A COMPARATIVE STUDY OF DEEP LEARNING AND ENSEMBLE METHODOLOGIES WITH KEYPOINT DETECTION FOR 2D IMAGE-BASED YOGA POSE RECOGNITION

A MAJOR PROJECT-II REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by

DEBASHREE DEBALAXMI

2K22/ISY/06

Under the supervision of

PROF. DINESH K. VISHWAKARMA and DR. VIRENDER RANGA

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY, 2024

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I, Debashree Debalaxmi, Roll No – 2K22/ISY/06 student of M.Tech (Information Systems), hereby declare that the project Dissertation titled "**A Comparative Study of Deep Learning and Ensemble Methodologies with Keypoint Detection for 2D Image-Based Yoga Pose Recognition**" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **DEBASHREE DEBALAXMI**

Date: 30/05/2024

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE

Certified that Debashree Debalaxmi (2K22/ISY/06) has carried out her research work presented in this thesis entitled "A Comparative Study of Deep Learning and Ensemble Methodologies with Keypoint Detection for 2D Image-Based Yoga Pose Recognition" in partial fulfilment of the requirement for the award of the degree of Master of Technology from Department of Information Technology, Delhi Technological University, Delhi, under our supervision. The thesis embodies results of original work, and studies are carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature	Signature
Prof. Dinesh Kumar Vishwakarma	Dr. Virender Ranga
SUPERVISOR	**CO-SUPERVISOR**
Head of Department	Associate Professor
Department of Information Technology	Department of Information Technology
DELHI TECHNOLOGICAL UNIVERSITY	DELHI TECHNOLOGICAL UNIVERSITY

Place: Delhi

Date: 30/05/2024

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to PROF. DINESH KUMAR VISHWAKARMA and DR.VIRENDER RANGA for their continuous guidance and mentorship that they provided during the project. They showed the path to achieve the targets associated with this project by explaining all the tasks to be done and the importance of this project as well as its industrial relevance. They were always ready to help and clear my doubts regarding any hurdles in this project. Without their constant support and motivation, this project would not have been successful.

Place: Delhi Debashree Debalaxmi

Date: 30/05/2024

Abstract

Yoga is a 5000 year old Indian spiritual practise that seeks to achieve balance between the body and mind, with the use of asanas, meditation, plus a variety of breathing exercises/techniques. Yoga is now widely used as a remedy for dealing with the rising levels of tension and anxiety while adjusting to modern lifestyles.

We can reap the greatest health benefits from yoga by adopting the ideal postures and adhering to the recommended techniques and sequencing. A multitude of techniques can be used to learn yoga, such as through attending courses at yoga studios, viewing films, perusing photographs, or reading books. The majority of people prefer self-learning at home due to their fast-paced existence. However, many find it challenging to see defects in their own yoga positions. However, adopting improper postures while practising yoga can result in a number of health issues, including short-term chronic issues and acute muscle discomfort.

Therefore, there is a need for scientific evaluation of incorrect yoga posture detection and correction through providing feedback in order to help people practise yoga effectively. To support self-learning, we'll provide a model for classifying poses using posture detection in this research. Users will pick a yoga stance to practise here and upload a picture or a video of it. The user pose is supplied to training models, which figure out and output the differences between the user pose's body angles and the real pose that was detected. With this output, the model instructs the user on how to enhance the position by pointing out the areas of their yoga posture that need work.

Contents

Candidate's Declaration i

Certificate ii

Acknowledgement iii

Abstract iv

Content vi

List of Tables vii

List of Figures viii

List of Symbols, Abbreviations ix

1 INTRODUCTION 1
1.1 Motivation and Objective of the project 1
1.2 Importance of Yoga: Health Benefits and Associated Injuries 3
1.3 Computer Vision in Pose Estimation 3

2 LITERATURE REVIEW 4
2.1 Microsoft Kinect-Based Approaches 4
2.2 CNN and Transfer Learning Approaches 5
2.3 RNN and LSTM Approaches 5
2.4 Keypoint-Based Approaches 6
2.5 Hybrid Approaches 7
2.6 Research Gap 8

3 METHODOLOGY 9
3.1 Tools used 9
3.1.1 Python 3.10 9
3.1.2 Tensorflow 10
3.1.3 Pandas 10
3.1.4 Matplotlib 10
3.1.5 Keras 10
3.1.6 Google Colab 10
3.2 Dataset Used 11
3.3 Proposed Approach 1 13
3.3.1 Data Pre-Processing and Augmentation 13
3.3.2 Model Architecture and Training 14

3.3.2.1 ResNet50 . . . 14
3.3.2.2 VGG16 . . . 14
3.3.2.3 VGG19 . . . 15
3.3.2.4 EfficientNetB0 . . . 15
3.3.2.5 MobileNetV2 . . . 16
3.3.2.6 InceptionV3 . . . 16
3.3.2.7 Xception . . . 17
3.3.2.8 DenseNet121 . . . 18
3.3.2.9 DenseNet201 . . . 18
3.3.3 Model Evaluation and Fine-Tuning . . . 19
3.3.4 Comparative Study . . . 19
3.4 Proposed Approach 2 . . . 20
3.4.1 Dataset Collection and Preprocessing . . . 20
3.4.2 Keypoint Extraction . . . 20
3.4.2.1 MediaPipe . . . 20
3.4.2.2 YOLOv8 . . . 21
3.4.3 Data Augmentation . . . 22
3.4.4 Model Training . . . 22
3.4.4.1 Linear SVM . . . 22
3.4.4.2 Decision Tree . . . 23
3.4.4.3 K-Nearest Neighbors . . . 23
3.4.4.4 Multi-Layer Perceptron . . . 24
3.4.4.5 GaussianNB . . . 24
3.4.4.6 Bagging . . . 25
3.4.4.7 Random Forest . . . 25
3.4.4.8 XGBoost . . . 26
3.4.4.9 LightGBM . . . 27
3.4.5 Model Testing & Evaluation . . . 27

4 RESULTS and DISCUSSION **28**
4.1 Approach 1 Result . . . 29
4.1.1 Output . . . 33
4.2 Approach 2 Result . . . 34
4.2.1 Output . . . 37
4.3 Web Deployment . . . 38

5 CONCLUSION AND FUTURE SCOPE **40**

Bibliography **42**

List of Publications **46**

List of Tables

4.1 Accuracy, Precision, Recall, and F1-Score for different Deep Learning Models 29
4.2 Posewise Accuracy for DenseNet201 Model 31
4.3 Model Parameters for YOLOv8 and MediaPipe using Grid Search 34
4.4 Comparison of Different Models' Accuracy (YOLOv8 vs. MediaPipe) . . . 34
4.5 Performance Metrics of Different Algorithms 35

List of Figures

1.1 Diverse yoga poses 2

3.1 Distribution of images across the dataset 11
3.2 Yoga Poses in the Dataset 12
3.3 Proposed Approach 1 13
3.4 ResNet50 Architecture 14
3.5 VGG16 Architecture 15
3.6 VGG19 Architecture 15
3.7 EfficientNetB0 Architecture 16
3.8 MobileNetV2 Architecture 16
3.9 InceptionV3 Architecture 17
3.10 Xception Architecture 17
3.11 DenseNet121 Architecture 18
3.12 Code-specimen for DenseNet201 customization 18
3.13 DenseNet201 Architecture 19
3.14 Proposed Approach 2 20
3.15 Landmark detection in a yoga image 21
3.16 Linear SVM Working 22
3.17 Decision Tree Working 23
3.18 K-Nearest Neighbors Working 23
3.19 Multi-Layer Perceptron Working 24
3.20 GaussianNB Working 24
3.21 Bagging Working 25
3.22 Random Forest Working 26
3.23 XGBoost Working 26
3.24 LightGBM Working 27

4.1 Confusion Matrix of DenseNet201 model 30
4.2 Accuracy and Loss Curves of DenseNet201 model 32
4.3 AUC-ROC Curve of DenseNet201 model 32
4.4 Actual vs Predicted labels of test yoga images 33
4.5 Confusion Matrix using MediaPipe+LightGBM model 35
4.6 Accuracy and Loss curves using MediaPipe+LightGBM model 36
4.7 Learning Curve for LightGBM model 36
4.8 AUC-ROC curve using MediaPipe+LightGBM model 37
4.9 True vs Predicted labels of some test images of poses 38
4.10 Code-specimen of gradio for web-demo 38
4.11 Web-demo view using Gradio 39

List of Symbols

API	Application programming interface
AUC	Area Under The Curve
CNN	Convolutional Neural Network
$CNTK$	Microsoft Cognitive Toolkit
GAP	Global Average Pooling
GBM	Gradient Boosting Machine
GNN	Graph Neural Network
GPU	Graphics Processing Unit
IoT	Internet of Things
$JSON$	JavaScript Object Notation
KNN	K-Nearest Neighbors
$LSTM$	Long Short-Term Memory
MLP	Multilayer Perceptron
NB	Naïve Bayes
$PTSD$	Post-Traumatic Stress Disorder
RGB	Red Green Blue
RNN	Recurrent Neural Network
$SMOTE$	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
VGG	Visual Geometry Group
$XGBoost$	Extreme Gradient Boosting
$YOLO$	You Only Look Once

Chapter 1

INTRODUCTION

In today's globe, the search of alternative health has actually expanded in relevance, and also yoga exercise is a lighthouse that supplies a way to locate equilibrium in the center of the turmoil. Presently there more than 300 million individuals exercising yoga exercise worldwide. With proper positions by adhering to appropriate methods coupled with series of yoga exercise, we can acquire out limit health and wellness advantages. There are more than 80+ yoga asanas in the world. Some yoga poses are shown in figure 1.1 There are different means to find out yoga exercise e.g. by going to courses in yoga exercise centres, seeing video clips considering photos or checking out publications. With hectic way of lives the majority of individuals like exercising in the house. However it is rather bothersome for them to locate problems in their very own positions. Nevertheless embracing inaccurate positions while exercising yoga exercise can bring about numerous health issue like joint discomfort as well as muscle mass stress or strains and so on [1]. As a result to aid individuals to execute yoga exercise in a reliable way there is a demand for clinical analysis of improper yoga exercise stance recognition and also adjustment by offering comments. To sustain self-learning our research study takes on a crucial part of self-learning in yoga exercise technique which is the demand for exact yoga exercise setting acknowledgment.

1.1 Motivation and Objective of the project

The inspiration behind this task is driven from the merging of 2 significant locations: the expanding appeal of yoga exercise as an all natural health method as well as the fast growth in computer system vision as well as deep discovering modern technologies. Typical strategies for position recognition often depend on by hand generated attributes as well as standard designs, that make it tough to precisely find keypoints in difficult yoga exercise presents. Convolutional semantic networks (CNNs) particularly have actually shown considerable efficiency in deep discovering for a selection of computer system vision applications consisting of present price quote. To boost efficiency set approaches incorporates lots of versions; as a result it is needed to examine as well as contrast the success price of deep discovering versions with these methods. This sort of relative analysis has the capacity to improve not just the area of yoga exercise setting recognition yet additionally the wide understanding of set strategies and also deep discovering in computer system vision applications.

The vital purpose of this argumentation research is to carry out a relative evaluation in between deep understanding together with set approaches with keypoint discovery for 2D image-based yoga exercise present acknowledgment. The research study intends to attain the complying with purposes:

1. **Efficiency Evaluation**: Evaluate the efficiency of state-of-the-art deep understanding designs consisting of CNNs as well as their variations in precisely spotting key-points related to different yoga exercise presents in 2D photos.

2. **Set Methodologies**: Investigate the efficiency of set strategies, such as bagging, improving as well as piling, in enhancing the precision as well as strength of present acknowledgment contrasted to private deep understanding designs.

3. **Information Acquisition and also Annotation**: Curate or take advantage of existing information having annotated 2D photos of yoga exercise presents making sure variety in postures, body kinds plus ecological problems to assist in thorough analysis as well as generalization of the designs.

4. **Attribute Representation**: Explore various function depiction strategies consisting of raw pixel worths, handmade functions, coupled with discovered depictions to catch spatial plus temporal details essential for exact position acknowledgment.

5. **Benchmarking as well as Comparison**: Conduct strenuous benchmarking experiments to contrast the efficiency of deep understanding designs plus set techniques in regards to precision, computational effectiveness as well as strength throughout various yoga exercise presents along with variants.

6. **Sensible Applications**: Investigate the useful usefulness as well as use of the suggested methodologies in real-world situations, such as yoga exercise training applications, interactive comments systems as well as automated present adjustment devices.

By resolving these goals, this research intends to offer beneficial understandings right into the staminas and also constraints of deep discovering as well as set techniques for 2D image-based yoga exercise position acknowledgment eventually adding to the growth of a lot more efficient plus available modern technologies for yoga exercise professionals and also trainers.

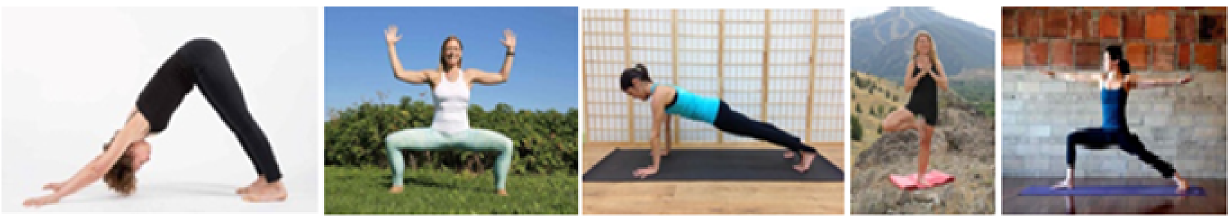


Figure 1.1: Diverse yoga poses [2]

1.2 Importance of Yoga: Health Benefits and Associated Injuries

From enhancing versatility, toughness as well as equilibrium to minimizing anxiety, anxiousness plus clinical depression the method of yoga exercise has actually been clinically verified to boost numerous facets of wellness. In addition yoga exercise advertises mindfulness as well as self-awareness, promoting a much deeper link in between mind, body and also spirit. Among the essential advantages of yoga exercise is its capacity to enhance physical conditioning. Yoga exercise presents or asanas assistance to extend together with enhance muscle mass enhance joint adaptability and also boost position. Routine technique of yoga exercise can result in raised muscle mass tone, boosted endurance, as well as boosted general physical fitness degrees.

Along with physical advantages, yoga exercise additionally has a favorable effect on psychological wellness. The technique of yoga exercise includes breath control as well as reflection, which can assist to relax the mind, decrease anxiety, as well as boost psychological quality. Yoga exercise has actually been revealed to be reliable in minimizing signs and symptoms of anxiousness, clinical depression together with PTSD and also can boost general psychological health.

Regardless of its lots of advantages exercising yoga exercise can likewise position dangers especially when presents are done inaccurately or without correct advice. Usual yoga-related injuries consist of pressures, stress and also overuse injuries, especially in locations such as the wrists, shoulders as well as reduced back. It is essential for professionals to obtain appropriate guideline together with assistance and also to pay attention to their bodies to stay clear of injury.

1.3 Computer Vision in Pose Estimation

The use of computer vision systems for the purpose of pose estimation, an area of expert system, concentrates on making it possible for makers to analyze along with comprehend aesthetic info from the real life. Pose evaluation a subfield of computer system vision, includes identifying and also tracking the placements of vital keypoints in images/video clips. By precisely approximating positions from aesthetic information, computer system vision systems can help in numerous applications, consisting of human-computer communication, sporting activities evaluation, plus health care.

In the context of yoga exercise posture acknowledgment, computer system vision methods play an essential function in immediately recognizing as well as evaluating yoga exercise postures from pictures or video clips. These methods make use of deep understanding formulas such as Convolutional Neural Networks (CNNs) plus Recurrent Neural Networks (RNNs) to discover intricate patterns as well as partnerships in yoga exercise postures. In addition, keypoints discovery formulas such as OpenPose and also PoseNet make it possible for the accurate localization of vital body joints assisting in exact present price quotes.

By leveraging developments in computer system vision, scientists together with designers objective to produce durable coupled with effective yoga exercise posture acknowledgment systems that can give beneficial responses to experts, improving their yoga exercise experience and also advertising more secure as well as extra reliable method sessions.

Chapter 2

LITERATURE REVIEW

To have a strong understanding of the numerous methods for approximating yoga exercise presents a few of the more recent operate in this area are examined. A selection of strategies have actually been used to approximate yoga exercise pose. Skeletal as well as non-skeletal strategies are generally used in yoga exercise present acknowledgment as well as category systems. Skeletal yoga exercise present acknowledgment recognizes joint placements utilizing designs like OpenPose, PoseNet and so on while non-skeletal category identifies presents based upon general aesthetic functions utilizing designs like ResNet, DenseNet and so on. The literary works includes a lot of documents referring to yoga exercise professional systems plus position category making use of numerous information mining methods consisting of SVM, Random Forest, Decision Trees along with various other Deep Learning structures based upon CNN.

2.1 Microsoft Kinect-Based Approaches

Microsoft Kinect is commonly utilized for position discovery in the bulk of yoga exercise present acknowledgment study. Chen et al. [3] recommended a yoga exercise training system in their research study that researches individual's poses utilizing the Y-system. Popular axes, skeleton-based attributes and also contourbased landmarks were all included in this system. An overall of 300 movies with 5 yoga exercise professionals implementing 12 various placements 5 times each were collected. At first, the body map was taken from the motion pictures and also the body shape was gotten. For efficiency contrast, Microsoft Kinect as well as the OpenNI collection were utilized to construct skeletal representation for yoga exercise placements. The system utilized a deepness sensing unit along with RGB video camera to collect deepness information for monitoring together with approximating body pose. This strategy was great however its basic drawback was the expensive price plus absence of easy to use nature of the Kinect tool.

