BOHR International Journal of Internet of things,
Artificial Intelligence and Machine Learning
2024, Vol. 3, No. 1, pp. 1–15
DOI: 10.54646/bijiam.2024.19
www.bohrpub.com

RESEARCH

Yoga Pose Recognition (YPR) using ML-DL and android application

Partha Ghosh[1]*, Sitam Sardar[1], Riya Mondal[1], Ayush Jha[1] and Aniruddha Sarkar[1]

[1]Department of Computer Science and Engineering Government College of Engineering and Ceramic Technology, Kolkata

***Correspondence:**
Partha Ghosh,
parth_ghos@rediffmail.com

Received: 14 March 2024; **Accepted:** 05 September 2024; **Published:** 12 October 2024

The study aimed to create a Human Activity Recognition (HAR) model for Yoga Pose Recognition and Classification using datasets gathered through smart sensor technologies and imaging and filming devices to read various human actions, recognize various poses, analyze them, and then predict and classify the Yoga pose with minimum error. Pre-recorded data was fed to the model for the initial run and thereafter the model would learn and re-learn new inputs and outputs by supervised learning methods. A collection of data from cameras present in smart smartphones and other devices were used to create a dynamic dataset of posture photos and videos to predict the most feasible output and add the mapping in the dataset to recognize particular Yoga poses. Yoga is a methodical way of attaining balance and harmony both inside oneself and outside the body. It has its roots in ancient India. Its history spans millennia, with the word "yoga" being first used in the Rig Veda, an ancient Indian scripture, which dates back to around 1500 BC. The Atharva Veda, which was written about 1200–1000 BC, places a strong emphasis on breath regulation. Indus-Saraswati seals and fossils depicting yoga sadhana practitioners have also been discovered. These artifacts date back to 2700 BC (10). Nowadays, yoga is performed by millions of people worldwide. It provides mental and physical health advantages, such as lowering stress, anxiety, and depression, as well as physical benefits like better flexibility, strength, and posture. Yoga has grown popular as more individuals try to live healthier lives.
The study investigated various human postures and actions to predict the possible Yoga pose performed by that particular human through ML/DL (Machine Learning and Deep Learning) approaches. The proposed system or model that learned and evolved by obtaining new data and through supervised learning. We have used single-user pose recognition to create personalized datasets. Our aim was to provide a self-instruction system that allows people to learn and practice yoga correctly by themselves. This development laid the foundation for building such a system by discussing various ML and DL approaches to accurately classify Yoga poses on pre-recorded videos and photos.

Keywords: Human Activity Recognition (HAR), CNN, Transfer Learning, Yoga Pose Recognition

1. Introduction

1.1. Human Activity Recognition (HAR)

This wide-ranging topic of study uses machine learning and deep learning to determine a person's precise movement or activity from sensor data. It has greatly influenced many modern research avenues on humans, their surroundings, and the interactions between them. It is being heavily relied on in the modern healthcare sector for health informatics and predictions.

There are three types of HAR:

(1) sensor-based single-user activity recognition.
(2) sensor-based multi-user activity recognition.
(3) sensor-based group activity recognition.

CC BY

2. Objective:

In our work, we focused on sensor-based single-user pose recognition. As smartphones, handheld devices, and other wearable devices have become more common nowadays, the focus of HAR dataset collection has shifted to the sensors present in these devices.

Pose recognition was accomplished in our project using both probabilistic and logical reasoning. Logic-based methods record all reasonable and coherent explanations for the observed behaviors. Thus, every consistent and conceivable result needs to be taken into account. More recently, activity recognition has used statistical learning models and probability theory to reason about activities and probable consequences under ambiguity.

2.1. Human pose recognition and estimation

We focused on Human Pose Recognition and Estimation as the only HAR component. One of the more difficult problems in computer vision is human pose estimation. In order to create a skeletal representation, it deals with the localization of human joints in an image or video. It is challenging to automatically identify a person's stance in an image since it depends on a variety of factors, including the image's quality and scale, lighting, background clutter, clothes, surrounds, and how people interact with their environment.

An application of pose estimation that has attracted many researchers in this field is exercise and fitness. One form of exercise with intricate postures is Yoga, which is an age-old exercise that started in India but is now famous worldwide because of its many physical, mental, and spiritual benefits.

2.2. About yoga

But the thing with yoga is that, like any other kind of exercise, it requires proper technique; otherwise, a yoga session may be counterproductive and even harmful. This means that a teacher is required to oversee the session and adjust the student's posture. An artificial intelligence-based program may be used to recognize yoga postures and offer individualized feedback to assist people improve their form, as not all users have access to an instructor.

2.3 Our focus

This work focused on exploring the different approaches for Yoga pose classification and sought to attain insight into the following: What is pose estimation? What is deep learning? How can deep learning be applied to Yoga pose classification in real time? This project used references from conference proceedings, published papers, technical reports, and journals.

TABLE 1 | Comparing the accuracy of different models on the public dataset.

Deep learning models	Accuracy rate (%)
LSTM	90.47
CNN	91.53
S-LSTM	95.81
LSTM	85.83
BLSTM	84.54
CNN	85.40
BLSTM	95.70
DBLSTM	96.75
HDL	97.95

TABLE 2 | The SLR, HAR, and SHAR datasets' respective results.

Experiment	Window size	IMU-CNN accuracy (%)	IMU-transformer accuracy
SLR	50	96.4	97.3
HAR	50	86.3	89.7
SHAR	50	83.4	85.2
Overall	50	88.6	90.6

3. Literature review

To convert smartphone readings into different kinds of physical activity, Marcin Straczkiewicz et al. (1) have presented a number of Human Activity Recognition (HAR) systems. They took out data on the sensors, body position of smartphones, kinds of physical activity that were researched, data processing methods, and classification systems applied to activity detection. Transitioning from data gathering to data analysis is the primary problem in this discipline. The methods utilized for feature extraction, activity categorization, data preprocessing, and data gathering were the main subjects of their investigation. They spoke on methods' generalizability and repeatability, or their capacity to apply key components to broad and varied research participant groups. Finally, the obstacles that must be overcome to hasten the broader use of smartphone-based HAR in public health studies.

Users can obtain publicly accessible datasets as detailed in a section by Binh Nguyen et al. (2). A framework covering the state-of-the-art research and new directions in HAR applications is proposed. This research aimed to examine the current state of the art for HAR power usage and categorization. HAR's power needs are discussed in detail. To the best of the authors' knowledge, this is the first review article discussing power use in HAR. **Table 1** shows the

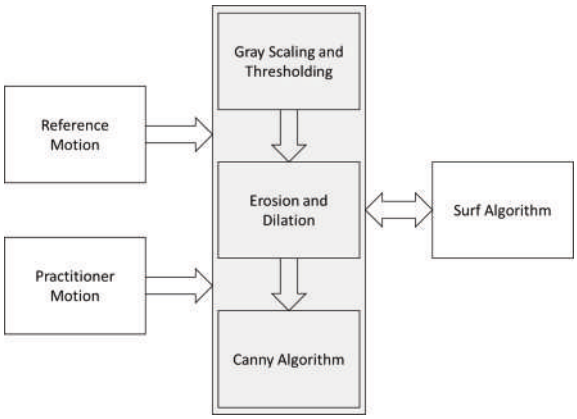

FIGURE 1 | Workflow Diagram.

