

# Performance Evaluation of Different Deep Learning Models for Automatic Yoga Pose Classification

Agnish Paul<sup>1</sup> Shuvam Ghosh<sup>1</sup>, Pawan Kumar Singh<sup>1,2</sup>

<sup>1</sup>Department of Information Technology, Jadavpur University, Jadavpur University Salt Lake Campus, Plot No. 8, Salt Lake Bypass, LB Block, Sector III, Kolkata 700106, West Bengal India

<sup>2</sup>Metharath University, 99, Moo 10, Bang Toei, Sam Khok, Pathum Thani, Thailand, 12160

{[agnishpaul2002@gmail.com](mailto:agnishpaul2002@gmail.com), [shuvamghosh1234@gmail.com](mailto:shuvamghosh1234@gmail.com), [pawansingh.ju@gmail.com](mailto:pawansingh.ju@gmail.com)}

## Abstract:

Yoga is a discipline originating from ancient India, concentrating on overall well-being, incorporating physical, mental & spiritual activities. When performing yoga postures without expert guidance, one must concentrate on form and technique to prevent injuries, impelling a significant empirical study in this regard. The main objective of our study is to assess still image-based classification of yoga postures using five of the most popular deep learning models, namely, Vision Transformer, DenseNet201, ResNet50, InceptionV3 and, VGG19. These models are tested on two recently developed standard yoga pose classification datasets. Niharika Pandit created the first standard dataset, while the second one was recently developed by A. Mohan Kumar. The comparison results conclude that the InceptionV3 model showed the best classification accuracy of 98.29% on the Niharika dataset, and the DenseNet201 model showed the best at 87.54% on the other one. This in-depth examination highlights the relative strengths and weaknesses present in each model, helping choose the best one for classifying yoga poses. This groundbreaking study validates the proposed method and marks a new era where high-quality yoga instructions are available to everyone regardless of where they are or how much money they have.

**Keywords:** Yoga pose classification, Deep learning, Vision Transformer, DenseNet201, ResNet50, InceptionV3, VGG19.

## 1. Introduction

Human action recognition is a crucial aspect of machine learning, with significant implications across multiple domains such as healthcare, sports, and fitness. One area gaining traction is the

identification and classification of yoga poses. This task involves sorting various yoga postures using machine learning algorithms, offering considerable advantages in fitness tracking, virtual yoga instruction, and therapeutic settings.

Yoga, an ancient discipline renowned for its physical, mental, and spiritual benefits, consists of precise postures known as "asanas." Traditionally, mastering these poses required guidance from a yoga teacher. However, there's a growing trend in automating this process to make yoga more accessible. Machine learning steps in by enabling the creation of systems that can accurately recognize and categorize yoga poses.

By utilizing extensive datasets containing images, videos, and data from wearable sensors, these systems learn the intricacies of different poses. Visual data like images and videos capture body alignment and positioning details while wearable sensors track movements and body orientations continuously.

Yoga pose detection has seen significant progress recently. It involves predicting body coordinate locations through computer vision techniques. Despite the numerous benefits of yoga practices, ensuring correct pose execution often requires costly personal trainers. Many turn to smartphones for guidance but struggle to assess their pose accuracy. This project aims to address this gap by identifying and correcting yoga poses for individuals striving to practice effectively.

Nevertheless, challenges persist in yoga pose detection. Issues, like pose diversity, individual variations, real-time processing requirements, occlusions, and clutter complexity, are common obstacles. Pose variability presents difficulties for software systems and individuals alike in maintaining knowledge of numerous poses accurately. Furthermore, variations in pose adoption based on comfort levels may hinder accurate classification. Efficient algorithms capable of processing image or video inputs effectively are essential to tackle these challenges. Additionally, overlapping body parts in input sources can complicate pose identification processes significantly.

In our study, to address a variety of issues, particularly in the area of yoga position identification, we use the Vision Transformer (ViT) model's transformational powers in conjunction with four different pre-trained models constructed using Convolutional Neural Networks (CNN) in our study. Two publicly accessible datasets from the Kaggle data repository are utilized in our study through the application of an intricate architectural framework. In addition, our work includes critical processes including transformer-based encoding, embedding construction, and patch-based picture segmentation, all of which are intended to improve the models' ability to recognize yoga poses accurately.

The following are the contributions made by our work:

- With the use of photos of common yoga poses, we have conducted a detailed evaluation of five well-known deep learning models: DenseNet201, VIT, ResNet50, InceptionV3, and VGG19. This analysis has sought to discover the best model for precisely recognizing yoga positions while highlighting the advantages and disadvantages of each model.
- Two standard datasets, the yoga positions dataset by Niharika Pandit [8] and A. Mohan Kumar [9], have been used to assess these models. This dual-dataset strategy demonstrates the models' flexibility across many data sources and guarantees a strong assessment of their performance.
- Our analysis shows that InceptionV3 has reached the highest accuracy rate of 98.29% on the Niharika Pandit dataset while the DenseNet201 model achieves a classification accuracy of 87.54% on the second dataset. These findings provide important guidance for future implementations and improvements by illuminating which models work best for categorizing yoga positions.
- This study opens the door to the democratization of excellent yoga education. We can provide everyone with exact yoga coaching, regardless of geography or financial situation, by utilizing the powers of deep learning models. The goal of this revolutionary method is to raise the standard of living for yoga practitioners everywhere.

The parts that follow outline the foundation that our research was built upon. With the purpose of shedding light on the larger background of our research field, Section 2 presents a summary of earlier research done on the datasets used in our study. The detailed architecture of each and every model we have utilized is then covered in detail in Section 3, which also clarifies the important phases of its creation and functioning. The empirical findings from our studies are presented in Section 4, which highlights the effectiveness and performance of our strategy. The conclusions from our research efforts are finally summarized in Section 5, which also emphasizes the importance and ramifications of our findings.