Edwin et al. [4] have recommended a yoga exercise acknowledgment system that makes use of Kinect v2 as well as Ada-boost category to identify 6 asanas with an precision rating of 94.78%. Nevertheless the approach takes even more handling time for even more challenging designs. They made use of a video camera that is based upon a deepness sensing unit which is often out of grab for clients.

Shruti et al. [5] used Microsoft Kinect to record video clip information. Kinect consists of a skeletal monitoring device efficient in identifying twenty joints in the body. They picked 10 certain joint indicate for their estimations making use of the information collected from these suggest produce a referral framework for each and every yoga exercise present. They after that determined the resemblance of vectors identifying the angles linking any type of 2 joint factors. If the determined angle drifted greater than a predefined limit worth, the present was identified as wrong.

2.2 CNN and Transfer Learning Approaches

Ajay et al. [6] consisted of 5 yoga exercise positions, specifically Natarajasana, Virbhadrasana 1, Virbhadrasana 2, Vrikshasana, Trikonasana, together with Utkatasana in a dataset that was constructed making use of internet sources as well as payments from system programmers. In order to recognize pose blunders, the recommended deep knowing method included a human joints Localization design right into its deep semantic networks (CNN) for yoga exercise present recognition. The constraint of this research study was the information made use of which makes up an extremely couple of groups.

The Yoga-82 dataset was used by Verma et al. [7] to execute keypoint acknowledgment. The dataset initially consisted of a range of yoga exercise creates that were collected from the web. The positions were by hand cleaned up and also organized right into a superclass power structure. For network understanding this ordered annotation provided abundant information that consisted of present information, spinal column results and also body placements. 3 ordered layers, having 6, 20 as well as 82 courses at each degree, were utilized to structure their dataset. DenseNet-201 executed extremely well, achieving 74.91% top-1 category precision.

To determine 5 various yoga exercise placements with marginal information, Ashraf et al. [8] recommended a deep discovering approach influenced from Xception version. Their YoNet design purposefully leverages dimensional and also spatialinformation out of the picture and also utilizes them to assist with category.

Chirumamilla et al. [9] made use of some prominent CNN designs like AlexNet, VGG16 as well as ResNet18 in their research study. The AlexNet design was constructed with 8 layers, as well as the very first 5 layers are semantic network layers adhered to by 2 completely linked layers plus a SoftMax layer. 8 AlexNet layers, 16 VGG layers, as well as 18 ResNet layers were educated and also examined with 13 courses of yoga exercise with a dataset of 2129 images. In this method, deep understanding standards other than AlexNet do even worse than various other versions. This version categorizes the yoga exercise presents with an precision of 30% in VGG16 60% in ResNet18 and also 83.05% in AlexNet.

Long et. al. [10] made use of a. dataset collected from 8 volunteers to create a yoga exercise. training system that used 6 transfer understanding. versions. They separated the dataset right into train, recognition and also. examination establishes at a proportion of 3:1:1 in order to educate their design. They. likewise made use of approaches for information enhancement on the dataset. The most effective version was figured out to be the TL-MobileNet-DA version as indicated by its 98.43% complete precision.

2.3 RNN and LSTM Approaches

Debabrata et al. [11] recommended a system for tracking the. various yoga exercise positions. It begins by determining 33 factors of an individual utilizing the MediaPipe collection catching important works with conserved in JSON layout. Ultimately a collection of 45 frameworks is produced in time. Fed right into the design. The version which incorporates CNN coupled with LSTM removes includes via CNN. Evaluates the series of frameworks making use of LSTM. Ultimately the Softmax layer establishes the chance of every yoga exercise present for the framework series as well as picks the present with the possibility. Furthermore each structures outcome is examined over 45 frameworks to determine along with offer the setting as the outcome to the customer. This system shows a precision price of 99.53%, on the examination dataset.

A system that might identify. coupled with change yoga exercise placements by contrasting the expert. video clip with the referral video utilizing the SURF formula. was provided by Fazil et al. [12]. Utilizing Mask RCNN the scientists transformed the dataset right into reefs. A CNN & LSTM network generated ideal precision of 0.9996.

Kothari et al. [13] carried out an extensive examination on the category of yoga exercise poses utilizing deep knowing algorithms as well as human present price quote. They utilized OpenPose to pre-process the ""Yoga exercise Vid Collected"" data source in order to remove pose keypoints. A contrast of deep understanding and also artificial intelligence methods for approximating yoga exercise presents was executed as component of the research study. In regards to misclassifications the crossbreed CNN-LSTM design subdued the SVM plus standalone CNN designs, displaying the least mistakes with recognition accuracy of 0.9987.

Sumeet et al. [14] explores different deep discovering versions for acknowledging yoga exercise presents in real-time. The research contrasts a crossbreed CNN & LSTM design plus numerous 3DCNN versions pre-trained on the Sports1M data source. Making use of a public yoga exercise position data source, the versions were examined for precision, accuracy, recall together with F1-score. The most effective version attained high precision coupled with a structure price of 31 frames/sec on an Nvidia GPU as well as 8 FPS on an Nvidia Xavier system showing their sensible application capacity for reliable yoga exercise position acknowledgment.

2.4 Keypoint-Based Approaches

Gradient Boosting (XGBoost) was put on the OpenPose device by Kakulapati et al. [15] in order to classify. pictures of 5 various positions. They separated the data source. where 80% were booked to educate the version, while the. staying is for screening dealing with the X plus Y works with as. important attributes. With a precision of 88%, the XGBoost. category made the best outcome.

In order to develop an arbitrary woodland category for examining. yoga exercise asanas, Mustafa et al. [16] utilized transfer human understanding. placement price quote strategies to determine 136 body-wide. crucial factors. They make use of the preferred position evaluation. approach AlphaPose in the initial phase and after that an arbitrary woodland. category in the 2nd. They developed a new data source that emphasizes various point of views in order to classify yoga exercise. positions. They depended on precise keypoint acknowledgment and also. present price quote.

Vivek et al. [17], established a deep learning-based Yoga setting evaluation. system that can identify position while likewise supplying. talk about exactly how to boost it. It makes use of 12. determined joints pressed right into 13 vectors to review. posture precision as well as draws out 18 keypoints. According to. their evaluation the MLP classifier had a remarkable screening. precision of 0.9958. However, inaccurate body map. segmentations lead to mistakes in shown standards.

Shubham et al. (2022) [18] recommended a CNN-based. method for categorizing yoga exercise presents in a various research study. They made use of the MediaPipe collection in their research to fix. essential areas as well as make a skeletal system layout. The. researchers downloaded and install a data source from Kaggle that was. offered to the general public and also they acquired the best. precision of 97.09%.

2.5 Hybrid Approaches

Aman et al. [19] offered the threestage Y-PN-MSSD standard for yoga exercise present acknowledgment. Initially, it utilizes an open-source data source with 7 postures and also documents the positions of 4 individuals to accumulate information. Throughout training, it after that links vital body parts to draw out. attributes. This version permits 99.88% precision in. real-time surveillance of individuals in yoga exercise presents. Nonetheless. application still encounters problems as a result of troubles. like blockage as well as unequal lighting.

Vallabhaneni et al. [20] presented a fabricated algae optimizer with a crossbreed deep learning-based yoga exercise position estimation design (AAOHDL-YPE). The AAOHDL-YPE version assesses yoga exercise video to approximate presents. It uses OpenPose, which makes use of Part Confidence Map as well as Part Affinity Field with balanced matching as well as parsing, to recognize joint areas. The deep idea network (DBN) version is after that made an application for yoga exercise position acknowledgment. Lastly the AAO formula is used in an attempt to improve EfficientNet's design acknowledgment efficiency. Detailed speculative evaluation reveals that the AAOHDL-YPE strategy surpasses existing techniques.

Santosh et al. [21] used a crossbreed of CNN & amplifier; LSTM in their research.CNN was frequently utilized for pattern acknowledgment problems while LSTM was made use of for time-series information applications. This CNN network is time-distributed, and was used to draw out the function from 2-dimensional worked with keypoints created within the coming before phase. That forecasted limit worth was utilized to recognize a structure in which the individual is not exercising yoga exercise as well as the effect on surveys of structures is being discovered. This version forecasts the yoga exercise presents in every video clip with precision at a good 99.04%. for frame-wise amplifier; 99.38% precision on the survey of 45 structures, and also it achieves 98.92% precision for a team of twelve varied individuals presenting its capacity to carry out 6 postures successfully.

Si et al. [22] presented Wiga, A sophisticated design that combines CNN and LSTM is used to extract high level features. By leveraging CSI data a comprehensive model is created to link signal variations caused by motion to tasks. Beginning with the CSI inputs Wiga and does signal filtering and recurring components. It then infers the underlying features with CNN. Captures the dependencies of the resource with LSTM. This version was reviewed with 17 yogic poses done by 7 experts. Wiga attains 97.7%. & amplifier; 85.6% precise yogic position leads to the skilled, and also. inexperienced yoga exercise individuals.

2.6 Research Gap

Regardless of the substantial improvements in yoga exercise position estimate along with category utilizing different computer system vision and also deep understanding strategies, numerous voids as well as difficulties stay:

1. **Limited Diversity of Yoga Poses**: Many existing research study has actually focused on a little collection of famous yoga exercise presents, usually 5 to 7. However real-world yoga exercise method includes numerous variants of yoga that are yet to be taken into consideration in the system for yoga pose recognition. For this, datasets of many diverse poses are required to educate plus examine the systems properly.

2. **Similarity in Yoga Poses**: Many yoga exercise presents have refined distinctions in pose as well as body placement, making them tough to identify. Present designs usually battle with these subtleties causing misclassifications. Improving the capacity of designs to manage these refined distinctions is crucial for boosting posture acknowledgment precision.

.

3. **Occlusion and Lighting Conditions**: Real-world applications commonly entail differing lights problems as well as partial occlusions of the expert's body. Existing designs which are mainly educated on suitable problems might not do well under these difficult situations. Research study requires to resolve these ecological variables to make the versions a lot more durable and also trusted.

.

4. **Real-Time Feedback and Guidance**: While some systems supply real-time responses plus advice, there is still a demand for even more interactive and also straightforward services that can supply prompt improvements and also changes throughout method. Boosting the interactivity as well as responsiveness of these systems can considerably boost the customer experience.

.

5. **Integration with Wearable Devices**: Incorporating information from wearable gadgets such as smartwatches plus physical fitness trackers can offer added understandings right into the specialist's motions together with enhance posture estimate precision. Nevertheless, research study in this field is still in its early stage, as well as additional expedition is required to incorporate these information resources efficiently.

6. **Ensemble Learning Approaches**: Although ensemble discovering methods have actually revealed assurance in various other domain names their application in yoga exercise present acknowledgment continues to be restricted. Checking out set techniques that incorporate several versions or formulas can possibly boost the precision and also dependability of present category.

.

7. **Scalability and Accessibility**: Many existing services call for specialized equipment such as deepness sensing units or high-performance computer sources which are not obtainable to the ordinary individual. Establishing scalable along with economical services that can work on common tools like mobile phones or laptop computers will certainly make yoga exercise present quote modern technology much more available to a more comprehensive target market.

Chapter 3

METHODOLOGY

The study utilised a two-step process to deal with the challenge of recognizing yoga poses in images. Given the job's intricacy, an extensive and meticulous strategy was established, integrating the strengths of both deep learning and ensemble based methods. By exploring these processes, the research intended to create a strong framework which is able to reliably identify yoga poses in variety of contexts. The following paragraphs dig into the process, emphasising the evaluation and its rationale.

The dataset was carefully set up to integrate a variety of yoga exercise presents caught in varied setups, with differing lights, point of views, people, as well as presents, assuring a sufficient depiction of real-world yoga exercise circumstances.

In the very first phase, deep knowing designs established for picture acknowledgment were related to the picture information. Designs were thoroughly educated as well as tweaked to allow accurate posture discovery and also their efficiency was evaluated making use of several metrics.

In the 2nd phase, keystone recognition approaches particularly YOLOv8 as well as MediaPipe, were used to discover the crucial spots of the yoga exercise presents. After that a variety of artificial intelligence along with ensemble-based strategies were educated on the keystones for categorising yoga exercise presents. A detailed evaluation and also contrast of the outcomes were done for both the methods.

The intricate nature of yoga exercise present acknowledgment triggered considerable examination to boost applications in this setup. The end-goal of performing this research was to develop a very accurate as well as reliable system efficient in precisely determining yoga exercise stances from images using light-weight methods that permit quick reasoning.

3.1 Tools used

Python 3.10 was used for the experimental part, and the results are reliable. In addition, various models from machine learning and ensemble learning were trained by an end-to-end method on the 20 class yoga pose dataset. For hardware and software requirements, a Tesla T4 GPU, two virtual CPUs, and 32GB of RAM were available on the Google Colab network.

3.1.1 Python 3.10

Python 3.10 is the language of choice for the experiments in this thesis. Python is known for its readability, simplicity, and large coollection of well-maintained libraries and frameworks for scientific computing, machine learning, and data analysis. Python 3.10

has several improvements in performance as well as syntax, which increases the clarity and maintainability of the code.

3.1.2 Tensorflow

The framework: TensorFlow is a tool or a framework in machine learning released by Google, used commonly in research and production in machine learning. It is open-soruce and offers a rich collection of libraries, community resources and tools. These help researchers develop fast and try to push the features of present systems to beat state-of-the-art in ML. It is possible to easily build and deploy applications powered by ML, for developers. We used TensorFlow to build and train the deep learning models for yoga pose recognition.

3.1.3 Pandas

Pandas. Pandas is an impressive library in Python used in data manipulation and also for data analysis. It gives numerous features and supports multiple data structures which are essential to read, clean and manipulate data. We utilized Pandas to handle the dataset, including loading the data, processing and transforming it into formats we need for training and evaluation.

3.1.4 Matplotlib

The Matplotlib is a quite a prominent tool used for creating interactive and dynamic graphics in Python.. It is also open-source library and one of the most frequently used in Python scientific computing for plotting and visualization of data. We utilized Matplotlib to plot the distribution of data, show model performance, as well as illustrate our results. This enabled us to observe how the training data appears and how it impacts model accuracy with different types of training.

3.1.5 Keras

Keras is a API made for deep learning workloads, written in Python and running on TensorFlow. It falls under open-source. It has the ability to run with CNTK, or Theano. It enables focusing on experimentating fast and supports both convolutional and recurrent networks. Being user-friendly, it allows rapid model prototyping. In the current work, the Keras module was used to develop and train deep learning models for yoga pose estimation.

3.1.6 Google Colab

Google Colab is an environment that supports Python coding and is built around Jupyter Notebooks. It is usually used for machine learning and data science tasks. The virtual environment provided by Google Colab allows users to write Python and then execute it right in the browser and share with others. Google Colab is a cloud service which is free and is known for providing free access to GPU resources. The training of deep learning models is very time-intensive. To do this, we impacted Google Colab for the experiments.

3.2 Dataset Used

We carefully selected a total of 1248 images of 20 yoga postures, each with a range of 50 to 90 photos, using the dataset we obtained from Kaggle [2] and its distribution is shown in figure 3.1. We present a consolidated dataset with a variety of yoga poses from varied backdrops, lighting conditions, and viewpoints. Important meta data is attached to every image, including the file path, pixel values, and label that corresponds to the asana index. The yoga poses used here are 'Virabhadrasana i', 'Vajrasana', 'Savasana', 'Balasana', 'Bakasana', 'Setu bandha sarvangasana', 'Salabhasana', 'Vriksasana', 'Paschimottanasana', 'Uttanasana', 'Trikonasana', 'Virasana', 'Tadasana', 'Matsyasana', 'Virabhadrasana ii', 'Ardha matsyendrasana', 'Kapotasana', 'Adho mukha svanasana', 'Bhujangasana' and 'Dhanurasana' as shown on figure 3.2. The proposed method seeks to include a greater variety of poses than existing work, including poses with similar alignments, like 'vajrasana' and 'virasana'. This varied dataset will aid in better understanding for evaluation of the models as they may misclassify and get confused.

