2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE) | 978-1-6654-9260-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IITCEE57236.2023.10090463

Correction and Estimation of Workout Postures with Pose Estimation using AI

Tanmay Hande,
G.H. Raisoni College of Engineering, Nagpur, India
tanmayhandeofficial@gmail.com

Bhargavi Kakirwar
G.H. Raisoni College of Engineering, Nagpur, India
bhargavikakirwar @gmail.com

Bharadwaja. A. V
Godavari institute of engineering and technology Andhrapradesh, India
, bharadwaja.akondi@gmail.com

Pravin Kshirsagar,
G.H. Raisoni College of Engineering, Nagpur, India
pravinrk88@yahoo.com

Aaditya Gupta
G.H. Raisoni College of Engineering, Nagpur, India
aadityag1000@gmail.com

P. Vijayakumar
Vellore Institute of Technology, Chennai, India
vijayrgcet@@gmail.com

***Abstract*— Strength workouts are effective and popular ways to achieve health benefits, but they can cause injury for newcomers if performed incorrectly without prior knowledge. The purpose of this research is to examine how the recent developments in pose recognition, estimation and correction can be used to estimate workout postures and provide valuable feedback on workout techniques to detect specific technique issues associated with a high risk of injury for common exercises. To provide a user with feedback, action recognition will be responsible for collecting, labeling, and organizing the data, as well as training and integrating with real-time data. Using a validation dataset of 218 workout images from the Penn Action dataset as a validation set, our best model scored 97.25 % accurate. The results of this research proves that the pose recognition, estimation and correction algorithm is accurate and can yield useful feedbacks when it comes to estimate workout techniques.**

***Keywords*—** *Computer Vision, Deep Convolutional Networks, Posture Analysis, Workout Recognition*

I. INTRODUCTION

The demands on people's material well-being will progressively increase as the times in today's society advance. People started pursuing healthy lifestyles at the same time, which mostly entail physical activity. Since the implementation of COVID, incorrect forms have grown to be an increasingly serious problem [26]. Since gyms are hubs for the spread of viruses, many novices choose to begin their fitness adventures at home. However, learning how to exercise properly can be a challenge without feedback from a personal trainer and little to no prior experience.

While several exercises, like deadlifts, squats, and shoulder presses, are beneficial to human physical fitness, they may also be seriously damaging if carried out incorrectly. These exercise's excessive weights might result in injuries to the muscles or ligaments. Many people do not hold the proper posture while consistently executing these exercises due to a lack of training or information. Muscle tension and fatigue might result from this.

The significant accomplishments attained by several computer vision and machine learning models served as the inspiration for this effort [27]. Particularly with its key-point challenge outcomes, posture estimation has offered a multitude of alternatives. In 2020, Xiong et al. [1] worked on reliable vision-based workout analysis. The effectiveness of their suggested 3D posture evaluation over earlier ones is demonstrated by test results. It recognizes improper motions but does not time these actions, which prevents it from giving consumers timely feedback. They were unable to incorporate video tutorials into their business. In 2019, Yadav et al. [2] used deep learning algorithms to tackle the challenge of correctly identifying different yoga positions. Six yoga asanas were collected into a dataset by 15 people. For yoga detection in real-time videos, a hybrid deep learning model was developed using CNN and LSTM; the system may be installed on a portable device for real-time predictions and self-training.

In 2019, Yiwen Gu et al. [3] used interactive computer vision systems to develop deep learning models for human position assessment and home-based physical rehabilitation. They were unable to create an algorithm that delivers more specific input on how the patient is doing rather than feedback based on overall performance or offer people a side-view choice. By separating the body shape, skeleton, dominant axes, and points, Chen et al. [4] proposed a system in 2018 that analyses the practitioner's stance from both the front and side perspectives. The methods of feature point detection and assistant axis generation for some poses can be improved or even completely redesigned to strengthen the system.

An algorithm was developed by Chen and Yang [5] to correct user's postures by producing the best workout motions. For posture estimation, Deep Neural Networks and OpenPose were employed, and machine learning and heuristic-based models were used to compare performance. The created application can only be accessed online and is supported by Windows and Linux machines with GPUs. Keshari [6] has employed SVM and RCNN for error detection and OpenCV for image processing. To find flaws, they created their own dataset, distributed it to understand optimal posture, and then used SVM and RCNN to identify incorrect posture.