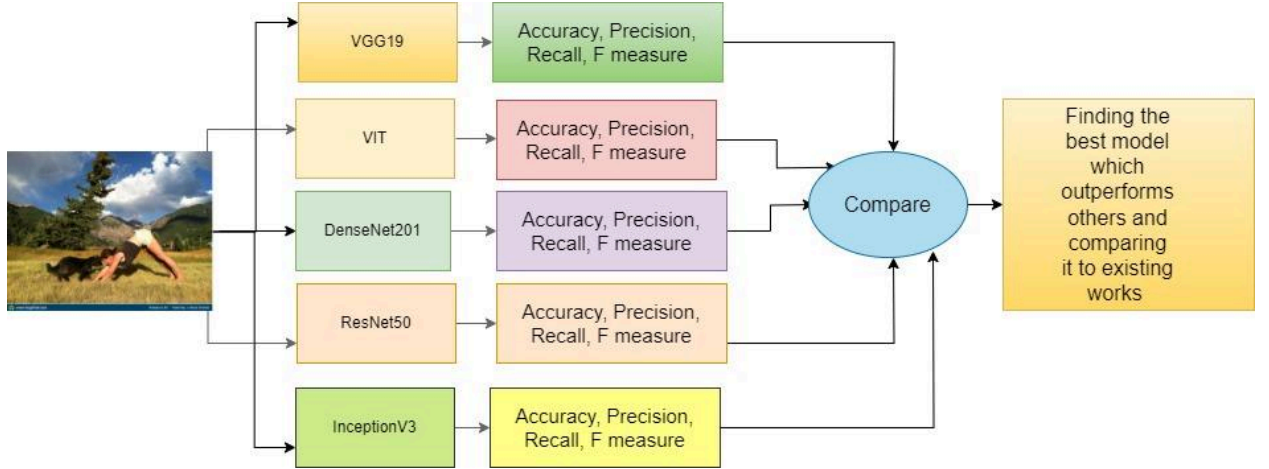


Fig. 1: Flow chart of our proposed study for accurate yoga pose classification problem.

## 2. Related Study

Yoga posture identification has received a great deal of scientific attention, with the main goal being to help practitioners improve their form and technique

In a work by [10], the authors classified yoga positions using a dataset similar to ours. They tested many machine learning models, such as Naive Bayes Updateable (NBU), Fisher's Linear Discriminant Analysis (FLDA), Instance-Based Classifier (IBK), and Logit Boost (LB), using an image enhancement CLF approach. Interestingly, NBU trained on a dataset that was split 90% for training and 10% for testing, and it demonstrated an impressive accuracy of 78.33%.

A thorough investigation of classification models trained on data from a human posture classification model [3] was carried out by the authors in [11]. Their methodology included methods for calculating joint angles as well as spatial structure-oriented strategies that combined body coordinates and relative locations. The authors trained their model in two stages using four different models: Multilayer Perceptron, Random Forest Classifier, Support Vector Machine, and XGBoost. Phase 2 had both coordinates and angles columns, whereas Phase 1 was only concerned with data that had coordinate columns. Notably, XGBoost performed exceptionally well in both phases, with remarkable accuracy rates of 94% in Phase 1 and 96% in Phase 2.

Yoga poses are categorized in [12] by using transfer learning techniques. The first part of the research is preparing the data using image histogram methods, such as the Auto Color Correlogram Filter and the Simple Histogram filter. After that, the processed pictures are used to

apply and evaluate machine learning meta-models, such as Classification via Regression and Iterative Classifiers. They have also used their models to classify yoga poses with noticeably excellent accuracy.

In [13] yoga and mudra poses are recognized using YOLOv7 and faster R-CNN. YOLOv7 relies on CNN architecture and supervised learning for precise predictions. This model detects emphysema in CT-scanned images. On the contrary, faster R-CNN capitalizes on a regional proposal network and a convolutional neural network. This has been implemented in this study to utilize CNN in the emphysema-affected region, resulting in more precise outcomes than YOLOv7.

In [6] a real time video from the webcam of the user's system is obtained and the yoga pose being performed in the video is predicted and information regarding the same i.e. the correctness of the technique or suggestive measure for adjustment and motivating the user to correct their positions and no pose, when no activity is performed, is displayed. A model which uses media pipe and ML framework, to obtain coordinates of certain positions in the body and then use ML like SVM, Random Forest etc algorithms to predict yoga poses.

In our study, we present a comparative analysis involving modern deep learning models, including ViT, DenseNet201, ResNet50, InceptionV3, and VGG19. We bring innovation by comparing these different architectures for classifying yoga poses. In ViT self-attention mechanisms are employed to grasp the overall context effectively, DenseNet201 incorporates dense connectivity to enhance feature reuse and training efficiency, ResNet50 utilizes skip connections to aid in training very deep networks, InceptionV3 employs a multi-scale architecture to capture diverse spatial features, and VGG19 applies a deep yet simple architecture with small convolutional filters for detailed feature extraction. However, modest accuracies have constrained past research, dependence on manually extracted features, complex preprocessing requirements, focus on medical imaging tasks, and potential inaccuracies in real-time pose estimation.



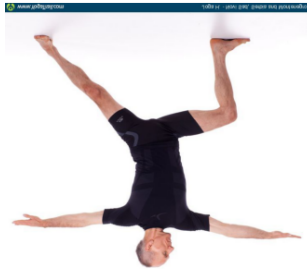
### **3. Methods and Materials**

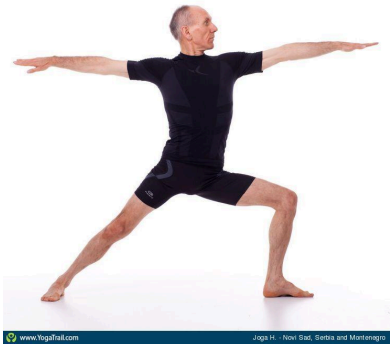

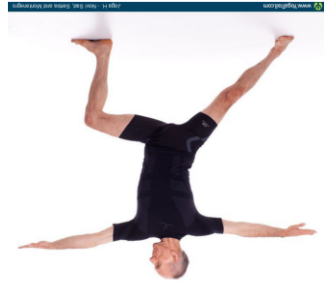
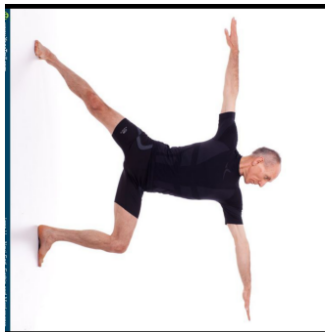
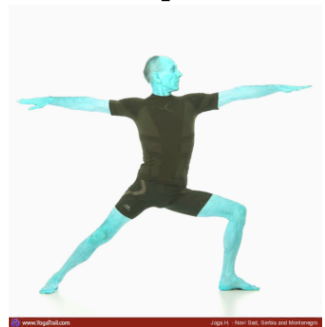
This section discusses the overall process of pre-processing the datasets along with a brief description of overall architectures of the five deep learning models used in the present study.


#### **3.1 Dataset Pre-processing**

To prepare the dataset, we built train and test dataloaders from the relevant data folders using the ImageFolder class with a batch size of 64. To improve speed, we applied certain modifications to enrich the data for every model. We employed augmentations like flipping and color jitter for the VIT model in order to improve generalization and resilience. The photos are scaled to 224x224 pixels and transformed into tensors for the ResNet50 and DenseNet201 models, making sure that the input dimensions fit the model specifications. To enhance performance, the InceptionV3 and VGG19 models additionally make use of augmentations including flipping, color jitter, and rotation. To provide better variety and resilience and produce more accurate and dependable findings, separate transformations are done for the test and train sets of data. As shown in Table1, the modifications alter depending on the unique requirements of each model, but the generation of train and test dataloaders is always the same, guaranteeing excellent performance across many architectures.

