International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Nov-2022)

https://dx.doi.org/10.12785/ijcds/1201104

Recognition of Yoga Asana from Real-Time Videos using Blaze-pose

Sharfuddin Waseem Mohammed[1], Vignesh Garrapally[2], Suraj Manchala[2], Soora Narasimha Reddy[1] and Santosh Kumar Naligenti[1]

[1]*Professor at Kakatiya Institute of Technology and Science, Warangal, Telangana, India*
[2]*Student at Kakatiya Institute of Technology and Science, Warangal, Telangana,India*

Received 10 Aug. 2021, Revised 28 Jul. 2022, Accepted 29 Nov. 2022, Published 30 Nov. 2022

Abstract: Yoga is a broad concept that connotes union. Considering yoga's spiritual and health benefits, it is now practiced by millions of people worldwide. This paper proposes a lightweight and robust architecture that could recognize yoga asana from video input. Most of the existing techniques use either expensive hardware configuration such as Kinect or specialized feature extraction techniques from raw inputs for each asana. Even though these produce decent accuracy in a controlled environment, they are complex to design and often fail in most real-time cases with complex backgrounds. The problem with the existing asana recognition methods from the literature is that they either demand high-end configurations or do not produce key points while recognition, which is crucial in pose correction employed at a later stage. The proposed model is so computationally efficient that it can be deployed even in entry-level smartphones, browsers, and smart TVs. Pose estimation is done initially using state-of-the-art Blaze Pose architecture. Transformations are applied after that to achieve scale and position independence. convolutional neural networks (CNN) and long-short-term memory (LSTM) networks are being used to train the model from the extracted key points. The CNN network from the novel architecture can be leveraged to extract spatial features, whereas LSTM networks understand the features through time stages. After precise tuning of hyperparameters, our system achieves a training accuracy of 95.29% and a test accuracy of 98.65% at 30 frames per second (FPS). To the best of our knowledge, this is the first computationally efficient work, which processes video input at 30 FPS and achieves decent accuracy compared to existing research works from the literature.

Keywords: Deep Learning, Computer Vision, Blaze Pose, Human Activity Recognition, Yoga, CNN, LSTM, Pose Estimation, Blaze pose, Recurrent Neural Networks

1. Introduction

Yoga is a broad concept that connotes union. It has been one of the spiritual practices in India for 5000 years. Yoga transports you to the absolute reality, where individual manifestations of life are surface bubbles in the creation process. Yoga in Sanskrit translates to the word union. It includes the techniques used to aim for the union of mind, body, and emotional self, with the ultimate objective of achieving transcendence and liberation [1]. Despite being an ancient technology, benefits obtained from yoga are also proved in many advanced studies. Chris C. Streeter et al., [2] contrasted the impact of yoga against walking on mood, anxiety, and brain gamma-Aminobutyric acid (GABA) levels.

The yoga group showed more mood changes and less anxiety than the walking group. Positive associations were observed between mood scales and changes in GABA levels in the yoga community. Another review by Guddeti, Raviteja R. et al., [3] indicates that yoga has been found to have beneficial effects on systemic inflammation, fatigue, cardiac autonomic nervous system, familiar and new cardiovascular risk factors. Exercising yoga produces healthy energy, vital for the immune system to work efficiently. Considering all the benefits of performing yoga by humans, the United Nations general assembly declared 21 June as "International Yoga Day." Another research by Luu et al., [4] showed that acute bursts of Hatha yoga and mindfulness meditation are proven to enhance organizational performance and mood. Catherine Woodyards' et al., [5] study indicates that yogic activities increase muscle strength and body endurance, encourage, and strengthen respiratory and cardiovascular capacity and alcohol rehabilitation, alleviate fatigue, stress, depression, and chronic pain, enhance sleep cycles, and improve sleep general health and quality of life. Most yoga asana involves performing complex movements in a sequence. All benefits could go in vain if one performs the asana incorrectly. This could also result in serious injuries that could even last a lifetime. Hence, while performing asana, it is mandatory to do asana as they were

E-mail address: waseem7602@gmail.com, garrapallyvignesh8055@gmail.com, suraj.manchala@gmail.com, snreddy75@gmail.com,santosh.naliganti@gmail.com

intended to with utmost care. With the growing popularity of yoga, we believe there would be a great advantage if a system exists that correct postures while assisting humans in training the asana correctly. Provided sufficient input data, computer vision (CV), and deep learning (DL) technologies would greatly help build a robust system that recognizes and then corrects postures by comparing them with expert poses, ultimately assisting in training humans in yoga asana. The research aims to develop a classification architecture for videos that can detect the asana being performed by the practitioner in real-time. To achieve this, we initially found the pose estimation of person using Blaze Pose, which was then processed through defined transformation techniques followed by CNN-LSTM architecture to predict the asana. High level design of the proposed architecture can be seen in the Figure 1. The model can also run on any low-end smartphone and personal computer (PC).

Human Activity Recognition (HAR) is a significant CV problem that attempts to distinguish events through a sequence of findings on subject's behavior and environmental factors. The recognition of human activities can support fitness control, monitoring of wellbeing, detecting falls, the consciousness of behavioral contexts, home, and work control, and self-management. Notably, vision-based HAR systems are pretty complex, which take input in the form of video or images and process them to recognize activities performed. A review by Shugang Zhang et al., [6] mentions various works done in the domain of HAR. HAR has been a significant problem within CV and augmented reality (AR) [7]. In the last decade, its applications, including human-machine interactions, intelligent video monitoring, environmentally supported living, entertainment, the interaction between human-robot and intelligent transport systems, have experienced tremendous development [8]. Great benefits also come with significant challenges. Many challenges imposed in vision-based HAR are mentioned in detail in an article by Imen Jegham et al., [9]. The author of the report states that the ambiguities in identifying behavior come from the issue of defining the movement of the part of the body and many other real-life issues, including camera movement, complex context, and bad weather. Research done by us and the results obtained is exhaustively discussed in the coming sections. The rest of the paper is organized as follows: section 2 includes a comprehensive literature survey of the various existing systems designed to recognize yoga asanas. Section 3 covers the details of the dataset that we have used to train our architecture. In section 4, the proposed methodology has been explained. Consequent sections 5, 6 and 7 discuss experimentation results, discussion, and conclusion.