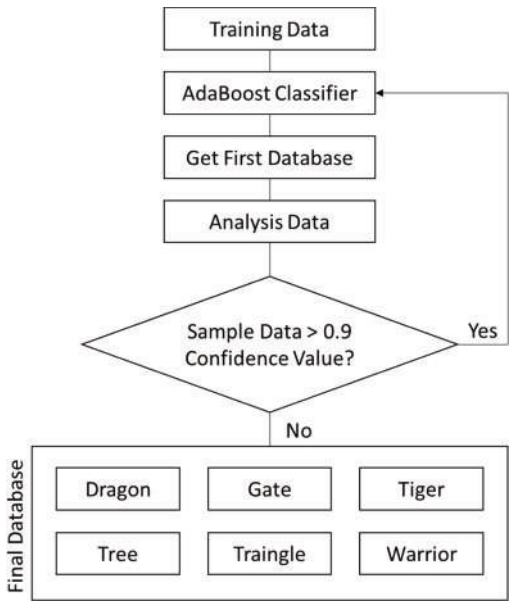

FIGURE 2 | Workflow Diagram.

Accuracy Comparison of several models using the publicly available dataset.

According to Shavit et al. (3), large short-term memory architectures or convolutional neural networks are used in the learning-based techniques currently used for activity recognition from inertial data. For sequence analysis tasks, transformers have recently been demonstrated to perform better than these structures. This study offers an enhanced and comprehensive framework for learning activity identification tasks: an activity recognition model based on Transformers. Across all investigated datasets and situations, the suggested method obtains consistently higher accuracy and improved generalization. The framework described above may be implemented in a codebase that can be found at (4). Relying on one or more of these sensors, HAR finds use in a wide range of applications, such as indoor navigation, gesture recognition, healthcare, and surveillance.

TABLE 3 | Confusion matrix of Asana recognition.

Ground truth	Recognition		
	Tree	Warrior	Dog
Tree	24	0	0
Warrior	1	25	0
Dog	0	0	25

**Warrior III* and *Downward-facing dog* are abbreviated as *Warrior* and *Dog.*

TABLE 4 | 2x2 Confusion matrix.

Predicted values	Actual values	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

The results obtained for the datasets SHAR, HAR, and SLR are presented in **Table 2**.

A lot of work has been done in the past in building systems that are automated or semiautomated which help to analyze exercise and sports activities such as swimming, basketball, etc.

S. Patil et al. (5) proposed a system for identifying Yoga posture differences between an expert and a practitioner using Speeded Up Robust Features (SURF), which uses information of image contours. However, describing and comparing the postures almost by using only the contour information is notsufficient. **Figure 1** illustrates the Workflow Diagram.

W. Wu et al. (6) have devised a system that uses tactors and inertial measurement units (IMUs) for yoga training. However, this may cause the user discomfort and interfere with the natural yoga stance.

E. Trejo et al. (7) presented a system for Yoga pose detection for six poses using Adaboost classifier and Kinect sensors and achieved an accuracy of 94.8%. However, they used a depth sensor-based camera that may not be always accessible to users. **Figure 2** demonstrates the Workflow Diagram.

Another system for Yoga poses correction using Kinect has been presented by H. Chen et al. (8) which takes into account three Yoga poses, warrior III, downward dog, and tree pose. However, their results are not very impressive, and their accuracy score is only 82.84%. The traditional method of skeletonization has now been replaced by deep learning-based methods. **Table 3** shows Confusion Matrix of Asana recognition.

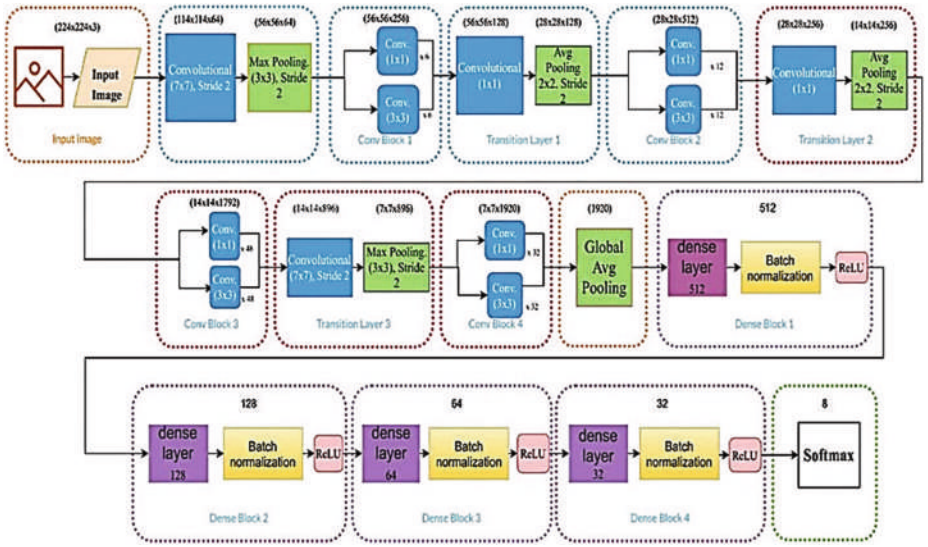

FIGURE 3 | Workflow Diagram of DenseNet201.

4. Motivation and problem formulation

Deep Learning is a promising domain where a lot of research is being done, enabling us to analyze tremendous data in a scalable manner. As compared to traditional Machine Learning models where feature extraction and engineering is a must, Deep Learning eliminates the necessity to do so by understanding complex patterns in the data and extracting features on its own.

4.1. About deep learning

When a model receives picture input and produces a prediction, deep learning is frequently employed for image classification tasks. In order to ascertain the relationship between the input and output, Deep Learning algorithms employ Neural Networks. In Pose Estimation tasks, an image including the individual's pose is used as input, and a Deep Learning model is trained to properly identify the various stances in order to categorize the images with accuracy. This could be a computationally expensive task if the number of images is large. Also, as we want accurate results, we would not want to compromise on the quality of the images as that could affect the features extracted by the model. Below are some basic deep learning models used for classification problems.

4.1.1. Multilayer Perceptron (MLP)

One input and one output layer make up the MLP, a traditional neural network. Known as hidden layers, these are the layers that lie between the input and output layers. There can be one or more hidden layers. MLPs form a fully connected network as every node in one layer has a connection to every node in another layer. A fully connected network is a foundation for Deep Learning. MLP is popular for supervised classification where the input data is assigned a label or class.

4.1.2. Recurrent Neural Network (RNN)

Neural Network Architectures, or RNNs, are employed in sequence prediction applications. One to many, many to one, and many to many are possible scenarios in sequence prediction. RNNs handle sequential data better since they retain a neuron's past information. RNNs are most commonly used for Natural Language Processing (NLP) problems where the input is naturally modeled in sequences.

In activity recognition or pose classification tasks too, there is a dependency between the previously performed action and the next action. In case of yoga as well, the context or information of initial or intermediary poses is important in predicting the final pose. Yoga can thus be thought of as a sequence of poses. This makes RNNs a suitable choice for Yoga pose recognition and classification.

4.1.3. Convolutional Neural Network (CNN)

Convolutional Neural Network is a type of Neural Network widely used in the computer vision domain. It has proved to be highly effective such that it has become the go-to method for most image data. CNNs consist of a minimum of one convolutional layer which is the first layer and is responsible for feature extraction from the images. The convolutional layer, through the use of convolutional filters, generates what is called a feature map. With the help of a pooling layer, the dimensionality is reduced, which reduces the training time and prevents overfitting. CNNs show a great promise in pose classification tasks, making them a highly desirable choice.