Using deep learning and OpenCV, Nagarkoti et al. [7] use a trainer's video that has already been captured. Dynamic time warping is employed to synchronize the motions of trainers and users, while optical flow tracking is used to track user's movements [28]. Only the user's posture is corrected; a suitable AI assistant for the user should also keep track of workout repetitions, mistakes, and report creation. Yoga positions were identified using a variety of classification algorithms by Agrawal et al. [8], with random forest classifiers producing the best outcomes. If the accuracy is calculated, the user will be able to track and improve performance. It detects and recognizes various yoga poses using this application.

Z. Cao carried out single-person and multi-person pose estimates [9]. They draw inferences for a single individual using local observations of body components and their spatial relationships. They have employed a top-down approach to recognize persons in multi-person scenarios before separately estimating each person's stance. Videos cannot be utilized with it; it only functions on photos. On the client's ongoing and recorded sessions, Kumar et al. [10] suggested using OpenPose to separate the joint areas using Part Confidence Maps and Part Affinity Fields. The user is then given feedback depending on the variance in angles. These only functions when the user provides images; real-time operation is not possible.

To get the important points, Chiddarwar et al. [11] gather a single ideal picture, which they then locally store on their system. The 17 crucial key points are then predicted by OpenCV using a pre-trained model, and the distance between each body component is determined using Euclidean distance. The software just recognizes and verifies the validity of the yoga stance. By creating a model that recognizes the user's stance while exercising, offers feedback, and proposes modifications as needed, we want to assist people in performing activities with proper posture. With this study, we hope to create a tool that will teach people how to do these lifts correctly and correct their own form. Pose estimation is used to recognize people in videos, and by going a step further, we can map out body joints and connect them to create skeletal representations [29].

The rest of this paper is structured as follows: Section II describes the procedures and materials used in the proposed methodology. Section III discusses the findings of the proposed investigation. Section IV discusses the research's findings and future directions.

II. DESIGN METHODOLOGY

A. *Dataset Gathering*

The Penn Action Dataset from the University of Pennsylvania contains 2326 photos or video sequences of 10 different activities, together with human joint comments about each sequence. It was essential to reach the proper angles as precisely as possible during recording in order to prevent excessive camera movement inaccuracies. The two viewpoints that will be used during filming were chosen based on the front and side views. The full person should be seen, and there should be space for safety around all corners. The athlete's full body need should be displayed in the center of the screen on the right side of the item, which is the Side View. The torso of the athlete should be angled so that their right side covers their left. Each video clip consists of a short narrative in which the person does the exercise while utilizing or omitting a certain method, as viewed from one perspective at a time. After completing one repetition of the workout, the user returns to the starting place where the video clip began. Each video clip is a small vignette that shows the individual performing the exercise while utilizing or without employing a certain method, from one perspective at a time. After completing one repetition of the workout, the user returns to the starting place where the video clip began.

Every participant completed every exercise successfully from one of the two suggested perspectives. This has to be done so that a subset of exercises that have been successfully completed may be evaluated similarly to how the other films are evaluated. Most commonly, this is done to expose false positives caused by faulty vector calculations. The system shouldn't flag any technique-related issues in addition to offering a baseline set for comparison.

While a small sample of athletes from both sexes may not account for all potential variations among athletes, it does account for enough factors to provide the model a lot of leeway. Each exercise was successfully done by every participant from both of the suggested angles. This will give a selection of exercises that have been successfully performed and may be evaluated similarly to the other films. The only discernible person in the photograph is the person working out.

B. *Data Preprocessing*

C. The material needed to undergo data filtering, which was the final and by far the most important data processing step. A low accuracy probability data point is removed using this method. In order to accomplish this, points forecast to be incorrect by human posture assessment techniques are removed along with points with excessively high variability. The precision of the result is entirely determined by this process. As a result of eliminating the potentially inaccurate estimated points, it is less likely that method problems will be found that don't exist. As a result, it is an essential step in avoiding false positives.