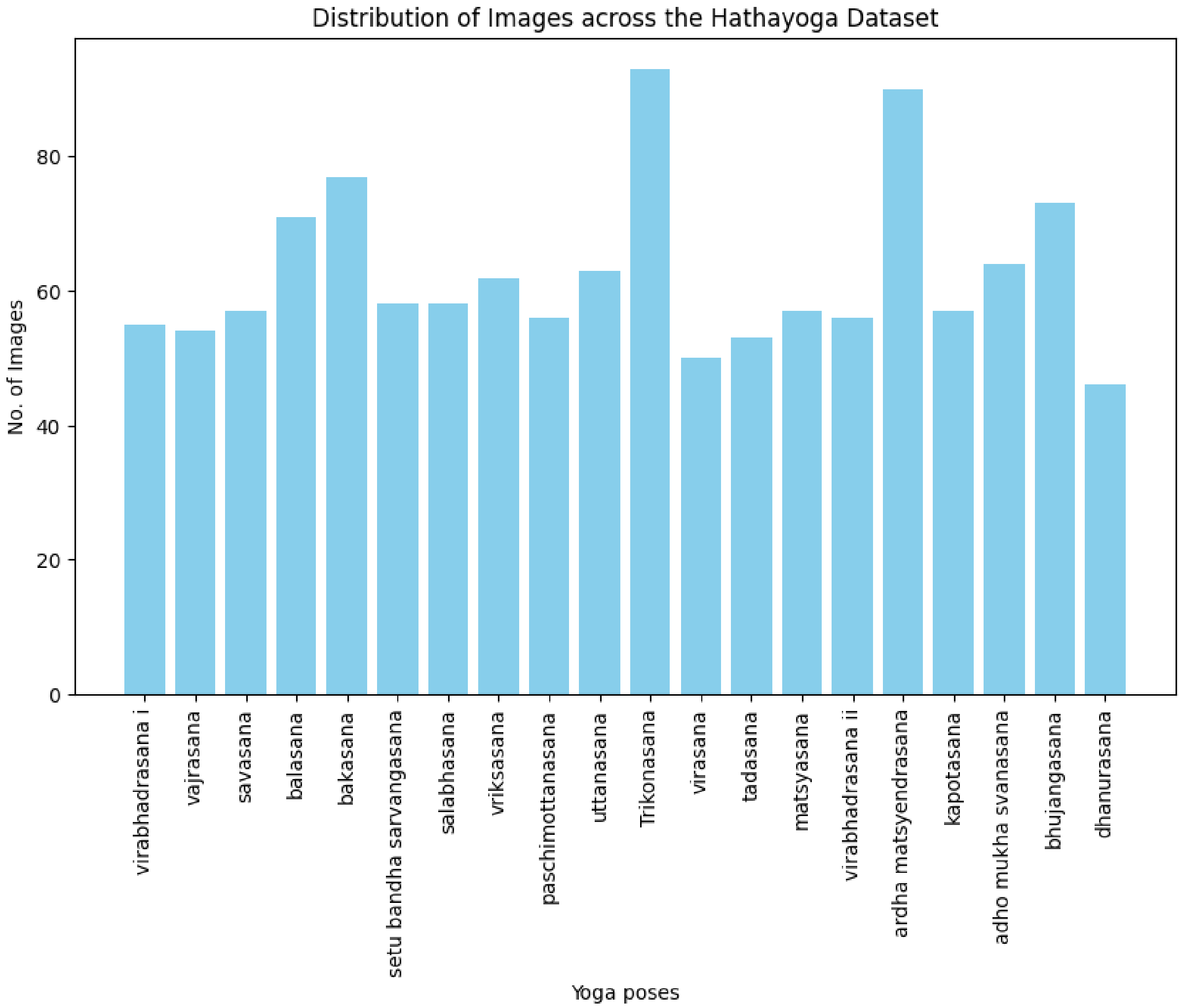


Figure 3.1: Distribution of images across the dataset

Figure 3.2: Yoga Poses in the Dataset [2] : (a) Adho Mukha Svanasana, (b) Ardha Matsyendrasana, (c) Bakasana, (d) Balasana, (e) Bhujangasana, (f) Dhanurasana, (g) Kapotasana, (h) Setu Bandha Sarvangasana, (i) Tadasana, (j) Uttanasana, (k) Vriksasana, (l) Virabhadrasana ii, (m) Vajrasana, (n) Virasana, (o) Virabhadrasana i, (p) Trikonasana, (q) Paschimottanasana, (r) Matsyasana, (s) Savasana and (t) Salabhasana

3.3 Proposed Approach 1

The process of creating a yoga posture identification system includes gathering a variety of picture datasets, preprocessing information, choosing models that have already been trained, fine-tuning, assessing performance, optimizing hyperparameters, deploying the best model as shown in figure 3.3.

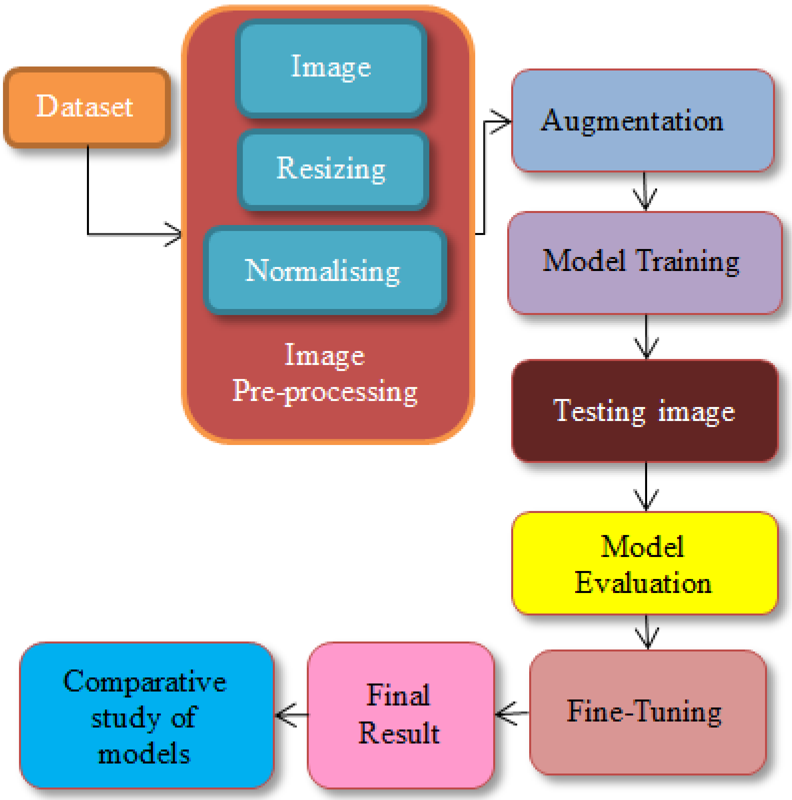


Figure 3.3: Proposed Approach 1

3.3.1 Data Pre-Processing and Augmentation

To guarantee consistency, images are loaded using OpenCV (cv2) and scaled to a standard size of 224x224 pixels. Each picture is given a label according to the asana to which it belongs, and the target variable is created by one-hot encoding these labels. To create a random sequence for the photos and labels, the dataset is shuffled. Then input data is normalized using the standard normalization values [-1, 1]. Then we created training and testing sets from the dataset. Around 80% of the time is usually spent on training and 20% on testing. Data augmentation is carried out using Keras' ImageDataGenerator. Techniques for augmentation include rotating, moving, zooming, and flipping both horizontally and vertically. Data augmentation exposes the model to different picture modifications, which improves its ability to generalize.

3.3.2 Model Architecture and Training

Weights from models which are pre-trained on ImageNet and other massive datasets, were used in transfer learning. As a result, the models were able to use information gleaned from generic picture attributes. Feature extraction involves extracting hierarchical features from input photos by means of the convolutional layers of pre-trained models. In order to decrease spatial dimensions and concentrate on important features, Global Average Pooling (GAP) layers were added. To build a custom classifier for yoga pose classification, dense layers with suitable activation functions, dropout, and batch normalization were included. Models were assembled using appropriate evaluation metrics (accuracy), loss functions (categorical crossentropy), and optimizers (e.g., Adam). Models were first trained using the yoga pose dataset for a predetermined number of epochs, early stopping to avoid overfitting. The best performing model was captured.

3.3.2.1 ResNet50

ResNet50, the most popular of the three, is a deep convolutional neural network with 50 layers (figure 3.4). It addresses the problem of vanishing gradient which makes training of deep networks unfeasible, by using residual learning where a network learns identity mappings by introducing skip connections that enable easier flow of gradients during backpropagation. The network architecture contains several convolutional layers, batch normalization, ReLU activations, that finally end with fully connected layers. ResNet50 has performed remarkably well on several image classification tasks, and has made it a highly popular model for extracting features from images.

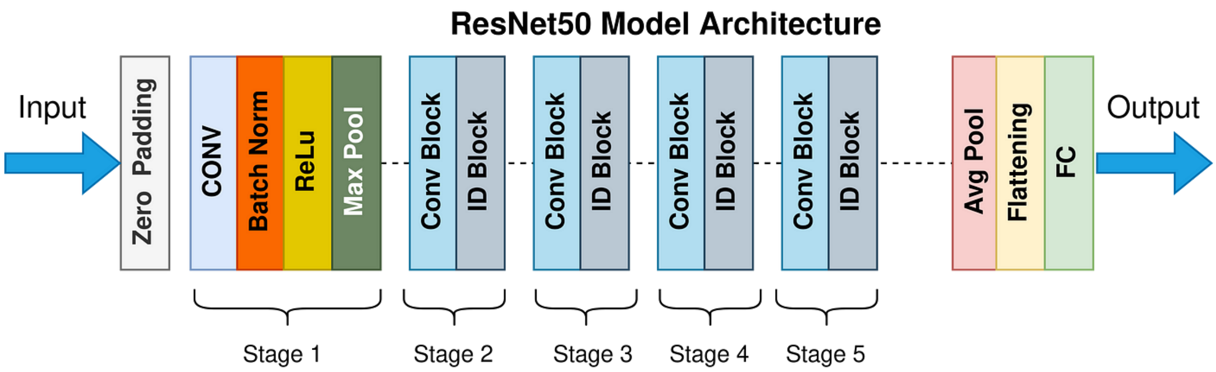


Figure 3.4: ResNet50 Architecture [23]

3.3.2.2 VGG16

VGG16 is an interesting deep learning architecture that has gained an unprecedented popularity due to its simple yet effective design. Developed by the Visual Geometry Group. VGG16 is known for its uniform architecture and uses 3x3 convolution filters that helps the network to have a deep architecture(figure 3.5). These small 3x3 convolution filters helps keep the architecture small and even sized as every filter reduces the spatial dimensions by 2 pixels. It is a deep network, easy to understand because it is uniformly built and uses small 3x3 filters. Even though it is known as being computationally and memory expensive, it is still state-of-the-art in many image classification benchmarks.

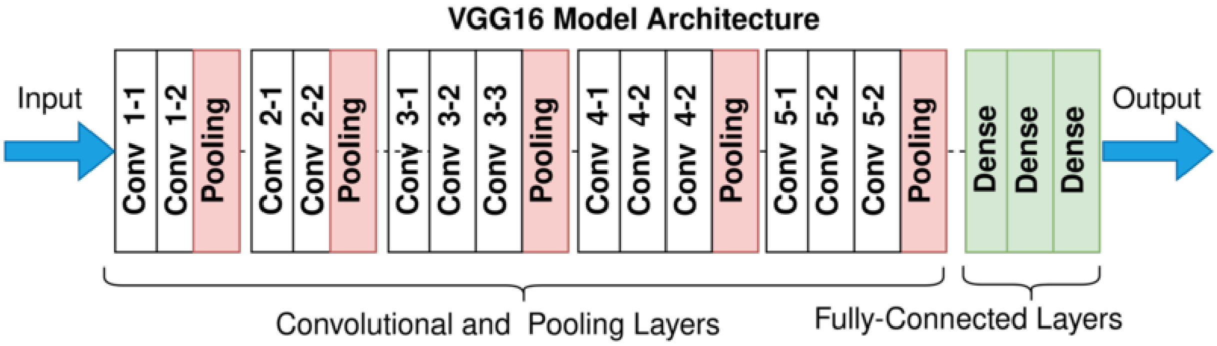


Figure 3.5: VGG16 Architecture [24]

3.3.2.3 VGG19

VGG19 builds on VGG16's architecture, adding three additional layers which are just convolutional layers (figure 3.6). That makes 19 weight layers in total. VGG19 is deeper than VGG16, because of which it gains additional learning ability and this allows capturing of more complex features. It uses the same structure as VGG16 with 3x3 filters. Though it is more computational heavy and is memory intensive because we have added more layers, VGG19 generally performs better than VGG16 on some image classification tasks, because of more depth.

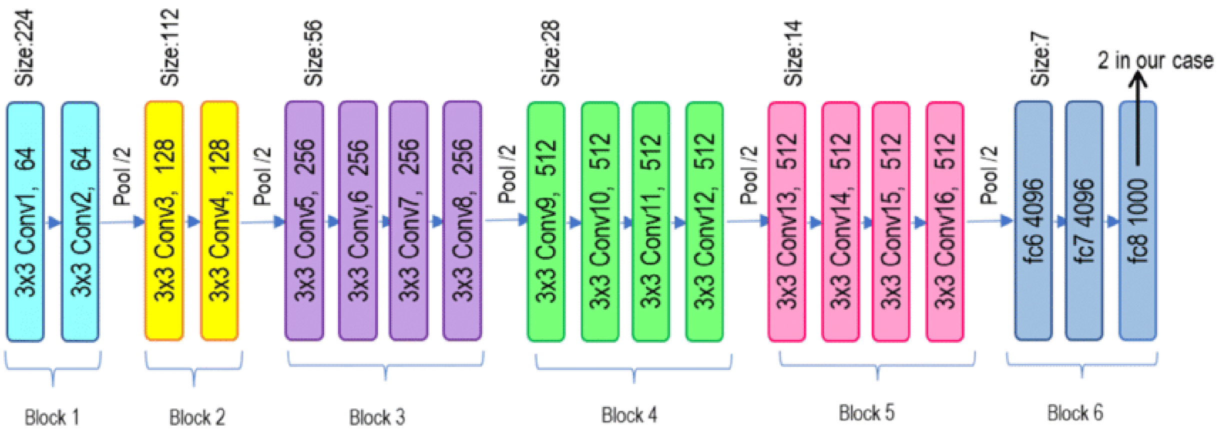


Figure 3.6: VGG19 Architecture [25]

3.3.2.4 EfficientNetB0

EfficientNetB0 is part of the entire EfficientNet family, and is the smallest one in terms of size. It introduced a novel idea of using compound scaling. This technique uniformly scales the width, depth and resolution of the network with resource constraints which are fixed in the beginning (figure 3.7). It is able to show state-of-the-art image performance in classification, while being up to 8x smaller and 6x faster than traditional models, which is remarkable. Because of this balance between performance and efficiency, EfficientNetB0 is an excellent choice for cases where high-performing and lightweight models are required such as edge devices and mobiles.

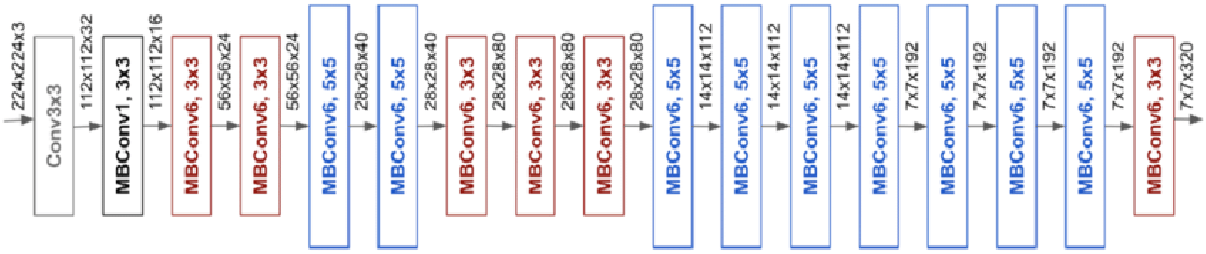


Figure 3.7: EfficientNetB0 Architecture [26]

3.3.2.5 MobileNetV2

MobileNetV2 is a model created for use, in mobile and embedded applications (figure 3.8). It puts significant emphasis on model efficiency to enable operations on even resource-constrained devices. The model leverages several techniques such as depth-wise separable convolutions, linear bottleneck layers and shortcut connections to bring down the large number of parameters and consequently the computational cost involved in training the network, resulting in a model that is both lightweight and powerful. Hence, it is appropriate for edge devices on which real-time processing is needed and reducing latency is critical, such as mobile applications and IoT devices.

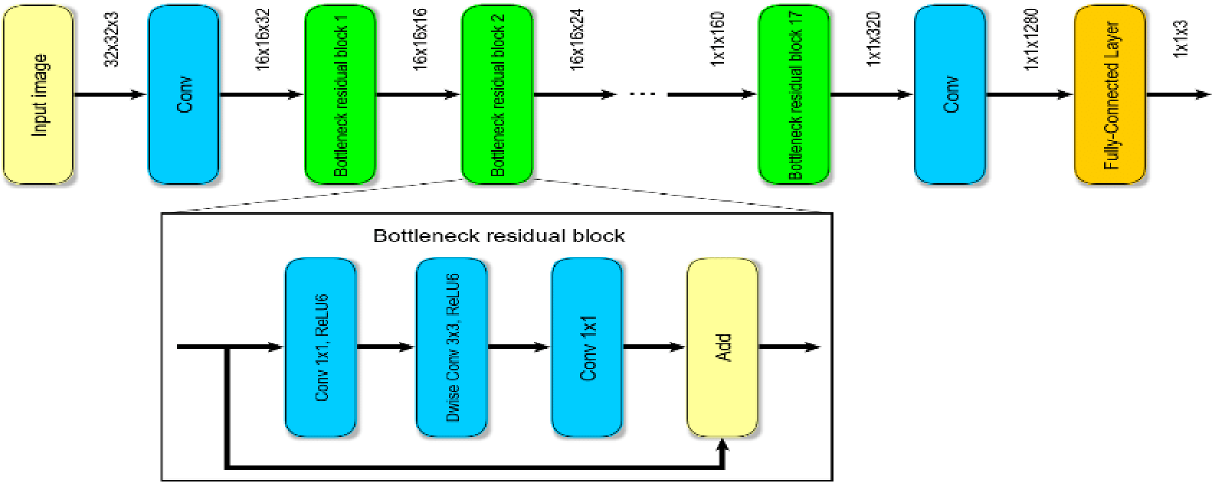


Figure 3.8: MobileNetV2 Architecture [27]

3.3.2.6 InceptionV3

InceptionV3 is based on the improved version of an Inception architecture called GoogLeNet (figure 3.9). Some of the optimization features of this model includes factorized convolutions, grid reduction and many that also makes it further accurate and computationally efficient. By using convolutional filters in the same module at different scales, it is really effective to deal with complex patterns in images. Even with characteristics showing high computational efficiency it still maintains its accuracy, InceptionV3 is the appropriate architecture for image recognition task in different types of environments including academia and industry.