2. Literature Survey

Our literature survey indicates that there are many robust recognition systems for different sports like swimming [10], hurdles racing [11], high jump [12], rugby [13], badminton [14], and basketball [15] [16] but very few for yoga asana. Patil et al., [17] introduced a yoga tutor project to detect a distinction between practitioners and specialists by speeding up robust features (SURF). However, this project cannot approximately describe the postures using only contour information and no critical point information. Luo et al., [18] proposed a yoga training system developed using an interface-suit technique to understand body movement accurately. This methodology consists of 16 inertial measurement units and tactors. However, it varies with the performance of the practitioner. To overcome this problem, Chen et al., [19] presented a yoga activity recognition system that uses features obtained from a Kinect device for extracting a user's body contour. But this system has only 82.84% of accuracy. The system is enhanced by using Kinect and Ada boost classification to increase overall accuracy, which increased the accuracy to 94.78%. However, this system's main problem is that depth sensor-based cameras have been used, which are costly and not open to many users.

Furthermore, Mohanty et al., [20] have developed a technique to solve this problem that identifies images of yoga postures using CNNs and autoencoder (SAE) algorithms. But it only works on still images and not on videos, which is the main drawback of this technique. Chen et al., [21] conducted another research on yoga self-training assistance, which helps correct postures while doing asana. However, it extracts features manually, so individual models must be built for each asana, which wouldn't be feasible considering thousands of asanas in the yogic system. All the above research works use conventional skeletonization approaches, demanding high computational cost. Deep Pose is another technique developed by Toshev et al., [22]. It is unique when compared to the above systems as it predicts the motion of a person by deep NN-based regressors to determine coordinates of joints. It also indicates hidden body parts and can anticipate a person's activity but suffers from the problem that performance will be delayed, making it ineffective for real-time predictions. Heish et al., [23] developed a yoga training system that compares distance variance between practitioner and trainer postures. The proposed method provides a score to the practitioner, indicating how correct the stance is when compared with the trainers. But this system has a problem of computational cost. This problem is then tried to negotiate using Open Pose developed by SK Yadav et al., [24]. This real-time multi-person recognition system jointly detects a human body with 18 critical points on single images using an RGB camera. Despite processing at 0.4 FPS, the significant disadvantages with this system are it requires high computational power and is not optimized for videos making it challenging to work on real-time videos, which generally stream at 24-30 FPS. Ji et al., [25] proposed an architecture using 3D convolutions to extract spatial, temporal features from videos. Shrajal Jain et al., [26] proposed a technique for yoga pose recognition using 3D CNN architecture. Compared to Karpathy et al., [27], this system [26] is slightly modified and implemented for sports action classification. This system is perfect in terms of accuracy and computational efficiency. Still, the major setback

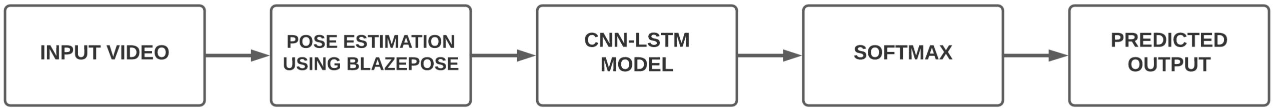

Figure 1. High Level Design

with this approach [26] is key points are not detected, which are crucial in correcting the practitioner pose.

To consolidate, some of the existing approaches [24], [26] available for yoga asana recognition overcame the challenges of specialized hardware sensor requirements, drastically decreasing the implementation cost using DL and machine learning algorithms. Yet, none of the approaches would overcome the challenges of speed and compatibility by maintaining accuracy. We focus on developing an end-to-end architecture that would address both problems. This is the first research work that addressed the performance and compatibility across devices in real-time to the best of our knowledge. We could achieve predictions at 30 FPS in a low-end CPU configuration in real-time. We have used Blaze Pose architecture for pose estimation, easily integrated into mid-range smartphones and entry-level PC browsers.

3. Data Collection

As CV and AI applications in the health and spiritual industry are still in their infancy and few publicly available datasets exist. But we could get access to one publicly available dataset published in research by SK Yadav et al., [24]. The dataset consists of six asanas, where each asana was performed by fifteen people (ten men and five women). Each video in the dataset represents a person performed a specific type of asana. Dataset is labeled for the classification task where each asana is marked to the name of asana being conducted by the practitioner in the video. More details about the dataset can be seen in table I. Yoga asana videos were captured with a Logitech HD 1080p (p means here progressive scan) web camera on a device with an NVIDIA TITAN X GPU and an Intel Xeon processor with 32GB RAM. Most of the yoga poses from the collected dataset are done next to the camera at 4–5 meters. Each asana has been performed in all possible variations by the users. All the videos were shot in an indoor setting with a frame rate of 30 FPS for more than 45 seconds. The cumulative length of 88 training videos is 1 hour, 6 minutes, and 5 seconds, or approximately 111,750 frames. The main problem with the dataset is that asanas aren't performed effectively. They were also recorded in a controlled environment, which increases the latent similarity and quickly makes the model overfitting. In addition, we have chosen not to apply any of the data augmentation techniques to the dataset. We believe augmentation techniques make the model less robust as most of the asanas have slight variances, which would also lead to inconsistency. We were able to download the dataset from [28].