Table 1 a)Horizontal Flip applied to original image b)Vertical Flip applied to original image c)Original image rotated by 90 degrees d)Original image rotated by 180 degrees e) Original image rotated by 270 degrees f)Colour Jitter applied to original image g) All the transformations applied together on original image

Original Image	Transformations	Result
	a) Horizontal Flip	
	b) Vertical Flip	

	c) Rotation By 90	
	d) Rotation By 180	
	e) Rotation By 270	
	f) Color Jitter	

	g) Horizontal Flip → Vertical Flip → Rotation By 90 → Color Jitter	
--	---	---

### **3.2 Model architectures:**

In this study, we have done a comparative analysis of five popular pre-trained deep CNN models on 2 yoga pose datasets available on Kaggle data repository. The five different models include Vision Transformer, Resnet50, Densenet201, InceptionV3, and VGG19, the detailed architectures of which are provided below.

#### **3.2.1 Vision transformer:**

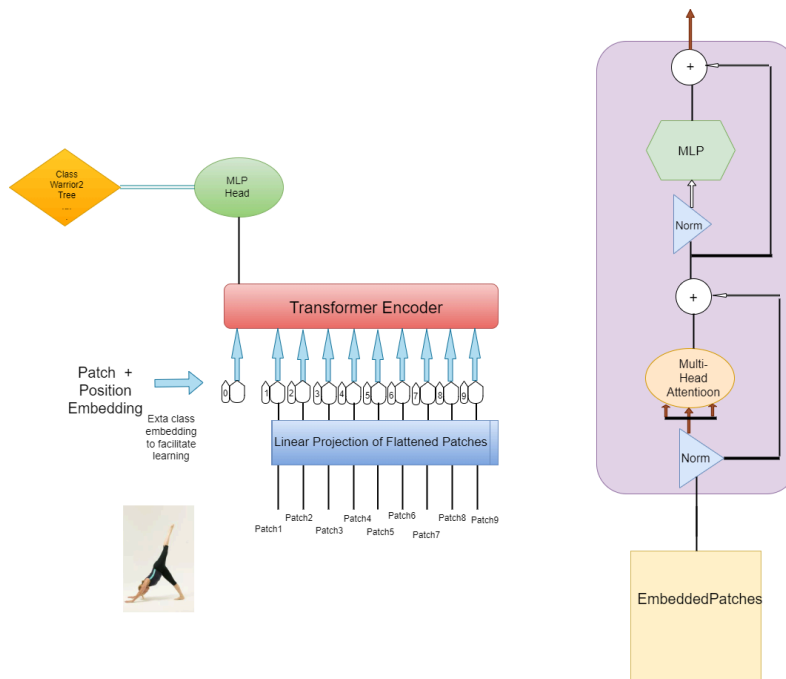
In the VIT model, images have been partitioned into segments of a fixed size, then linearly embedded and this resulting sequence of vectors is fed to a standard Transformer encoder after which an extra classification token is added to facilitate more learning. [14]

Numerous iterations of VIT models have surfaced in scholarly discourse. The fundamental framework of the vision transformer architecture entails the subsequent procedures:

To enable feature extraction and finer-grained analysis, pictures are first divided into distinct patches of consistent size. The spatial arrangement of the pixels is then flattened into a representation that is organized and suitable for further processing steps(12). The linkages and significant aspects of the flattened patches are preserved while they are converted into lower-dimensional linear embeddings, which successfully condenses the copious visual information into a more succinct representation. In order to maintain both spatial context and positional links, the embedding space incorporates positional embeddings. This enables the model to comprehend the relative arrangement of features within the picture.. The sequence of embeddings we've generated is fed into a cutting-edge transformer encoder[15]. This encoder is particularly good at understanding connections between distant elements and the context surrounding each element. Before being put into use, the vision transformer model goes through a



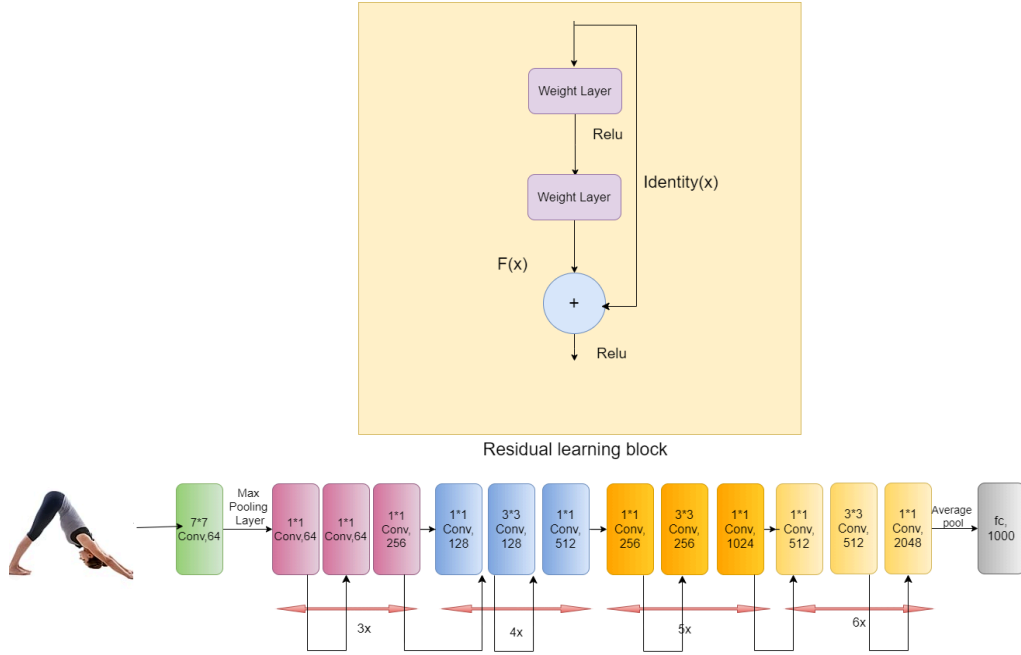
thorough pre-training phase. It basically trains itself in an autonomous fashion throughout this phase, learning how to extract salient characteristics from label-free pictures. This is an important stage since it configures the model's initial parameters and aids in its comprehension of the basic structure of visual input. The model then undergoes a procedure known as fine-tuning after pre-training[15]. This entails making use of a sizable dataset in which pictures have labels indicating the relevant classes or categories. By fine-tuning the model's parameters to more closely match the details of the target dataset, the adjustment process boosts the model's efficiency and increases its suitability for activities and applications in the real world.