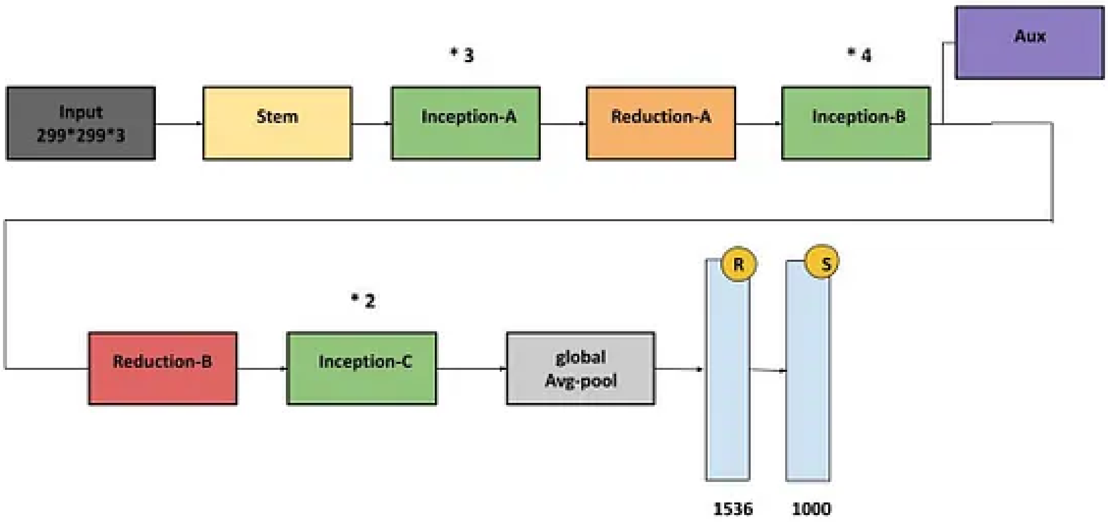


Figure 3.9: InceptionV3 Architecture [28]

3.3.2.7 Xception

The name Xception is a short form of "Extreme Inception". It may be considered as an extreme version of the Inception architecture (figure 3.10). This modification extends the model, simplifies it, and distributes the computational effort in one part of the learning system to better utilize resource to improved performance. Compared to Inception, Xception has a lot less number of parameters and achieves better accuracy as a result. This enables Xception to classify images more correctly and is a much faster model. As a result, Xception is a incredibly popular and standard choice for image classification, and all kinds of computer vision tasks.

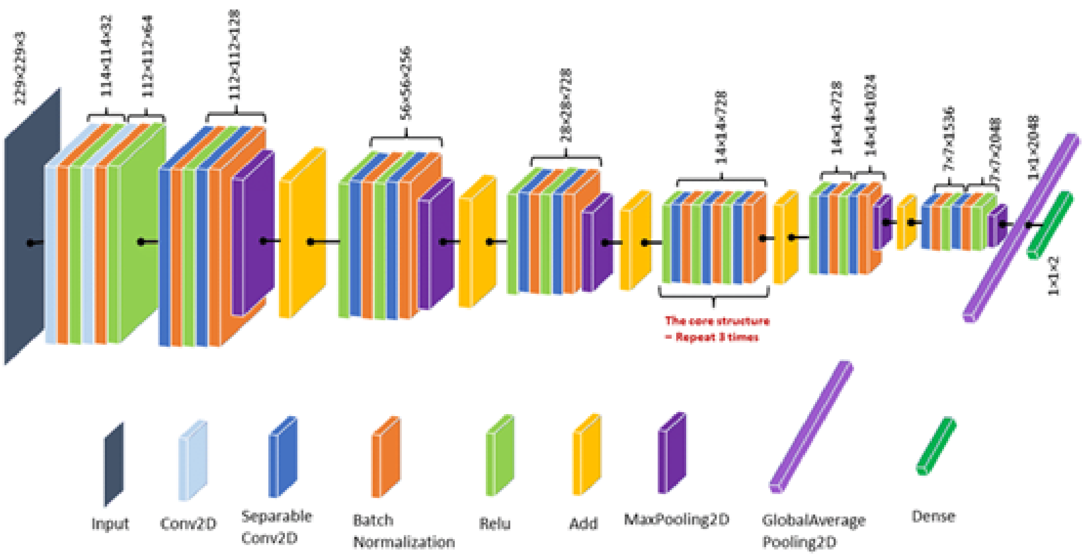


Figure 3.10: Xception Architecture [29]

3.3.2.8 DenseNet121

DenseNet121, which is Densely Connected Networks is a CNN with the kye feature as the output of every layer feeds to to all forward layers (feed-forward way) (figure 3.11). This connectivity enhances feature propagation and in turn enhances network performance. The dense connectivity leads to more information and gradients flow throughout the network, which means it can be more accurate and it is less likely that the training data overfits. DenseNet121 uses less number of parameters to reach good results as it reuses features from all the previous layers for all its dense layers. This increases the model efficiency and accuracy.

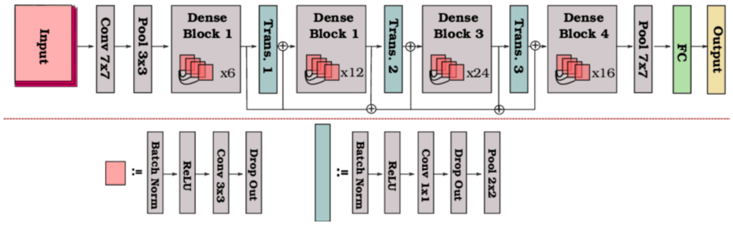


Figure 3.11: DenseNet121 Architecture [30]

3.3.2.9 DenseNet201

DenseNet201 is an even deeper variant of the DenseNet architecture, containing 201 layers. The dense connectivity allows it to be more accurate that its short depth counterparts. The architecture is the sam except that it has more layers and thus it can learn more complex features. Low level features are used and further built upon to extract even more complex features.

For adding more layers for fine tuning, top layers of the DenseNet201 model were freezed and the modified model is trained on it for yoga posture categorization whose code is shown in figure 3.12. The added layers includes the batch normalization, dropout layers (at 25% and 20% rates), and dense layers with ReLU activation (1024 and 512 units) (figure 3.13). Apart from this, the output layer is used to represent twenty different yoga positions with the softmax activation. In order to capture posture-specific features, few fine tuning layers are also unfreezed. Categorical cross-entropy loss function was used in the model compilation using Adam optimizer (lr: 0.001).

```
[ ] pretrained_model = tf.keras.applications.DenseNet201(input_shape=(100,100,3),include_top=False,weights='imagenet',pooling='avg')
    # Freeze the weights of the base model
    for layer in pretrained_model.layers:
        layer.trainable = False

[ ] inputs = pretrained_model.input
    x_layer = tf.keras.layers.Dropout(0.25)(pretrained_model.output)
    x_layer = tf.keras.layers.Dense(1024, activation='relu')(x_layer)
    x_layer = tf.keras.layers.BatchNormalization()(x_layer)
    x_layer = tf.keras.layers.Dense(512, activation='relu')(x_layer)
    x_layer = tf.keras.layers.BatchNormalization()(x_layer)
    x_layer = tf.keras.layers.Dropout(0.25)(x_layer)
    x_layer = tf.keras.layers.Dense(256, activation='relu')(x_layer)
    x_layer = tf.keras.layers.BatchNormalization()(x_layer)
    x_layer = tf.keras.layers.Dropout(0.20)(x_layer)
    outputs = tf.keras.layers.Dense(20, activation='softmax')(x_layer)

    # Create the final model
    model = tf.keras.Model(inputs=inputs, outputs=outputs)

[ ] optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
    model.compile(optimizer=optimizer,loss='categorical_crossentropy',metrics=['acc'])
    history = model.fit(datagen.flow(X_train,Y_train,batch_size=32),validation_data=(X_val,Y_val),epochs=50)
```

Figure 3.12: Code-specimen for DenseNet201 customization

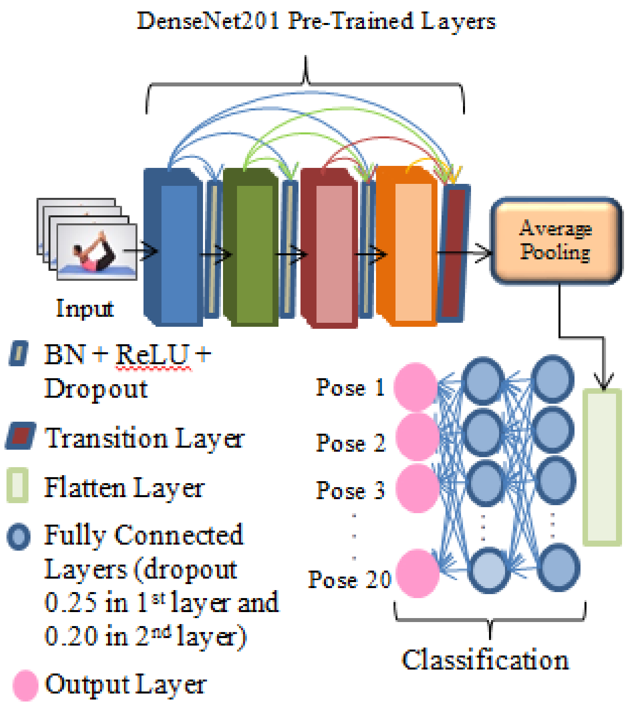


Figure 3.13: DenseNet201 Architecture

3.3.3 Model Evaluation and Fine-Tuning

We assessed the each model's performance on a different test set. For quantitative analysis, we created accuracy, confusion matrices, and classification reports. We also found out which pose is the most accurately detected for each model. To determine convergence, overfitting, and generalization abilities, we compared metrics from training and testing. Next, in order to improve alignment with the dataset, we unfroze and fine-tuned certain layers. Model architecture and performance play a role in the selection of the layers. After making adjustments, we recompile the models and keep practicing for more epochs.

3.3.4 Comparative Study

Taking into account training speeds, convergence patterns, and final metrics, a comparative analysis allowed for a more fine-grained and detailed understanding of the performance with respect to each model. This rigorous process resulted in the selection of the best-performing model, whose weights were saved for later deployment and which had the highest testing accuracy.

3.4 Proposed Approach 2

To properly categorize yoga postures, the suggested method includes dataset preprocessing, skeletal keypoint extraction, data augmentation, base models training and then applying ensemble approaches for improving performance as shown in figure 3.14.

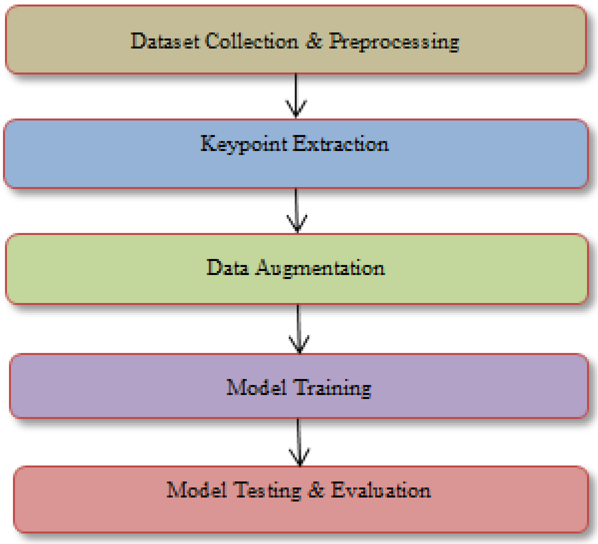


Figure 3.14: Proposed Approach 2

3.4.1 Dataset Collection and Preprocessing

To guarantee consistency and quality, certain preprocessing measures are implemented as soon as the dataset is gathered. These include image scaling, normalization, and label encoding. A split of 80:20 is followed, which means 80% data is for model traiing, and 20% is reserved for testing.

3.4.2 Keypoint Extraction

After preprocessing, two distinct keypoint extraction methods, namely YOLOv8 and MediaPipe are applied to extract skeletal keypoints from the images [2] as depicted in Figure 3.15. These landmarks stand in for important anatomical regions, including the shoulders, wrists, elbows, knees, hips, and ankles. These landmarks' x, y, and z coordinates are also kept in arrays.

3.4.2.1 MediaPipe

MediaPipe is a framework and it is open-source developed by Google. It provides cross-platform solutions and well-built pipelines for using ML in real-time use cases. MediaPipe can design ML models which accept one or more streams like media as input, and generate structured data. One popular model present in MediaPipe is the pose detection model, which uses video streams and focuses on identifying and monitoring poses detected. This has many applications across industries such as healthcare, fitness and sports. The pose detection architecture in MediaPipe is designed by a combination of ML with computer vision.

This model features an accurate pose detection pipeline that is also efficient, which can operate on mobile devices and the web. The pose estimation process involves 2 stages; detection and tracking. During detection, a MobileNetv3-based single shot pose detector is used to find if a person is present in the frame and estimate keypoint locations. Keypoint localization, for all 33 keypoints (wrists, elbows, shoulders, hips, knees and ankles of each individual) is performed during tracking stage. A keypoint tracking model refines the keypoint to provide an estimation. All keypoints are interconnected to form a skeletal structure which represents the body posture. The model is quite accurate and does real time execution on devices to track body movements for precise details.

3.4.2.2 YOLOv8

YOLOv8 (You Only Look Once version 8) is the most recent iteration of the YOLO object detection framework. YOLO is known for its speed and accuracy. YOLOv8 introduces many architectural improvements and optimizations over its predecessors and is designed for detecting objects in images or video streams in real time. It works for human pose estimation as well.

YOLOv8 uses only 1 neural network that takes as input the entire image, processes it in one pass, and predicts bounding boxes with class probabilities. To use YOLOv8 for human pose estimation the model can be extended to predict keypoints by outputting additional feature maps that correspond to the location of the human joints. The architecture commonly includes a backbone like CSPDarknet, a neck like PANet, and a head for keypoint detection.

With the introduced human pose estimation, YOLOv8 performs keypoint detection by dividing each image in the form of a grid and predict keypoint coordinates inside each grid cell. The model generates heatmaps for each keypoint which are used to pinpoint the location of every keypoint in the image. The result is a set of keypoints that form the human body skeleton.With its real-time high-speed processing, accurate keypoint detection, and pose estimation.

Figure 3.15: Landmark detection in a yoga image [2]

3.4.3 Data Augmentation

A variety of methods, including rotation, scaling, flipping, cropping, translating, adding noise, varying brightness and contrast, color jittering, random erasing, and cutting, are included in data augmentation. By producing synthetic examples, these techniques are used to diversity the dataset, improve the model's generalization, and lower the likelihood of overfitting. It also has the benefit of permitting the model to perform generalisation on future unidentified images.

One method that is widely used for this is SMOTE, which is an abbreviation for 'Synthetic Minority Over-sampling Technique'. By interpolating between the current instances and their closest neighbors, SMOTE creates artificial training records for the minority class. For each observation, after determining the k-nearest neighbors, we generate synthetic data points by interpolating between them. First, a stochastic sample is used to select the minority class. Then, SMOTE oversamples the minority class, and due to this successfully addresses class imbalance, decreases overfitting, and enhances model performance.

3.4.4 Model Training

The skeletal data is used to train a variety of machine learning and ensemble algorithms, to identify underlying patterns between the extracted keypoints and yoga position labels. GridSearchCV is employed to methodically adjust the models' hyperparameters to find the optimal values which provide the best results during prediction. It operates by sifting through a predetermined set of parameters and assessing the effectiveness of each combination using cross-validation. The models that were utilized are as follows.

3.4.4.1 Linear SVM

Linear Support Vector Machines (SVM) is a straightforward supervised learning models for various classification and also regression tasks. The premise is to find the best hyperplane in the feature space which separates different classes (figure 3.16). Here "best" means margin maximization between the classes, which helps generalize the model and reduces the problem of overfitting. It is widely successful when dealing with high-dimensional data that is linearly separable[31]. One of the features of a linear SVM is that it's not sensitive to outliers because it only uses the support vectors to determine the hyperplane that best separates the classes.

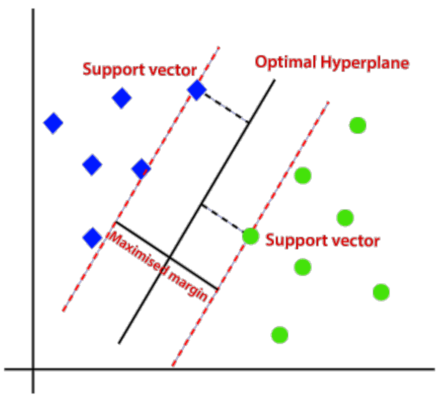


Figure 3.16: Linear SVM Working [32]

3.4.4.2 Decision Tree

A Decision Tree is a supervised learning algorithm, it is non-parametric and used for both classification and regression tasks. Decision Trees divide a dataset into subsets, or branches, and then split these branches into smaller ones using decision nodes. When the tree splits into sub-branches, it requires us to make a decision based on which path to take (figure 3.17). Each branch of the tree represents an input feature, or a decision to be made about that feature, and each leaf node represents a label for a class, or a number in the case of regression tasks. However, because a decision tree has too many branches it may overfit when the tree is too deep. Techniques such as pruning, using decision tree stumps, and dropout can help mitigate this.