TABLE I. Details of the Yoga Dataset

S.NO	Asana name	No. of Persons	No. of videos
1	Tadasana	15	15
2	Bhujangasana	15	16
3	Trikonasana	13	13
4	Shavasana	15	14
5	Padmasana	14	14
6	Vrikshasana	15	15
Total Number of Videos			**88**

4. Proposed Methodology

Our approach aims to develop a lightweight recognition system that can even run-on web browsers, smart TVs, and entry-level smartphones. The proposed architecture is split into three modules: pose estimation, feature transformations, and finally, the NN. Initially, the video is extracted into individual frames, and then pose estimation is done using Blaze Pose architecture. The architecture returns 33 landmarks located at various positions in the body, with each key point having four parameters. The four parameters for each key point are normalized X and Y coordinates, depth of the joint concerning hip, and visibility. Then, transformations are applied to the coordinates such that the key points become independent of the scale and position of the person being detected in the frame. Transformed key points are then fed into the novel CNN-LSTM model. Here, CNN is used for extracting spatial features and LSTM to understand the temporal parts. Finally, the data is then fed into the SoftMax layer, where the probability of each asana is determined. Proposed methodology can be seen in Figure 2. A detailed explanation of the steps mentioned above is discussed in the subsequent sections.

A. *Pose Estimation*

Pose estimation is one of the problems in the CV that has existed for decades, which aims to detect the joints and the person's orientation. Human pose estimation plays a crucial role in diverse fields such as health tracking, sign language processing, and gestural control. Despite the enormous implications of pose estimation in the real world, it is tough to estimate strong articulations, small and barely visible joints, occlusions, clothing, and lighting changes. Due to a wide variety of poses, numerous degrees of freedom, and occlusions, the problem of human pose estimation is even very challenging. However, significant advancements have been achieved in human pose prediction, which helps us

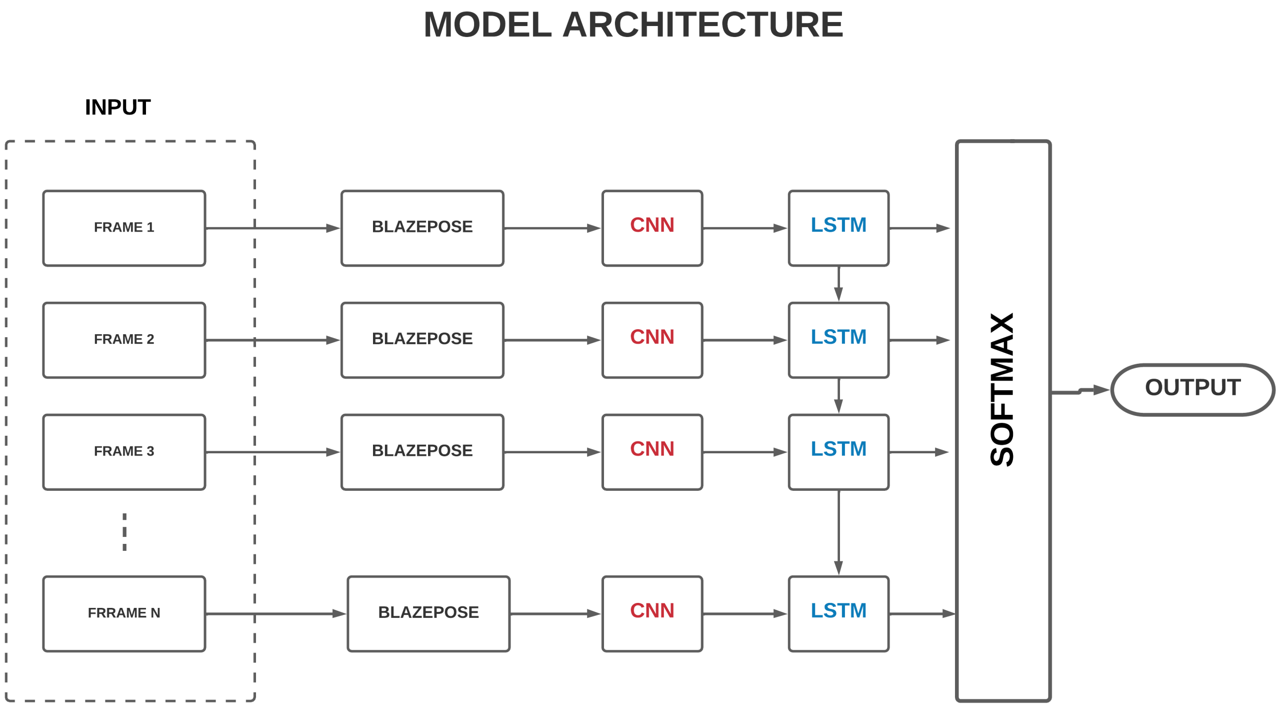

Figure 2. Proposed Model Architecture

best support the various feasible uses. In pose estimation, Blaze Pose is a lightweight, robust, and powerful CNN for mobile devices optimized for real-time inference. The architecture can produce 33 key points or joints in the body at FPS greater than 30, even on mid-tier smartphones or entry-level laptops. The Blaze Pose [29] leverages the offsets and heatmaps from previous frames for this to happen. Blaze Pose architecture is highly optimized for 1-2 people.

Many architectures before Blaze Pose generate critical points with great accuracy. Still, when used for real-time applications such as video input, they often fail as these architectures are computationally intensive. Open Pose [30] is one such architecture used by most of the pose estimation community. The Open Pose is a real-time multi-person pose estimation architecture designed for desktop configurations. In Open Pose, the image is first analyzed through the first ten layers of VGG-19 architecture, generating a feature representation. Thus, obtained feature representations are then passed into two branch multi-stage CNN which outputs confidence maps of parts and vector fields of part affinities. Branch one predicts a set of 2D confidence maps of body parts. Branch two indicates 2D vector fields of part affinities for parts association. Using these, K-partite graph matching is done for multi-person pose estimation. As shown in table 2, Valentin Bazarevsky et al., [29] work shows that depending on the requested quality; Blaze Pose performs 25–75 times faster on a single mid-tier phone CPU than Open Pose on a 20-core laptop CPU.