Each method provides a confidence score in addition to the data points indicating the likelihood of each guess being correct. By disregarding crucial details with a low level of confidence, inaccurate estimations can be weeded out quickly and easily. These videos have a higher chance of giving an accurate estimate because they were shot in a controlled environment. The action recognition and technique evaluation tasks were not solved when confidence ratings were less than 70%. As a result of the filter being set to exclude certain estimated points with uncertainty, yet still keep the most important points, 70% of the estimated points were allowed to pass. It is possible to discover method elements even if key points for some frames are ignored since the dataset has data coordinates for each action extracted from the video frame.

We adjusted various calculations to remove noise that might cause false positives by increasing the confidence score threshold to 90%. Side view important points had substantially lower confidence scores than the rest of the data. Therefore, all films produced from a side-on perspective will have to meet a 60% confidence score threshold.

D. ***Proposed Model***

In general, the system generates a table of identified technical problems for a workout video that is provided by the user. OpenPose, AlphaPose, or WrnchAl process the input video in the Pose Extraction System. This research compares the efficiency of each system in identifying technique problems during exercise using videos of each video. Document databases receive the processed data from different human pose estimation algorithms.

A classification neural model called MobileNet or InceptionV3 is used to identify the exercise completed and the camera angle after the important information is taken from the database by the Action Recognition System. A neural network model with Inception-V3 neural network, dynamic temporal warping, and MobileNet is used to categorize the exercise. A document database is then created and populated with the categorization outcome and relevant key points. The Technique Evaluation System culminates the process by downloading all relevant data from the database and calculating specialized vectors based on the anticipated workout and detection angle. It is then prepared to be shown to the user alongside the identified workout and the filming angle based on the technique analysis that gets saved in the database as shown in figure I.

This system's input is a raw, unedited, and unfiltered fitness video. First, the workout footage is analyzed by OpenPose, a human posture estimate algorithm. The resultant dataset is then purified of extraneous information, converted to a common format, and checked for erroneous estimates. The key points that are produced are then indexed and kept in the database.

The Action Recognition system begins by removing all significant points from the document database that have been processed. Additionally, each key point is analyzed as a time series to identify the exercise that was completed and the camera angle. Prior to doing the exercise detection and filming angle detection individually, data normalization and noise filtering are undertaken. Then, based on the anticipated workout and shooting angle, a selection of pertinent vector formulae is chosen. To determine if there are any methodological problems in the provided pose estimation dataset, all selected vector formulae are then computed. The result is a list of all technique flaws found in the dataset. An empty list is returned if none were discovered.

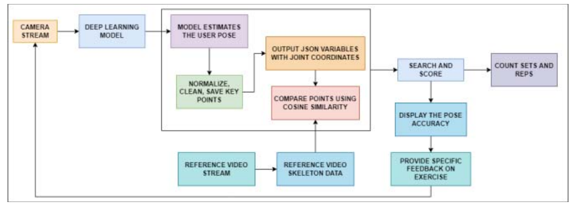

Fig I. Architecture for proposed Workout Detection and Classification model

MobileNet consists of two types of deep level convolutions: depth level convolutions and point level convolutions. The depth- and point-wise convolutions make up 28 layers in a MobileNet, but by adjusting the width multiplier hyper parameter, the 4.2 million parameters are further reduced. The preprocessed input image is 224 x 224 by x 3, in size.

This model is based on depth and pointwise separable convolutions, transforming a conventional convolution into a depth and pointwise convolution, known as depth and pointwise convolution [26]. With MobileNet depth-wise convolution, every input channel is treated to a single filter followed by a 1x1 convolution that merges the depth- and point-wise convolutions. By combining inputs and outputs in one step, a regular convolution produces a new set of outputs. By using depth-wise separable convolution, we separate this into two layers for filtering and merging. The size of the model and the computation are greatly reduced as a result of this factorization.

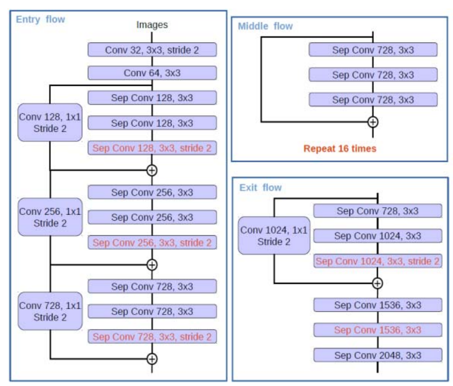

Fig II. Architecture for proposed MobileNet Model

A MobileNet structure, as opposed to the first layer, is composed of depth-wise separable convolution. The last fully connected layer in MobileNet does not have nonlinearity, but feeds into a SoftMax layer for classification, which has batch norms and ReLU nonlinearity.