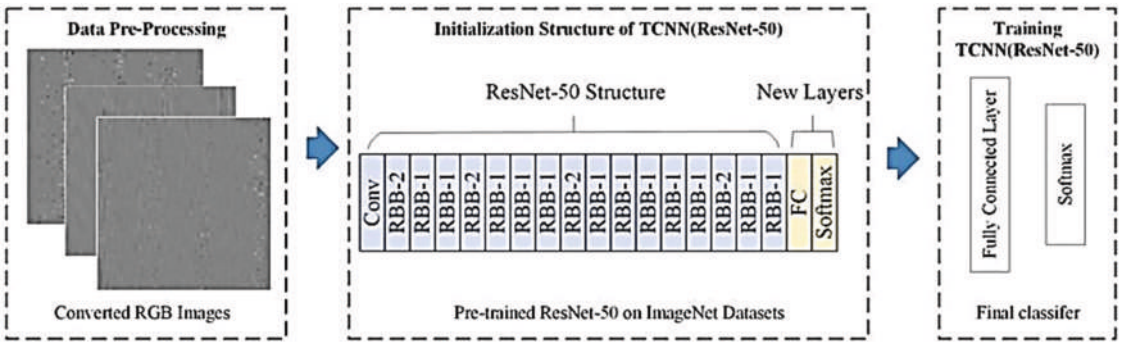

FIGURE 4 | Workflow Diagram of ResNet50.

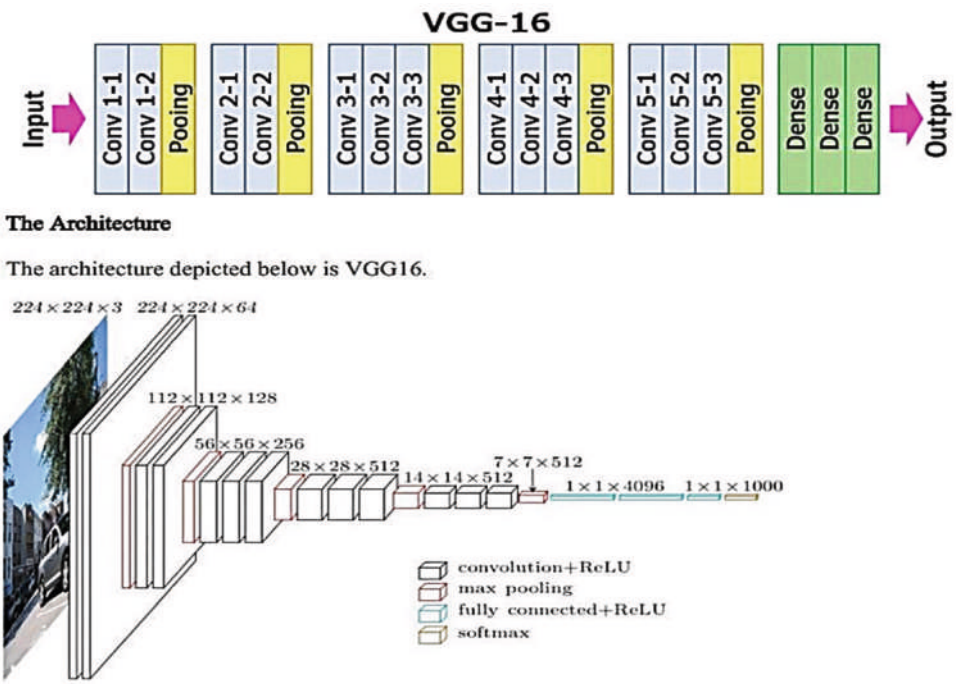

FIGURE 5 | Workflow Diagram of VGG16.

4.2. Problem formulation

As we can see, there is yet to be a robust methodology to identify and classify Yoga poses (Human poses in general) with minimum error, time, and computational power. We are hoping to use Deep Learning Algorithms to try and fill that gap.

4.3. Proposed solution

Through ML/DL approaches, the study of various human postures and actions is being conducted to predict the possible Yoga pose. Through ML/DL approaches, the new data obtained and the supervised learning continued to cause the system or model learn and evolve. We had used using single-user pose recognition to create personalized datasets. Our aim is to provide a self-instruction system that allows people to learn and practice yoga correctly by themselves. This project lays the foundation for building such a system by discussing various Machine Learning and Deep Learning approaches to accurately classify Yoga poses on prerecorded videos and images.

5. Evaluation metrics

5.1. Classification score

Classification score refers to what we usually mean by accuracy of the model. It can be described as the proportion of number of predictions that were correct to the total input samples. In case of multiclass classification, this metric gives good results when the number of samples in each class is

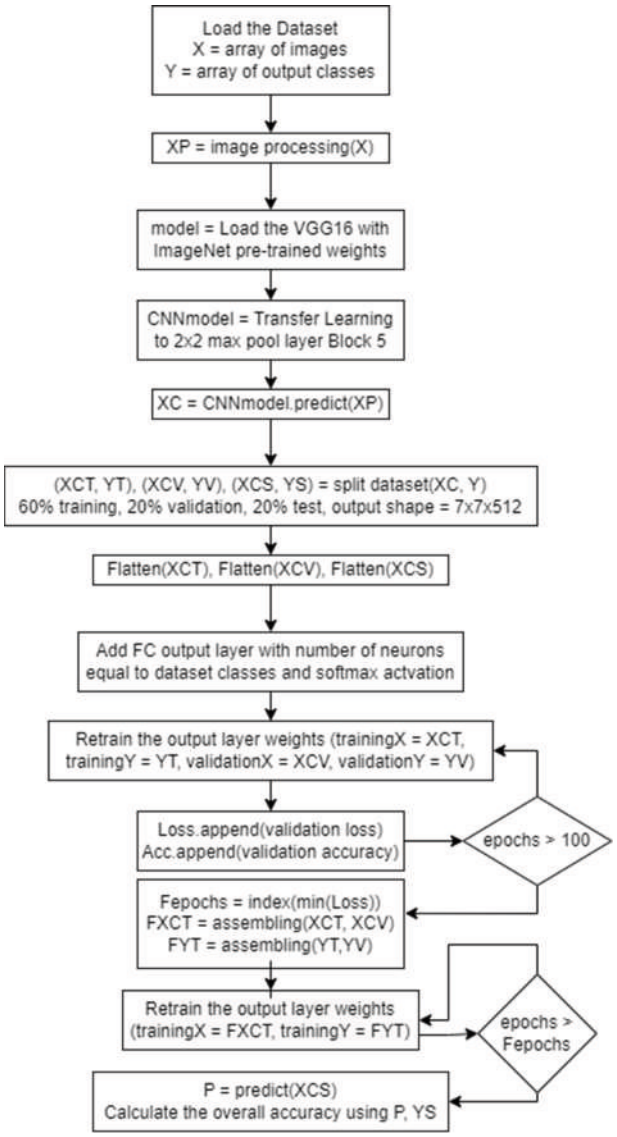

FIGURE 6 | Workflow of VGG16 model.

almost the same.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

5.2. Confusion matrix

Confusion matrix represents a matrix that explains the accuracy of the model completely. There are four important terms when it comes to measuring the performance of a model.