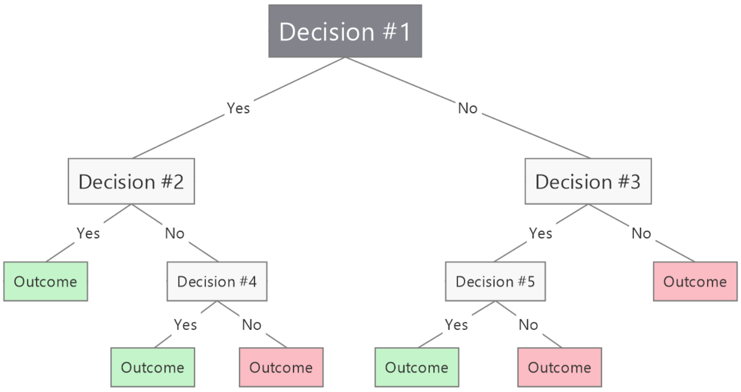


Figure 3.17: Decision Tree Working [33]

3.4.4.3 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is an example-based, non-parametric learning algorithm. It functions by finding the K training examples nearest to a given query and output predictions based on the maximum voting of the neighbors (for classification) or the average (for regression)[34] (figure 3.18). KNN is quite simple and easy to use, but as the data grows it will become very expensive for computation because we need to do the computations for distance between all points represented by the query. KNN, on the other hand, is flexible and can handle misinformation well.

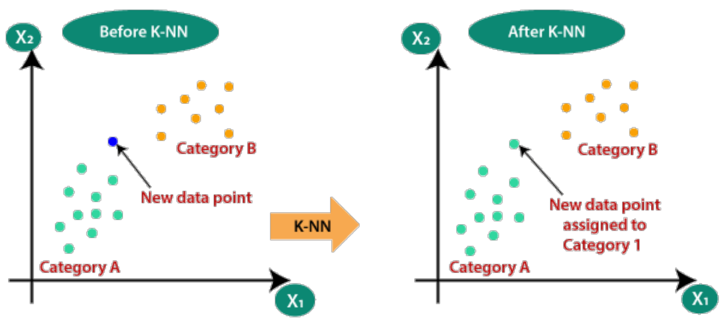


Figure 3.18: K-Nearest Neighbors Working [35]

3.4.4.4 Multi-Layer Perceptron

Multilayer perceptron (MLP) is a type of feedforward artificial neural network. MLP has at least 3 node layers (figure 3.19). Every node except the input layer nodes is a neuron which also consist of a nonlinear activation function. Backpropogation technique is used to train MLPs, which helps it to learn and discover complex patterns and relationships within the data [36]. However, the main downside for MLPs is that training these models are known for being quite computationally intensive and may require a large number of tuning for hyperparameters, for instance, number of hidden layers/neurons, learning rate, and activation functions in order to achieve good results.

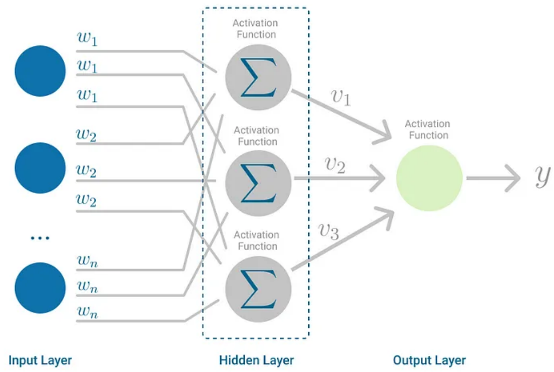


Figure 3.19: Multi-Layer Perceptron Working [37]

3.4.4.5 GaussianNB

Gaussian Naive Bayes (GaussianNB) is a Naive Bayes version that uses Gaussian (normal) Distribution to model continous random variables. It is a probabilistic classifier based on Bayes' theorem and is particularly suited for continuous data. GaussianNB is simple to implement and computationally efficient, making it a good first choice as a baseline model for classification tasks (figure 3.20). However, since the Naive Bayes model does not exploit any of the possible correlation between the features, it is not able to fully capture all the necessary information to make the best prediction.

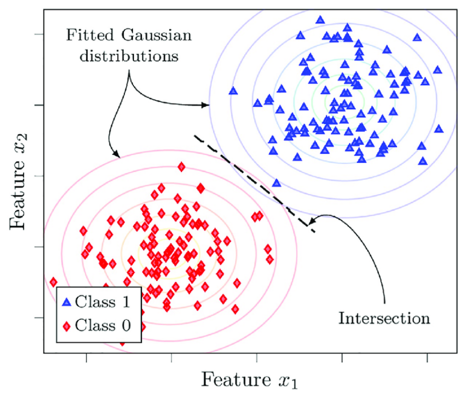


Figure 3.20: GaussianNB Working [38]

3.4.4.6 Bagging

Bagging or Bootstrap Aggregation is an ensemble learning method which aims at improving accuracy and stability of machine learning algorithms. Here, we train various versions of a model on varying subsets of the training data and then combine the predictions of all models (figure 3.21). In addition to tackling overfitting, bagging can also reduce the variance of models, especially those with high variance like decision trees. Since the predictions of the models are averaged, bagging can result in more reliable and generalized results. Bagging is most effective when individual models often overfit, and the ensemble method helps smooth the randomness or noise. This results in a more stable prediction, which can be made for a wide range of inputs.

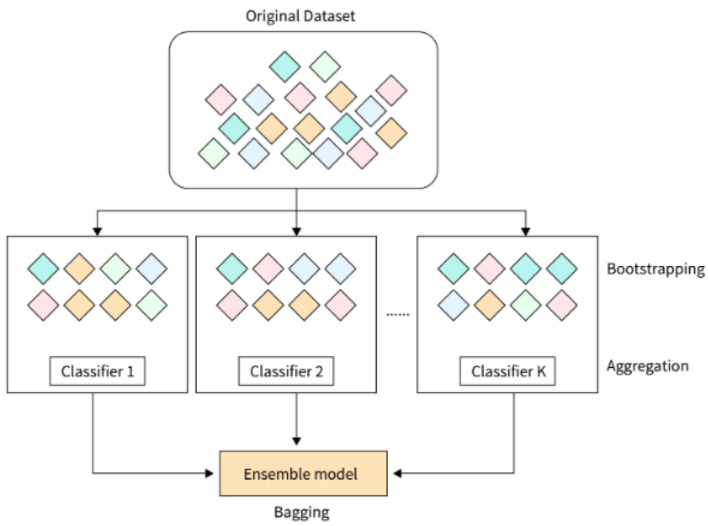


Figure 3.21: Bagging Working [39]

3.4.4.7 Random Forest

Random Forest Random Forest is arguably one of the most flexible, simple and powerful learning algorithms. It is also an ensemble learning method (figure 3.22). The functioning progresses as it constructs multiple decision trees while training. For each tree, the outputs class is the mode(classification) or mean(regression) of individual tree predictions. This technique brings an increase in classification accuracy with control over-fitting by averaging many decision trees, trained on different portions of the data. Random Forests are robust, highly accurate, scalable and can also be applied on data of higher dimensionality[40]. It can also handle missing values elegantly. Random Forest outputs an estimate of the importance of features in a classification problem and in some cases, it can be used for feature selection. Random Forest can handle large datasets which are very common in practical machine learning applications. Therefore, Random Forest is a must to have tool in the set of tools.

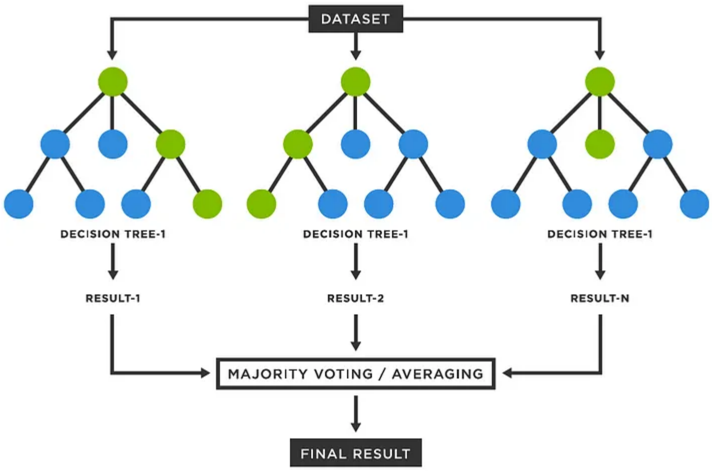


Figure 3.22: Random Forest Working [41]

3.4.4.8 XGBoost

XGBoost, short for Extreme Gradient Boosting, is a Gradient Boosting Machines that is considered state-of-the-art. It creates an ensemble of weak decision trees, each tree attempts to correct the misclassifications or errors made by its predecessors (figure 3.23). XGBoost is extremely fast, efficient and scalable, both because of its system-level optimizations and the algorithmic component - the learning algorithm has been augmented with a number of techniques, such as tree pruning, regularization and handling of missing values [42]. As a matter of fact, XGBoost often wins or ranks among the top spots in many machine learning competitions and has been successfully applied across a wide range of domains - especially in those areas where model performance is crucial.

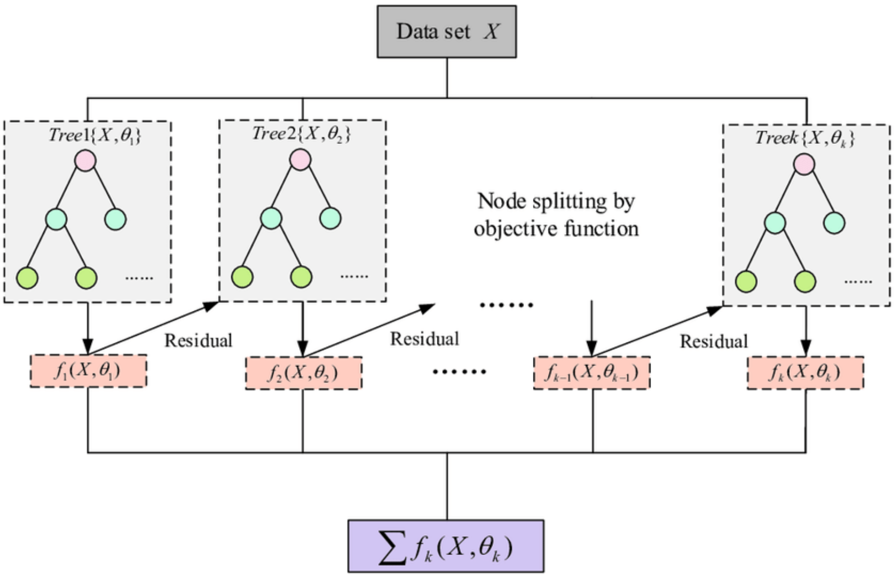


Figure 3.23: XGBoost Working [43]

3.4.4.9 LightGBM

LightGBM, or Light Gradient Boosting Machine, is a framework which makes use of tree-based learning algorithms (figure 3.24). LightGBM is focused on performance and can scale effectively to handle large datasets with millions of instances. It uses a novel technique for handling gradient-based boosting where decision tree learning is represented as a histogram-based algorithm[44]. This approach reduces memory requirements and speeds up training, and allows handling of even very large datasets efficiently. It is a strong competitor to any other gradient boosting framework for speed and memory utilization.

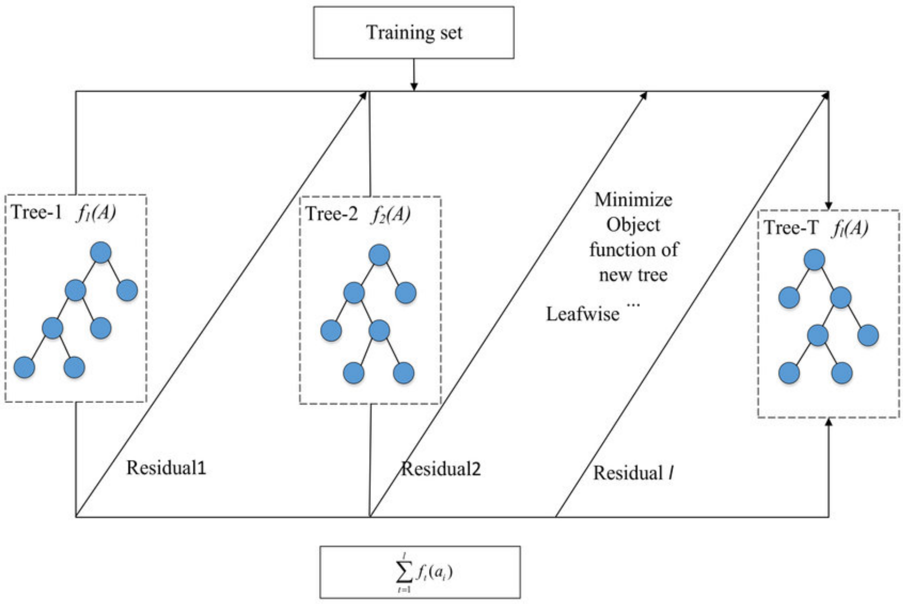


Figure 3.24: LightGBM Working [45]

3.4.5 Model Testing & Evaluation

After rigorous training phase, model performance is evaluated using the test set. Various measures like test accuracy, F1-score, precision, recall and ROC-AUC score are determined. To gain more insight into the performance of each model in relation to the landmark detection methods employed, a comparative study is performed. The weights of the model that outperformed the others in terms of accuracy highest test accuracy are then saved.

Chapter 4

RESULTS and DISCUSSION

Important Metrics Used:

In this chapter, we detail several standard metrics used for the performance evaluate of our models: accuracy, precision, recall, and F1-score. These help to understand the effectiveness of the model in predicting the correct classes, especially in cases where the class distribution might be imbalanced.

1. Accuracy

Accuracy measures the percentage of correct results (both true positives and true negatives) from the total number of classes. This is also the **Top-1 Accuracy**. It is defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

Top-5 Accuracy:
Top-5 accuracy calculates the percentage of times the true class label is among the top 5 predicted probabilities. It is defined as:

$$\text{Top-5 Accuracy} = \frac{\text{No. of correctly predicted samples in the top 5 predicted labels}}{\text{Total number of samples}} \tag{4.2}$$

2. Precision

Precision is a ratio of correct positive predicted classes to total positive predicted classes. It is described as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.3}$$

3. Recall It is a ratio of correctly predicted positive observations to all observations in the actual class. It is also known as Sensitivity or True Positive Rate, defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.4}$$

4. F1-Score The F1-score is the weighted average of Precision and Recall. It balances the two metrics and is particularly useful in imbalanced class distribution cases. The F1-score is defined as:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.5}$$

4.1 Approach 1 Result

We carried out a thorough analysis of many deep learning models for yoga position identification in our study. The table 4.1's accuracy results demonstrated DenseNet201's best performance, with the model attaining the maximum accuracy of 90.4% for top-1 classification out of all the models. Among the best performers, EfficientNetB0 stood out with a 98.4% accuracy rate for top-5 categorization and also presents information on each model's F1 score, precision, and recall.

Table 4.1: Accuracy, Precision, Recall, and F1-Score for different Deep Learning Models

Model Name	**Before Fine-Tuning Testing Accuracy**	**After Fine-Tuning**		**Precision**	**Recall**	**F1-Score**
		Top-1	**Top-5**			
ResNet50	72.0%	84.4%	96.8%	0.86	0.84	0.84
VGG16	62.4%	80.0%	96.2%	0.82	0.80	0.80
VGG19	57.6%	82.4%	97.6%	0.84	0.83	0.82
EfficientNetB0	74.4%	86.8%	**98.4%**	0.88	0.87	0.87
MobileNetV2	66.4%	80.0%	95.6%	0.82	0.80	0.79
InceptionV3	54.4%	80.8%	96.0%	0.82	0.82	0.80
Xception	60.4%	81.2%	94.8%	0.82	0.81	0.81
DenseNet121	74.0%	85.2%	97.6%	0.86	0.85	0.85
DenseNet201	79.2%	**90.4%**	97.2%	0.92	0.90	0.90

The table 4.1 displays the performance metrics for all deep learning models used, before and after fine-tuning.

Key observations include:

1. **Before Fine-Tuning Testing Accuracy**: This column shows the accuracy of each model before any fine-tuning. The values range from 54.4% (InceptionV3) to 79.2% (DenseNet201).

2. **After Fine-Tuning Top-1 Accuracy**: This column indicates the accuracy after fine-tuning when the model's top prediction is considered. The accuracies improve significantly post fine-tuning, with values between 80.0% (VGG16, MobileNetV2) and 90.4% (DenseNet201).

3. **After Fine-Tuning Top-5 Accuracy**: This shows the accuracy when any of the model's top-5 predictions are correct. The accuracies range from 94.8% (Xception) to 98.7% (EfficientNetB0).

4. **Precision, Recall, and F1-Score**: These columns present the precision, recall, and F1-score for each model after fine-tuning. Precision ranges from 0.82 (VGG16, MobileNetV2, Xception) to 0.92 (DenseNet201), recall ranges from 0.80 (VGG16) to 0.87 (EfficientNetB0), and F1-score ranges from 0.79 (MobileNetV2) to 0.90 (DenseNet201).