Figure II compares the factorized layer with depth-wise convolution, pointwise convolution, batch norm, and ReLU nonlinearity with the layer with conventional convolutions, batch norm, and ReLU nonlinearity after each convolutional layer. The first layer uses stride convolution to control down sampling, as do the depth-wise convolutions. The spatial resolution is reduced to 1 before the fully connected layer by a last average pooling. If depth- and point-wise convolutions are regarded as separate layers, MobileNet has 28 layers.

For instance, formless light matrix operations are frequently slower than thick matrix operations up until a very high level of sparsity. Dense convolutions do practically of the work in this model construction. This may be accomplished by using very effective generic matrix multiply (GEMM) methods. A GEMM typically implements convolutions; however, memory must first be reorganized in order to move a convolution to a GEMM. This method is utilized by the popular Caffe algorithm. This memory reordering is not necessary when using one of the most sophisticated numerical linear algebra methods, GEMM, to execute 1x1 convolutions directly.

Using TensorFlow, MobileNet models were trained using RMSprop and gradient descent asynchronously, similar to Inception V3. Rather than training large models, small models should be trained with less regularization and data augmentation techniques. For massive Inception training, we reduce distortions by using side heads or by smoothing labels. We also restrict the size of micro crops. Furthermore, we found that it is necessary to give depth wise filters very little to no weight decay because they have so few parameters.

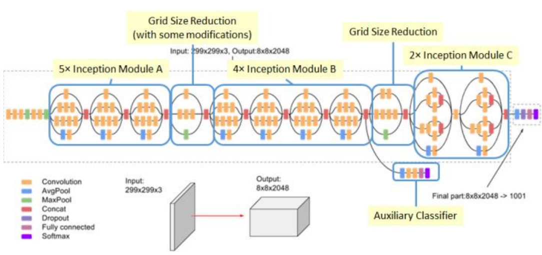

Fig III. Architecture for proposed InceptionV3 model

The training comes to an end once 10 different learning rates—0.01 for the first hundred, 0.001 for the next hundreds, 0.0001 for the next hundreds, and so forth—have been used. This process is repeated until 1000 epochs have been finished. We consider a weight loss of 0.00001 pounds. Additionally, in this study, the performance of SGD was compared to that of the training algorithms Adagrad, Adadelta, AdamW, and Adamax.

The dataset consists of data from multiple subjects. We split the data into train and test sets based on subjects. Subject-wise split ensures that the specific characteristics of a test subject do not leak into the training set and the performance on the test set can be generalized to any new subject. The suggested architecture's efficiency is estimated using 10-fold cross validation. As the system accuracy stabilizes after just 800 to 1000 epochs, 100 epochs are considered for training in each fold.

Our dataset includes information from a variety of disciplines. We categorized the data into train and test groups based on the topics. The performance on the test set may be generalized to any new subject thanks to the subject-wise split, which prevents the special traits of a test subject from leaking into the training set. The effectiveness of the proposed design is calculated using 10-fold cross validation. Only 800 to 1000 epochs are needed for the system accuracy to settle, hence 100 epochs are taken into account for training in each fold. We execute 698 training iterations before ending validation and store the model weights that resulted in the smallest validation set loss. Note that we split our data into multiple smaller batches since it was too large to transmit to the network all at once. The data is completed in batches of 500, requiring 6 iterations.

III. Result and Discussion

The performance of the networks is evaluated based on the classification accuracy, precision, specificity, and sensitivity throughout the experimentation. As determined by counting the number of correct predictions to the total number of predictions, classification accuracy is defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where, TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative.