- True Positive: Predicted value & the actual output are both 1.
- True Negative: Predicted value & the actual output are both 0.
- False Positive: Predicted value is 1 but the actual output is 0.

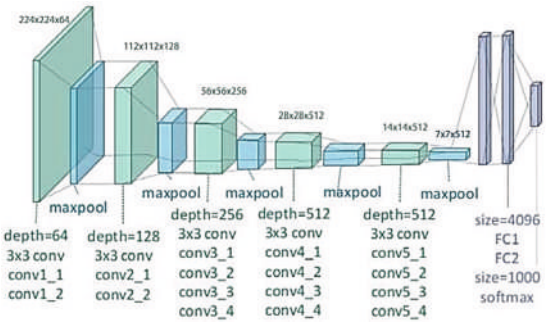

FIGURE 7 | Workflow Diagram of VGG19.

- False Negative: Predicted value is 0 but the actual output is 1.

Table 4 shows a basic confusion matrix for binary classification. The diagonal values represent the samples that are correctly classified and thus, we always want the diagonal of the matrix to contain the maximum value. In case of a multiclass classification, each class represents one row and column of the matrix.

5.3. Model accuracy and model loss curves

These curves are also referred to as learning curves and are mostly used for models that learn incrementally over time, for example, Neural Networks. They represent the evaluation on the training and validation data which gives us an idea of how well the model is learning and how well is it generalizing. The model loss curve represents a minimizing score (loss), which means that a lower score results in better model performance. The model accuracy curve represents a maximizing score (accuracy), which means that a higher score denotes better performance of the model. A good fitting model loss curve is one in which the training and validation loss decrease and reach a point of stability and have a minimal gap between the final loss values. On the other hand, a good fitting model accuracy curve is one in which the training and validation accuracy increase and become stable and there is a minimum gap between the final accuracy values.

6. Methodology

Proposed Models.

6.1. DenseNet201

Overview

Through a feed-forward connection, DenseNet links each layer to every other layer. They significantly lower the

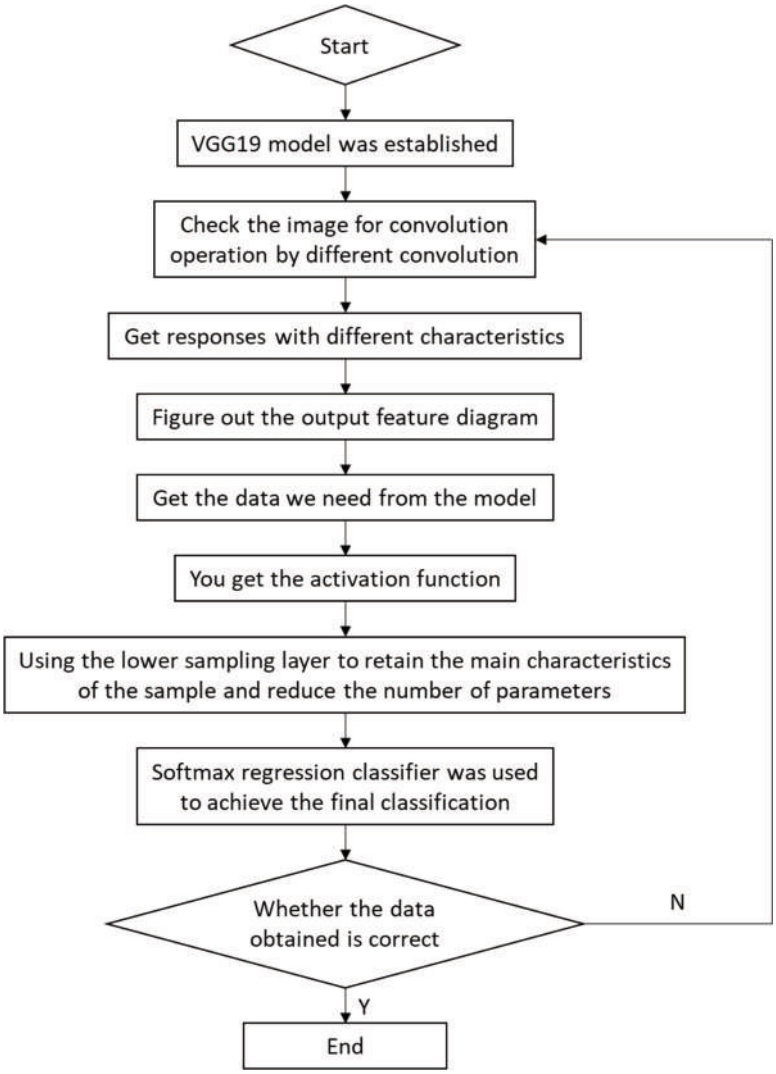

FIGURE 8 | Workflow of VGG19 model.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg19 (Model)                (None, 7, 7, 512)         20024384
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 5)                 125445
=================================================================
Total params: 20,149,829
Trainable params: 20,149,829
Non-trainable params: 0
_________________________________________________________________
```

FIGURE 9 | Total params and trainable params.

