JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 16, No. 4, December 2022, pp. 199-210

Recognition and Classification of Human Actions using 2D Pose Estimation and Machine Learning

Monika Dhiman*
Computer Science and Engineering, University Institute of Engineering and Technology, Panjab University, Chandigarh, India
dhimanmonika772@gmail.com

Akash Sharma
Computer Science and Engineering, University Institute of Engineering and Technology, Panjab University, Chandigarh, India
akashazarcosharma@gmail.com

Sarbjeet Singh
Computer Science and Engineering, University Institute of Engineering and Technology, Panjab University, Chandigarh, India
sarbjeet@pu.ac.in

Abstract
Recognition and classification of human actions is a fundamental but difficult computer vision task that has been studied by several researchers throughout the world in recent years. Pose estimation is a widely used technology to recognize human actions. It has several applications, especially in the field of computer vision, where it can be used to recognize basic as well as complex human actions. This research provides a novel framework for recognition and classification of human actions which includes five categories - standing, walking, waving, punching and kicking. The dataset used for the recognition and classification purpose is generated using the videos that are recorded by using a smart phone and 2D pose estimation technique has been applied to extract the features from the human body. The ML classifiers have been trained on a custom-built dataset. While all algorithms nearly performed well in classification task, LGBM outperformed the rest in terms of accuracy (98.80 %).

Category: Human-Computer Interaction

Keywords: Action Classification; Action Recognition; Openpose; Pose Estimation; Machine Learning

I. INTRODUCTION

Human activity recognition (HAR) systems have acquired a lot of traction in the field of computer vision in the last decade. Identifying human actions and classifying them from video sequences or still images is a difficult endeavour due to the large range of applications associated with this specific area of research. It has a wide range of uses, including video surveillance, human-computer interaction, violence detection, fall detection, sport performance analysis, etc. By recognizing the movements of a human body in a sequence of photos, we can deduce what a person is doing.

Pose estimation is a technique that estimates human

Open Access http://dx.doi.org/10.5626/JCSE.2022.16.4.199 http://jcse.kiise.org

Received 22 July 2022; **Accepted** 06 December 2022
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

poses from key points which are extracted from skeleton. It detects different parts of the body that include the arm region, legs region, facial region, etc. Recent study [1] indicates that human pose estimation can be used for action recognition, but this does not imply that the two tasks are unrelated or that action recognition is a secondary task. In fact, action recognition may serve as a catalyst for further advancements in human pose estimation. Pose estimation makes it possible to determine that a pose is obviously distinct from another by using the joints of the body; hence, if activity recognition is carried out using the joint data, the activity can be more precisely identified.

The HAR methods have been mainly classified into three categories: template-based, probability statistics-based and semantic-based methods[2]. Traditional HAR algorithms usually have high requirements on source images and consume too many computational resources. In single image identification tasks, less processing is required, making classification easier; but, in real-time videos, there is a correlation between all previous image frames and the present image frame, making the task more challenging because these frames must be analyzed at the same time.

In this paper, we have proposed a real time body pose estimation and human action classification by using open-source pose identification library Openpose (https://github.com/CMU-Perceptual-Computing-Lab/openpose) having less computational cost. By using this library, we analyzed the coordinates of human skeletons, to locate the region of a human figure. As indicated in Fig.1, we consider five specific actions for our HAR problem: standing, walking, punching, waving and kicking. All these actions are collected by two people at a time (where one is performing the action and the other one holding the equipment that is used to capture videos) on our university campus. However, using only the skeleton information of humans, this study aims to develop a system that accurately classifies basic human postures in images as well as videos.

Our work is greatly influenced by work carried out by Chen et al. [2] and Rao [5]. We have created a system that detects a variety of human activities including violent as well as non-violent actions. Violent actions consist of punching and kicking, while non-violent actions consist of standing, walking and waving.

This paper is organized as follows. In Section II, we discuss the existing approaches for the human action recognition system. The dataset generated for the classification of five primary actions is presented in Section III. Section IV presents the whole architecture and technique used to identify and classify actions. Section V then presents the final evaluation findings utilizing a variety of machine learning classifiers. Finally, Section VI wraps up this research and offers suggestions for the future.

II. RELATED WORK

Recognizing human activity is still a key problem that is being researched. In contrast to existing HAR systems, the HAR approach we offer is implemented on our unique real-time dataset and incorporates a crucial component of pose estimation in the processing of photos and videos.

Cao et al. [4] proposed Openpose, the first open source library to obtain the information of body posture using the bottom up method of association scores by using Part Affinity Fields (PAFs), a set of 2D vector fields that encode the location and orientation of limbs over the image domain. As a result, the bottom-up approach employing Openpose, which uses PAFs, a set of 2D vector fields that encode the placement and alignment of parts over the image domain, is recommended.

Due to the shared focus on comprehending human motion, human pose estimation and action identification are two tasks that are closely related. Chen et al. [2] proposed an action recognition system that used the extracted skeleton images instead of source images and they had also used the openpose library for skeleton extraction from source images. The four human activities i.e., squat, walk, stand and work were considered by the authors for further classification and their extracted skeleton images were fed into CNN.

Singh et al. [3] proposed surveillance system using drones for detection of violent human actions using SHDL network. The suggested method employs a feature pyramid network (FPN) to detect individuals in aerial images while, SHDL network is utilized to learn pose estimation by using an annotated Aerial Violent Individual (AVI) dataset that can identify a total of 14 key points using proposed network. In addition, for the identification and classification purpose, SVM (Support Vector Machine) has been used to detect violent and non-violent classes.

An efficient real time activity identification and classification using pose estimation system has been implemented by Rao [5] where the cost of computation and accuracy are considered as the two key obstacles it has confronted. The author used the Openpose that can extract the person's pose from a folder of images and then applied preprocessing and feature extraction on the joints produced by the Openpose. Then, to classify the actions various machine learning algorithms are used and the most accurate one to check the test data.