The following formula may be used to measure precision, a mathematical concept that examines the number of genuine true positives to the true positive forecasts.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

The true positive rate is obtained by calculating sensitivity, which is the percentage of true positive cases that is perfectly classified according to its class. The sensitivity can be mathematically written as

$$Sensitivity = \frac{TP}{TP + FN} \quad (3)$$

We also assess specificity by computing true negative values, which measure the percentage of true negative cases that is correctly categorized as based on its class as shown by

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

The training data are divided into 10 different folds of same size so as the model learns effective mapping of inputs and outputs. The fine tuning and hyper-parameter tuning of features are carried out on these folds. The efficiency of this deep learning is tested using various performance metrics. Table I and Table II elaborates the accuracy, specificity, sensitivity, and precision scores across the 10-folds for MobileNet architecture and Inception V3 architecture. The class overlap represents data samples that appear to be valid instances of multiple classes, which could be causing noise in data sets.

The training stops with 10 alternative learning rates, starting with 0.01 for the first hundred, 0.001 for the following hundreds, 0.0001 for the next hundreds, and so on. We repeat this method until we have completed 100 epochs. We take a 0.00001 weight loss into account. Furthermore, in this study we have also compared SGD performance with other training algorithms namely Adagrad, Adadelta, AdamW, and Adamax.

The model is validated and tested in three stages, with training taking up 70% of the time, validation taking up 20%, and testing taking up the remaining 40%. The dataset is made up of information from several disciplines. We split the data into train and test sets based on subjects. Subject-wise split ensures that the specific characteristics of a test subject do not leak into the training set and the performance on the test set can be generalized to any new subject. The suggested architecture's efficiency is estimated using 10-fold cross validation. As the system accuracy stabilizes after just 800 to 1000 epochs, 100 epochs are considered for training in each fold.

We execute 698 training iterations before ending validation and store the model weights that resulted in the smallest validation set loss. Note that we split our data into multiple smaller batches since it was too large to transmit to the network all at once. The data is completed in batches of 500, requiring 6 iterations. Training is stopped early if validation loss stops decreasing for a sustained period of time using early stopping.

Key point data from each of the three posture estimate algorithms, employing all 82 movies, made up the assessment dataset. There is a related video of each user performing the exercises correctly for each perspective and each exercise. Only 5 videos were taken in this category since, as was already noted, not all users could do a squat with a knee extension on purpose. There are 252 total files utilized in this assessment, or 82 files for each pose estimate technique.

Accurately predicting the filming angle and carrying out exercise were the objectives of the Action Recognition System. This included the two sub-tasks of workout detection and angle detection. Both Table I and Table II separately display the outcomes of each. These squats were captured on video from the side, which by nature included key points with poor confidence ratings. The remaining 247 files were properly predicted, yielding a quite good accuracy. Only OpenPose had accurate predictions for every one of the 84 videos.

Folds	Performance Metrics (MobileNet)			
	Specificity	Sensitivity	Accuracy	Precision
Fold-I	98.12	98.17	98.17	98.45
Fold-II	97.24	98.36	98.21	97.41
Fold-III	98.35	97.54	97.27	97.52
Fold-IV	97.46	97.72	97.35	97.36
Fold-V	98.75	97.93	98.74	98.47
Fold-VI	97.52	98.11	98.45	97.69
Fold-VII	98.63	97.13	97.19	96.14
Fold-VIII	97.87	97.74	97.73	97.25
Fold-IX	98.20	97.25	98.46	96.36
Fold-X	97.07	98.36	98.18	96.42
Overlapped Data	NULL	NULL	NULL	NULL
Heat Map Features	98.09	97.14	98.19	98.26
Average	98.11	97.42	98.37	97.75

Table I : Performance Metrics for MobileNet Model

Folds	Performance Metrics (InceptionV3)			
	Specificity	Sensitivity	Accuracy	Precision
Fold-I	97.22	97.19	96.19	96.48
Fold-II	96.17	96.30	96.25	96.46
Fold-III	96.48	96.52	96.24	96.56
Fold-IV	96.54	96.72	96.35	96.38
Fold-V	96.81	96.99	96.79	98.45
Fold-VI	96.89	96.19	96.46	96.69
Fold-VII	96.19	96.17	96.18	96.19
Fold-VIII	96.13	96.70	96.71	96.25
Fold-IX	96.16	96.27	96.41	96.31
Fold-X	97.74	96.36	96.11	96.46
Overlapped Data	NULL	NULL	NULL	NULL
Heat Map Features	96.37	96.25	96.20	96.22
Average	96.17	96.35	96.34	96.24

