden layers, one input layer, two convolutional layers, and two pooling layers. Fig. 2 illustrates the structure of DCNN technique.

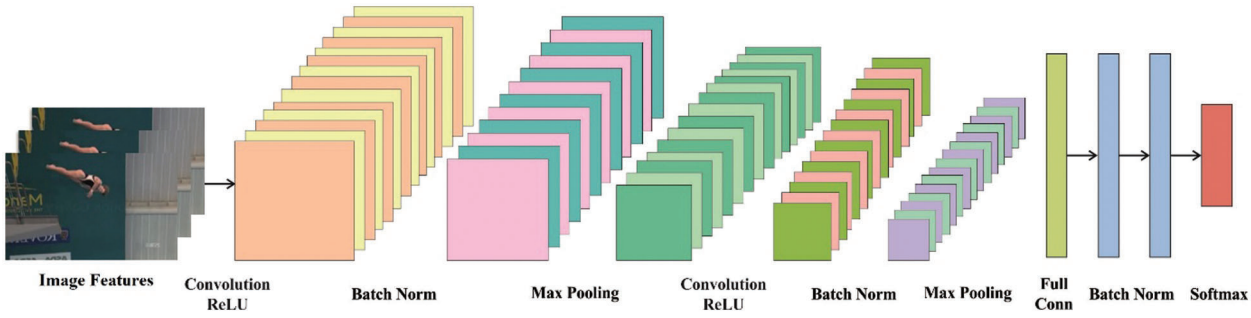

Figure 2: Structure of deep CNN

Input Layer. The input layers transform the information to a matrix of embedding, represented as $W \in R^{(k+n)\times m}$ as the input of network, in which n and k represent the maximal amount of key points and concepts, correspondingly. Also, m represent the dimension of key point embedding. Then attain W by concatenating the embedding of concepts and key points: $W = W_w \oplus W_c$. Now, W_w and W_c denotes the embedding of concepts and key points, correspondingly. Also, $\oplus$ represent the concatenation process. The way to construct W_w is relatively easy: Assume the data contains n key points, and $v_i^w \in R^m$ denotes an m-dimension vector of i^{th} key point in the data:

$$W_w = v_1^w \oplus v_2^w \oplus \ldots \oplus v_n^w \; n \tag{1}$$

In order to get the representations of W_c, assume the weight of concept simultaneously. For all the embedding vectors $v_i^c \in R^m$ of concept c_i, multiply it with the constant w_i to represent the weights of concept:

$$W_c = w_1 v_1^c \oplus w_2 v_2^c \oplus \ldots \oplus w_k v_k^c \tag{2}$$

When the concept or data vector is no longer sufficient, apply 0 as padding. Then get the embedding v_i^w and v_i^c by looking up the pretrained key point embeddings.

Convolution Layer. The process of convolutional layer is to extract high level features from the input matrix. In order to get distinct types of features, use filters with distinct sizes. Like earlier studies, fix the width of all the filters as m and process the height h as a hyperparameter. Assumed a filter $\omega \in R^{h\times m}$, a feature s_i is created from a window of concepts and key points $[v_i : v_{i+h-1}]$ as:

$$s_i = g(\omega \cdot [v_i : v_{i+h-1}] + b) \tag{3}$$

where $b \in R$ denotes a bias term. g indicates a nonlinear function. In the study, apply $ReL \cup$ as nonlinear function for convolutional layer. The filter is employed for each probable window of concepts and key

points in W to generate a feature map $s \in R^{n+k-h+1}$. This method is iterated for several filters with distinct heights to improve the feature coverage.

Pooling Layer. The process of pooling layers is to abstract the feature created from convolutional layer by gathering the scores for all the filters. In the study, use max-over-time pooling process over all the feature maps. The concept is to select the maximum values on all the dimensions of vector for capturing the essential feature. With pooling layer, induce fixed length vectors from feature map.