```
55/55 [==============================] - 76s 1s/step - loss: 0.0374 - accuracy: 0.98
6/6 [==============================] - 7s 1s/step - loss: 0.0210 - accuracy: 0.989
final train accuracy = 98.96 , validation accuracy = 98.91
```

FIGURE 10 | Accuracy.

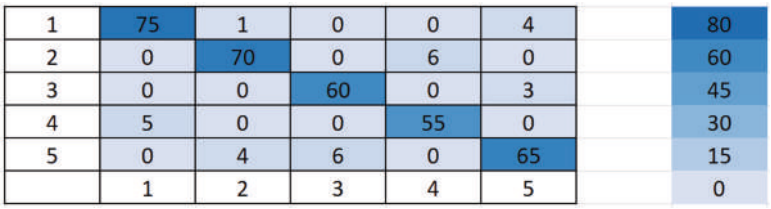

FIGURE 11 | Confusion Matrix: 1. Down dog 2. Goddess 3. Plank 4. Tree 5. Warrior II.

TABLE 5 | Result Chart.

Class	n(truth)	n(classified)	Accuracy	Precision	Recall	F1 score
1	80	80	97.18%	0.94	0.94	0.94
2	75	76	96.89%	0.92	0.93	0.93
3	66	63	97.46%	0.95	0.91	0.93
4	61	60	96.89%	0.92	0.90	0.91
5	72	75	95.20%	0.87	0.90	0.88

number of parameters, improve feature propagation, resolve the vanishing-gradient issue, and promote feature reuse. The concept behind DenseNet is that convolutional networks with fewer connections between input and output layers may be trained to be significantly deeper, more accurate, and more efficient.

201 layers deep CNN is called DenseNet-201. DenseNet is a reasonably tiny and low power consumption model, which is why we chose it.

Results

Training Accuracy: 0.8066

Validation Loss: 2.1614

Validation Accuracy: 0.5377

Workflow Diagram

Figure 3 illustrates Workflow Diagram of DenseNet 201.

6.2. ResNet50

Overview

When we add more layers to our deep neural networks, the performance becomes stagnant or starts to degrade. This happens due to the vanishing gradient problem. When gradients are back propagated through the deep neural network and repeatedly multiplied, this makes gradients extremely small causing the vanishing gradient problem. ResNet solves the vanishing gradient problem by using Identity shortcut connection or skip connections that skip one or more layers. Shortcut connections are connecting output on layer N to the input of layer N+Z.

ResNet-50 is a 50 layers deep CNN. It has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations. It is a type of Artificial Neural Network (ANN) that forms networks by stacking residual blocks.

Results

Training Accuracy: 0.7555

Validation Loss: 1.6325

Validation Accuracy: 0.5590

Workflow Diagram

Figure 4 shows Workflow Diagram of ResNet50.

6.3. VGG16

Overview

There are 16 layers in the convolutional neural network VGG-16. It is thought to be among the greatest computer vision models available today. It mostly concentrates on using the maxpool layer of a 2x2 filter with stride 2 and identical padding for convolution layers of a 3x3 filter with stride 1. Over the complete design, it maintains this configuration of convolution and max pool layers. The final component is a softmax output, which is followed by two FCs (completely connected layers). 16 layers with weights is what the 16 in VGG16 stands for. Approximately 138 million (approximately) trainable parameters make up this huge network.

Results

Training Accuracy: 0.9545

Validation Loss: 0.9325

Validation Accuracy: 0.9245

Workflow Diagram

Figure 5 shows Workflow Diagram of VGG16

Figure 6 illustrates Workflow of VGG16 model.

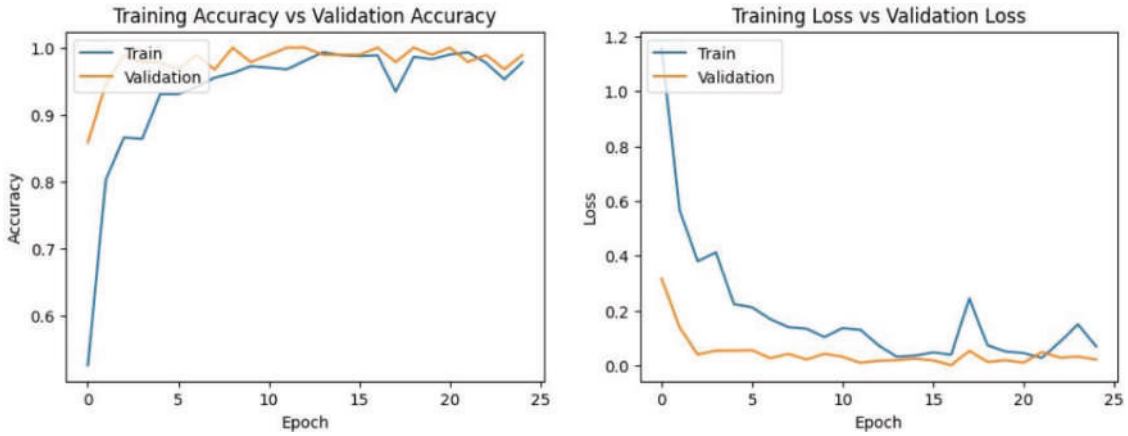

FIGURE 12 | Training Accuracy vs. Validation Accuracy and Training Loss vs. Validation Loss.

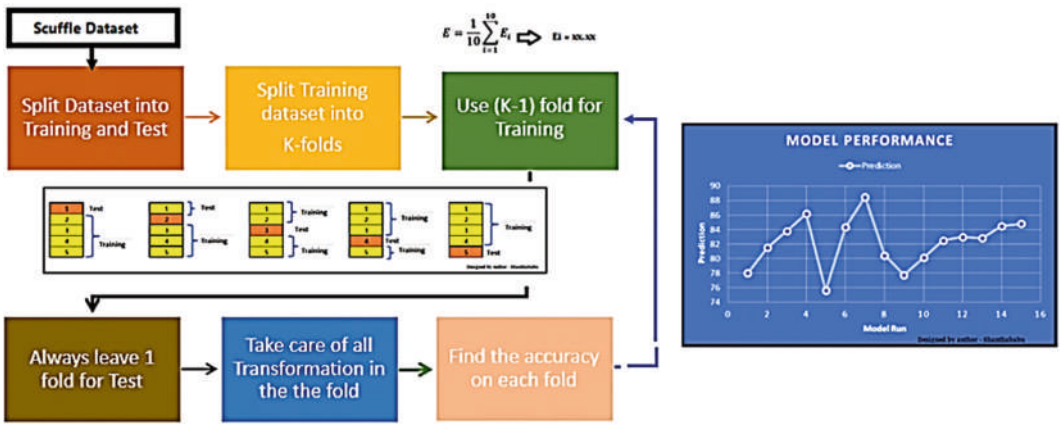

FIGURE 13 | Workflow Diagram of K-fold Cross Validation.

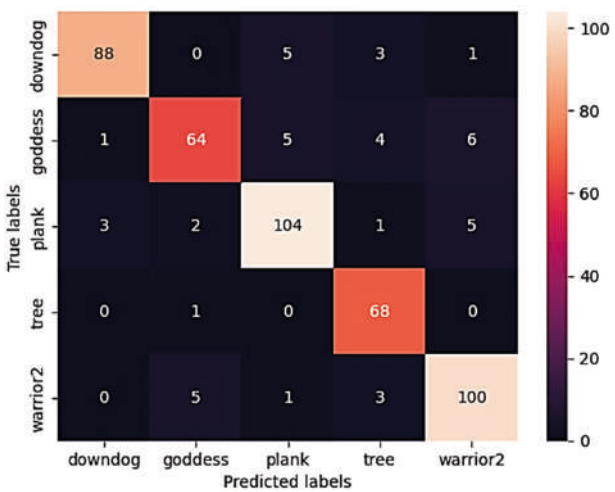

FIGURE 14 | Confusion Matrix.

6.4. VGG19

Overview

A convolutional neural network with 19 layers is called VGG-19. There are sixteen convolution layers, five MaxPool layers, three fully linked layers, and one SoftMax layer. It makes use of (3x3) kernels with a 1-pixel stride size. 19.6 billion FLOPs are in VGG19.

To maintain the image's spatial resolution, spatial padding was applied. Using stride 2, max pooling is carried out over a 2x2 pixel frame. The Rectified Linear Unit (ReLu), which introduces non-linearity to improve the model's classification and computing efficiency, comes next. Previous models relied on sigmoid or tanh functions.

One big advantage of VGG19 is that the weights are easily available with other frameworks like keras so they can be tinkered with and used for as one wants.

Workflow Diagram

Figure 7 describes Workflow Diagram of VGG19.

Figure 8 exemplifies Workflow of VGG19 model.

Results

Training Accuracy: 0.9896

Validation Loss: 0.0210

Validation Accuracy: 0.9891

Figure 9 shows total params and trainable params.

Figure 10 demonstrates Accuracy.

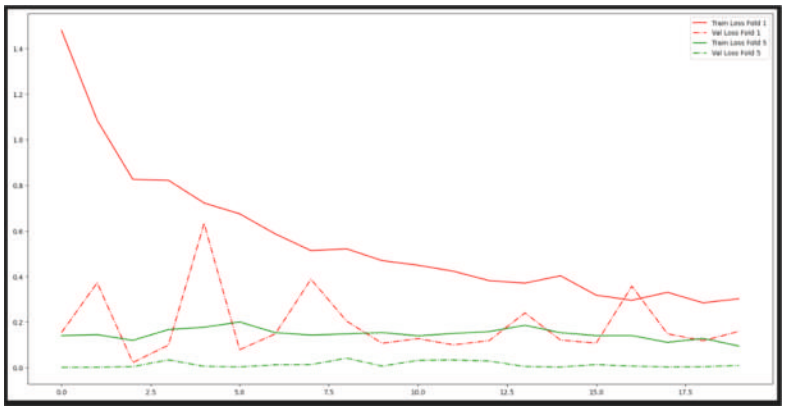

FIGURE 15 | Training Accuracy vs. Validation Accuracy vs. Training Loss vs. Validation Loss.

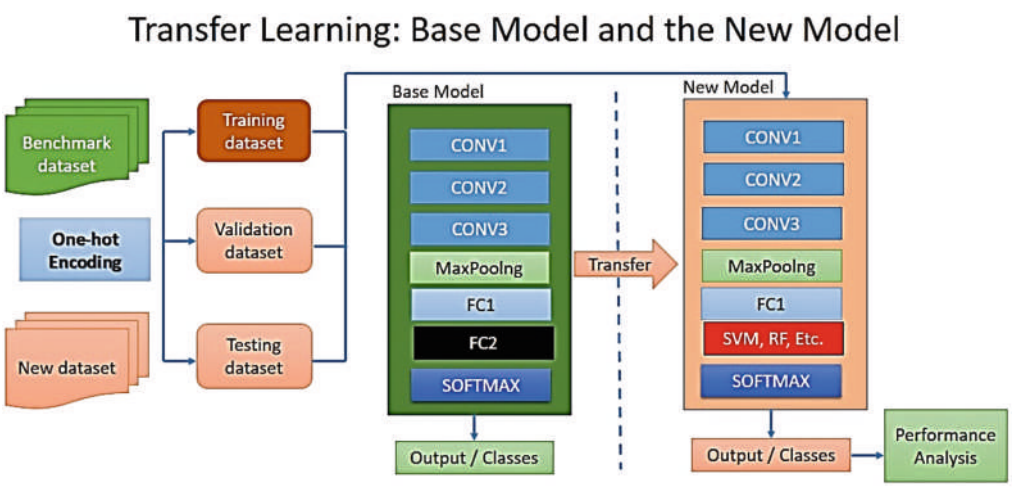

FIGURE 16 | Workflow Diagram of Transfer Learning.