**Fig 2::ViT model's architecture**

### **3.2.2 Resnet50**

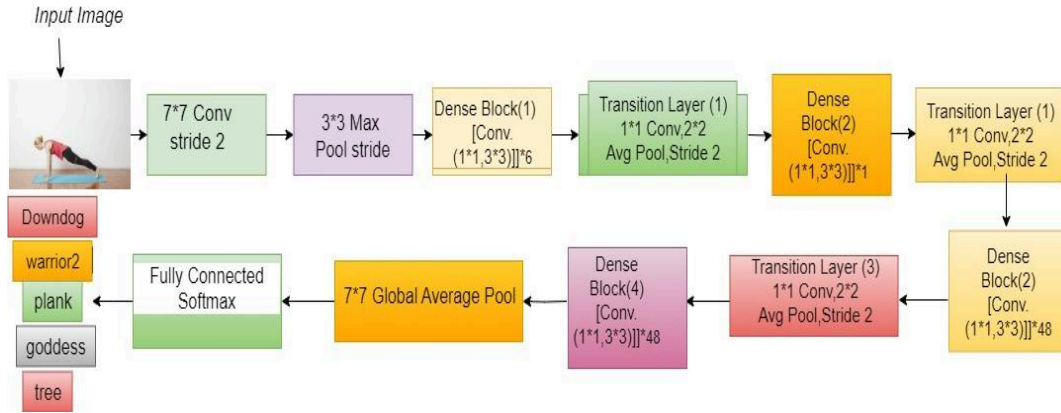
Each of the five blocks that make up ResNet-50's 50 layers is built of a series of residual blocks. The network's ability to derive meaningful representations from the input data is enhanced by these residual blocks, which help preserve important information from prior levels. Residual blocks, fully connected layers, and convolutional layers are the primary parts of ResNet-50[16]. A Maxpooling layer comes after the convolutional layers, which are the first layers of the network. Two convolutional layers make up each residual block, and before the residual block passes through the ReLU activation function, the output of the second convolutional layer is added to the input of the residual block[16]. The next block receives the output that was produced in the leftover block. The ultimate classification is performed by the fully connected layer, and the final class probabilities are obtained by processing the output through a Softmax activation function.



**Fig 3:**Architecture of the Resnet50 model

### **3.2.3 Densenet201**

DenseNet can be seen as a natural progression from ResNet in which each convolutional layer is directly connected to every other layer in a feed-forward manner, which effectively prevents the potential issue of gradient vanishing, reduces the number of parameters needing training, encourages feature reuse, and allows each layer to receive input from all preceding layers, along with that it also has less susceptibility to overfitting when used with unaugmented datasets[17].. We have used pretrained Densenet201 in this paper among many other densenet architectures.

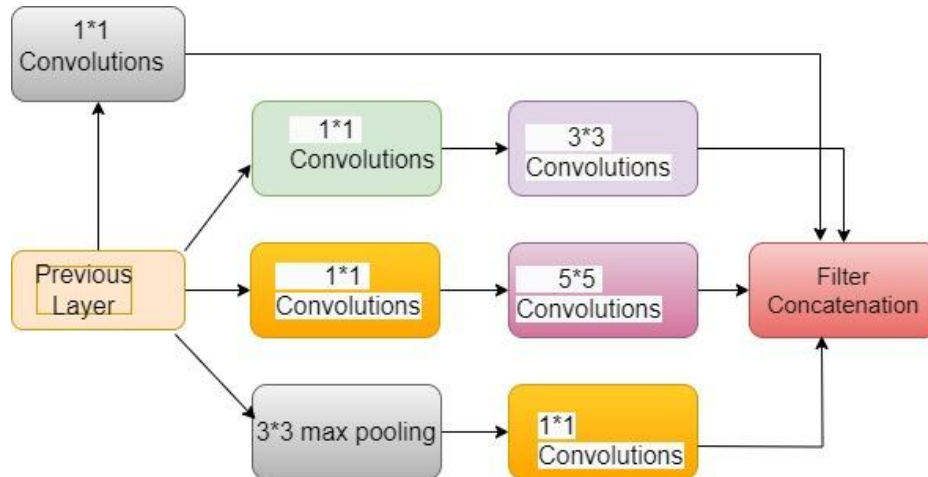


**Fig 4:**Architecture of the Densenet201 model.

After respective models are trained on the training set and then they are assessed on a separate set of data for testing purposes. We saved the model in 3 separate directories for using them later by loading them again from their respective directories.

### **3.2.4 InceptionV3**

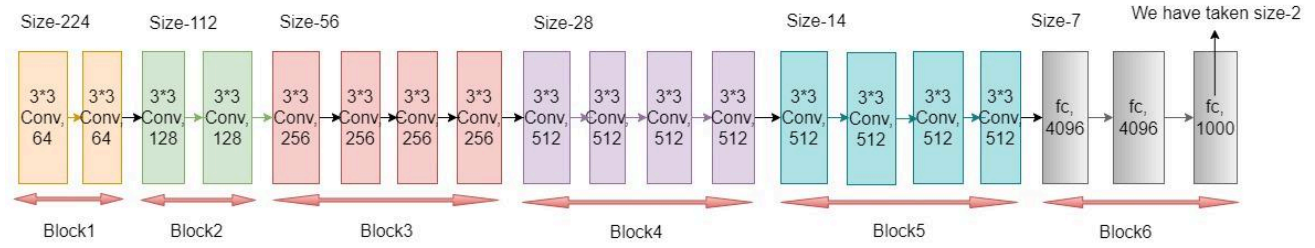
Inception v3 is a brilliant picture awareness diagram reaching nearly 78.1% correctness on the ImageNet database, using a very tricky arrangement of convolutions, pooling, concatenations, and dropout layers with much batch normalization and softmax waste [18]. It is executed utilizing the TPUEstimator API for effective TPU utility, with a clear separation between model delineation and input tubes. The diagram manages input pictures through a thorough pipeline concerning storage, pre-refining, and transfer to TPUs, and employs different optimization methodologies, involving RMSProp, to enrich performance [19]. Efficient pre-refining, likely random chopping and color misunderstanding, significantly boosts diagram correctness, showing the importance of robust input data enhancement in profound learning diagrams.



**Fig 5:**Architecture of the InceptionV3 model.

### **3.2.5 VGG19**

VGG19 is a VGG16 architectural expansion of 19 layers, 16 of which being convolutional layers with the remaining levels being fully linked layers. To efficiently collect spatial data, it makes use of 3x3 convolutional filters with a stride of 1 pixel. A Rectified Linear Unit (ReLU) activation function is subsequently applied to each convolutional layer, speeding up the training process[20]. To minimize spatial dimensions, these convolutional layers are arranged into blocks and have max-pooling layers added between them. Unlike earlier models such as AlexNet, VGG19 does not use Local Response Normalization (LRN) because of its low efficacy. With the exception of the last layer, which includes 1000 channels representing the outputted classes, the network ends with completely linked layers at the end, each with 4096 channels[20]. VGG19 is an effective model for image recognition tasks because of its deep architecture and compact convolutional filters, which let it capture complicated information.