Hidden Layer. To utilize rich features attained from the pooling layer, we employ nonlinear hidden layers to integrate distinct pooling features. Then, employ tanh as the activation function. In this layer, employ dropout as a means of normalization by setting randomly to zero a proportion of element of the feature vectors. In the same way, the low subnetworks consist of two hidden layers, one input layer, two convolution layers, and two pooling layers. The input of subnetwork is a sequence of encoded characters. The encoding is performed by firstly producing an alphabet of each character in the data and randomly initializing the embedding of all the characters with m_c dimension. Next, the sequence of characters is converted to matrix $W_c \in R^{L \times m_c}$. Now, L represents a hyperparameter which limits the maximal size of the sequence. In the study, fix the values of L to be 256. Lastly, integrate the output vector of the two sub networks by concatenating them. Next, use output layer on joint vector to transform the output number into probability for classification.

3.3 Hyperparameter Tuning Using RMSProp

For improving the training efficiency of the DCNN method, the RMSProp model [22] is used to tune the hyperparameters such as learning rate, batch size, and epoch count. The DL methods are full of hyperparameters and finding the optimal configuration for this parameter in such a higher dimension space is not an insignificant problem. RMSProp is an optimization model designed by Geoffrey E. Hinton in Coursera. Further, optimize the loss functions in the upgrade of extreme swings and accelerate the convergence function, RMSProp approach employed the differential square weight average for the gradients of weight Wand bias b. Therefore, it makes greater development in the direction either the variable space is gentler. The number of squares of historical gradient is small owing to gentler direction that results in small learning drop.

$$s_{dw} = \beta s_{dw} + (1 - \beta) dW^2, \tag{4}$$

$$s_{db} = \beta s_{db} + (1 - \beta) db^2, \tag{5}$$

$$W = W - \alpha \frac{dW}{\sqrt{s_{dw}} + \varepsilon}, \tag{6}$$

$$b = b - \alpha \frac{db}{\sqrt{s_{db}} + \varepsilon}, \tag{7}$$

In which s_{dw} and s_{db} denotes the gradient and gradient momentum accumulated with the loss function in the preceding iteration t − 1, and β vector is exponential of gradient accumulation. In order to avoid the denominator, which becomes zero, ε becomes small value. RMSProp helps to eliminate the direction of large swings and is applied to correct the swings to help the swings in all the dimensions is small. Simultaneously, making the network function converge fast.

4 Experimental Validation

The proposed KPE-DCNN technique has been simulated using Python tool. The performance validation of the KPE-DCNN technique takes place using the UCF sports action dataset [23,24], which includes several

actions (class labels) such as Diving, Golf_Swing, Kicking, Lifting, Riding_Horse, Run_Side, Skate_Boarding, Swing and Walking_Front of 720 × 480 resolution. Figs. 3–4 shows the sample test images that exist in the diving and kicking classes respectively.

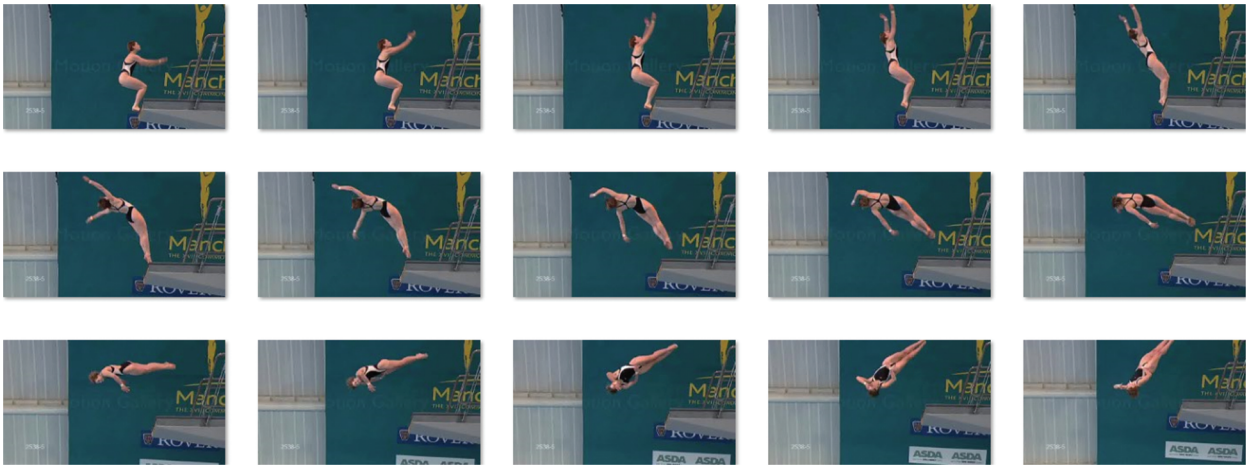

Figure 3: Sample image-diving

Figure 4: Sample image-kicking

Fig. 5 visualizes the sample results obtained by the OpenPose technique. The figure clearly shows the key point's extraction process for the kicking process.