```
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
image-encoder (Model)        (None, 2048)              23564800
_________________________________________________________________
dense (Dense)                (None, 128)               262272
=================================================================
Total params: 23,827,072
Trainable params: 23,781,632
Non-trainable params: 45,440
```

FIGURE 17 | Total params and trainable params.

Figure 11 explains Confusion Matrix: for 1. Down dog 2. Goddess 3. Plank 4. Tree 5. Warrior II and Table 5 shows the Result Chart.

Figure 12 shows Training Accuracy vs. Validation Accuracy & Training Loss vs. Validation Loss.

6.5. CNN (K-fold Cross Validation)

Overview

A resampling technique called cross-validation is used to assess machine learning models on a small sample of data. The process takes a single parameter, k, which is the number of groups into which a given data sample should be divided. As such, k-fold cross-validation is a common name for the process. When a particular number for k is selected, it may be substituted for k in the model's reference; for example, k = 10 would become 10-fold cross-validation.

In applied machine learning, cross-validation is mostly used to assess a model's proficiency on hypothetical data. That is, to assess the model's projected overall performance using a small sample in order to make predictions on data that were not utilized during the training of the model.

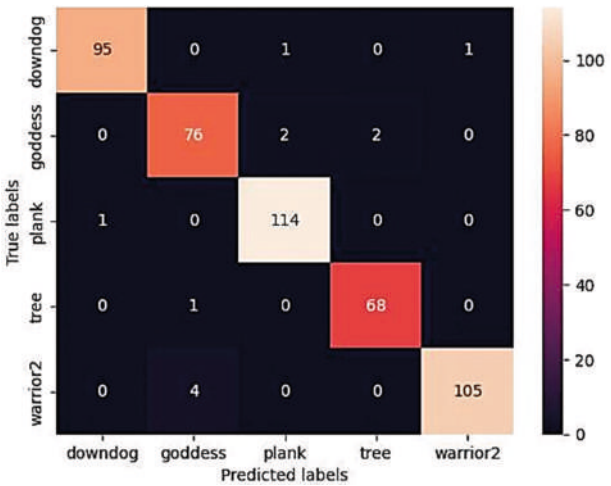

FIGURE 18 | Confusion Matrix.

TABLE 6 | Result chart.

Class	Accuracy	Precision	Recall	F1 score
1	98.60%	0.95	0.95	0.98
2	97.18%	0.93	0.94	0.97
3	97.92%	0.96	0.92	0.96
4	97.45%	0.94	0.90	0.97
5	96.30%	0.89	0.91	0.90

The general procedure is as follows:

(1) Randomly arrange the dataset.
(2) Divide the collection into k groups.
(3) For every distinct group:

a. Use the group as a test or holdout set of data.
b. Create a training data set from the remaining groups.
c. Use the training set to fit a model, then assess it using the test set.
d. Save the assessment result and throw away the model.

4. Compile the model's skill set using a sample of model evaluation results.

Workflow Diagram

Figure 13 illustrates Workflow Diagram of K-fold Cross Validation.

Results

Training Accuracy: 0.9675

Validation Loss: 0.1210

Validation Accuracy: 0.8620

Figure 14 shows Confusion Matrix.

Figure 15 shows Training Accuracy vs. Validation Accuracy vs. Training Loss vs. Validation Loss.

6.6. CNN (Transfer Learning)

Overview

Using Transfer Learning, a Convolutional Neural Network technique, a model created for one task is applied to another as the foundation for a new model.

Given the enormous computation and time resources required to develop neural network models on these problems, as well as the significant skill jumps, they provide on related problems, pre-trained models are frequently used as the starting point on computer vision and natural language processing tasks in deep learning.

Transfer learning involves training a base network on a base dataset and task first, then repurposing the acquired features to train a second target network on a target dataset and task. When characteristics are general—that is, appropriate for both base and target tasks—rather

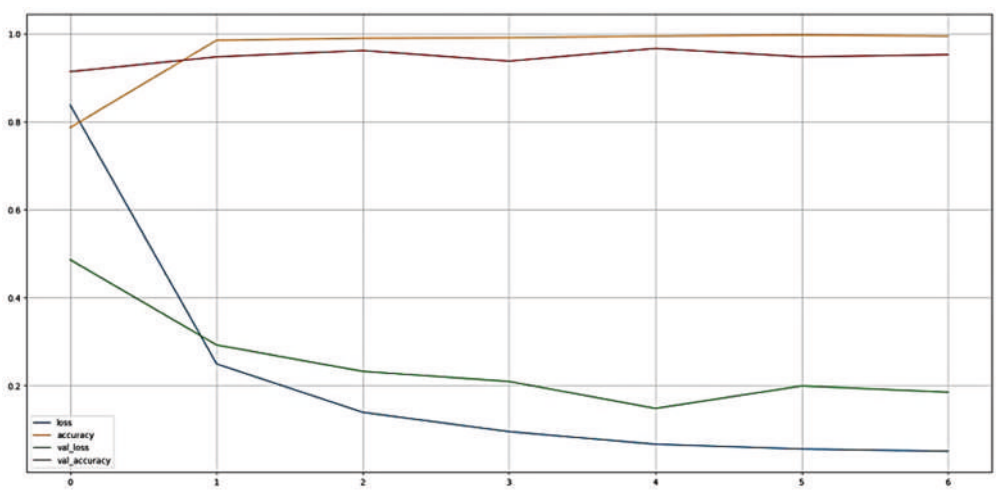

FIGURE 19 | Training Accuracy vs. Validation Accuracy vs. Training Loss vs. Validation Loss.

than particular to the base task, this procedure is more likely to succeed.

Workflow Diagram

Figure 16 demonstrates Workflow Diagram of Transfer Learning

Results

Training Accuracy: 0.9749

Validation Loss: 0.0907

Validation Accuracy: 0.9745

F1 Score: 0.974531880367235

Figure 17 shows total params and trainable params

Figure 18 depicts Confusion Matrix.

Table 6 describes the Result Chart.

Figure 19 shows Training Accuracy vs. Validation Accuracy vs. Training Loss vs. Validation Loss.

7. Dataset

The Dataset used in the project is taken from Kaggle (9) and is publicly available. It consists of images of 5 Yoga poses namely—Downdog (Adho Mukha Svanasana), Goddess (UtkataKonasana), Plank (Phalakasana), Tree (Vrikshsasana), and Warrior II (Virbhadrasana II).