In contrast, the current proposed architecture can even process videos on mid-range smartphones. Blaze Pose is optimized explicitly for video input. For this to happen effectively, Blaze pose architecture leverages the information from previous frames. In addition, the Blaze Pose predicts key points precisely up to three people, though, beyond three, the architecture can regress key points.

Moreover, Open Pose architecture always assumes humans to be in an up-straight position. But, while performing asana, it can be inferred that many asanas require humans to maintain an inverted pose. So, we consider Open Pose not a reliable solution for real-world applications. Examples of pose estimation by Blaze Pose are shown in Figure 4, 5 and 6. In the first pose estimation step, Blaze Pose uses a fast on-device face detector, as shown in Figure 3, to estimate the person's region for the first frame. This data is then used to generate heatmaps and offsets using an encoder-decoder network, as shown in Figure 7.

As a result, the heatmap is used to supervise the minimal embedding, which the regression encoder network (Figure 7) can then use. This architecture was partially inspired by the layered hourglass approach proposed by Newell et al., [31]. To achieve a compromise between high and low-level functionality; the Blaze Pose architecture approach deliberately uses skip-connections as shown in

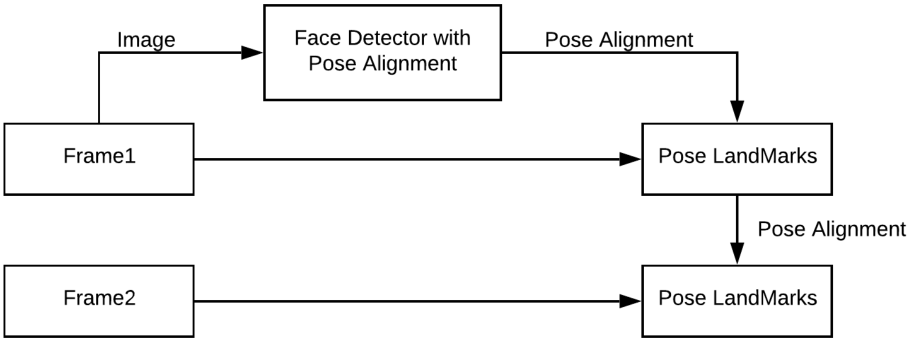

Figure 3. Blaze Pose Detector

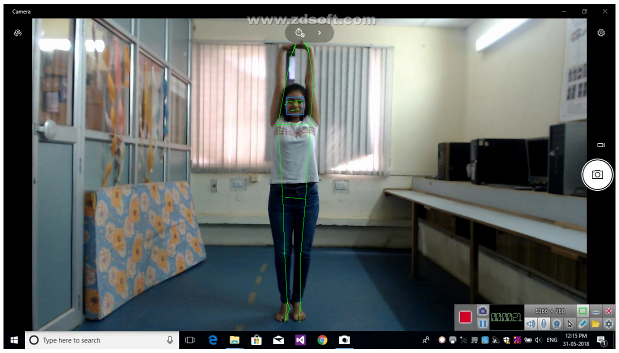

Figure 4. Tadasana pose detected by Blaze Pose

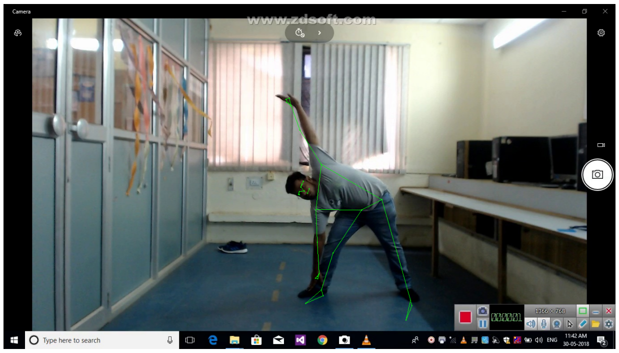

Figure 5. Trikonasan pose detected by Blaze Pose

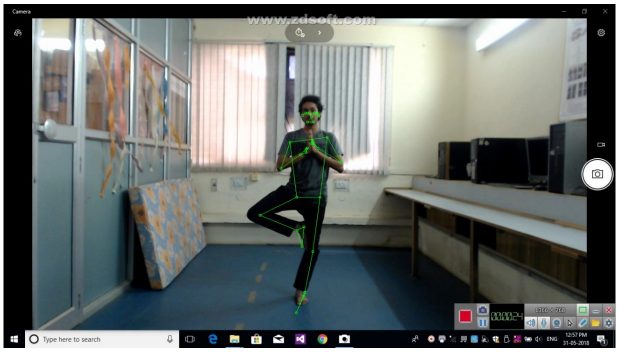

Figure 6. Vrikshasana pose detected by Blaze Pose

Figure 7 between all stages of the network. The regression encoder's gradients, on the other hand, are not passed on to the heatmap-trained features. Not only does this improve heatmap predictions, but it also improves coordinate regression accuracy significantly. For a frame, the Blaze pose architecture returns a list of 33 landmarks. Each landmark has four values, namely (x, y, z, visibility). (x, y) values represent the position of a specific key point in the range of (0, 1) normalized to image height and width. Z represents landmark depth, with depth at the midpoint of the hips being origin. Visibility defines whether the landmark is visible or not. The dataset to train Blaze Pose architecture includes 60 thousand images with one or few people in traditional positions and 25 thousand pictures with a person in the frame conducting fitness exercises. Manual annotation is done on each frame in the entire dataset.