Tab. 1 offers the sample set of extracted key points obtained by the KPE-DCNN technique. Totally, a set of key points are demonstrated in Tab. 1. Fig. 6 illustrates the confusion matrix generated by the KPE-DCNN technique on the classification of distinct human activities. The figure stated that the KPE-DCNN technique has identified 59 instances into class 0, 1374 instances into class 1, 383 instances into class 2, 728 instances into class 3, 677 instances into class 4, 667 instances into class 5, 1077 instances into class 6, 1116 instances into class 7, and 3185 instances into class 8.

Figurer 5: Key point extraction-kicking

Table 1: Extracted key points

NoseX	NoseY	NeckX	NeckY	RShoulderX	RShoulderY	RElbowX	RElbowY
0.00000	0.00000	0.65741	0.29891	0.67130	0.28804	0.68056	0.25000
0.64352	0.29891	0.64815	0.32065	0.66204	0.32609	0.68056	0.28261
0.62500	0.32609	0.65741	0.35870	0.64815	0.35870	0.67593	0.33152
0.62500	0.35326	0.66204	0.40217	0.65741	0.40217	0.00000	0.00000
0.62037	0.37500	0.64815	0.42391	0.63889	0.41848	0.00000	0.00000
0.62500	0.38044	0.66204	0.45109	0.66204	0.44565	0.00000	0.00000
0.62500	0.36413	0.66204	0.43478	0.66204	0.42935	0.62963	0.51087
0.63426	0.33696	0.66204	0.40217	0.67130	0.40217	0.61111	0.43478
0.64352	0.29891	0.66667	0.35326	0.67593	0.35870	0.00000	0.00000
0.66667	0.17935	0.68056	0.24457	0.67593	0.23370	0.68982	0.19022
0.63889	0.97283	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.68519	0.23370	0.68519	0.29891	0.68519	0.29348	0.67130	0.23370
0.68056	0.26630	0.67130	0.33152	0.66204	0.32065	0.58796	0.26630

(Continued)

Table 1 (continued)

NoseX	NoseY	NeckX	NeckY	RShoulderX	RShoulderY	RElbowX	RElbowY
0.50000	0.45109	0.53704	0.39674	0.00000	0.00000	0.00000	0.00000
0.49537	0.36957	0.55093	0.35326	0.55093	0.34783	0.00000	0.00000
0.51389	0.30435	0.55556	0.33696	0.00000	0.00000	0.00000	0.00000
0.55093	0.28261	0.57870	0.36413	0.61111	0.36957	0.00000	0.00000
0.59722	0.29348	0.58796	0.35870	0.56944	0.34783	0.54630	0.39674
0.48148	0.51630	0.52315	0.52717	0.53241	0.51630	0.00000	0.00000
0.50463	0.51087	0.53704	0.55978	0.53241	0.56522	0.49074	0.64130
0.54167	0.53261	0.55556	0.59239	0.55093	0.58152	0.00000	0.00000
0.60185	0.63044	0.59259	0.69565	0.59259	0.67391	0.57407	0.63587
0.35648	0.95109	0.35648	0.96739	0.34259	0.96739	0.00000	0.00000
0.37037	0.78261	0.37963	0.80978	0.36574	0.80978	0.35648	0.85870
0.01852	0.61957	0.01852	0.65217	0.00000	0.00000	0.00000	0.00000