Hidalgo et al. [6] described the first single-network solution to whole-body pose estimation that works in real time regardless of the number of persons in the image. Their approach also improved the present state-of-the-art Openpose by significantly increasing its run-time execution while also somewhat enhancing key point accuracy.

Ghazal et al. [7] presented an algorithm for activity

recognition system for humans on still images using 2D pose information and activities that were taken under consideration were sitting and standing. The algorithm was implemented on their own dataset that were collected using images from internet and for feature extraction, openpose was used. Openpose extracted a total of 18 key points, out of which only 6 key points (Left hip, left knee, left ankle, right hip, right knee and right ankle) were used to extract information to distinguish between sitting and standing poses. The model was then tested on two different datasets i.e., INRIA and Freiburg Datasets (contains images of sitting and standing) and MPII Human Pose Dataset (contains images of people doing daily life activities like, running, exercising, etc.) to check the accuracy of the algorithm.

Similarly, Osokin [8] proposed a lightweight openpose that performed 2D pose estimation on the CPU without the use of a graphic card and provided analysis of the original openpose.

Sultani et al. [9] also proposed real time videos dataset using drone for the action identification purpose, where they have used the game action videos to tackle the challenges in real life scenario videos. For that, they presented two datasets: 1) Aerial ground game dataset and 2) Real aerial dataset that contains actions corresponding to eight actions of UCF-101.

Recently, Dileep et al. [10] proposed an approach on HAR to classify the suspicious actions and normal actions using 2D pose estimation algorithms such as Openpose and Mediapipe [11], as well as CNN, were implemented on their own prepared customized dataset, which included two main actions wall climbing or trespassing and fall detection. They have also presented an alarming system that has to occur whenever suspicious activity is detected and a message is sent to the desired person through email, so that any further immediate action could be taken.

Nale et al. [12] proposed an action detection system for both normal and abnormal activities using skeleton data that is reduced to geometrical features. For classification, they used LSTM, a modified form of RNN (Recurrent Neural Network), to distinguish between different medical suspicious activities. The dataset used for training is NTU-D 60, which consists of 25 joints of the human skeleton. It was tested on NTU RGB+D dataset.

As described in [13] by Tripathi et al., due to the wide

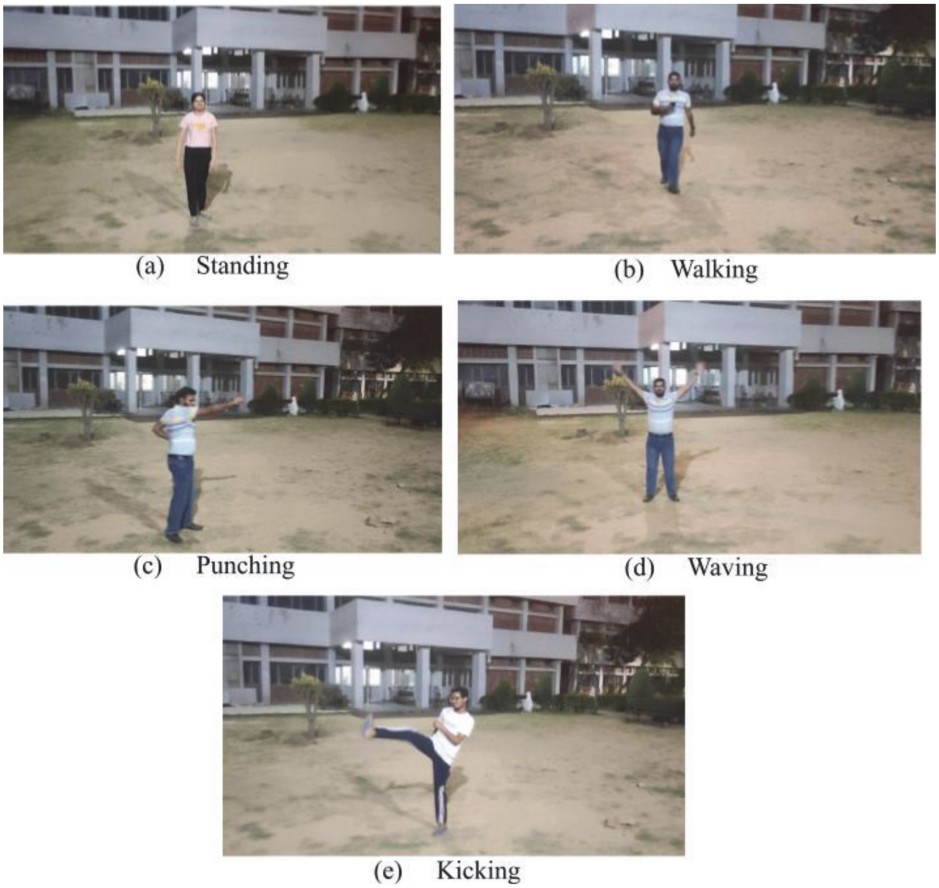

Fig. 1. Sample Snapshots of collected actions.

range of recognition of suspicious human actions from cameras, it may be utilized to many domains including university campuses and academic institutions, public infrastructure, retail commerce, airports, railways, and bus stations.

The fundamental takeaway from this research work is the creative, minimal approach to categorizing and identifying human activity that we implement to resolve the HAR dilemma.

III. DATASET FOR RECOGNITION OF HUMAN ACTIONS

This research uses an annotated dataset for the purpose of recognizing and classifying various types of human actions. The dataset is composed of 21715 samples, where each sample is labeled manually with each action class. The complete dataset consists of five people performing each action from one of the following activities: Walking, Punching, Waving, Kicking and Standing as shown in Fig. 1.

These activities have been collected using smart phone mounted on tripod stand from the same height of 9 ft (ft: feet), two videos of each action are recorded, while the duration for each video remains fifty seconds to two minutes. The videos are captured with static background, having proper stabilization, without any illumination changes and total 25 key points (using 2D pose estimation [14]) for every human are labeled manually with each action for further classification purpose.

The human action identification and classification task from the targeted dataset is a challenging task to perform with less computation time and minimal cost. The dataset is trained using various ML classifiers resulting in achieving satisfactory accuracy.