**Fig 6:**Architecture of the VGG19 model.

## 4. Performance Evaluation

### 4.1 Dataset Used

We employed two publicly accessible datasets from Kaggle in this research to train and assess our models for classifying yoga poses.

#### Dataset #1: Niharika Yoga Pose Classification Dataset

The first dataset we used is the Niharika Yoga Pose Classification dataset, which contains labeled images of various yoga poses. The data set has been separated into train and test subfolders, each containing 5 subdirectories corresponding to the five categories of yoga poses. They are ‘Down-dog’, ‘Goddess’, ‘Plank’, ‘Tree’ and ‘Warrior2’.

#### Dataset #2: A. Mohan Kumar Yoga Pose Classification Dataset

To further validate our model, we employed the A Mohan Yoga Pose Classification dataset. This dataset, also available on Kaggle, comprises nine folders, each representing a different yoga pose. Each folder contains approximately 250 images extracted from Google, providing a diverse and comprehensive collection of yoga pose images. The nine yoga poses included in this dataset are: ‘Bridge’, ‘Child’, ‘Cobra’, ‘Downward-Dog’, ‘Pigeon’, ‘Standing-Mountain’, ‘Tree’, ‘Triangle’, and ‘Warrior’.

### 4.2 Experimental setup

The numerical experiments are conducted on an HP PAVILION LAPTOP with an AMD Ryzen 7 5800H processor, 16GB RAM, 64-bit Windows 10 (version 22H2), a 512GB NVMe SSD, and

an NVIDIA GeForce RTX 3050 GPU with 4GB VRAM. Python (version 3.9.13) is used to implement all of the Yoga Pose Prediction models and various analyses. The respective models have been trained and evaluated with the help of PyTorch on GPU.

### 4.3 Evaluation Metrics and Hyperparameters

The list of hyperparameters we used in this study for our best-performing models on two yoga pose classification datasets is shown in Table 2. Various evaluation metrics (namely precision, recall, F-measure, and confusion matrix) are used. To assess the efficacy of the methods for predicting yoga poses. The mathematical equations defining precision, recall, and F1 Score are represented by Equation (1), Equation (2), and Equation (3) correspondingly.

$$\text{Precision: - } \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \quad (1)$$

$$\text{Recall: - } \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \quad (2)$$

$$\text{F1 Score : - } 2 \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

$\text{TP}_c$  is the number of true positives for class c

$\text{FP}_c$  is the number of false positives for class c

$\text{FN}_c$  is the number of false negatives for class c

**Table 2:** Values of hyperparameters for our best models used in this study on two yoga pose classification datasets.

Hyperparameters	Dataset1	Dataset2
Optimizer	<b>Adam</b>	<b>Adam</b>
Loss Function	<b>CrossEntropyLoss</b>	<b>CrossEntropyLoss</b>
Batch size	<b>32</b>	<b>64</b>
Gradient Descent	N/A	N/A
Epochs	<b>20</b>	<b>25</b>
Scheduler	N/A	<b>ReduceLROnPlateau</b>
Learning rate	<b>0.001</b>	<b>0.001</b>

Weight Decay	N/A	1e-4
--------------	-----	------

#### 4.4 Results on Dataset #1

The pre-processed Dataset #1's yoga pose samples are divided into 70% for training and 30% for testing. Table 2 displays the overall results of accuracy, precision, recall, and F1 from five deep-learning models applied to Dataset #1. The highest classification accuracy, 98.29%, is achieved by our best model, InceptionV3. Comparatively, ViT, Densenet201, Resnet50, and VGG19 models exhibit classification accuracies of 96.09%, 97.02%, 93.82%, and 91.70% respectively, as depicted in Table 3.

**Table 3:** Overall results produced by five standard deep learning models on Dataset #1.

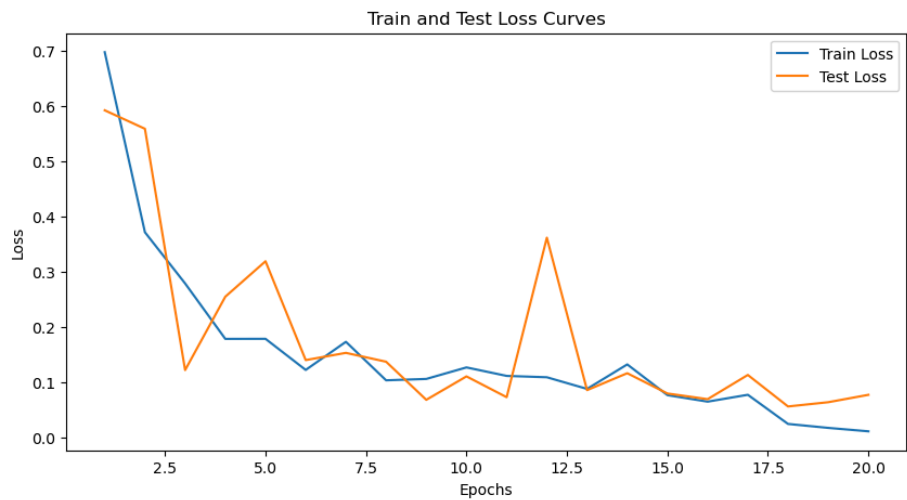
Model	Precision (%)	Recall (%)	F1 Score (%)	Test Accuracy (%)
ViT	95.80	95.74	95.74	96.09
Densenet201	96.94	97.03	96.98	97.02
ResNet50	93.90	93.82	93.78	93.82
InceptionV3	<b>98.30</b>	<b>98.29</b>	<b>98.29</b>	<b>98.29</b>
VGG19	91.72	91.70	91.61	91.70

The loss curve graph produced by InceptionV3 model on Dataset#1 is shown in Fig. 7. This Figure illustrates that the training curve starts at a high loss, eventually decreasing over time highlighting the fact that with more epochs, InceptionV3 starts to fit better on the training set. The test loss curve fluctuates more than the training loss curve and is sensitive to specific data points. Fig.8 clearly depicts that the InceptionV3 model improves its performance with more epochs, however, the fact that the test loss is high suggests that the model does not generalize perfectly to unseen data.