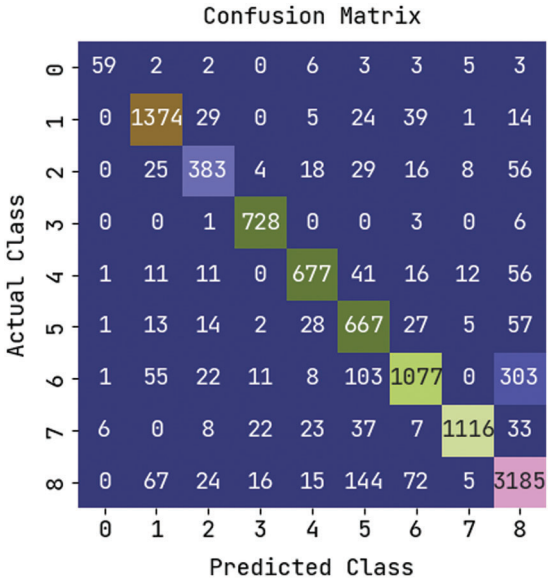

Figure 6: Confusion matrix of KPE-DCNN technique

Tab. 2 and Fig. 7 list the classification results provided by the KPE-DCNN technique under distinct class labels. The experimental values denoted that the KPE-DCNN technique has the ability to attain improved classification results under every class. For instance, under diving class, the KPE-DCNN technique has obtained$prec_n$, $reca_l$, and F_{score} of 86.76%, 71.08%, and 78.15% respectively. Likewise, under kicking class, the KPE-DCNN technique has attained $prec_n$, $reca_l$, and F_{score} of 77.53%, 71.06%, and 74.15% respectively. Similarly, under lifting class, the KPE-DCNN technique has reached $prec_n$, $reca_l$, and F_{score} of 92.98%, 98.64%, and 95.73% respectively. Eventually, under run_side class, the KPE-DCNN technique has obtained $prec_n$, $reca_l$, and F_{score} of 63.65%, 81.94%, and 71.64% respectively. Meanwhile,

under swing class, the KPE-DCNN technique has accomplished $prec_n$, $reca_l$, and F_{score} of 96.88%, 89.14%, and 92.85% respectively.

Table 2: Result analysis of KPE-DCNN technique with different classes

Classes	Precision	Recall	F-score
Diving	86.76	71.08	78.15
Golf_swing	88.82	92.46	90.60
Kicking	77.53	71.06	74.15
Lifting	92.98	98.64	95.73
Riding_horse	86.79	82.06	84.36
Run_side	63.65	81.94	71.64
Skate_boarding	85.48	68.16	75.85
Swing	96.88	89.14	92.85
Walking_front	85.78	90.28	87.97
Average	84.96	82.76	83.48

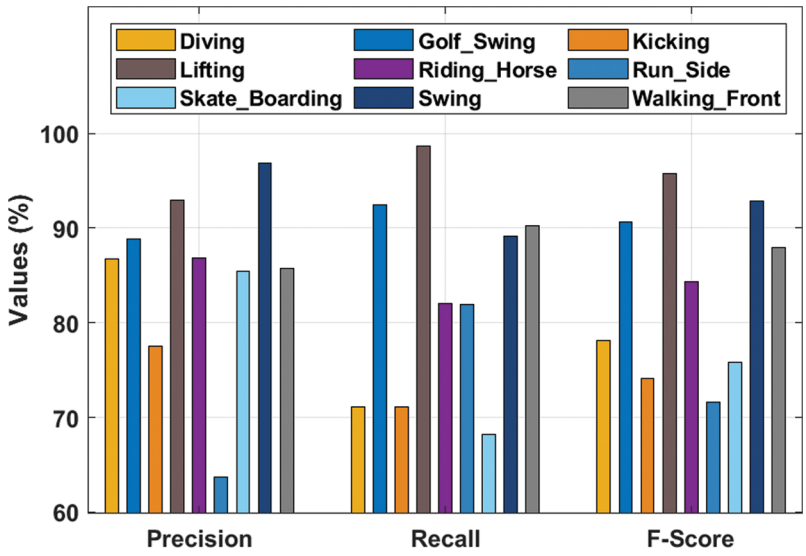

Figure 7: Result analysis of KPE-DCNN technique with different classes

Tab. 3 and Fig. 8 reports the average classification result analysis of the KPE-DCNN technique interms of different measures. The experimental values shown that the KPE-DCNN technique has obtained improved $accu_y$ of 85.44%, $prec_n$ of 84.96%, $reca_l$ of 82.76%, F_{score} of 83.48%, and ROC score of 98.50%.

The accuracy outcome analysis of the KPE-DCNN methodology on the test data is portrayed in Fig. 9. The results outperformed that the KPE-DCNN approach has accomplished higher validation accuracy related to training accuracy. It can be also observable that the accuracy values get saturated with the epoch counts. The loss outcome analysis of the KPE-DCNN technique on the test data is demonstrated in Fig. 10. The figure demonstrated that the KPE-DCNN technique has denoted the lesser validation loss over the training loss. It can be additionally stated that the loss values get saturated with the epoch counts.