IV. DESIGN AND METHODOLOGY

Initially, five different types of human activities are collected, annotated, and preprocessed, then the pose estimation technique is used to generate human poses and extraction of key points from skeleton data of human body. After that, a set of 25 joint key points (from the COCO + foot model [4] of OpenPose) per frame has been nominated as features that are manually annotated with their related actions corresponding to the human poses. A total of 21,715 frames were gathered and retrieved using the 2D OpenPose method of pose estimation; Fig. 4

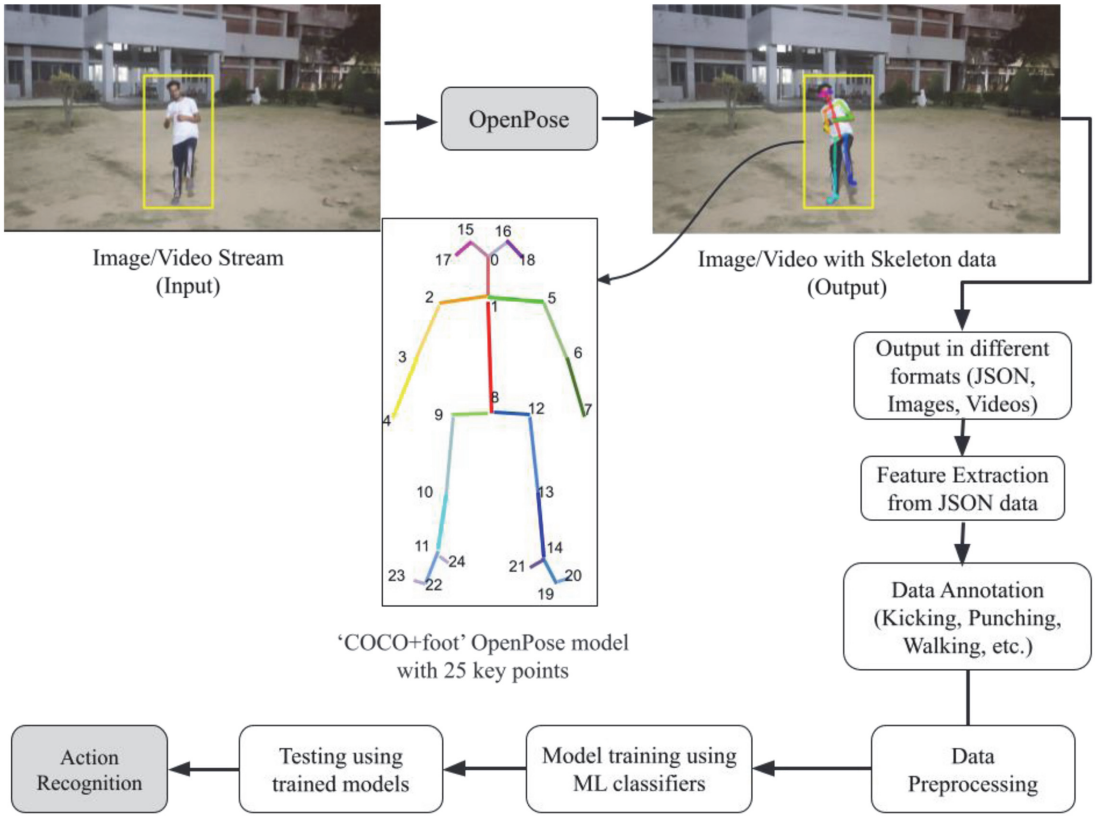

Fig. 2. Illustration shows the workflow for Human Action Recognition system based on skeleton data.

Table 1. Dataset Statistics of videos

Parameters	Values
Frame Rate	30 fps
Resolution	1920x1080 pixels
Length	50 sec to 2 min

shows the data distribution in more detail. Then, each action class is manually assigned to the joint positions (key points) of each frame. The annotated dataset was then trained using various machine learning techniques such as SVM, KNN, XGBoost, and others, with the most accurate classifier being chosen for testing. Finally, we compared the results of various machine learning classifiers that demand the least amount of money and computational time by using these strategies. The whole workflow of human activity identification system using joint positions is shown in Fig 2.

A. Data Collection

In this study, we have gathered five types of actions in the form of videos in our university campus that were recorded in several situations. These videos were captured at ground level by using a smart phone, mounted on a tripod stand having resolution of 1920x1080 pixels and a frame rate of 30 frame per second, ensuring that they were quick enough to capture the entire movement of an activity. Each video has a varied length, ranging from 50 seconds to 2 minutes, and thus a distinct number of frames. We have collected fight as well as normal human activities, where Punching and Kicking are incorporated into the category of fight whereas Standing, Waving and Walking falls under normal category. Table 1 shows the dataset statistics for the collected videos.

B. Pose Estimation and Feature Extraction

Recognizing body parts of person and the difficulty of identifying structural key points or "parts" has been a major focus of human pose estimation. There are a variety of pose estimation techniques available, including Mask R-CNN, Alpha-Pose, Openpose, and others [15] from which we chose Openpose because it solves the concerns that other libraries have. Openpose allows us to retrieve the key points of body parts by providing input as