The total number of images is 1081. All the images have been taken in an indoor and an outdoor environment. **Table 7** describes the Dataset.

TABLE 7 | Dataset.

Sl. No.	Yoga pose	Regional name	No. of images
01	Downdog	Adho Mukha Svanasana	223
02	Goddess	UtkataKonasana	180
03	Plank	Phalakasana	266
04	Tree	Vrikshsasana	160
05	Warrior II	Virbhadrasana II	252
Total Yoga Pose Images			1081

TABLE 8 | Comparative study.

Sl. No.	Model	Training Accuracy	Validation Loss	Validation Accuracy
1	DenseNet201	0.8066	2.1614	0.5377
2	ResNet50	0.7555	1.6325	0.5590
3	VGG16	0.9545	0.9325	0.9245
4	VGG19	0.9896	0.0210	0.9891
6	CNN (K-Fold Cross Validation)	0.9675	0.1210	0.8620
5	CNN (Transfer Learning)	0.9749	0.0907	0.9745

8. Results or finding

Human Activity Recognition has been studied extensively in the past years and resulted in the advancement of Machine Learning and Deep Learning methodologies as well as giving rise to new techniques. Further development has been seen in the particular aspect we worked on, i.e., Human Pose Recognition and Estimation to assist in prevention of injuries in sports and exercises and improves performance.

Our Yoga Pose Recognition and Classification system, which will lead to a Yoga self-instruction system, carries the potential to make Yoga more accessible and approachable, making it popular and making sure it is performed in the right manner.

Deep Learning methods are promising because of the vast research being done in this field. The use of the CNN models on the given dataset is seen to be highly effective and classifies all the 5 yoga poses perfectly. **Table 8** illustrates the Comparative study.

9. Discussions

According to **Table 8** one of the highest accuracies is shown by the Transfer Learning model with the least validation loss. Also, we see from **Table 6** that this model has the best consistent Accuracy, Precision, Recall and F1-score for all of the observed Yoga poses though the later ones have a minor drop in the values.

From **Figure 21** we can show that our model reflects a quick rise in both training and validation accuracy that stabilizes early at a certain max value. Unlike VGG19, there is no further dip in the value and is consistent throughout. Inversely, we can show that our model has a quick fall in training and validation loss that gradually stabilizes at a certain min value. It is a bit unstable but stabilizes eventually. Transfer Learning is a bit less accurate than VGG19 but it reaches stability faster and gives better Precision, Recall, and F1-score. It also produces a better Confusion Matrix than the previous, as shown in **Figure 20**. So, it can be trained faster and with a smaller optimized dataset with greater accuracy.

Thus, we conclude that Transfer Learning is the best model for Yoga Pose Recognition and Classification with minimum error we want to achieve.

10. Results

Android Implementation

Android is a mobile operating system based on a modified version of the Linux Kernel and other open-source software, designed primarily for touchscreen mobiles.

Why have we chosen android?

Now a days android smart phones are the most popular across the globe. Almost everyone has a smart phone.

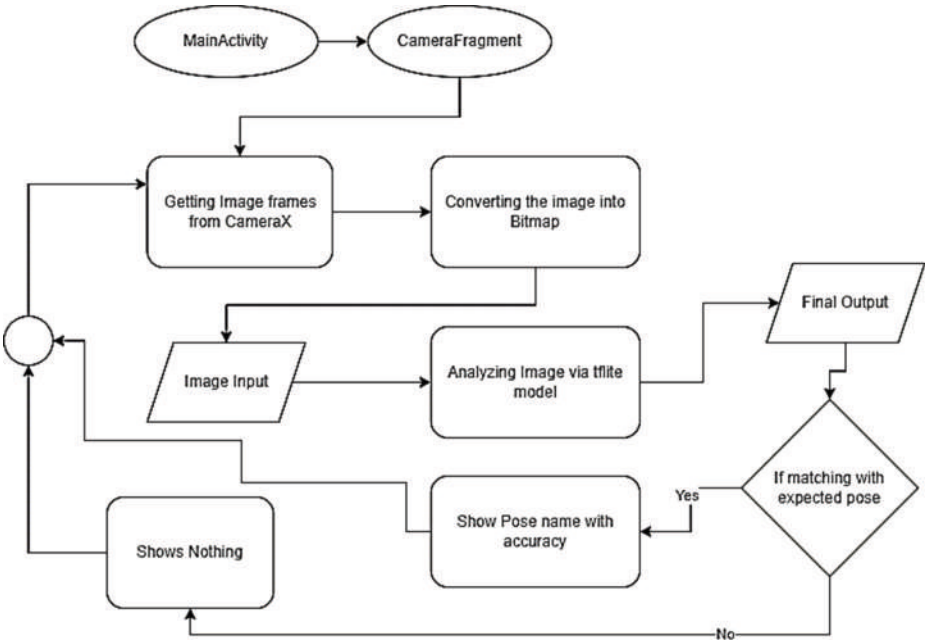

FIGURE 20 | Workflow Diagram of Android Application.

FIGURE 21 | The QR code.

Therefore, through our app, Yoga Trainer, we can deliver our product to a huge audience and also as the android is an open-source platform, it's easy to access and work with camera and sensors. There isn't any restriction in android during development. It gives more freedom, control, and customization.

Till now our implementation

- At the opening of our main screen (MainActivity() ? CameraFragment()) we are opening camera to take real-time video shots. To do so we are using CameraX Library, which is a Jetpack library, built to help make camera app development easier.

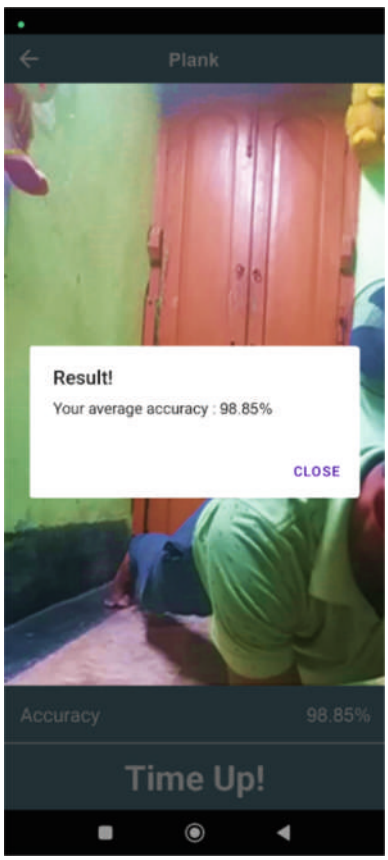

FIGURE 22 | Plank pose.

- Using CameraX whatever image frames we are getting in real time we are converting them into a required Bitmap, which is simply a rectangle of pixels.