As a summary,we can say fine-tuning proves a significant improvement in both top-1 and top-5 accuracies across all models. DenseNet201 shows the highest performance in all metrics after fine-tuning, with the highest precision (0.92), recall (0.88), and F1-score (0.90). MobileNetV2 and VGG16 have the lowest F1-scores (0.79 and 0.80 respectively) after fine-tuning, indicating comparatively lower performance. Overall, the table 4.1 highlights the impact of fine-tuning on model performance, showing substantial improvements in accuracy and other evaluation metrics.

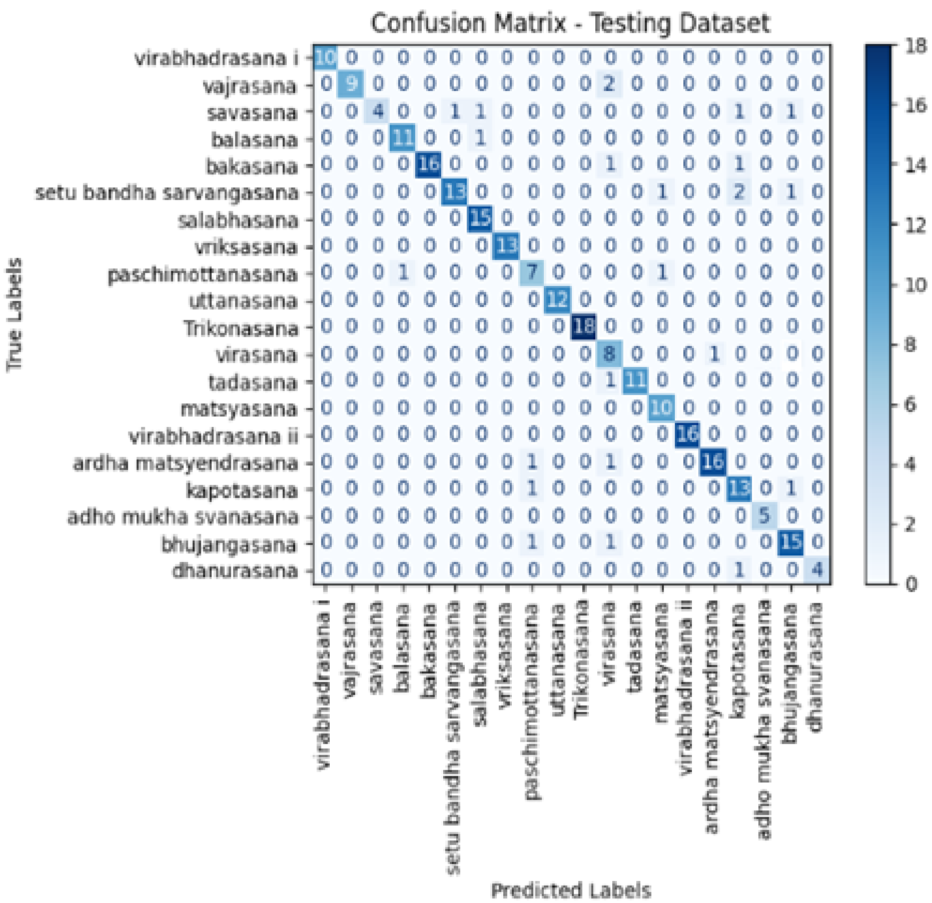


Figure 4.1: Confusion Matrix of DenseNet201 model

The confusion matrix in figure 4.1 illustrates the performance of a yoga pose recognition model on a testing dataset, showcasing the true labels on the vertical axis and the predicted labels on the horizontal axis. The matrix's diagonal elements, representing correct predictions, demonstrate that the model has achieved high accuracy across most classes, with poses like "virabhadrasana i," "vajrasana," "savasana," and "adho mukha svanasana" exhibiting perfect prediction accuracy, each with 10, 16, 10, and 15 correct predictions, respectively. This strong diagonal presence indicates a robust model performance, as the majority of predictions are accurate. However, some poses show moderate accuracy, such as "setu bandha sarvangasana," "matsyasana," and "tadasana," which have significant correct predictions but also a few misclassifications. The matrix also reveals specific patterns of misclassification, where certain poses are occasionally confused with similar-looking ones. For instance, "virabhadrasana i" was misclassified as "virabhadrasana ii" once, and "virasana" and "paschimottanasana" were each misclassified as "vajrasana" twice. Table 4.2 shows accuracy of each pose obtained from DenseNet201 model.

These errors also point to the directions on how one can either improve the feature extraction of the model or simply better differentiate between poses that are visually or contextually similar. In conclusion, this model performs well and the matrix shows a strong diagonal line that indicates many of the frames are classified correctly. In future iterations we could focus on improving this by modifying our training set to include more examples of poses that tend to be confused (the bottom left), or continuing to tune our model architecture such that subtle differences between some poses cause less misclassification (top row).

These errors also point to the directions on how one can either improve the feature extraction of the model or simply better differentiate between poses that are visually or contextually similar. In conclusion, this model performs well and the matrix shows a strong diagonal line that indicates many of the frames are classified correctly. In future iterations we could focus on improving this by modifying our training set to include more examples of poses that tend to be confused (the bottom left), or continuing to tune our model architecture such that subtle differences between some poses cause less misclassification (top row).

Table 4.2: Posewise Accuracy for DenseNet201 Model

Pose	**Accuracy**
Adho mukha svanasana	100.00%
Ardha matsyendrasana	88.88%
Balasana	91.66%
Bakasana	88.88%
Bhujangasana	88.23%
Dhanurasana	80.00%
Kapotasana	86.67%
Matsyasana	100.00%
Paschimottanasana	77.77%
Salabhasana	100.0%
Savasana	50.0%
Setu bandha sarvangasana	76.47%
Tadasana	91.66%
Trikonasana	100.0%
Uttanasana	100.00%
Vajrasana	81.82%
Virabhadrasana i	100.0%
Virabhadrasana ii	100.0%
Virasana	88.88%
Vriksasana	100.00%

The figure 4.2 shows the training and validation accuracy and loss curves of a model which are trained over 30 epochs. As we can see, the training accuracy curve is smoothly increasing and closes to 100% after the training process. It tells us that the model is effectively learning from the training data. A huge jump in validation accuracy is also observed, which plateaus at about 90% by epoch 30. This suggests that the model is not only performing well on the training data but also generalizing to unseen data.

The loss curve visualizes the learning progress of the model: we find that training loss consistently goes down and error is almost near zero at the end of training period. This highlights the model's efficiency in reducing errors. Also, the same thing is happening with validation loss, stabilizing at a slightly higher value but following a similar decreasing trend. And these curves show that the model continue to learn from experience and reducing their errors with respect to time for both training set as well as validation set. All in all, the model demonstrates good learning ability with high accuracy coupled with small loss on both datasets, indicating effective training and generalization.

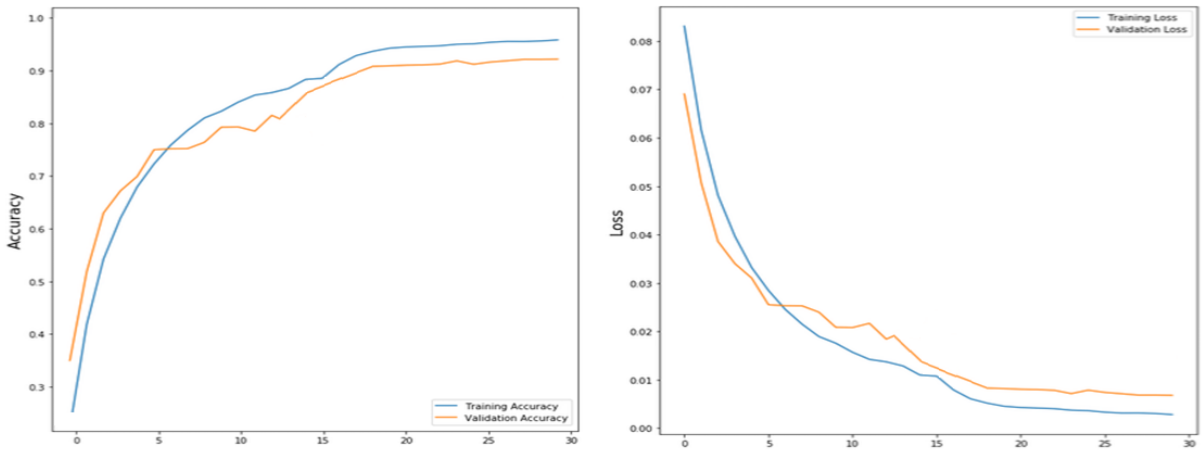


Figure 4.2: Accuracy and Loss Curves of DenseNet201 model

The ROC curve in figure 4.3 displays the performance of DenseNet201 model, with most classes showing near-perfect Area Under the Curve (AUC). The majority of classes, including "virabhadrasana i," "savasana," have an AUC of 1.00, signifying perfect classification for all images in these categories. Micro-average ROC curve, which combines performance for all classes comes to a high AUC (0.99), displaying the model's robustness and overall effectiveness. There are few classes like "vajrasana" and "virasana" with lower AUC scores of 0.98 and 0.94, respectively. But these still can be considered good performance. In conclusion, the model demonstrates exceptional ability in classifying different classes, making it highly reliable.

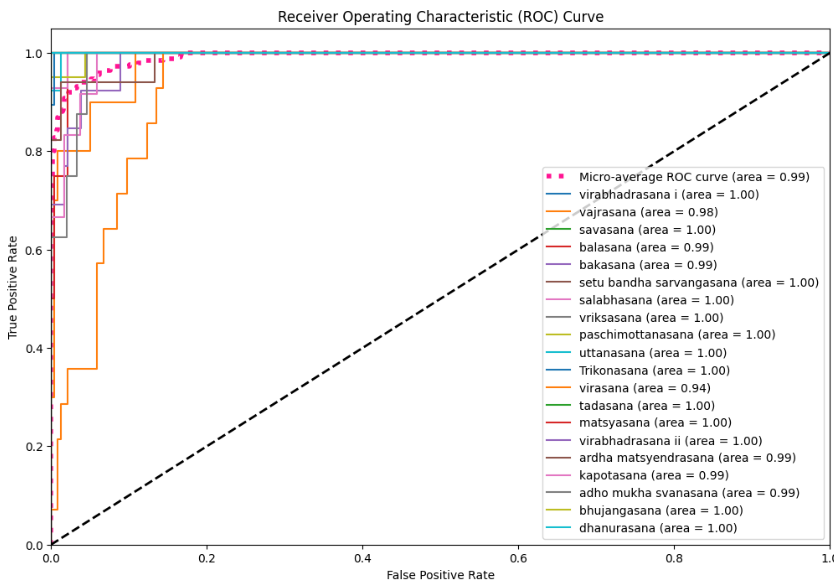


Figure 4.3: AUC-ROC Curve of DenseNet201 model

4.1.1 Output

The figure 4.4 compares the labels predicted by the DenseNet201 model with the actual labels of test images in the artwork featuring yoga positions. This comparison provides a visual evaluation of the model's posture recognition accuracy. It provides a succinct but useful illustration of how closely the model matches the real world, which helps assess how well it performs in terms of classifying yoga poses.

Figure 4.4: Actual vs Predicted labels of test yoga images

4.2 Approach 2 Result

For pose estimation, YOLOv8 detects 32 keypoints, while the MediaPipe framework detects 33 keypoints. After training on each set of skeletal data, a list of optimal hyperparameters for each model is provided in table 4.3. These were obtained using Grid Search technique and the given values yielded the best performance.

Table 4.3: Model Parameters for YOLOv8 and MediaPipe using Grid Search

Model	Parameter	YOLOv8	MediaPipe
Linear SVM	penalty	L1	L2
	loss	hinge	hinge
	multi_class	ovr	ovr
Decision Tree	min_samples_leaf	2	1
	max_depth	40	30
	criterion	entropy	entropy
	splitter	best	best
K-Nearest Neighbors	p	2	2
	n_neighbors	3	3
	algorithm	auto	auto
Multi-Layer Perceptron	activation	relu	relu
	solver	adam	adam
	early_stopping	True	False
GaussianNB	priors	None	None
	var_smoothing	1e-09	1e-09
Random Forest	bootstrap	False	False
	max_features	sqrt	sqrt
	criterion	gini	entropy
	oob_score	False	False
Bagging	oob_score	False	False
	n_estimators	164	100
XGBoost	max_depth	8	6
	learning_rate	0.005	0.001
LightGBM	learning_rate	0.005	0.001
	boosting_type	gbdt	rf
	num_leaves	31	31

The accuracy of trained models for yoga position categorization utilizing keypoints found by MediaPipe and YOLOv8 is summarized in the table 4.4. All things considered, MediaPipe routinely beats YOLOv8 for all the models trained. With YOLOv8 keypoints, the LightGBM obtains the best accuracy (82%) but with MediaPipe landmarks, it shows remarkably higher accuracy (96.52%), closely followed by Random Forest (95.65%). All other ensemble methods also showed consistently good accuracy of over 90% using MediaPipe keypoints. Multi-Layer Perceptron also exhibits notable accuracy gains with both YOLOv8 and MediaPipe keypoints. It is also observed that ensemble learning methods consistently outperformed the base ML models.

Table 4.4: Comparison of Different Models' Accuracy (YOLOv8 vs. MediaPipe)

Base Models	Accuracy (YOLOv8)	Accuracy (MediaPipe)
Linear SVM	65.28%	85.65%
Decision Tree	64.94%	81.73%
K-Nearest Neighbors	70.00%	70.00%
Multi-Layer Perceptron	80.00%	87.39%
GaussianNB	65.45%	88.69%
Random Forest	81.00%	95.65%
Bagging	80.94%	94.78%
XGBoost	79.28%	93.91%
LightGBM	**82.00%**	**96.52%**

Table 4.5: Performance Metrics of Different Algorithms

Algorithm	**Precision**	**Recall**	**F1 score**
Linear SVM	0.86	0.85	0.85
Decision Tree	0.82	0.81	0.81
K-Nearest Neighbors	0.70	0.69	0.69
Multi-Layer Perceptron	0.87	0.87	0.86
GaussianNB	0.90	0.89	0.88
Random Forest	0.96	0.96	0.96
Bagging	0.95	0.95	0.95
XGBoost	0.94	0.94	0.94
LightGBM	0.97	0.96	0.96

The table 4.5 shows performance metrics like precision, recall, F1-score for models trained with MediaPipe keypoints. With LightGBM being the best, other ensembles- Random Forest, XGBoost and Bagging also exhibit remarkable results, with precision, recall and F1-scores over 0.94. Looking at table 4.4 and table 4.5, overall it was discovered that LightGBM with MediaPipe keypoints was the best model.

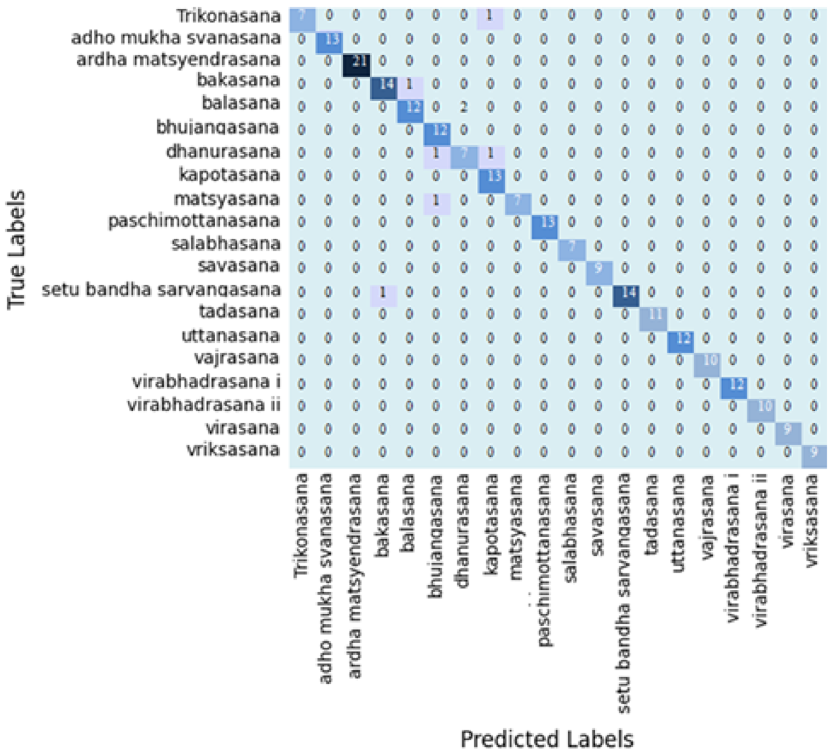


Figure 4.5: Confusion Matrix using MediaPipe+LightGBM model

The figure 4.5 shows the confusion matrix, wherein a visualization of pose-wise accuracy of each yoga pose is shown. The diagonals in the matrix counts the number of correct predictions. 14 poses, namely 'Adho mukha svanasana', 'Ardha matsyendrasana', 'Bhujangasana', 'Kapotasana', 'Paschimottanasana', 'Salabhasana', 'Savasana', 'Tadasana', 'Uttanasana', 'Vajrasana', 'Virabhadrasana i', 'Virabhadrasana ii', 'Virasana', 'Vriksasana' show an accuracy of 100%, meaning all test images were classified correctly. The other elements in the matrix denote incorrect classifications for every exercise. It is because some poses are too intricate to be photographed from one single shot. Some poses carry many similarities like how the torso is oriented and the legs are positioned. For instance, in the 9th row, 'matsyasana' has been correctly predicted 7 times and misclassified as 'bhujangasana' once. Accuracy of the other yoga poses are: 'bakasana' (93.33%), ;setu bandha sarvangasana' (93.33%), 'matsyasana' (87.5%), 'trikonasana' (87.5%), 'balasana' (85.71%) and 'dhanurasana' (77.77%). For the exercises where comparatively lower accuracy is obtained, more images can be added to the dataset taken from various angles and direction for overall understanding of the pose by the model.