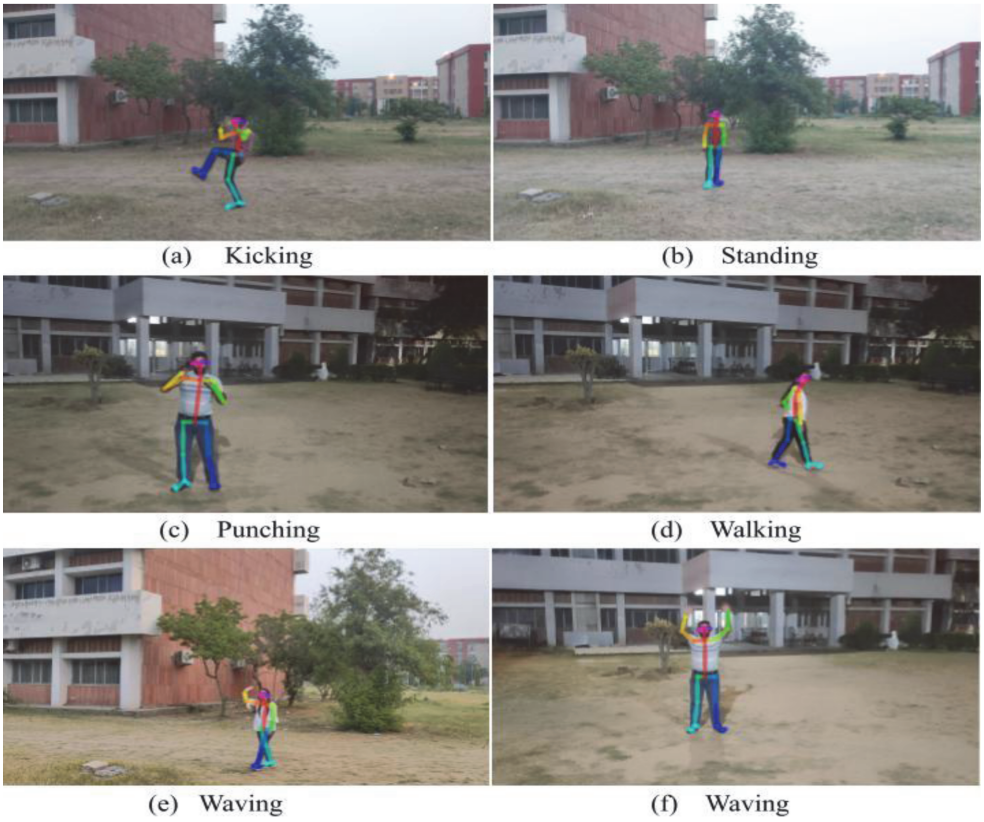

Fig. 3. Illustration shows the resultant frames using Openpose.

one video at a time. To get the features from the input, the image is passed across the CNN output network. After that, the feature map is processed in multi-stage CNN sequential layers to yield Part Affinity Fields (PAF) and Confidence Map.

To capture human pose in the image, the PAFs and confidence map obtained above are passed via a bipartite graph matching method [16]. Sample frames of the estimated human pose are shown in Fig 3, that are saved in our custom files, where key points of the skeleton are represented by x and y coordinates of every joint position of the human body.

The total numbers of extracted frames of each action and the distribution of data are indicated in Fig 4. The resultant joint key points are then saved in a JSON file (one JSON file per resultant frame), from which 25 key points (x-axis and y-axis co-ordinates of each joint position) are preprocessed and chosen for feature extraction. The 25 key points are illustrated as follows where R and L in the features indicates the right and left region of the human body respectively: 'Nose', 'Neck', 'REye', 'LEye', 'REar', 'LEar' 'RShoulder', 'LShoulder', 'RElbow', 'LElbow', 'RWrist', 'LWrist', 'RHip', 'MidHip', 'LHip', 'RKnee', 'LKnee', 'RAnkle', 'LAnkle', 'RHeel', 'LHeel', 'RBigToe', 'LBigToe' 'RSmallToe', 'LSmallToe'.

C. Data Annotation and Preprocessing

After all the relevant features from the input videos have been extracted by performing pose estimation, data annotation or labelling of all five action classes has been done manually by awarding each skeleton key point data a corresponding action class label. In order to convert the action class field from categorical to numerical form, we used an in-built encoder called LabelEncoder loaded from the Sklearn package that converts the data in machine readable format and it assigns a unique number to each class of the data. In our dataset, after applying label encoding, each action class is encoded as follows: 0 – 'Kicking', 1 – 'Punching', 2 – 'Standing', 3 – 'Walking', 4 – 'Waving'.

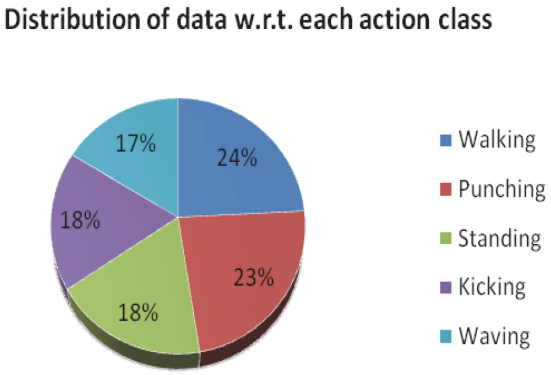

Fig. 4. Illustration shows the distribution of frames extracted with respect to each action.

D. Classification

Our dataset comprises five classes, as shown in Fig 1, leading to a multiclass classification task. The dataset is divided in proportion, with 70% of the data used for training and 30% of the data used for testing; among these, 17372 rows are used for training and the rest 4343 are used for testing. Ten machine learning classifiers are used to train the model after splitting the dataset, all these classifiers are imported from the Python library Scikit-learn, while some are installed using the pip package. Each action class is accurately and precisely classified by using these classifiers. In the next section, we have outlined each classifier along with its associated confusion matrix and evaluation results.

V. RESULTS AND DISCUSSION

In this section, we compare the accuracy results of machine learning classifiers and present the evaluation findings from the machine learning algorithms in terms of performance parameters.

A. Performance Measures

Accuracy, precision, recall, and F1 score are some of the metrics we consider while evaluating the performance of machine learning algorithms. Table 2 illustrates the equations for each performance measure. Where, the true positives (TP) reflect the number of actions that were correctly assigned to the class to which they belong, and the true negatives (TN) represent the number of actions that were correctly assigned to the class that they do not belong to. Additionally, the false negatives (FN) and false positives (FP) show the number of actions that were incorrectly identified as not belonging to a class and as belonging to a particular action, respectively.

***1) Accuracy*:** The ratio of correctly categorized action classes to the total number of samples represents accuracy as indicated in Table 2. The maximum accuracy we could obtain for our model was 0.988, which equates to a 98 percent accuracy rate.