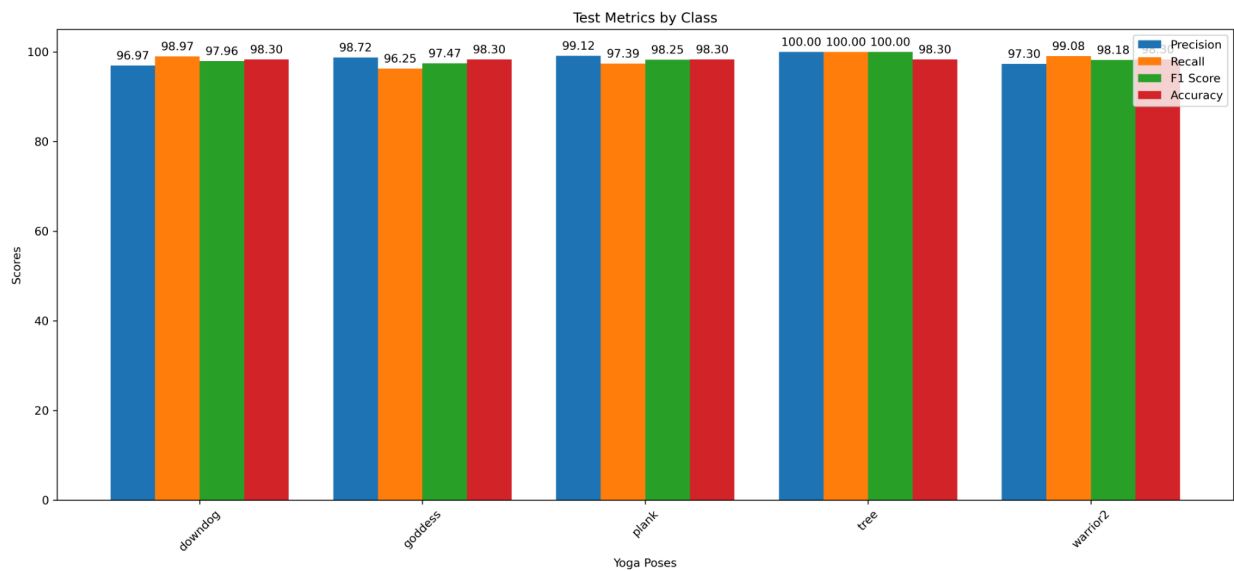
The class-wise recall, F1 Score, precision and accuracy achieved by the best performing InceptionV3 model after the evaluation on Dataset#1 is shown in Fig. 8. From that figure, it can be easily seen that the 'downdog' and 'tree' pose have been classified with the highest accuracy, precision, recall, and accuracy.

The confusion matrix generated by the most successful InceptionV3 model on Dataset#1 is illustrated in Figure 9. Examining this matrix, it is evident that the InceptionV3 model boasts a classification accuracy exceeding 97% across all five yoga pose categories. Notably, with the exceptions of 'Goddess' and 'Plank', the model accurately identifies the other three yoga poses.

An observation from the matrix reveals that some 'Goddess' pose samples are mistakenly classified as 'Warrior2', while certain 'Plank' pose samples are erroneously categorized as 'Down-dog' due to their similar body alignments

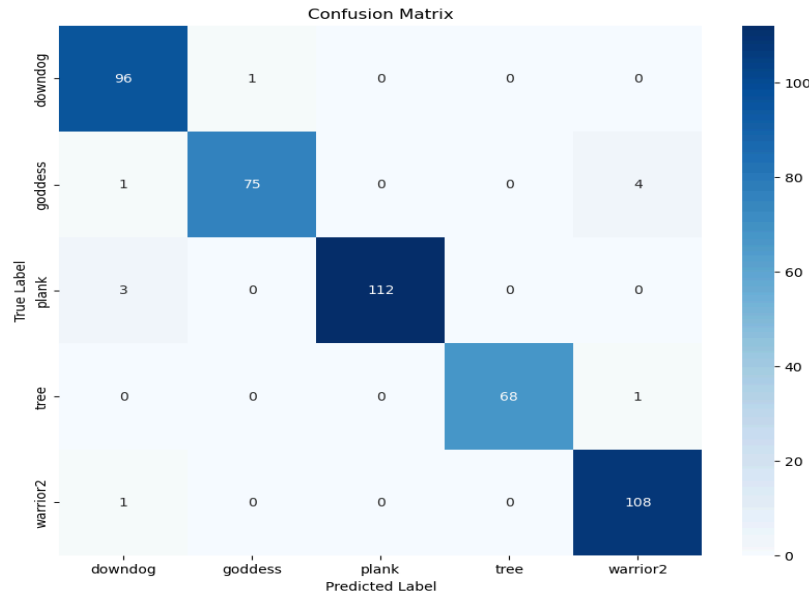


**Fig. 7:** Graph showing the loss curve for the best performing InceptionV3 model on Dataset #1.



**Fig. 8:** Visual comparison of the best-performing InceptionV3 model, based on , Precision, Recall, F1 Score, Accuracy on Dataset #1.





**Fig 9:** Confusion matrix achieved by the best performing InceptionV3 model on Dataset#1.

## **4.5 Results on Dataset 2**

The yoga poses found in Dataset#2 after preprocessing are divided into training (70%) and testing (30%) sets for training and evaluating five deep learning models. Table 3 displays the collective outcomes of accuracy, precision, recall, and F1 Score by these models on Dataset#2. The top-performing Densenet201 model achieves a classification accuracy of 87.54%, while the ViT Resnet50, InceptionV3, VGG19 models achieve accuracies of 85.80%, 73.55%, 86.01%, and 81.61% respectively as illustrated in Table 4.

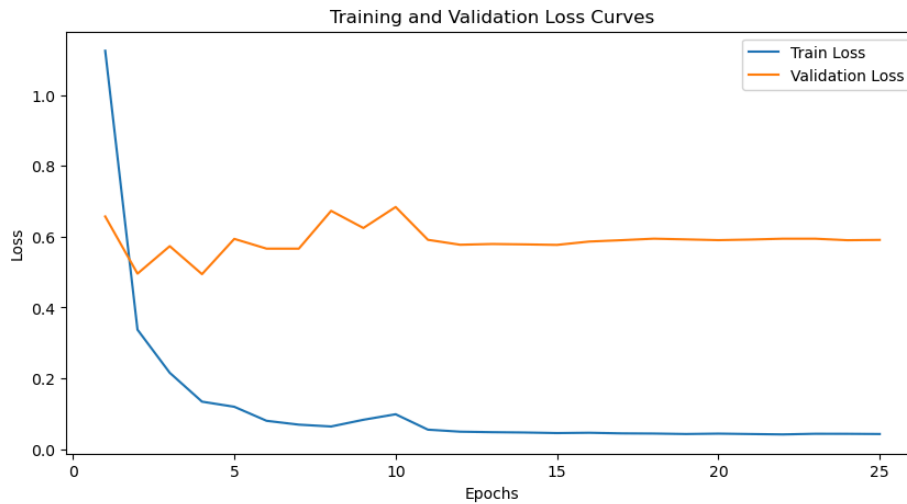
The loss curve graph produced by the best performing Densenet201 model on Dataset#2 is shown in Fig.10. This Figure illustrates that the training curve starts at a high loss, eventually decreasing over time, highlighting that the model begins to fit better on the training set with more epochs. The test loss curve, however, is widely spaced apart from the training loss curve suggesting that our best performing model Densenet201 needs to be simplified for our dataset.