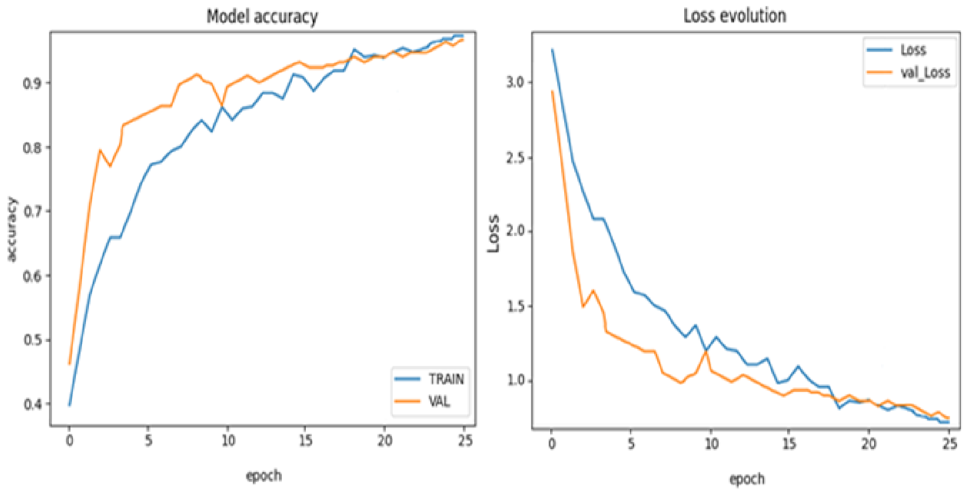


Figure 4.6: Accuracy and Loss curves using MediaPipe+LightGBM model

The figure 4.6 shows the accuracy and loss graphs of the MediaPipe+LightGBM model. The number of epochs, or one whole run across the training set, is shown on the x-axis. The left graph's y-axis denotes accuracy whereas the right one's represents loss. The accuracy curve starts at around 0.4 and increases to nearly 0.96 over the 25 epochs. This shows that the predictive algorithm is boosting its rate of accuracy in identifying the data points. The initial high loss (around 3.0) signifies significant discrepancies across the predictions to the true data points. The loss steadily decreasing towards 0 over 25 epochs indicates the model is progressively refining its predictions to better match the actual data.

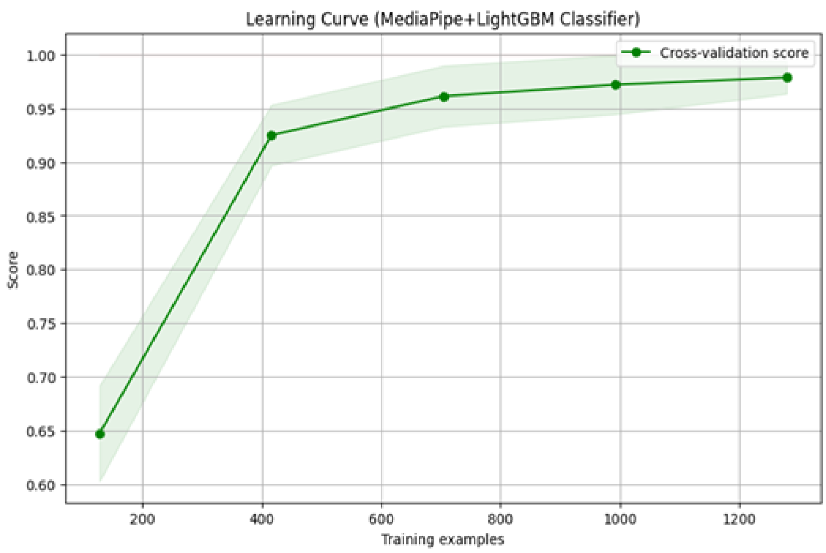


Figure 4.7: Learning Curve for LightGBM model

The learning curve of the LightGBM model with MediaPipe keypoints is shown in figure 4.7, where the x-axis is the count of training images and y-axis shows cross-validation score, which rises quickly from roughly 0.65 to 0.96 as more training samples are added. Beyond a score of 0.96 improvement flattens and settles around a remarkable score of 0.97, which signifies a well-trained model. Also, the flattening of the curve indicates that after around 1200 samples, the model performance stays the same. As more samples are used for training, the shaded region encompassing the cross-validation score line diminishes, signifying a drop in uncertainty in the model's performance.

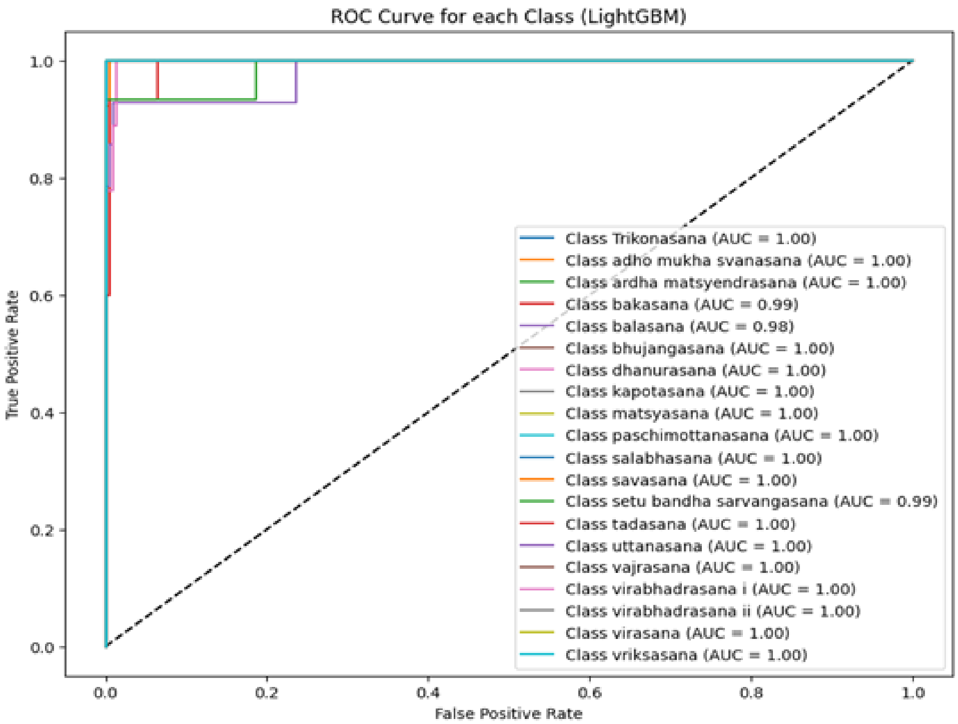


Figure 4.8: AUC-ROC curve using MediaPipe+LightGBM model

A graph of Receiver Operating Characteristic (ROC) is also shown for each class in figure 4.8. AUC, which is the Area Under the Curve, quantifies classifier performance, with 1.0 indicating perfection classification. Classes like 'Trikonasana', 'adho mukha svanasana', and 'ardha matsyendrasana' have AUCs of 1.0. 'Bakasana' and 'setu bandha sarvangasana' achieve AUCs of 0.99, indicating high performance. Analysis of ROC curves and AUC values guides identification of classes necessitating improvement or more training data.

4.2.1 Output

The true and predicted labels with landmarks indicated for pictures from the testing dataset showing different yoga positions are shown in figure 4.9. From the six test images, five of them are correctly classified and one image of 'Trikonasona' pose is misclassified as 'kapotasana'.

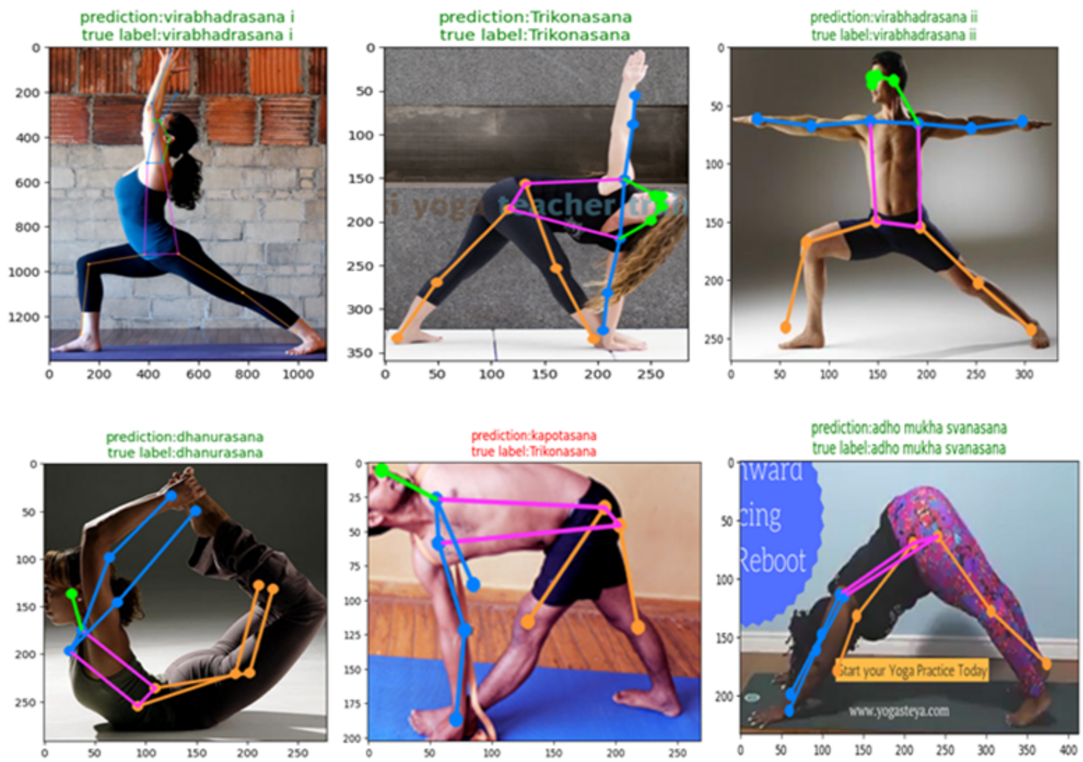


Figure 4.9: True vs Predicted labels of some test images of poses

4.3 Web Deployment

Integrate the trained models into an application using Python gradio library (figure 4.10) that can receive input 2D images, process them and output the predicted yoga pose. Ensure that the input images undergo the same preprocessing steps as during training and testing. The web-view demo works as shown in figure 4.11.

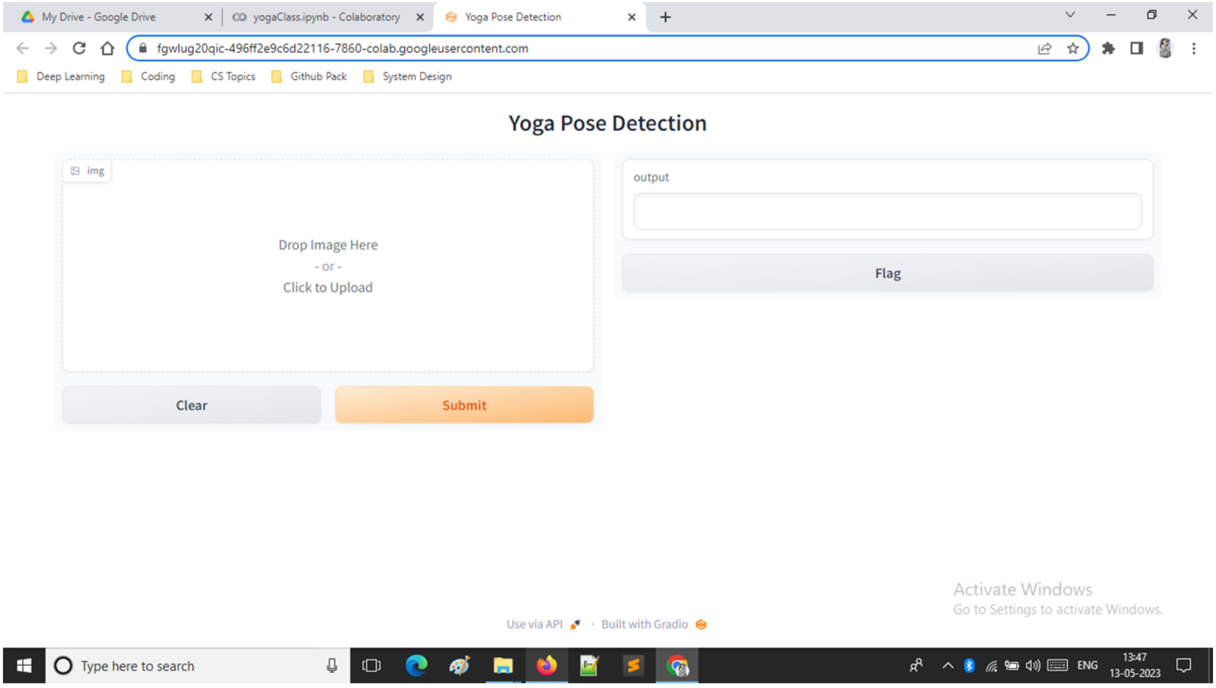
```
[ ]  !pip install fastapi>=4.0
```

```
[ ]  def predict_image(img):
       img_4d=img.reshape(-1,224,224,3)
       prediction=model.predict(img_4d)[0]
       #return {labels[i]: float(prediction[i]) for i in range(5)}
       for i in range(len(labels)):
         if prediction[i] == 1:
           return "Predicted Class: "+labels[i]
```

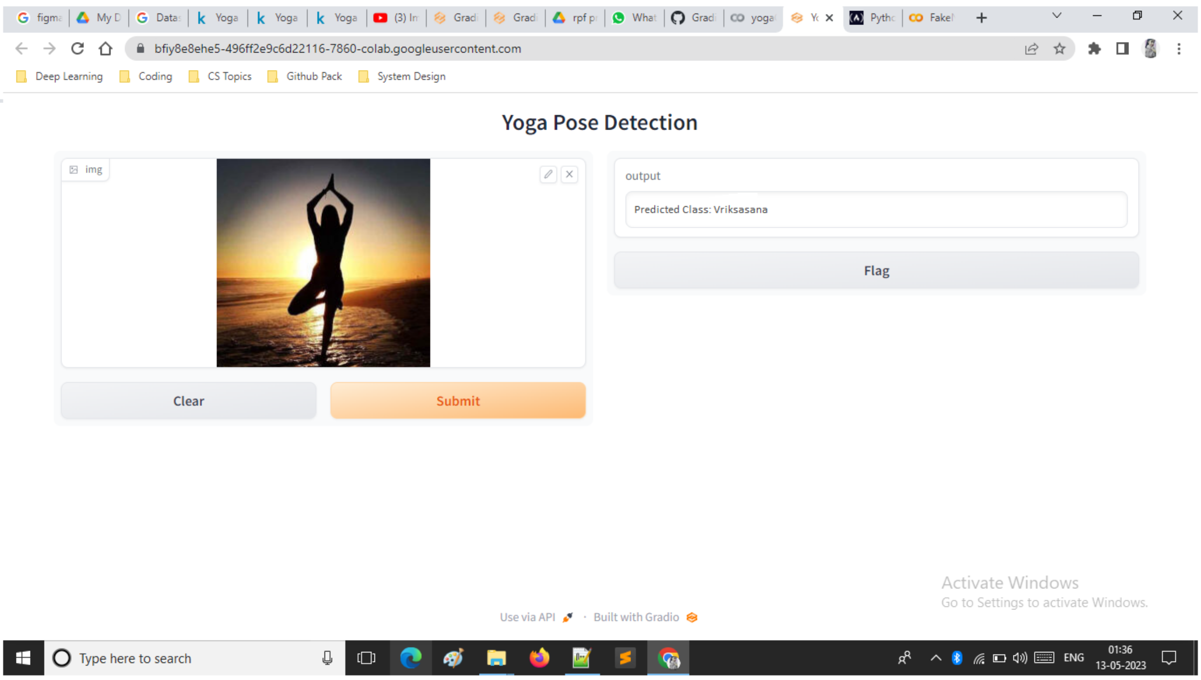
```
[ ]  imageimage = gr.inputs.Image(shape=(224,224))

     gr.Interface(fn=predict_image, inputs=image, outputs='text',title="Yoga Pose Detection").launch(debug='True')
```

Figure 4.10: Code-specimen of gradio for web-demo



(a) Web-page view of a yoga image submission



(b) Web-page view showing predicted output of the give pose

Figure 4.11: Web-demo view using Gradio

Chapter 5

CONCLUSION AND FUTURE SCOPE

In our comprehensive study, we explored multiple state-of-the-art deep learning architectures and techniques for yoga pose classification. Approach 1 involved evaluating DenseNet201 and DenseNet121, which demonstrated superior performance compared to other models, while EfficientNet and MobileNet excelled in processing efficiency. This highlights the inherent trade-offs between accuracy and computational economy, providing valuable guidance for model selection in practical applications. Our findings establish a solid baseline and indicate significant directions for future research in yoga pose categorization.