Table II: Performance Metrics for CNN-LSTM Model

Optimizer	30 Epochs	75 Epochs	150 Epochs	300 Epochs
SGD	17.68 ± 0.19	14.63± 0.12	11.04± 0.21	14.63± 0.12
Adam	17.88 ± 0.17	14.45 ± 0.06	11.07 ± 0.14	14.45 ± 0.06
Adadelta	16.49 ± 0.16	12.37 ± 0.15	11.14 ± 0.16	12.37 ± 0.15
Adagrad	16.51 ± 0.4	15.41 ± 0.20	11.06 ± 0.18	15.41 ± 0.20
Adamw	18.35 ± 0.12	13.67 ± 0.17	11.12 ± 0.10	13.67 ± 0.17
Adamax	10.65 ± 0.05	8.86 ± 0.19	8.02 ± 0.21	8.86 ± 0.19

TABLE III: Performance of Different Optimizers on Train and Test Data

The system may have been examined more accurately by altering some procedures. Since various methodologies are being used in this study, a much larger dataset, which includes more individuals and even more videos, would be needed to adequately address the research question. It is fortunately apparent that the findings indicate that there is a good resolution to the problem. However, the study's objective required that this software be accepted by the market for consumers to use it. Using a 2-dimensional video, we determine whether the user is following good form during his or her workouts by looking at the key points. A weightlifting exercise must be performed with the correct technique and target the right muscles.

According to the study, two dimensional pose detection and estimation is a useful tool in reducing injury risk when used on healthy individuals when seen from the front. For more precise identification of whether front view success may be extended to side views, more research is needed, as well as optimizations to the detection technique. For these sorts of technique characteristics, dynamic time warping could also be a more appropriate solution than Pose Trainer results for side viewing angles.

The algorithm is nevertheless able to give some relevant findings in this situation even if rotation-intensive activities seem to be more challenging to recognize. When comparing these results to the Pose Trainer, it appears that the current method produces results that are comparable but with a bigger dataset and more accurate feedback on exercises that are more difficult. This study builds on previous work on Pose Trainer and firmly demonstrates how basic approaches combined with crucial aspects from 2D Human Pose Estimation may provide feedback on weightlifting form.

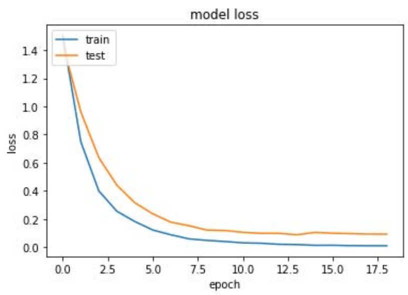

Fig IV. Loss Model for MobileNet Architecture

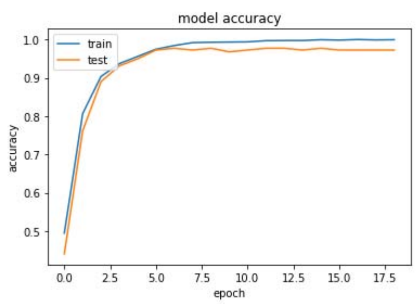

Fig V. Accuracy Model for MobileNet Architecture

Pose Trainer compared the outcomes according to the motions that were executed correctly or incorrectly. Since they are provided separately for each technique element in this study, the results are difficult to compare to one another.

The machine learning technique used by Pose Trainer produced positive results, with front raises being accurately identified as either correct or wrong for all exercises. The results from the other activities were likewise encouraging, with an F1 score of about 0.8. But the machine learning approach is the only source of this data. With the exception of the bicep curls, which could detect 80% of incorrect executions, none of the results of the geometric technique utilized were made public. In comparison to side view detection, side view evaluation in this study produced noticeably lower ratings for the shoulder press, front raise, and bicep curls. This implies that a technique that takes advantage of the dynamic distortion of the system would be more advantageous in this case.

The angle prediction is the first significant step in evaluating the training videos; if it is inaccurate, the system's other components will also provide inaccurate findings. The algorithm was able to correctly forecast every viewing angle by employing a distance vector between the two shoulders. Independent of the user's movement, the space between a person's shoulders is relatively constant. With the same distribution of front and side angle of view, it follows that identical results would be found with bigger datasets.