In order to illustrate the performance of the classifier model, we additionally employ other performance metrics (precision, recall, and F1-score).

***2) Precision*:** Precision is a measurement of the proportion of correctly predicted positive action classes to all positive classification predictions that are positive.

***3) Recall*:** Recall which measures the proportion of real positive, correctly expected action classes.

***4) F1-Score*:** F1-score that calculates the average of recall and precision.

Table 2. Equations for each Performance Measure

Performance metric	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1-Score	$\frac{2\ x\ Precision\ x\ Recall}{Precision+Recall}$

B. Experimental Setup

All the experiments for the machine learning algorithms were implemented in Python. Specifically, the Scikit-learn library is used to implement, train, and test the machine learning algorithms. The dataset is split into training and test datasets, containing 70% of the samples for training and 30% of the samples for testing. In addition, we also performed data preprocessing and set different parameters in the machine learning algorithms to promote accuracy. These parameters are related to each machine learning algorithm.

C. Evaluation of Machine Learning Algorithm

We implement and compare ten machine learning algorithms: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), Gaussian Naïve Bayes (GNB), Adaptive Boosting (AdaBoost), Random Forest (RF), MultiLayer Perceptron (MLP), Linear

Table 3. Accuracy, precision, recall and F1-score values for the classification algorithms

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	97.55	97.42	97.30	97.34
SVM	92.13	92.21	91.69	91.75
DT	96.92	96.69	96.71	96.70
GNB	47.89	52.20	49.47	38.70
AdaBoost	97.36	97.14	97.18	97.16
RF	98.58	98.46	98.46	98.46
MLP	97.34	97.19	97.26	97.20
LDA	81.19	81.26	80.28	80.43
LGBM	98.80	98.71	98.68	98.69
XGBoost	98.67	98.56	98.57	98.56

Table 4. Accuracy, precision, recall and F1-score values for the classification algorithms using stratified k-fold cross validation

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	96.93	96.86	96.77	96.79
SVM	90.95	91.16	90.66	90.71
DT	96.51	96.39	96.37	96.37
GNB	48.01	52.57	48.95	38.31
AdaBoost	96.49	96.38	96.35	96.36
RF	98.52	98.44	98.45	98.44
MLP	96.50	96.43	96.44	96.42
LDA	79.51	80.01	78.76	78.91
LGBM	98.46	98.37	98.37	98.36
XGBoost	98.43	98.35	98.34	98.34

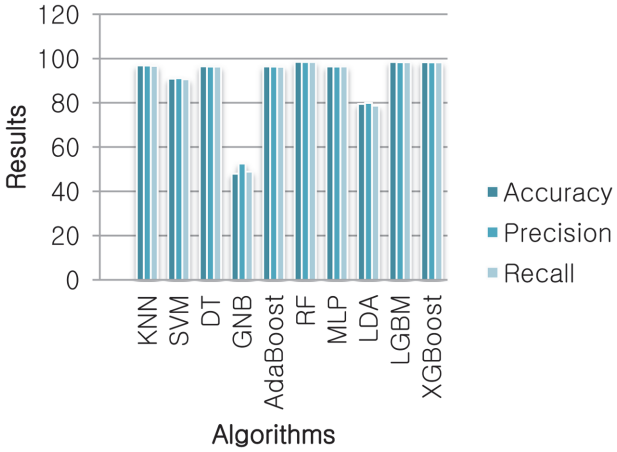

Fig. 5. Performance of various machine learning algorithms in terms of accuracy, precision and recall.

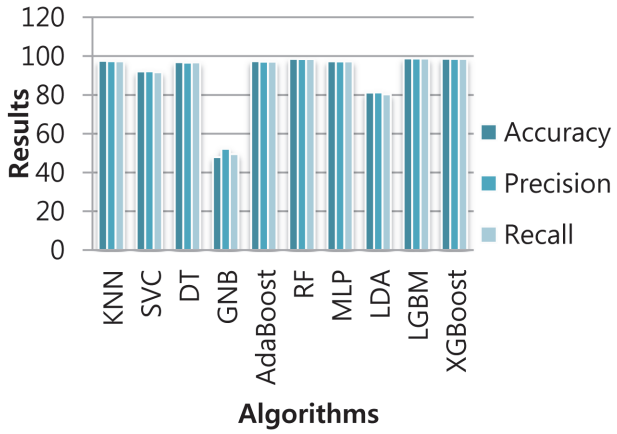

Fig. 6. Performance of various machine learning algorithms using stratified k-fold cross validation.

Discriminant Analysis (LDA), Light Gradient Boosting Machine (LGBM) and eXtreme Gradient Boosting (XGBoost) as mentioned in Table 3.

In addition, stratified k-fold cross-validation (k-fold CV) is also used to evaluate the performance of these algorithms, in which the training set is split into k smaller sets (where k= 10, estimator is a classifier and target variable is multiclass) and the resulting model is validated on the remaining part of the data. The evaluation results of cross validation are also shown in Table 4. We set different model parameters for each algorithm to improve accuracy. The performance of these algorithms in terms of accuracy, precision and recall is illustrated in Fig 5 and Fig. 6.

1) K-Nearest Neighbors: In the KNN algorithm, we use the minkowski distance that provides high accuracy and performance. The KNN algorithm classifies data by finding the closest k neighboring data points where the maximum accuracy is obtained when we set k as 3. The data class prediction is then based on majority voting between the neighbors according to distance. The average accuracy of the KNN algorithm reached 97.55%, which is remarkable performance and confusion matrix is also shown in Fig 7.

2) Support Vector Machine: The SVM performs supervised learning and represents data samples in a high-dimensional space. In the SVM implementation, we set a linear kernel and penalty parameter 'C' is set as 1. This space determines a hyperplane to optimally separate the samples and maximize the margin between classes, using support vectors. The average accuracy of SVM reached 92.13% and confusion matrix for the same is also shown in Fig 8.