Table 3: Average analysis of KPE-DCNN technique with respect to various measures

Metrics	Average values (%)
Accuracy	85.44
Precision	84.96
Recall	82.76
F-Score	83.48
ROC Score	98.50

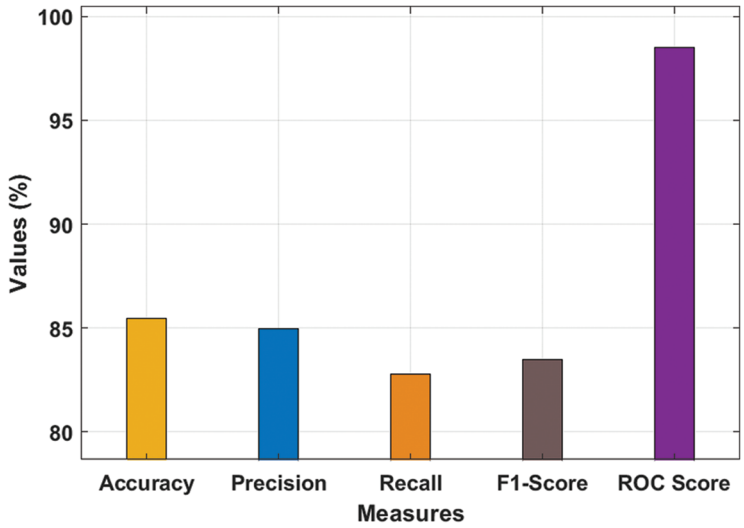

Figure 8: Average analysis of KPE-DCNN technique with various measures

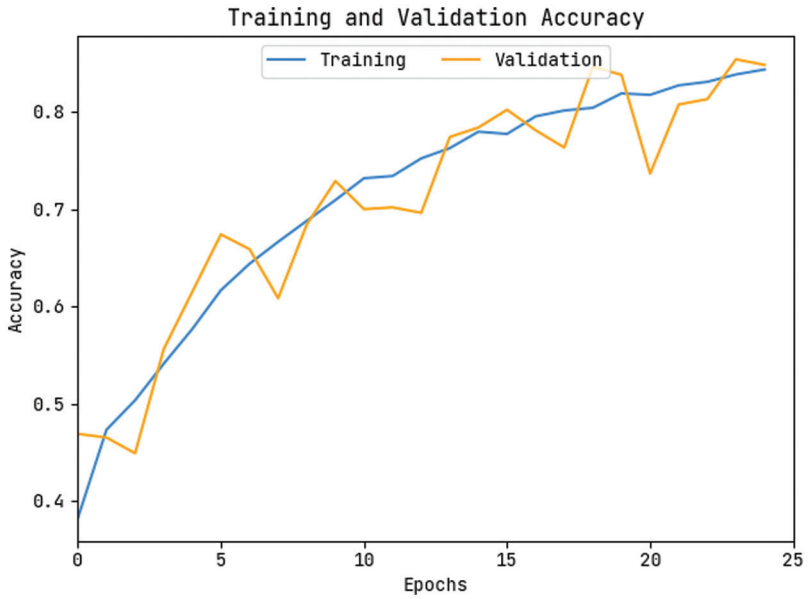

Figure 9: Accuracy graph analysis of KPE-DCNN technique

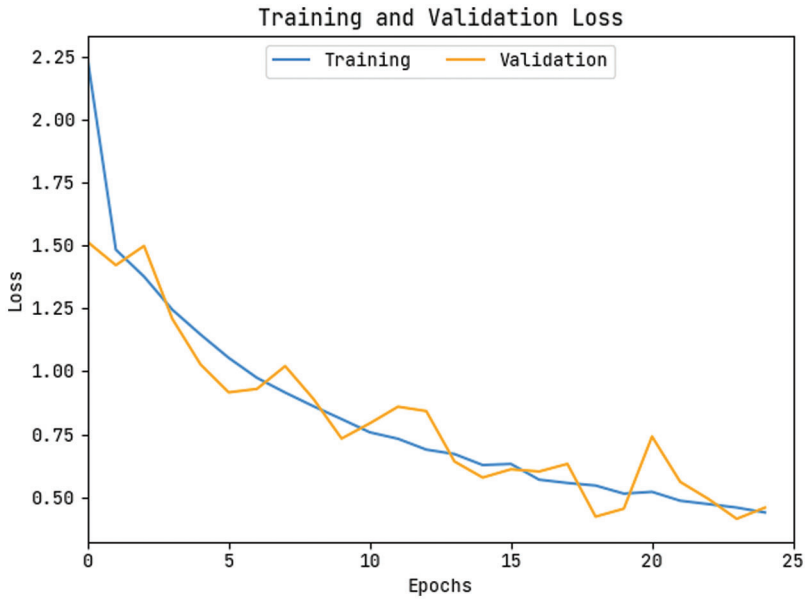

Figure 10: Loss graph analysis of KPE-DCNN technique

Fig. 11 demonstrates the precision-recall curve analysis of the KPE-DCNN technique on the test dataset. The figure portrayed that the KPE-DCNN technique has obtained effective classification results under all class labels. The AUC analysis of the KPE-DCNN technique takes place under varying class labels are provided in Fig. 12. The results show that the KPE-DCNN technique has resulted in maximum AUC of 99.91 under class label 3.