The results of evaluating our most effective model, Densenet201, on the test dataset can be seen in Figure 11 through a confusion matrix. Based on the data from this matrix, the Densenet201 model classifies all poses almost correctly but confuses many poses with others. For instance, the model faces the most difficulty when classifying between ‘Tree’ and ‘Pigeon’ poses. It confuses

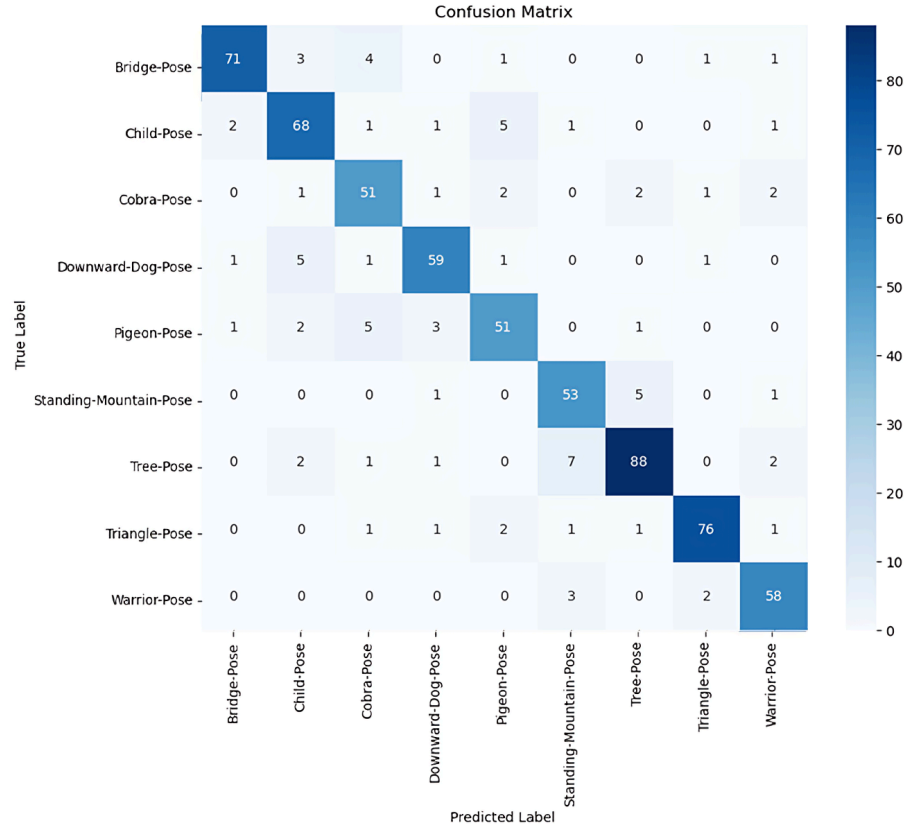
the ‘Tree’ pose most with the ‘Standing-Mountain’ pose whereas the ‘Pigeon’ pose is also incorrectly classified as the ‘Cobra’ pose.

**Table 4:** Comparison of overall results of five deep learning models on Dataset#2.

Model	Precision (%)	Recall (%)	F measure (%)	Test Accuracy (%)
ViT	84.82	84.80	84.73	85.80
Densenet201	<b>87.08</b>	<b>87.39</b>	<b>87.19</b>	<b>87.54</b>
ResNet50	73.35	73.55	73.36	73.55
InceptionV3	86.67	86.01	86.06	86.01
VGG19	82.53	81.61	91.81	81.61

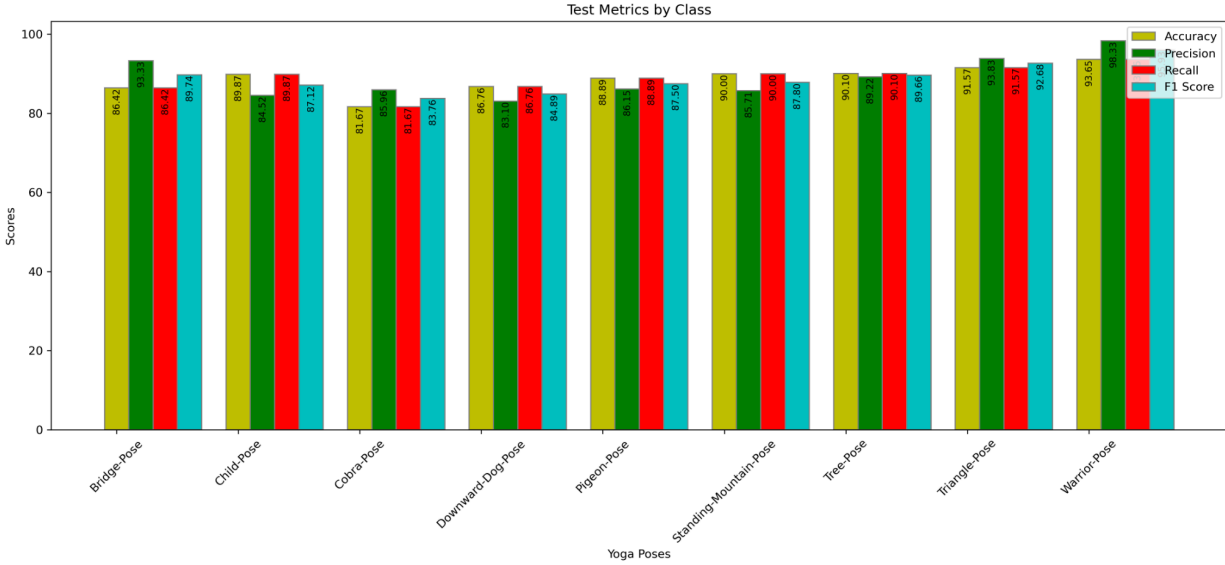


**Fig. 10:** Graph showing the loss curve for the best-performing Densenet201 model on Dataset #2.



**Fig 11:** Confusion matrix achieved by the best performing Densenet201 model on Dataset#2.

The class-wise recall, precision, accuracy and F1 Score, achieved by the best performing Densenet201 model after the evaluation on Dataset#2 are shown in Fig. 12. From Fig. 12, we can observe that class '8' has been classified with the highest accuracy, precision, recall, and accuracy whereas class '2' scores the lowest classification accuracy.



**Fig 12.** Visual comparison of the best-performing InceptionV3 model, based on, Precision, Recall, F1 Score, and Accuracy on Dataset #2.

#### **4.4 Comparison of our best model with previous works:-**

Table 5 compares all the previous works that have been done on yoga pose classification with our best-performing model, in terms of recall, accuracy, precision, and F measure. To ensure equitable comparison, we have compared the best results of the previous works with our best-performing model.

From Table 5, it's clear that the best-performing model on Dataset#1, Inceptionv3 has greater accuracy than almost all the other models, even the values of recall, precision, and F measure that we have achieved are greater than almost all the values of the previous models.