3) Decision Tree: The DT algorithm consists of multiple nodes and conditions until reaching its leaves to predict classes. The average accuracy of the DT algorithm reached 96.92%, which is the remarkable performance among the evaluated classification algorithms and confusion matrix is shown in Fig 9.

***4) Gaussian Naïve Bayes*:** Naïve Bayes classifier is a basic but effective classification model that draws influence from Bayes Theorem. In this implementation, we have used Gaussian Naïve Bayes to predict the classes. GNB model is not very suitable for numerical data. It works best when features are independent of each other and may not work well suitable when the feature-values are nominal. The average classification accuracy of the GNB model reached 47.89%, which is lower than that of other algorithms and confusion matrix is shown in Fig. 10.

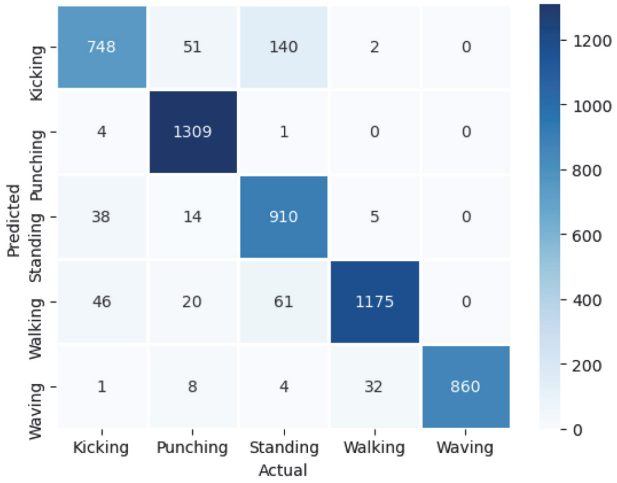

Fig. 8. Confusion matrix for SVM with linear kernel.

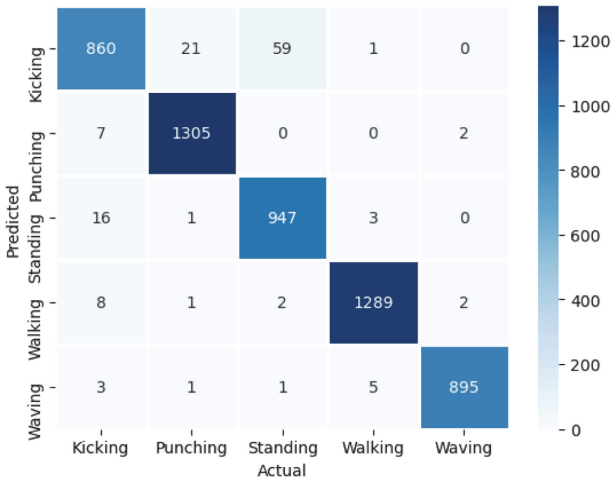

Fig. 7. Confusion matrix for KNN classification.

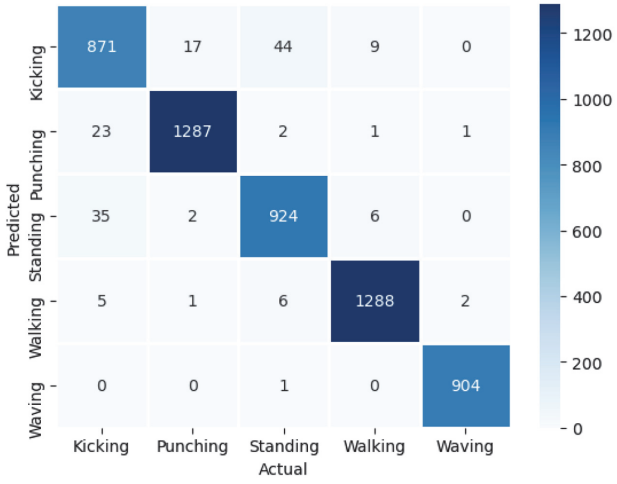

Fig. 9. Confusion matrix for DT classification.

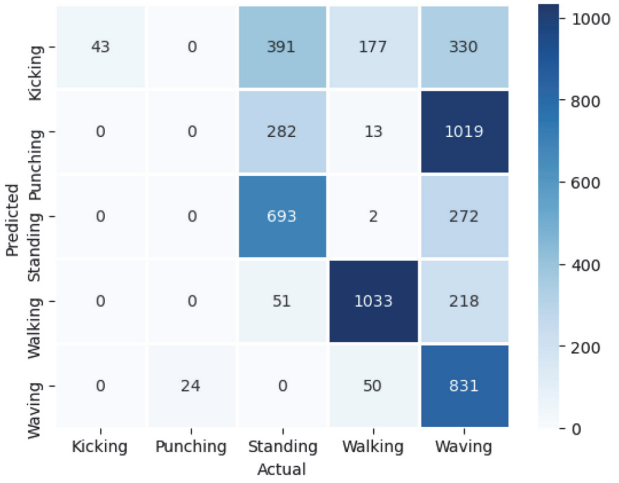

Fig. 10. Confusion matrix for GNB classification.

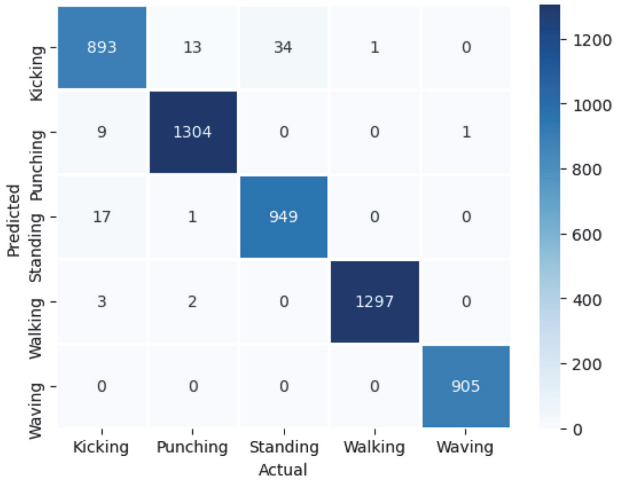

Fig. 12. Confusion matrix for RF classification.