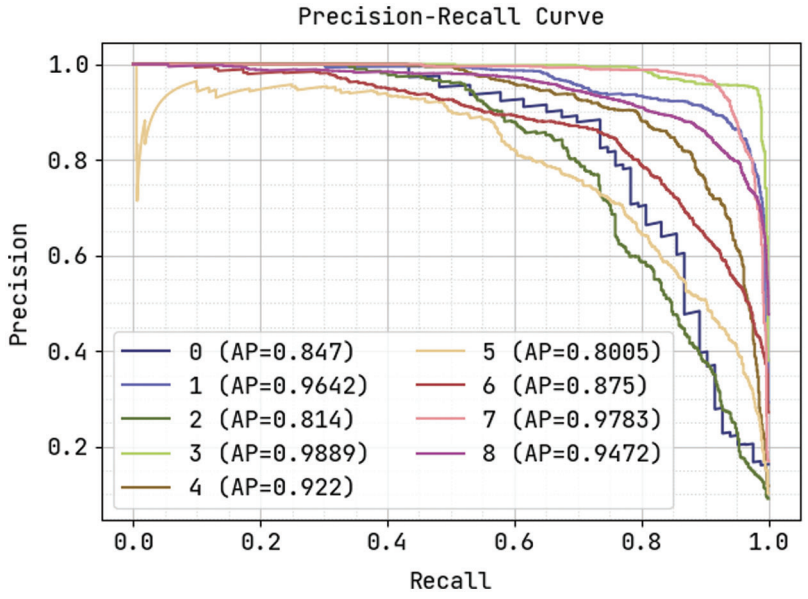

Figure 11: Precision-recall curve analysis of KPE-DCNN technique

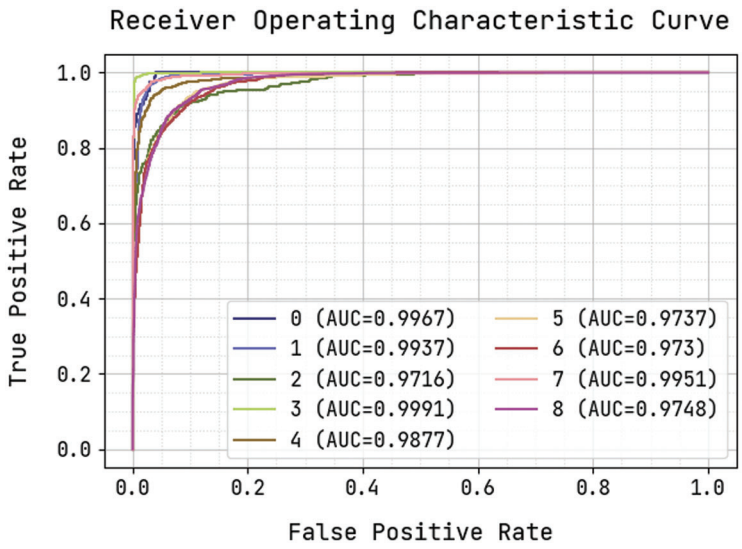

Figure 12: AUC analysis of KPE-DCNN technique

Finally, a comparative result analysis of the KPE-DCNN technique is offered with recent methods [25–27] in Tab. 4 and Fig. 13. The results demonstrated that the CNN and SVM models have obtained lower accuracy values of 76.57% and 78% respectively. At the same time, the DT and T-CNN techniques have attained slightly improved accuracy values of 80% and 80.56% respectively. Though the STAL technique has resulted in near optimal accuracy of 81.21%, the presented KPE-DCNN technique has accomplished maximum accuracy of 85.44%. From the above results and discussion, it is confirmed that the KPE-DCNN technique has the ability of accomplished effectual outcomes on pose estimation and HAR.

Table 4: Comparison analysis of KPE-DCNN technique with existing methods

Methods	Accuracy (%)
CNN	76.57
DBN	83.75
Decision Tree	80.00
SVM	78.00
STAL	81.21
T-CNN	80.56
KPE-DCNN	85.44

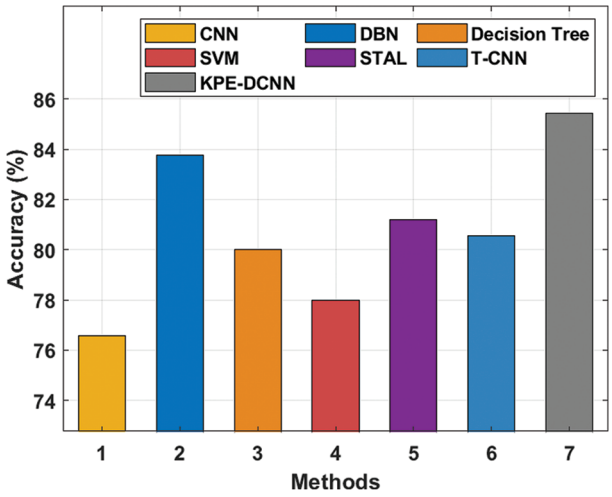

Figure 13: Comparative analysis of KPE-DCNN technique with existing methods

5 Conclusion

In this study, a new KPE-DCNN model has been developed for activity recognition. The KPE-DCNN technique encompasses three major processes as OpenPose based key point extraction, DCNN based activity classification, and RMSProp based hyperparameter tuning. The application of the RMSProp model helps to properly tune the hyperparameters of the DCNN model and it helps to considerably boost the detection performance. In order to demonstrate the promising performance of the KPE-DCNN technique, a wide range of experimental