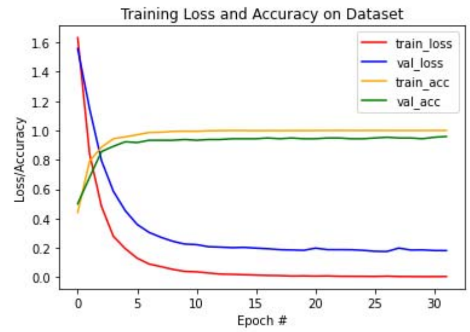

Fig VI. Accuracy Model for InceptionV3 Architecture

In order to attempt to address the study topic, two physical systems were chosen, along with learning more about various Human Pose Estimation Systems and investigating the danger associated with weight training. In order to automatically check for technique elements related to the recognized activity and shooting angle and for evaluating technique, an action recognition system can be implemented that is capable of recognizing the exercise and shooting angle of a video automatically to pinpoint its elements that are strongly associated with injury risk. Hence warning users when they are engaging in exercise in a way that puts them at danger of harm.

The ability of the angle detector to identify angles that aren't directly frontal or oblique hasn't been tested. Angles that don't match the vector calculations cannot be ruled out by the system. In the detection technique, shoulders are not normalized, so someone standing extremely far away from the camera and looking forward may have a projected shoulder position that is too close to the minimum for the algorithm to detect and accurately forecast that they are being videotaped from the side. An independent threshold value was used to compare the average shoulder width to a predefined threshold.

IV. CONCLUSION

This study used human pose estimation to identify specific methodology issues in everyday workouts. According to the research, the methodology can be used to evaluate a wide variety of fitness training methods for a wide range of users, exercises, and technique problems. In order to do this, several methods of human pose estimation were evaluated on a limited number of individuals. Pose estimation is capable of detecting technique problems in weightlifting considering every dimensional view. The algorithm performs considerably better estimating the video frame from the frontal view rather than from the side, due to the pose estimation tool's inaccuracy in pose detection and angle calculation on side view videos.

Even though the results can't be generalized due to the small number of data samples, this methodology offers a modern perspective by comparing the similar methodology for users with varying body types and body pose estimation techniques. Variances may be taken into consideration and generalized in order to make accurate deductions about weightlifting technique. Prior research in the field has mostly concentrated on using depth cameras or several sensors to gather data in three dimensions. In order to capture the subject's position for this research, only a single RGB camera was utilized. This led to a more accessible service that, in theory, needs nothing more than a user's own smartphone camera.

V. References

[1] H. Xiong, S. Berkovsky, R. V. Sharan, S. Liu, and E. Coiera, "Robust vision-based workout analysis using diversified deep latent variable model," in 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), pp. 2155–2158, 2020.

[2] S. Yadav, A. Singh, A. Gupta, and J. Raheja, "Real-time yoga recognition using deep learning," Neural Computing and Applications, vol. 31, pp. https://link.springer.com/article/10.1007/s00521–019, 12 2019.

[3] Y. Gu, S. Pandit, E. Saraee, T. Nordahl, T. Ellis, and M. Betke, "Home-based physical therapy with an interactive computer vision system," in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 2619–2628, 2019.

[4] C. Huang, Y.-Z. He, and C.-C. Hsu, "Computer-assisted yoga training system," Multimedia Tools and Applications, vol. 77, 09 2018.

[5] S. Chen and R. Yang, "Pose trainer: Correcting exercise posture using pose estimation," 03 2018.

[6] P. Keshari, "Wrong posture detection using opencv and support vector machine," 01 2020.

[7] A. Nagarkoti, R. Teotia, A. K. Mahale, and P. K. Das, "Realtime indoor workout analysis using machine learning amp; computer vision," in 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1440–1443, 2019.

[8] Y. Agrawal, Y. Shah, and A. Sharma, "Implementation of machine learning technique for identification of yoga poses," in 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), pp. 40–43, 2020.

[9] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "Openpose: Realtime multi person 2d pose estimation using part affinity fields," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 43, pp. 172–186, jan 2021.

[10] D. Kumar and A. Sinha, "Yoga pose detection and classification using deep learning," International Journal of Scientific Research in Computer Science Engineering and Information Technology, 11 2020.

[11] G. Chiddarwar, A. Ranjane, M. Chindhe, R. Deodhar, and P. Gangamwar, "Aibased yoga pose estimation for android application," International Journal of Innovative Science and Research Technology, vol. 5, pp. 1070–1073, 10 2020.