5) Adaptive Boosting: AdaBoost is a statistical classification meta-algorithm that makes 'n' number of decision trees during the data training period. As the first decision tree is made, the incorrectly classified record in the first model is given priority. Only these records are sent as input for the second model. The process goes on until we specify several base learners we want to create. The average accuracy of AdaBoost algorithm reached 97.36%, which is also a remarkable performance, and its confusion matrix is shown in Fig. 11.

6) Random Forest: The RF algorithm creates DTs trained on the data flows and then aggregates the different results to predict the class. In the RF implementation, we consider 10 trees to maintain reasonable execution time and accuracy. We also set entropy as a parameter that yields the best results. The average classification accuracy of RF reached 98.58%, which is a remarkable performance with its many DTs improving class prediction and confusion matrix is shown in Fig. 12.

7) MultiLayer Perceptron: MLP is a feed forward artificial neural network classification algorithm that generates a set of outputs from a set of inputs. In MLP implementation, we consider rectified linear as activation function, adam (stochastic gradient-based optimizer) as optimizer since we have thousands of training samples, and the number of epochs is considered as 100. The average accuracy of the MLP algorithm reached 97.35% which is a remarkable performance and confusion matrix for the same is shown in Fig. 13.

8) Linear Discriminant Analysis: LDA is one of the most popular dimensionality reduction techniques used for supervised classification problems in machine learning. It is used to project the features in higher dimension

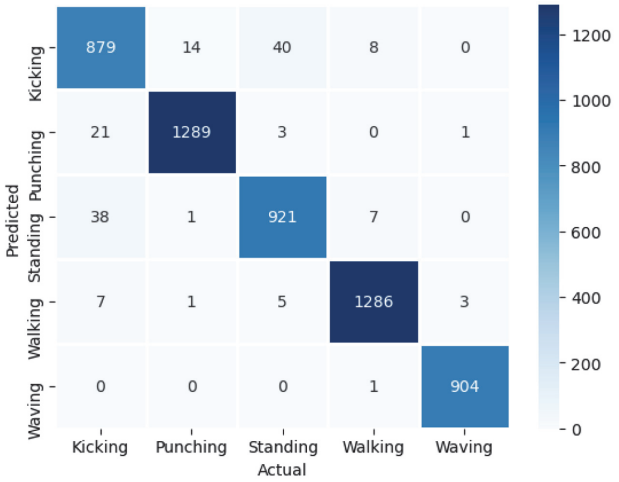

Fig. 11. Confusion matrix for AdaBoost classification.

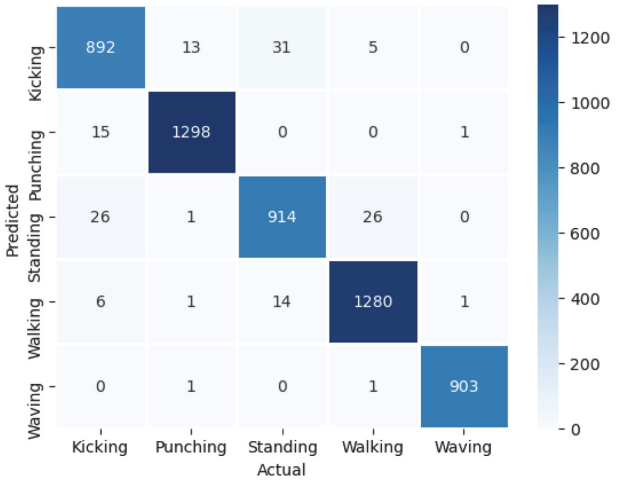

Fig. 13. Confusion matrix for MLP classification.

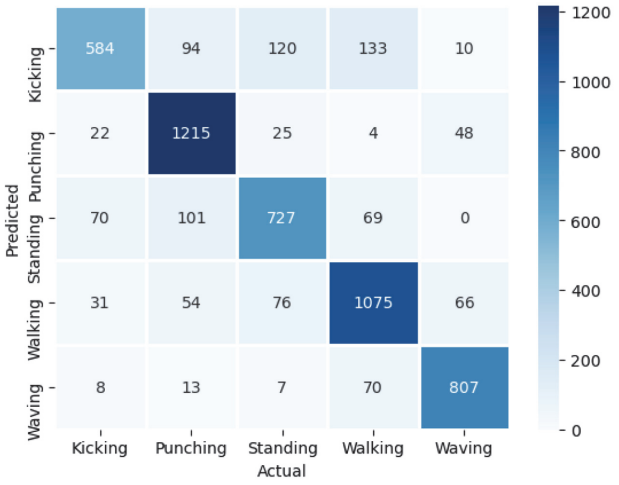

Fig. 14. Confusion matrix for LDA classification.

Fig. 16. Confusion matrix for XGBoost classification.

space into a lower dimension space. The average accuracy of LDA algorithm reached 81.19% and confusion matrix is shown in Fig. 14.

***9) Light Gradient Boosting Machine*:** LGBM is a gradient boosting framework based on decision trees, used for classification, increasing the efficiency of the model and many other machine learning tasks. The average accuracy of LGBM algorithm reached 98.80% that is the highest among all other algorithms showing high measured values for all the classes and confusion matrix is also shown in Fig. 15.

10) eXtreme Gradient Boosting Machine: XGBoost is a scalable, distributed gradient boosted decision tree machine learning library. It provides parallel tree boosting and is the leading machine learning library for

Fig. 15. Confusion matrix for LGBM classification.

classification problems. In this implementation, boosting is done by using gbtree and we consider 100 trees to maintain accuracy and execution time. The average accuracy of the model is 98.67% that is second highest among all other algorithms, and confusion matrix is also shown in Fig. 16.

The experiments conducted on custom dataset using different classification algorithms have shown remarkable performance. By analysing the overall results, we have investigated that the recognition results have best performance with LGBM and RF (using cross-validation) in terms of accuracy, precision, recall and f1-score.