analyses is carried out on the benchmark videos. The extensive comparative analysis reported better outcomes of the KPE-DCNN technique over the recent approaches. In future, the detection performance of the KPE-DCNN technique can be improvised by the design of hybrid metaheuristic optimization algorithm based hyperparameter tuning process.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Kim, H. Lee, Y. Kim, S. Lee and H. Myung, "Weighted joint-based human behavior recognition algorithm using only depth information for low-cost intelligent video-surveillance system," *Expert System Application*, vol. 1, no. 45, pp. 131–141, 2016.

[2] L. N. Irvin and A. M. Melendez, "Human action recognition based on low- and high-level data from wearable inertial sensors," *International Journal of Distributed Sensor Networks*, vol. 15, no. 12, pp. 1–12, 2019.

[3] K. Aurangzeb, I. Haider, T. Kha, T. Saba, K. Javed *et al.,* "Human behavior analysis based on multi-types features fusion and von nauman entropy based features reduction," *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 4, pp. 662–669, 2019.

[4] I. Yang, X. Gao X. Gao and S. Peng, "A novel activity recognition system for alternative control strategies of a lower limb rehabilitation robot," *Applied Sciences*, vol. 9, no. 19, pp. 3986–3938, 2019.

[5] Z. Hu, S. Y. Park and E. J. Lee, "Human motion recognition based on spatio-temporal convolutional neural network," *Journal of Korea Multimedia Society*, vol. 23, no. 8, pp. 977–985, 2020.

[6] A. Fuentes and S. Yoon, "Deep learning-based hierarchical cattle behavior recognition with spatio-temporal information," *Computers and Electronics in Agriculture*, vol. 1, no. 177, pp. 105627–105632, 2020.