We have compared our best-performing model InceptionV3 with previous works done on Dataset. However, Dataset2 has been recently developed due to which previously many works have not been done, that is why we did not compare Densenet201, our best-performing model on Dataset2, with any other existing work.

**Table 5: Comparison of our best-performing model with previous works on Dataset 1.**

Authors	Algorithm	Accuracy (%)	Recall	Precision	F1 score
Sasi Kumar D et al. [12]	SCHF+IC O	98.47	0.98	0.98	0.98

V.Loganathan et al.[10]	NBU	78	0.78	0.78	0.78
Hema Krishnan et al.[6]	SVM	91	0.9071	0.9243	0.9141
Hema Krishnan et al[6]	Gaussian Naive Bayes	71	0.7294	0.6810	0.6906
Hema Krishnan et al. [6]	KNeighbors Classifier	81	0.786	0.7972	0.7901
Hema Krishnan et al. [6]	Random Forest Classifier	85	0.8145	0.7998	0.8071
Hema Krishnan et al. [6]	Gradient Boosting	89	0.8549	0.8146 0.	0.8342
P Charith et al.[11]	XG Boost	96	0.98	1	0.99
P Charith et al.[11]	Multilayer Perceptron	88	0.97	1	0.98
<b>Our model</b>	<b>Inception V3</b>	<b>98.29</b>	<b>0.9829</b>	<b>0.9830</b>	<b>0.9829</b>

## 5. Conclusion & Future Works

In our study, a comparative analysis was conducted on modern models like ViT, DenseNet201, ResNet50, InceptionV3, and VGG19 for yoga pose classification. The aim was to improve the form and technique of practitioners. Interestingly, distinct features of each model were revealed, providing insight into how well they classify poses. The versatility of these models has been demonstrated in our study, showing their potential to identify yoga poses accurately. Our models have been tested on 2 different datasets namely the Niharika Pandit dataset and the Mohan Dataset. According to our study, InceptionV3 has performed the best with an accuracy of 98.29% on Dataset#1 whereas the DenseNet201 model has demonstrated superior performance compared to others with a classification accuracy of 87.54 % on a very recently developed Dataset#2. Thus, a single model could not be found that outperforms all other models, on both datasets. There's still room for improvement and further exploration of other models, which can guarantee better accuracies across both the datasets and the finding of a single model capable of performing

optimally on all yoga pose datasets.. Moreover, our findings emphasize the wide range of applications these models have beyond yoga pose recognition, including tasks like facial recognition, emotion detection, and food identification, physical therapy, dance, and other activity recognition tasks. We may include ensemble learning models to combine two or more models to improve the accuracy and robustness of individual models for pose classification. The dataset can be expanded to include a wide variety of yoga poses along with diverse environments to model generalization and performance. More advanced augmentation techniques can be used such as synthetic data generation and domain adaptation. Other future works include the incorporation of wearable Tech like smartwatches and motion sensors to keep an eye on yoga practices consistently, making models easier to understand so that practitioners can grasp why the system gives feedback and corrections, and platforms for learning together where practitioners can work together, share progress, and get feedback from the community while using the model's classification features to build a supportive learning space.

## References

- 1.Sukrit Bhattacharya, Vaibhav Shaw, Pawan Kr. Singh, Ram Sarkar, Debotosh Bhattacharjee: “*SV-NET: A Deep Learning Approach to Video-Based Human Activity Recognition*”, In: Proc. of 11<sup>th</sup> International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019), Advances in Intelligent Systems and Computing, Vol. 1182, pp. 10-20, Springer, Cham, 2019
- 2.Swarnava Sadhukhan, Siddhartha Mallick, Pawan Kumar Singh, Ram Sarkar, Debotosh Bhattacharjee: “*A Comparative Study of Different Feature Descriptors for Video-Based Human Action Recognition*”, In: Intelligent Computing: Image Processing Based Applications, AISC 1157, pp. 35-52, 2020
- 3.Saikat Chakraborty, Riktim Mondal, **Pawan Kumar Singh**, Ram Sarkar, Debotosh Bhattacharjee: “*Transfer learning with fine tuning for human action recognition from still images*”, In: Multimedia Tools and Applications, Springer Publishers, Vol. 80, No. 13, pp. 20547-20578, 2021
- 4’Avinandan Banerjee, Sayantan Roy, Rohit Kundu, Pawan Kumar Singh, Vikrant Bhateja, Ram Sarkar: “*An ensemble approach for still image-based human action recognition*”, In: Neural Computing and Applications, Springer Publishers, Vol. 34, pp.19269–19282, 2022

5. Wang, X. (2022, March). Multi-classification for yoga pose based on deep learning. In *CIBDA 2022; 3rd International Conference on Computer Information and Big Data Applications* (pp. 1-4). VDE.
6. Krishnan, H., Jayaraj, A., Anagha, S., Thomas, C., & Joy, G. M. (2022, November). Pose estimation of yoga poses using ml techniques. In *2022 IEEE 19th India Council International Conference (INDICON)* (pp. 1-6). IEEE.
7. Agrawal, Y., Shah, Y., & Sharma, A. (2020, April). 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). Ieee.
8. <https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset>
9. <https://www.kaggle.com/datasets/amohankumar/yoga-pose-classification-dataset>
10. Loganathan, V., Suganthi, N., Mary, L. J., & Mohanaprakash, T. A. Classifications of Yoga Poses by through Image enhancement CLF Technique.
11. Spatial Structure-oriented and Angle-based Human Pose Estimation for Pose Classification - P Charith, Prajwal Kumar B R, Anitha M - IJFMR Volume 5, Issue 6, November-December 2023.
12. Sasi, K. D., Venkatachalam, K., Saravanan, P., Mohan, E., & Nagarajan, M. (2023, April). Meta Models of Yoga gestures by ACCF and SCHF with ML techniques. In *2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)* (pp. 1-5). IEEE.
13. Bodempudi, S., Nikhitha, K. D., Niharika, T., Snehit, N., Shameem, S., & Namgiri, J. V. (2023, November). Mudras and Yoga Positions Detection and Recognition using YOLOv7 and Faster R-CNN. In *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1254-1259). IEEE.
14. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30(2017).
16. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
17. Jaiswal, A., Gianchandani, N., Singh, D., Kumar, V., & Kaur, M. (2021). Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. *Journal of Biomolecular Structure and Dynamics*, 39(15), 5682-5689.
18. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

19. Kim, Y., Hwang, I., & Cho, N. I. (2017). A new convolutional network-in-network structure and its applications in skin detection, semantic segmentation, and artifact reduction. *arXiv preprint arXiv:1701.06190*.

20. Lagunas, M., & Garces, E. (2018). Transfer learning for illustration classification. *arXiv preprint arXiv:1806.02682*.