[12] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learning based 2d human pose estimation: A survey," Tsinghua Science and Technology, vol. 24, no. 6, pp. 663– 676, 2019.

[13] F. Sajjad, A. F. Ahmed, and M. A. Ahmed, "A study on the learning based human pose recognition," in 2017 9th IEEE-GCC Conference and Exhibition (GCCCE), pp. 1–8, 2017.

[14] H.-T. Chen, Y.-Z. He, and C.-C. Hsu, "Computer-assisted yoga training system," Multimedia Tools and Applications, vol. 77, no. 18, pp. 23969–23991, 2018.

[15] S. Jain, A. Rustagi, S. Saurav, R. Saini, and S. Singh, "Three dimensional CNN-inspired deep learning architecture for Yoga pose recognition in the real-world environment," Neural Computational Intelligence and Neuroscience, Computing & Applications, vol. 33, no. 12, pp. 6427–6441, 2021.

[16] Mais Yasen and Shaidah Jusoh. "A systematic review on hand gesture recognition techniques, challenges and applications". In: PeerJ Computer Science 5 (2019), e218

[17] H. Tang, Q. Wang and H. Chen, "Research on 3D Human Pose Estimation Using RGBD Camera," 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2019, pp. 538-541, doi: 10.1109/ICEIEC.2019.8784591.

[18] G.R.S. Murthy and R.s Jadon. "A review of vision based hand gesture recognition". In: International Journal of Information Technology and Knowledge Management 2 (Aug. 2), pp. 405–410

[19] Santiago Riofrio et al. "Gesture Recognition Using Dynamic Time Warping and Kinect: A Practical Approach". In: Nov. 2017, pp. 302–308. doi: 10 . 1109/INCISCOS.2017.36. .

[20] Zhe Cao et al. "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields". In: IEEE Transactions on Pattern Analysis and Machine Intelligence 43.1 (2021), pp. 172–186. doi: 10.1109/TPAMI.2019.2929257.

[21] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang and C. Yang, "The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation," in IEEE Access, vol. 8, pp. 133330-133348, 2020, doi: 10.1109/ACCESS.2020.3010248.

[22] Y. Cheng, P. Yi, R. Liu, J. Dong, D. Zhou and Q. Zhang, "Human-robot Interaction Method Combining Human Pose Estimation and Motion Intention Recognition," 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2021, pp. 958-963, doi: 10.1109/CSCWD49262.2021.9437772.

[23] Imam Riadi, Sunardi Sunardi, and Arizona Firmansyah. "Forensic Investigation Technique on Android's Blackberry Messenger using NIST Framework". In: International Journal of Cyber-Security and Digital Forensics 6 (Oct. 2017), pp. 198–205.

[24] Henrik Sj¨oberg et al. "Content Validity Index and Reliability of a New Protocol for Evaluation of Lifting Technique in the Powerlifting Squat and Deadlift". In: Journal of Strength and Conditioning Research (Sept. 2018). doi: 10.1519/ JSC.0000000000002791.

[25] Zhe Cao et al. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: (Dec. 2018).

[26] Bajaj, Nidhi & Kshirsagar, Pravin & Akojwar, Sudhir. (2018). A hybridised neural network and optimisation algorithms for prediction and classification of neurological disorders. International Journal of Biomedical Engineering and Technology. 28. 307. 10.1504/IJBET.2018.10017200.

[27] Kshirsagar, Pravin & Akojwar, Sudhir. (2016). Hybrid Heuristic Optimization for Benchmark Datasets. International Journal of Computer Applications. 146. 11-16. 10.5120/ijca2016910853.

[28] Kshirsagar, Pravin & Karve, Swagat & Chipppalkatti, Pranav. (2018). Performance optimization of neural network using GA incorporated PSO. Journal of Advanced Research in Dynamical and Control Systems. 10. 156-169.

[29] Liang, Nanying & Saratchandran, Paramasivan & Huang, Guang-Bin & Sundararajan, Narasimhan. (2006). Classification of Mental Tasks from EEG Signals Using Extreme Learning Machines. International journal of neural systems. 16. 29-38. 10.1142/S0129065706000482.