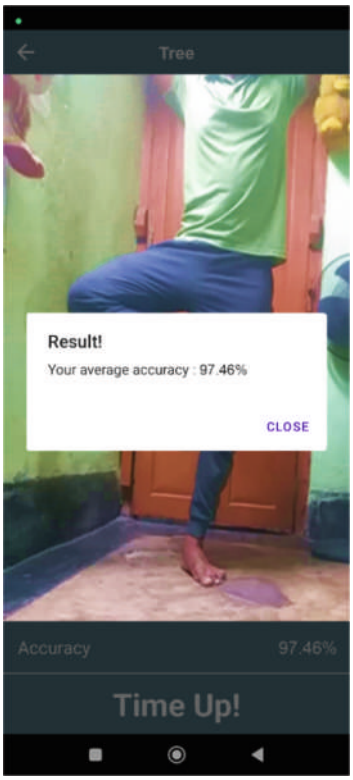

FIGURE 23 | Tree pose.

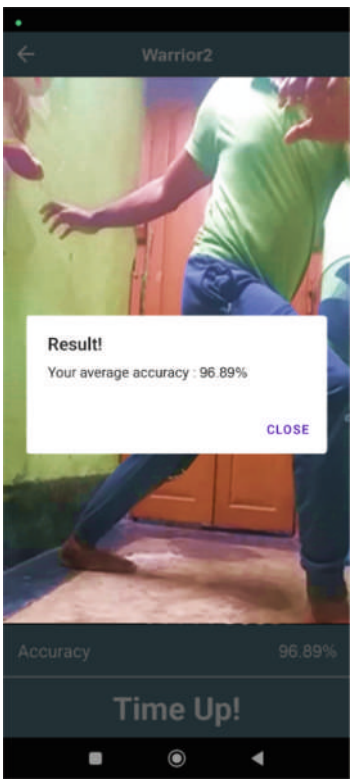

FIGURE 24 | Warrior pose.

- Now we are analyzing each image in real time via our trained tflite model with the help of image analyzer of cameraX and tflite android support.

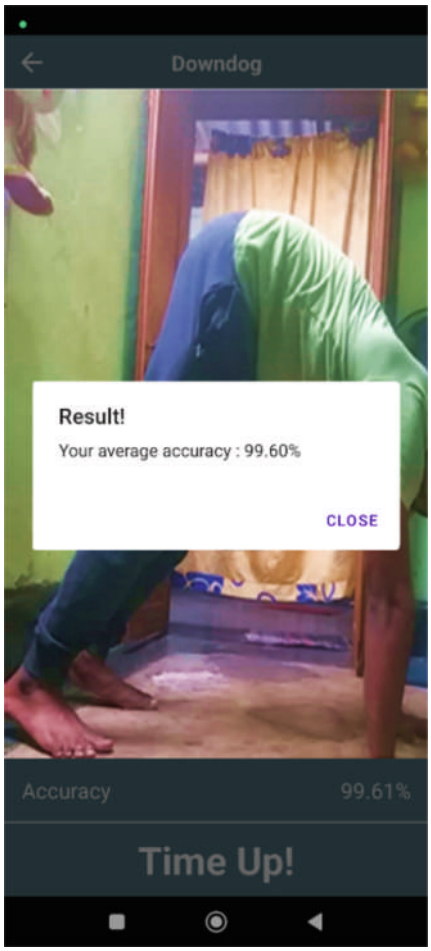

FIGURE 25 | Downdog pose.

- Then according to the analyzed data our result is shown. If any poses are matching, the name of the pose and accuracy are displayed.

Workflow Diagram

Figure 20 shows Workflow Diagram of Android Application.

QR Code to download the app

Figure 21 illustrates The QR Code

Download Link: https://yogatrainer-finalyearproject.github.io/

Working (Shown through four images)

Figures 22–25 show detection of different yoga poses.

Technology Used

Framework

- Android

Language

- Kotlin
- XML
- Python

Tools

- TFLite
- Figma
- Android Studio

11. Conclusion

The work aimed to develop a self-instruction system for practicing yoga by individuals using machine learning and deep learning approaches. It uses single-user pose recognition to create personalized datasets, allowing users to learn and practice yoga independently. The system evolved through new data and supervised learning. The project discusses various ML and DL approaches for accurately classifying yoga poses from pre-recorded videos and photos.

The proposed model currently classifies only 5 Yoga asanas. There are a number of Yoga asanas, and hence creating a pose recognition and classification model that can be successful for all the asanas is a challenging problem. The dataset can be expanded my adding more Yoga poses performed by individuals not only in indoor settings but also in outdoor settings. A portable device for self-training and real-time predictions can be implemented for this system. This work demonstrates Human Activity Recognition for practical applications.

We will also like to extend the idea to build a full-fledged Yoga Tutorial and Guidance system in real time at home with step-by-step tutorials and will also provide posture error detection, posture correction guidance, and other healthcare-related services like personalized Yoga regime, online real-time guidance, personalized diet charts, etc. through a dedicated app.

Author contributions

PG, SS, and AJ conceived of the presented idea. PG, SS, and AJ developed the theory and performed the computations. AJ and SS verified the analytical methods. PG, RM, and AS encouraged us to investigate and supervised the findings of this work. All authors discussed the results and contributed to the final manuscript.

References

1. Straczkiewicz M, Peter J, Jukka-Pekka O. A systematic review of smartphone-based human activity recognition methods for health research. *NPJ Digit Med.* (2021) 4:1–15.
2. Nguyen B, Yves C, Teodiano B, Sridhar K. Trends in human activity recognition with focus on machine learning and power requirements. *Mach Learn Applic.* (2021) 5:100072.
3. Shavit Y, Itzik K. Boosting inertial-based human activity recognition with transformers. *IEEE Access.* (2021) 9:53540–7.
4. GitHub, Inc. *yolish/har-with-imu-transformer*. (2023). Available online at: https://github.com/yolish/har-with-imu-transformer
5. Patil S, Amey P, Aditya P, Aamir NA, Arundhati N. Yoga tutor visualization and analysis using SURF algorithm. *2011 IEEE control and system graduate research colloquium*. IEEE (2011).
6. Wu W, Yin W, Guo F. Learning and self-instruction expert system for Yoga. *2010 2nd international workshop on intelligent systems and applications*. IEEE (2010).
7. Trejo EW, Yuan P. Recognition of Yoga poses through an interactive system with Kinect device. *2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE (2018).
8. Chen H, He Y, Chou C, Lee S, Lin BP, Yu J. Computer-assisted self-training system for sports exercise using kinects. *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE (2013).
9. Kaggle. *Yoga Poses Dataset*. (2023). Available online at: https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset
10. Jayasuriya, Kasun S. (2012) Discuss evidence of the Yoga practices in the Pre-Vedic Indus-Saraswati Valley.