B. Applying Transformations on Key Points

The generated key points are now dependent on the person's position and scale. So, the key points are now transformed to become independent of the scale and position in the frame. Transformations are necessary to eliminate the challenge of positional variance. Any pose estimation model would typically give the coordinates of the key points of various joints it supports. Therefore, the same person with the same pose would result in different key points depending on the person's position in the frame. This may at times increase the variance of the model. We have developed a transformation technique that eliminates the positional variance to mitigate this. The following equations 1 and 2 are used to generate new landmark positions. Now, every co-ordinate of the rendered landmarks is in the range of (0, 256). These landmarks are then reshaped into a 1-dimensional feature vector, resulting in the shape (132,1).

$$arr[i][j] = \left(\frac{256}{max_x - min_x}\right) * (arr[i][j] - min_x) \tag{1}$$

$$arr[i][j] = (\frac{256}{max_y - min_y}) * (arr[i][j] - min_y) \qquad (2)$$

C. Training a Neural Network (NN) Model

By now, the transformed key points are normalized and ready to feed into the NN architecture. CNN's are feed-forward NN primarily designed to solve image recognition tasks. These networks take an image as input and use convolutional structures to extract features, update its parameters, and classify the image based on recognized common patterns.

It would be computationally costly when an image with more dimensions is fed to a standard feed-forward network. Hence, a CNN architecture has multiple pairs of alternating convolutional and pooling layers that extract the most important features to describe the input image. This representation is then encoded and passed to fully connected layers or passed to another network. Recurrent Neural Networks (RNN's) are NN's used to process sequential data like many words, time-series data, machine translation, and speech recognition. The main advantage of RNN over non-sequence-based NN's is its performance. In RNN, the output of the current state depends not only on the current state but also on the production of the previous state, thereby making it suitable for sequential data processing. But the disadvantage of RNN's is it becomes very tedious to process long sequences. Hence, we use LSTM, a particular type of RNN which effectively solves this problem. LSTM networks are specific RNN's intended to remember long input sequences. Its primary purpose is to avoid the long-term dependency problem. In standard RNNs, NN's cell state will have a straightforward structure with a single tanh layer. But in the case of LSTMs, cell states will have four layers interacting especially.

The proposed architecture is a combination of CNN and LSTM networks. For feature extraction on input data, CNN layers are paired with LSTMs to facilitate sequence prediction in the CNN-LSTM architecture. CNN-LSTMs is a type of model that is both spatially and temporally deep and may be used to a wide range of vision problems with sequential inputs and outputs. CNN-LSTMs effectively solve activity identification, picture, and video description challenges. Each training sample passed to the NN collects key points from 64 subsequent video frames. Such training samples are made with 75% overlap. That is key points from the last 48 frames in the previous window become the initial 48 frames in the current window. This way, the model can be used in real-time environments where input is typically a live stream. The output of the LSTM is sent to dense layers for further encoding and finally passed to a SoftMax layer which outputs the probability of the input belonging to a specific class. The class with the highest chance is being considered as the prediction. Each video in the dataset is a person or practitioner performing a particular asana. A label, also known as ground truth, is given to a video clip indicating respective asana. Asanas predicted by the model are compared against the ground truth to determine the correctness of the video.

5. Experimental Results

The entire dataset has been initially split into training and testing datasets in the ratio of 80 and 20, respectively (Figure 8). Again, 20% of dataset has been used for validation during training itself in training data. Training has been done for 100 epochs. While training the model, the learning rate is kept as 0.0001 and the dropout rate as 0.2. The optimizer used is Adam, and the loss function used to calculate loss is categorical cross-entropy loss. After training, the model resulted in training accuracy of 95.29% and testing accuracy of 98.65%. Loss and accuracy on train and validation data are plotted and displayed in Figure 9 and Figure 10. After that, a normalized confusion matrix is developed from the predictions done by the model on the test dataset. A confusion matrix is a table used to describe the performance of a classifier on a set of data for which accurate labels are known. The main problem with classification accuracy for multi-class scenarios is the model performance cannot be understood perfectly at the class level. The confusion matrix helps better in visualizing the model performance on class level. A normalized confusion matrix will have all its values within the range 0-1. The confusion matrix and normalized confusion matrix are displayed in Figure 11 and Figure 12. In addition to accuracy and confusion matrix, we have also calculated metrics like precision, recall as shown in II

6. Discussion

Though many literary works have demonstrated higher accuracy numbers, neither have disclosed the dataset [26] nor mentioned whether the accuracy metrics are obtained on the train, test, or validation datasets. A similar dataset was used by SK Yadav et al., [24], where he developed a recognition system using DL and Open Pose, which resulted in an accuracy of 99.38%. However, our accuracy is 98.65%, which is comparable. The main problem with Open Pose is that it requires very high computational power to estimate key points. Despite high computational power, the usage of Open Pose allows us to process frames at only 0.4 FPS, which is not feasible for real-time applications. In contrast, our recognition system processes video input at 30 FPS, which most real-time cameras capture information. In addition, all the methods take an average of all frame predictions in the entire video. This might result in higher accuracy metrics but is not feasible for real-time applications. The model is supposed to continuously give forecasts of the asana rather than per video basis. We strongly feel that our proposed methodology can classify yoga asanas with decent accuracy in real-time compared to other literary works. Moreover, the proposed architecture can also be deployed on entry-level smartphones, web browsers, and smart TVs. In comparison, the model proposed by SK Yadav et al., [24] can only work on desktop configuration. Another research work presented by Shrajal et al., [26] also used the same dataset. Their proposed system has