[7] S. Nabi A. F. Alkaim and Z. Adel, "An innovative synthesis of deep learning techniques (DCapsNet&DCOM) for generation electrical renewable energy from wind energy," *Soft Computing*, vol. 24, no. 14, pp. 10943–10962, 2020.

[8] N. Dawar, S. Ostadabbas and N. Kehtarnavaz, "Data augmentation in deep learning-based fusion of depth and inertial sensing for action recognition," *IEEE Sensor Letters*, vol. 3, pp. 1–4, 2018.

[9] T. Jayasankar and A. Vijayaselvi, "Prediction of syllable duration using structure optimised cuckoo search neural network (SOCNN) for text-to-speech," *Journal of Computational and Theoretical Nanoscience,* vol. 1, no. 1, pp. 7538–3544, 2016.

[10] Y. Yang, P. Hu and X. Deng, "Human action recognition with salient trajectories and multiple kernel learning," *Multimedia Tools and Applications*, vol. 18, no. 77, pp. 17709–17730, 2018.

[11] X. Li and M. C. Chuah, "Rehar: Robust and efficient human activity recognition," in *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, USA, vol. 18, no. 6 pp. 362–371, 2018.

[12] Y. Guo and X. Wang, "Applying TS-DBN model into sports behavior recognition with deep learning approach," *the Journal of Supercomputing*, vol. 7, no. 4, pp. 1–17, 2021.

[13] A. Nadeem., A. Jalal and K. Kim, "Accurate physical activity recognition using multidimensional features and markov model for smart health fitness," *Symmetry*, vol. 12, no. 11, pp. 1766–1778, 2020.

[14] S. Agahian, F. Negin and C. Kose, "An efficient human action recognition framework with pose-based spatiotemporal features," *Engineering Science and Technology, an International Journal*, vol. 23, no. 1, pp. 196–203, 2021.

[15] J. Kim and D. Lee, "Activity recognition with combination of deeply learned visual attention and pose estimation," *Applied Sciences*, vol. 11, no. 9, pp. 4153–4168, 2021.

[16] H. H. Pham, A. Crouzil, S. A. Velastin and P. Zegers, "A unified deep framework for joint 3d pose estimation and action recognition from a single rgb camera," *Sensors*, vol. 20, no. 7, pp. 1825–1836, 2020.

[17] H. Luo, M. Wang, P. K. Y. Wong and J. C. Cheng, "Full body pose estimation of construction equipment using computer vision and deep learning techniques," *Automation in Construction*, vol. 12, no. 110, pp. 103016–103025, 2020.

[18] K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask r-cnn," in *Proc. Int. Conf. on Computer Vision*, Venice, Italy, pp. 1–12, 2017.

[19] H. S. Fang, S. Xie, Y. W. Tai and C. Lu, "RMPE: Regional multiperson pose estimation," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 2334–2343, 2017.

[20] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 1812–1824, 2017.

[21] J. Wang, Z. Wang, Z. Zhang and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification," in *Proc. of Twenty-Sixth Int. Joint Conf. on Artificial Intelligence*, Melbourne, Australia, vol. 350, pp. 2915–2921, 2017.

[22] F. Zou, L. Shen, Z. Jie, Z. Zhang and W. Liu, "A sufficient condition for convergences of adam and rmsprop," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, CA, USA, vol. 3, pp. 11127–11135, 2019.

[23] T. Lan, Y. Wang and G. Mori, "Discriminative figure-centric models for joint action localization and recognition," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, Wahington, United Sates of America, pp. 1–21, 2011.

[24] M. D. Rodriguez, J. Ahmed and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, USA, vol. 4, pp. 1–8, 2008.

[25] S. RuiHou, C. Chen, and M. Sha,"Tube convolutional neural network (t-cnn) for action detection in videos," in *Proc. of the IEEE International Conference on Computer Vision*, Chen, vol. 4, no.17, pp. 1703.10664, 2017.

[26] P. Weinzaepfel, Z. Harchaoui and C. Schmid, "Learning to track for spatio-temporal action localization," *Proc of the IEEE International Conference on Computer Vision*, vol. 15, no. 8, pp. 3164–3172, 2015.

[27] W. Sun, L. Dai, X. R. Zhang, P. S. Chang and X. Z. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Applied Intelligence*, vol. First Online, pp. 1–16, 2021.