VI. CONCLUSION

In this paper, we have implemented a human action recognition system for identifying actions performed by humans that were trained and evaluated on our own dataset. The pose estimation approach was utilized to extract essential key points of human body by using 2D openpose library. The extracted key points were then nominated as the features, from which 25 features were selected for the construction of model. Walking, Waving, Standing, Punching, and Kicking are the five actions that were gathered, which were first manually labeled and then identified using solely human skeleton key point dataset. After that, action recognition and classification operations were carried out by evaluating ten machine learning algorithms: KNN, SVM, DT, GNB, AdaBoost, RF, MLP, LDA, LGBM and XGBoost, out of which LGBM algorithm provided the highest accuracy (98.80%). In contrast, the GNB algorithm provided the lowest average accuracy (47.89%).

In the future, we will work to improve the outcomes by adding more layers and fine-tuning. To gain a better insight of the complicated dataset for both single and

multiple persons, future work will examine a noticeably more varied collection of action categories.

ACKNOWLEDGMENTS

The authors are grateful to the All-India Council for Technical Education (AICTE) for GATE PG Fellowship Scheme. The authors are also thankful to Amit Kumar, Deepali Sharma, Monika Rasodey and Divyanshi Chhabra (Research scholars at University Institute of Engineering and Technology, Panjab University, Chandigarh) for capturing the videos that had later been used for making dataset.

REFERENCES

1. L. Song, G. Yu, J. Yuan, Z. L.-J. of V. C. and Image, and undefined 2021, "Human pose estimation and its application to action recognition: A survey," *Elsevier*, Accessed: Jul. 12, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320321000262?casa_token=dh6RDowi1pMAAAAA:VRNG3aEeLpAylYmtDLkpCB_LLVIpCAOccN2Z1ZnuvPscJusvnCP4ZdTQrXzhJHD6P-IxjAzHTcBs
2. Y. Chen, W. Ke, K. H. Chan, and Z. Xiong, "A human activity recognition approach based on skeleton extraction and image reconstruction," in *ACM International Conference Proceeding Series*, Jun. 2021, pp. 1–8. doi: 10.1145/3474906.3474909.
3. A. Singh, D. Patil, and S. N. Omkar, "Eye in the sky: Real-time drone surveillance system (DSS) for violent individuals identification using scatternet hybrid deep learning network," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2018-June, pp. 1710–1718, 2018, doi: 10.1109/CVPRW.2018.00214.
4. Z. Cao *et al.*, "OpenPose : Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," vol. XXX, no. XXX, 2019, doi: 10.1109/TPAMI.2019.2929257.
5. A. Rao and A. T. Data, "Efficient Min-Cost Real Time Action Recognition using Pose Estimates," pp. 1–6, 2020.
6. G. Hidalgo *et al.*, "Single-Network Whole-Body Pose Estimation".
7. S. Ghazal and U. S. Khan, "Human Posture Classification Using Skeleton Information," pp. 1–4, 2018.
8. D. Osokin, "Real-time 2D multi-person pose estimation on CPU: Lightweight OpenPose," *ICPRAM 2019 - Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, pp. 744–748, 2019, doi: 10.5220/0007555407440748.
9. W. Sultani and M. Shah, "Human action recognition in drone videos using a few aerial training examples," *Comput. Vis. Image Underst.*, vol. 206, no. September 2020, p. 103186, 2021, doi: 10.1016/j.cviu.2021.103186.
10. A. S. Dileep, S. S. Nabilah, S. Sreeju, K. Farhana, and S. Surumy, "Suspicious Human Activity Recognition using 2D Pose Estimation and Convolutional Neural Network," pp. 19–23.
11. C. Lugaresi *et al.*, "MediaPipe: A Framework for Building Perception Pipelines".
12. R. Nale, M. Sawarbandhe, N. Chegogoju, and V. Satpute, "Suspicious Human Activity Detection Using Pose Estimation and LSTM," no. September, pp. 197–202, 2021.
13. R. Kumar, T. Anand, S. Jalal, and S. C. Agrawal, "Suspicious human activity recognition : a review," *Artif. Intell. Rev.*, vol. 50, no. 2, pp. 283–339, 2018, doi: 10.1007/s10462-017-9545-7.
14. M. Nasr, "Realtime Multi-Person 2D Pose Estimation," no. June, 2020, doi: 10.35444/IJANA.2020.11069.
15. C. Wang, F. Zhang, and S. Sam, "Engineering Applications of Artificial Intelligence A comprehensive survey on 2D multi-person pose estimation methods," *Eng. Appl. Artif. Intell.*, vol. 102, no. March, p. 104260, 2021, doi: 10.1016/j.engappai.2021.104260.
16. A. Gupta, K. Gupta, K. Gupta, and K. Gupta, "Human Activity Recognition Using Pose Estimation and Machine Learning Algorithm," pp. 323–330, 2021.

Monika Dhiman

Monika Dhiman is a student in master course at University Institute of Engineering and Technology, Panjab University, India. She received her bachelor's degree in Computer Science and Engineering from Himachal Pradesh Technical University, Hamirpur, India in 2020. Her research interests include action recognition in surveillance, machine learning and pattern recognition.

Akash Sharma

Akash Sharma is a student in master course at University Institute of Engineering and Technology, Panjab University, India. He received his bachelor's degree in Computer Science and Engineering from Chandigarh University, Punjab, India in 2019. His research interests include computer vision, action recognition in surveillance, machine learning and cyber security.

Sarbjeet Singh

Sarbjeet Singh is a professor at University Institute of Engineering and Technology, Panjab University, India. He received his B.Tech degree in Computer Science and Engineering from Punjab Technical University, Jalandhar, India, in 2001 and the M.E. and Ph.D. degrees in Computer Science and Engineering from Thapar University, Patiala, India, in 2003 and 2009 respectively. His research area includes Machine Learning, Deep Learning, Object Detection, Activity Recognition, Cloud Computing, Social Network Analysis and Sentiment Analysis.