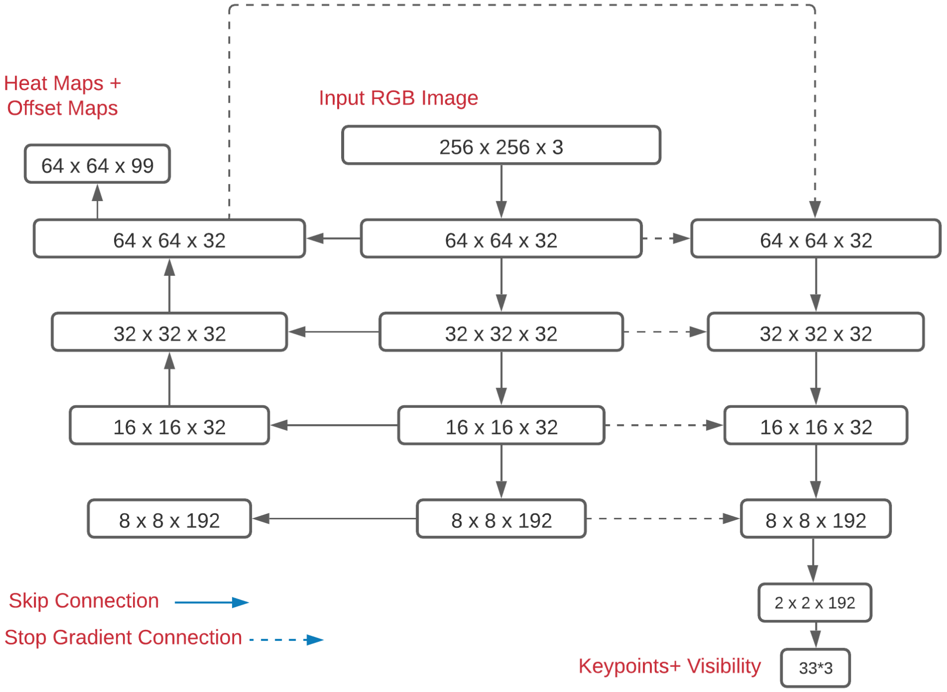

Figure 7. Co-ordinate Regression Model

TABLE II. Performance metrics

Metric	bhujangasan	padmasan	shavasan	tadasan	trikonasan	vrikshasan	micro-average
Precision	0.96	1	1	1	1	0.94	0.98
Recall	1	0.98	0.92	0.97	1	1	0.98
Sensitivity	1	0.98	0.92	0.97	1	1	0.98
Specificity	0.99	1	1	1	1	0.98	0.99
F1 Score	0.98	0.99	0.98	0.99	1	0.99	0.99

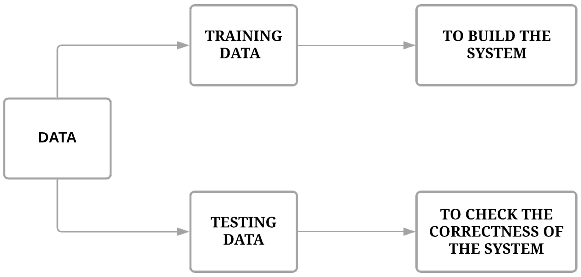

Figure 8. Train and Test Splits of the Dataset

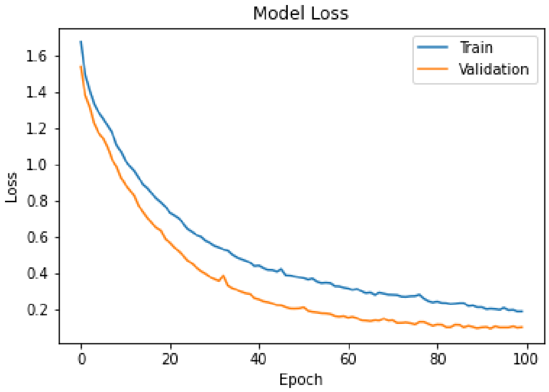

Figure 9. Loss vs Epoch

resulted in an accuracy of 99.39%. However, this project [26] doesn't generate key points during recognition, which will become difficult at later stages when these systems are used for correcting yoga postures. Key points can be leveraged by finding various angles between essential joints

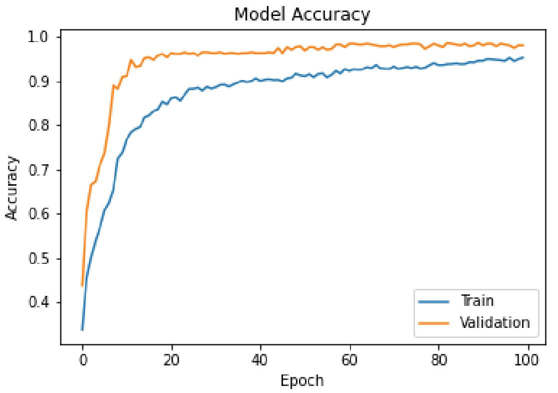

Figure 10. Accuracy vs Epoch

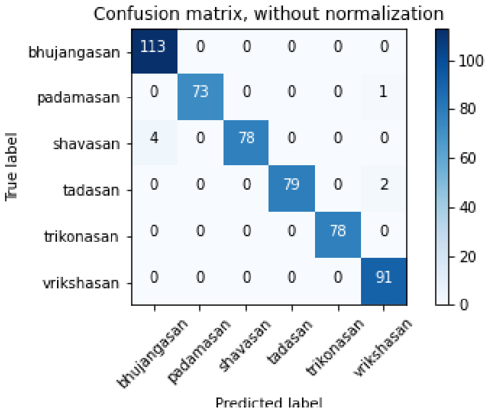

Figure 11. Confusion Matrix without Normalization

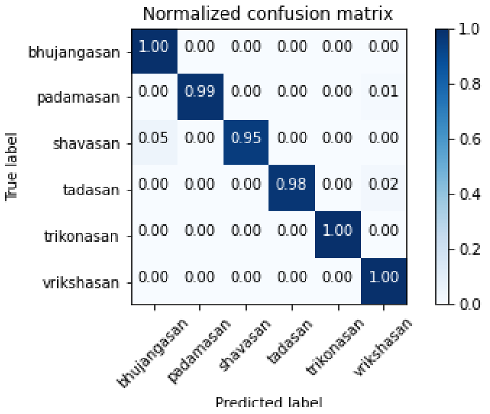

Figure 12. Confusion Matrix with Normalization

of an asana and then fixing them by comparing them with trainer data if available. Though the accuracy is very high, these systems cannot be leveraged for real-time yoga asana training. Other earlier research works do not use DL concepts for recognition. All these systems use specialized hardware, which may not be easily affordable. Usage of DL concepts will ease the entire process. Though these systems generate decent accuracy, they cannot effectively be implemented in real-time scenarios. Our model uses Blaze Pose architecture, which regresses the coordinates of various joints in the body at 30 FPS. We have applied transformation technique on key points of the person's position in the frame to make it scale independent. After training on large datasets, we firmly believe that our approach will give robust results in real-time scenarios. The research objective is to provide a classification architecture that can further assist an individual in training various yoga asanas. We have specifically chosen the Blaze Pose architecture in our pipeline for pose estimation for the speed and accuracy it regresses to coordinates of multiple joints of a human for an individual. Our ablation studies observed that the Blaze Pose architecture could detect key points of various persons up to three persons precisely. Beyond three, we have noticed a decrease in the pose estimation accuracy. In addition, any architecture to detect key points for more than three persons would require high-end configurations.

TABLE III. Accuracy and Performance on OpenPose vs BlazePose

Model	FPS	Accuracy	Dataset used
Open Pose	0.4[1]	99.04%	[28]
Proposed method (Blaze Pose)	30[2]	98.65%	[28]

[1] Desktop CPU with 20 cores (Intel i9-7900X)
[2] Pixel 2 Single Core via XNNPACK backend

7. Conclusion

This paper proposed a robust classifier system that recognizes the yoga asana using an RGB camera at 30 FPS. The developed system works effectively with desktop configurations, web browsers, smart TVs, and entry-level smartphones. The recognition system initially uses Blaze Pose architecture to detect key points from the input stream. Obtained key points are then transformed using the feature extraction technique mentioned in the methodology so that the resulting key points are frame and resolution-independent. After that, the processed key points are passed into a novel DL model of CNN and LSTM. CNN in the architecture is used to extract spatial features of the image. The LSTM network is used to extract the temporal characteristics of the asana over the sequence of the images. The classifier is made robust by applying feature engineering techniques that are not employed by previous works. Also, to the best of our knowledge, this is the first recognition system deployed even in mid-tier smartphones and web browsers, which processes input frames at 30 FPS. Despite high accuracy and decent precision, we still see some limitations that can be addressed in future works which are as follows. 1. publicly available datasets are scarce. As CV and Artificial Intelligence applications in the health industry are still in their infancy; very few public benchmark datasets exist. A large dataset will open up many challenges. 2. The currently available datasets are almost latently similar, with all the videos having the same background. In addition, the dataset doesn't have detailed annotations of whether a user is performing asana correctly or not, which now limits the model only to predict just the name of asana being performed by the practitioner. 3. As of now, our proposed architecture can precisely identify the yoga asana being performed by the practitioner. Once the model detects the asana being performed, this information can be used to correct the asana performed by the practitioners. Further research must be invested in creating suitable architectures and datasets.

References

[1] "A glimpse at yoga, history, philosophy, sanskrit, mantra, mudra, asana, pranayama," http://spot.pcc.edu/~lkidoguc/Yoga/Yoga01.htm, online, accessed: 22 December-2021.

[2] C. C. Streeter, T. H. Whitfield, L. Owen, T. Rein, S. K. Karri, A. Yakhkind, R. Perlmutter, A. Prescot, P. F. Renshaw, D. A. Ciraulo, and J. E. Jensen, "Effects of yoga versus walking on mood, anxiety, and brain GABA levels: A randomized controlled MRS study," *The Journal of Alternative and Complementary Medicine*, vol. 16, no. 11, pp. 1145–1152, nov 2010.

[3] R. R. Guddeti, G. Dang, M. A. Williams, and V. M. Alla, "Role of yoga in cardiac disease and rehabilitation," *Journal of Cardiopulmonary Rehabilitation and Prevention*, vol. 39, no. 3, pp. 146–152, may 2019.

[4] K. Luu and P. A. Hall, "Examining the acute effects of hatha yoga and mindfulness meditation on executive function and mood," *Mindfulness*, vol. 8, no. 4, pp. 873–880, dec 2016.

[5] C. Woodyard, "Exploring the therapeutic effects of yoga and its ability to increase quality of life," *International journal of yoga*, vol. 4, pp. 49–54, 07 2011.

[6] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li, "A review on human activity recognition using vision-based method," *Journal of Healthcare Engineering*, vol. 2017, pp. 1–31, 2017.

[7] D. R. Beddiar, B. Nini, M. Sabokrou, and A. Hadid, "Vision-based human activity recognition: a survey," *Multimedia Tools and Applications*, vol. 79, no. 41-42, pp. 30 509–30 555, aug 2020.

[8] A. Bux, P. Angelov, and Z. Habib, "Vision based human activity recognition: a review," *Advances in Computational Intelligence Systems*, pp. 341–371, 2017.

[9] I. Jegham, A. B. Khalifa, I. Alouani, and M. A. Mahjoub, "Vision-based human action recognition: An overview and real world challenges," *Forensic Science International: Digital Investigation*, vol. 32, p. 200901, mar 2020.

[10] N. B. Nordsborg, H. G. Espinosa, and D. V. Thiel, "Estimating energy expenditure during front crawl swimming using accelerometers," *Procedia Engineering*, vol. 72, pp. 132–137, 2014.

[11] K. Przednowek, K. Wiktorowicz, T. Krzeszowski, and J. Iskra, "A web-oriented expert system for planning hurdles race training programmes," *Neural Computing and Applications*, vol. 31, no. 11, pp. 7227–7243, may 2018.

[12] U. Yahya, S. M. N. A. Senanayake, and A. G. Naim, "A database-driven neural computing framework for classification of vertical jump patterns of healthy female netballers using 3d kinematics–EMG features," *Neural Computing and Applications*, vol. 32, no. 5, pp. 1481–1500, aug 2018.

[13] M. Waldron, C. Twist, J. Highton, P. Worsfold, and M. Daniels, "Movement and physiological match demands of elite rugby league using portable global positioning systems," *Journal of Sports Sciences*, vol. 29, no. 11, pp. 1223–1230, aug 2011.

[14] C. Z. Shan, E. S. L. Ming, H. A. Rahman, and Y. C. Fai, "Investigation of upper limb movement during badminton smash," in *2015 10th Asian Control Conference (ASCC)*, 2015, pp. 1–6.

[15] P.-F. Pai, L.-H. ChangLiao, and K.-P. Lin, "Analyzing basketball games by a support vector machines with decision tree model," *Neural Computing and Applications*, vol. 28, no. 12, pp. 4159–4167, apr 2016.

[16] L. Bai, C. Efstratiou, and C. S. Ang, "wesport: Utilising wrist-band sensing to detect player activities in basketball games," in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2016, pp. 1–6.

[17] S. Patil, A. Pawar, A. Peshave, A. N. Ansari, and A. Navada, "Yoga tutor visualization and analysis using surf algorithm," in *2011 IEEE Control and System Graduate Research Colloquium*, 2011, pp. 43–46.

[18] Z. Luo, W. Yang, Z. Q. Ding, L. Liu, I.-M. Chen, S. H. Yeo, K. V. Ling, and H. B.-L. Duh, ""left arm up!" interactive yoga training in virtual environment," in *2011 IEEE Virtual Reality Conference*, 2011, pp. 261–262.

[19] H.-T. Chen, Y.-Z. He, C.-C. Hsu, C.-L. Chou, S.-Y. Lee, and B.-S. P. Lin, "Yoga posture recognition for self-training," in *MultiMedia Modeling*, C. Gurrin, F. Hopfgartner, W. Hurst, H. Johansen, H. Lee, and N. O'Connor, Eds. Cham: Springer International Publishing, 2014, pp. 496–505.

[20] A. Mohanty, A. Ahmed, T. Goswami, A. Das, P. Vaishnavi, and R. R. Sahay, "Robust pose recognition using deep learning," in *Proceedings of International Conference on Computer Vision and Image Processing*. Springer, 2017, pp. 93–105.

[21] H.-T. Chen, Y.-Z. He, and C.-C. Hsu, "Computer-assisted yoga training system," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 23 969–23 991, feb 2018.

[22] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[23] B.-S. Wu, C.-C. Hsieh, and C.-C. Lee, "A distance computer vision assisted yoga learning system," *JCP*, vol. 6, pp. 2382–2388, 11 2011.

[24] 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/s00 521–019, 12 2019.

[25] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[26] 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 Computing and Applications*, vol. 33, pp. 1–15, 06 2021.

[27] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[28] “Yoga vid collected : Free download, borrow, and streaming : Internet archive.” https://archive.org/details/YogaVidCollected, accessed:22 December-2021.

[29] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, “Blazepose: On-device real-time body pose tracking,” 2020.

[30] 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, no. 1, pp. 172–186, 2019.

[31] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 483–499.

Sharfuddin Waseem Mohammed is currently pursuing his Ph.D. (CSE) from NIT-Tiruchirappalli, Tamil Nadu, India, and is an Assistant Professor, Dept. of CSE KITSW, Warangal, Telangana State, India. He has a total of 11 years of experience as an academic. He researches Image Processing, pattern recognition, and Deep Learning, emphasizing object detection. He is also active in doing real-time projects in Java and Web Development. Apart from research and teaching, he creates innovative solutions to real-world problems and has received grants to build projects that can be of excellent service to society. Email ID: waseem7602@gmail.com

Vignesh Garrapally is a computer vision engineer currently working as a freelancer. He had completed his graduation from the Kakatiya Institute of technology in computer science. He had worked in developing real-time AI Applications for the Aviation and Agriculture Industries. He has a total experience of a year in computer vision and deep learning. His research interests are computer vision and data analytics. Email: garrapallyvignesh8055@gmail.com ORCID ID: https://orcid.org/0000-0001-5437-4225

Soora Narasimha Reddy received his post-graduate degrees (Ph.D.(CSE)) from VNIT, Nagpur, Maharashtra, India in 2017and (M.Tech.(CSE)) from JNTU, Hyderabad, Telangana, India in 2007 and his under-graduate degree (B.E.(CSE)) from Osmania University, Hyderabad, Telangana, India in1999. He has 21 years of experience, nine years in the IT industry, and 12 years as an academic. His research interests are in Image Processing, machine learning, and Deep Learning. He is a life member of ISTE, CSI, India, and a Fellow of IETE, India. Email ID: snreddy75@gmail.com, ORCID: https://orcid.org/0000-0002-2268-0022, Scopus author id: 56766080900, Web of Science ResearcherID: AAF-6622-2019, Vidwan-ID: 181740.

Santosh Kumar Naligenti C received his M. Tech in Software Engineering from Kakatiya Institute of Technology and Science, Kakatiya University, Warangal. Currently, he is working as Assistant Professor in the department of CSE at Kakatiya Institute of Technology and Science and has more than 12 years of Teaching Experience. His Research Interest is Biomedical Image Processing and He published 04 research articles in International Journals and one research article in International The conference, and one research article National Journal till date in the field of Biomedical Image Processing, Spatial Data Mining, and Software Testing, respectively. Currently, he is a research Scholar at GIT, GITAM University, Vishakhapatnam, A.P, India.

Suraj Manchala is an Application Development Associate at Accenture. He had completed his graduation from the Kakatiya Institute of technology in computer science. He has a total 6 months experience in deep learning. His research interests are image