Approach 2 focused on leveraging YOLOv8 and MediaPipe for keypoint detection, followed by classification using various base models and ensemble learning techniques on a diverse custom dataset of yoga poses. The results highlighted that models using MediaPipe keypoints outperformed YOLOv8 models. This is an important fact to understand the importance of precise keypoint identification in classification accuracy improvement. Ensemble learning classifiers comfortably beat base models, with LightGBM's combination with MediaPipe keypoints showcasing the best performance. This model was also rigorously analyzed with multiple metrics as can be seen above, for authenticity, correctness and verification. Furthermore, this approach shows the ability to accurately distinguish between similar-looking poses.

In conclusion, our study emphasizes different architectures strengths and importance of keypoint detection methods in yoga pose classification. By highlighting accuracy and computational efficiency trade-offs, proving the benefits of keypoint identification and ensemble learning, we contribute valuable insights. Particularly in this domain, this provides a robust foundation for future research.

Future research should be aimed at improving real-time applications of the yoga posture identification algorithms. This will enable integration into user-friendly platforms, providing immediate feedback during yoga practice for users to correct their postures.

1. Real-Time Applications and Deployment: Examine real-time applications with deployment of established algorithm for yoga pose detection. Also look into possible integratation techniques into user-friendly platforms for immediate feedback.

2. Dataset Expansion and Model Efficacy: Testing the model's performance and possible improvements on a larger datasets. This includes more postures and people with varying expertise.

3. Addressing Real-World Challenges: A real-world problem could be obstacles such as occlusion and light fluctuations. Overcoming these will improve adaptability and is key to a better system. We can incorporate user input and preferences too, for customization of the yoga training experience, for instance their expertise level.

4. Advanced Deep Learning Architectures: Explore intricate deep learning architectures like Graph Neural Networks (GNNs) for pose categorization. Examine more ensemble learning combinations as deep learning potential alternatives

5. Power Efficient Mobile Applications: Develop power-efficient, computationally lighter mobile applications in order to the application more accessible. We believe that such a system should reach the masses without restriction on their devices' computational power.

6. Incorporation of Biomechanical Knowledge and Health Measurements: Incorporate biomechanical knowledge and health metrics to design a more holistic system. Then deliver personalized suggestions on objectives such as desired physical ability and wellness.

By improving on these areas, the yoga posture classification models can become more robust, adaptable, accessible and ultimately enhance the overall yoga experience for all users, regardless of their expertise level.

Bibliography

[1] F. Rishan, B. De Silva, S. Alawathugoda, S. Nijabdeen, L. Rupasinghe, and C. Liyanapathirana, "Infinity yoga tutor: Yoga posture detection and correction system," in *2020 5th International Conference on Information Technology Research (ICITR)*, 2020, pp. 1–6.

[2] S. Saxena, "Yoga pose image classification dataset," 2020, accessed: May 23, 2024. [Online]. Available: https://www.kaggle.com/datasets/shrutisaxena/yoga-pose-image-classification-dataset

[3] 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: 20th Anniversary International Conference, MMM 2014, Dublin, Ireland, January 6-10, 2014, Proceedings, Part I 20*. Springer, 2014, pp. 496–505.

[4] E. W. Trejo and P. Yuan, "Recognition of yoga poses through an interactive system with kinect device," in *2018 2nd international conference on robotics and automation sciences (ICRAS)*. IEEE, 2018, pp. 1–5.

[5] S. Kadbhane, K. Datir, T. Jagdale, S. Dhongade, and G. Jagtap, "Yoga posture recognition," *Int. J. Adv. Res. Comput. Commun. Eng.(IJARCCE)*, vol. 10, pp. 143–147, 2021.

[6] A. Gupta and A. Jangid, "Yoga pose detection and validation," in *2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA)*. IEEE, 2021, pp. 319–324.

[7] M. Verma, S. Kumawat, Y. Nakashima, and S. Raman, "Yoga-82: a new dataset for fine-grained classification of human poses," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 1038–1039.

[8] F. B. Ashraf, M. U. Islam, M. R. Kabir, and J. Uddin, "Yonet: A neural network for yoga pose classification," *SN Computer Science*, vol. 4, no. 2, p. 198, 2023.

[9] C. Nagalakshmi and S. Mukherjee, "Classification of yoga asanas from a single image by learning the 3d view of human poses," *Digital techniques for heritage presentation and preservation*, pp. 37–49, 2021.

[10] C. Long, E. Jo, and Y. Nam, "Development of a yoga posture coaching system using an interactive display based on transfer learning," *The Journal of Supercomputing*, pp. 1–16, 2022.

[11] D. Swain, S. Satapathy, B. Acharya, M. Shukla, V. C. Gerogiannis, A. Kanavos, and D. Giakovis, "Deep learning models for yoga pose monitoring," *Algorithms*, vol. 15, no. 11, p. 403, 2022.

[12] F. Rishan, B. De Silva, S. Alawathugoda, S. Nijabdeen, L. Rupasinghe, and C. Liyanapathirana, "Infinity yoga tutor: Yoga posture detection and correction system," in *2020 5th International conference on information technology research (ICITR)*. IEEE, 2020, pp. 1–6.

[13] S. Kothari, "Yoga pose classification using deep learning," *Master's Projects. 932.*, 2020.

[14] S. Saurav, P. Gidde, and S. Singh, "Exploration of deep learning architectures for real-time yoga pose recognition," *Multimedia Tools and Applications*, pp. 1–43, 2024.

[15] V. Kakulapati, S. Kokkirala *et al.*, "Xgboost analyses of classification of yoga," *Pavan sai and Kokkirala, Savyasachi, Xgboost Analyses of Classification of Yoga (October 22, 2021)*, 2021.

[16] M. Chasmai, N. Das, A. Bhardwaj, and R. Garg, "A view independent classification framework for yoga postures," *SN computer science*, vol. 3, no. 6, p. 476, 2022.

[17] V. A. Thoutam, A. Srivastava, T. Badal, V. K. Mishra, G. Sinha, A. Sakalle, H. Bhardwaj, and M. Raj, "Yoga pose estimation and feedback generation using deep learning," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[18] S. Garg, A. Saxena, and R. Gupta, "Yoga pose classification: a cnn and mediapipe inspired deep learning approach for real-world application," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 12, pp. 16 551–16 562, 2023.

[19] A. Upadhyay, N. K. Basha, and B. Ananthakrishnan, "Deep learning-based yoga posture recognition using the y_pn-mssd model for yoga practitioners," in *Healthcare*, vol. 11, no. 4. MDPI, 2023, p. 609.

[20] N. Vallabhaneni and P. Prabhavathy, "Artificial algae optimizer with hybrid deep learning based yoga posture recognition model," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–11, 2023.

[21] S. K. Yadav, A. Singh, A. Gupta, and J. L. Raheja, "Real-time yoga recognition using deep learning," *Neural computing and applications*, vol. 31, pp. 9349–9361, 2019.

[22] S. Huang, D. Wang, R. Zhao, and Q. Zhang, "Wiga: A wifi-based contactless activity sequence recognition system based on deep learning," in *2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*. IEEE, 2019, pp. 69–74.

[23] S. Mukherjee, "The annotated resnet-50," 2022, accessed: May 23, 2024. [Online]. Available: https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758

[24] Gorlapraveen, "Vgg16.png," 2021, accessed: May 23, 2024. [Online]. Available: https://commons.wikimedia.org/wiki/File:VGG16.png

[25] A. Khattar and S. Quadri, "Generalization of convolutional network to domain adaptation network for classification of disaster images on twitter," *Multimedia Tools and Applications*, vol. 81, no. 21, pp. 30 437–30 464, 2022.

[26] F. J. P. Montalbo and A. S. Alon, "Empirical analysis of a fine-tuned deep convolutional model in classifying and detecting malaria parasites from blood smears," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 15, no. 1, pp. 147–165, 2021.

[27] U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E. T. Matson, "Real-time and accurate drone detection in a video with a static background," *Sensors*, vol. 20, no. 14, p. 3856, 2020.

[28] A. BRITAL, "Inception v3 cnn architecture explained," 2021, accessed: May 23, 2024. [Online]. Available: https://medium.com/@AnasBrital98/inception-v3-cnn-architecture-explained-691cfb7bba08

[29] A. Mehmood, "Efficient anomaly detection in crowd videos using pre-trained 2d convolutional neural networks," *IEEE Access*, vol. 9, pp. 138 283–138 295, 2021.

[30] N. Radwan, "Leveraging sparse and dense features for reliable state estimation in urban environments," Ph.D. dissertation, University of Freiburg, Freiburg im Breisgau, Germany, 2019.

[31] J. Palanimeera and K. Ponmozhi, "Classification of yoga pose using machine learning techniques," *Materials Today: Proceedings*, vol. 37, pp. 2930–2933, 2021.

[32] "Support vector machine algorithm," accessed: May 23, 2024. [Online]. Available: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[33] L. Armstrong, "Decision tree diagrams: what they are and how to use them," 2021, accessed: May 23, 2024. [Online]. Available: https://blog.mindmanager.com/decision-tree-diagrams/

[34] D. Shah, V. Rautela, C. Sharma *et al.*, "Yoga pose detection using posenet and k-nn," in *2021 International conference on computing, communication and green engineering (CCGE)*. IEEE, 2021, pp. 1–4.

[35] "K-nearest neighbor(knn) algorithm for machine learning," accessed: May 23, 2024. [Online]. Available: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[36] P. Szczuko, "Deep neural networks for human pose estimation from a very low resolution depth image," *Multimedia Tools and Applications*, vol. 78, no. 20, pp. 29 357–29 377, 2019.

[37] C. Bento, "Multilayer perceptron explained with a real-life example and python code: Sentiment analysis," 2021, accessed: May 23, 2024. [Online]. Available: https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141

[38] K. Shyrokykh, M. Girnyk, and L. Dellmuth, "Short text classification with machine learning in the social sciences: The case of climate change on twitter," *Plos one*, vol. 18, no. 9, p. e0290762, 2023.

[39] B. Gupta, "Bagging in machine learning," 2023, accessed: May 23, 2024. [Online]. Available: https://www.scaler.com/topics/machine-learning/bagging-in-machine-learning/

[40] U. Bahukhandi and S. Gupta, "Yoga pose detection and classification using machine learning techniques," *Int Res J Mod Eng Technol Sci*, vol. 3, no. 12, pp. 13–15, 2021.

[41] D. Gunay, "Random forest," 2023, accessed: May 23, 2024. [Online]. Available: https://medium.com/@denizgunay/random-forest-af5bde5d7e1e

[42] J. Guo, L. Yang, R. Bie, J. Yu, Y. Gao, Y. Shen, and A. Kos, "An xgboost-based physical fitness evaluation model using advanced feature selection and bayesian hyper-parameter optimization for wearable running monitoring," *Computer Networks*, vol. 151, pp. 166–180, 2019.

[43] R. Guo, Z. Zhao, T. Wang, G. Liu, J. Zhao, and D. Gao, "Degradation state recognition of piston pump based on iceemdan and xgboost," *Applied Sciences*, vol. 10, no. 18, p. 6593, 2020.

[44] J. Z. Tan, C. P. Lee, K. M. Lim, and J. Y. Lim, "Yoga pose estimation with machine learning," in *2023 11th International Conference on Information and Communication Technology (ICoICT)*. IEEE, 2023, pp. 260–265.

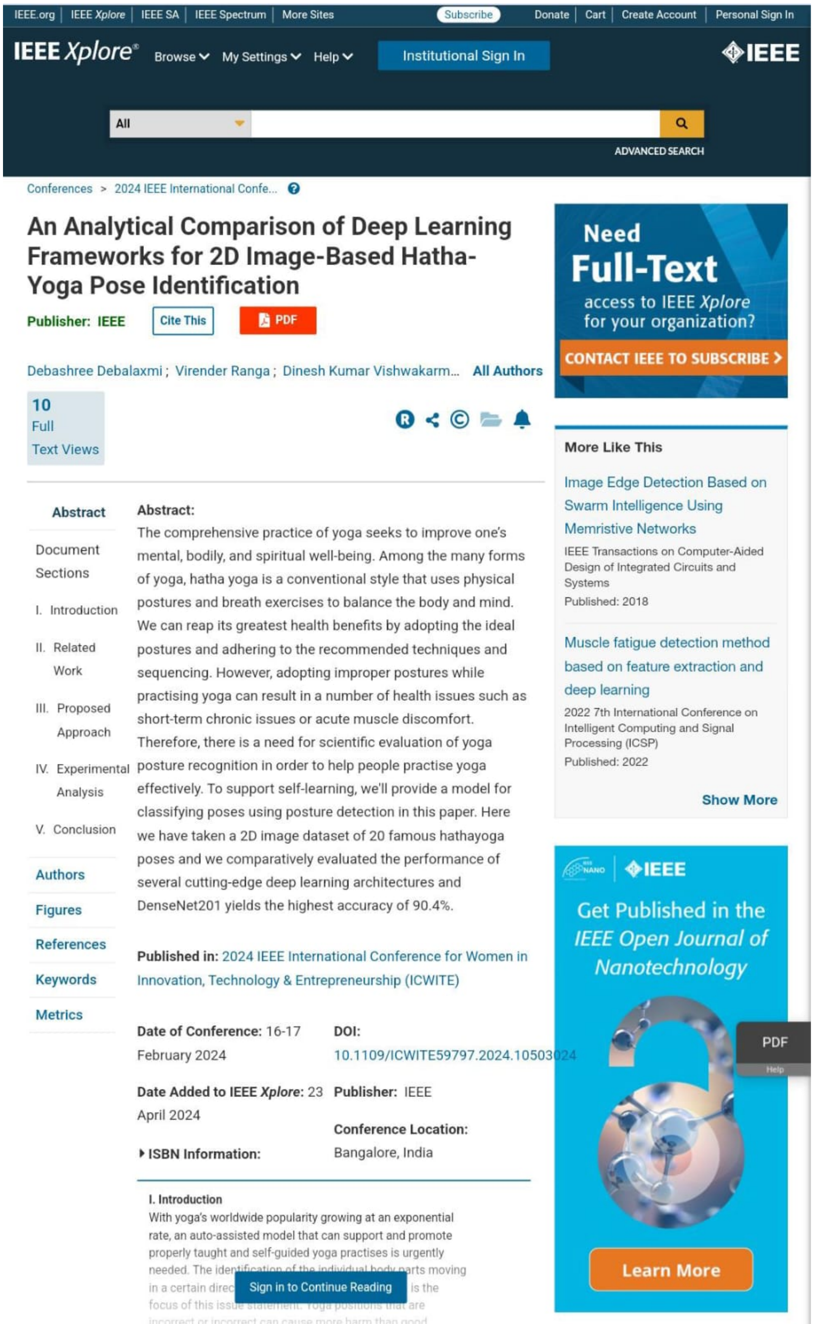
[45] Z. Su, Q. Liu, C. Zhao, and F. Sun, "A traffic event detection method based on random forest and permutation importance," *Mathematics*, vol. 10, no. 6, p. 873, 2022.

List of Publications

1. D. Debalaxmi, V. Ranga and D. K. Vishwakarma, *"An Analytical Comparison of Deep Learning Frameworks for 2D Image-Based Hatha-Yoga Pose Identification"*, **2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)**, Bangalore, India, 2024, pp. 271-276, doi: 10.1109/ICWITE59797.2024.10503024.

2. D. Debalaxmi, D. K. Vishwakarma and V. Ranga, *"Analyzing Yoga Pose Recognition: A Comparison of MediaPipe and YOLO Keypoint Detection with Ensemble Techniques"*, **2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)**, Salem, India, 2024, pp. 1011-1017, doi: 10.1109/ICAAIC60222.2024.10574984.

Publication in conference 1: ICWITE 2024



IEEE.org | IEEE *Xplore* | IEEE SA | IEEE Spectrum | More Sites   Subscribe   Donate | Cart | Create Account | Personal Sign In

IEEE *Xplore*® Browse My Settings Help Institutional Sign In IEEE

All

ADVANCED SEARCH

Conferences > 2024 IEEE International Confe...

An Analytical Comparison of Deep Learning Frameworks for 2D Image-Based Hatha-Yoga Pose Identification

Publisher: IEEE Cite This PDF

Debashree Debalaxmi ; Virender Ranga ; Dinesh Kumar Vishwakarm... **All Authors**

10 Full Text Views

Abstract

Document Sections

I. Introduction

II. Related Work

III. Proposed Approach

IV. Experimental Analysis

V. Conclusion

Authors

Figures

References

Keywords

Metrics

Abstract:
The comprehensive practice of yoga seeks to improve one's mental, bodily, and spiritual well-being. Among the many forms of yoga, hatha yoga is a conventional style that uses physical postures and breath exercises to balance the body and mind. We can reap its greatest health benefits by adopting the ideal postures and adhering to the recommended techniques and sequencing. However, adopting improper postures while practising yoga can result in a number of health issues such as short-term chronic issues or acute muscle discomfort. Therefore, there is a need for scientific evaluation of yoga posture recognition in order to help people practise yoga effectively. To support self-learning, we'll provide a model for classifying poses using posture detection in this paper. Here we have taken a 2D image dataset of 20 famous hathayoga poses and we comparatively evaluated the performance of several cutting-edge deep learning architectures and DenseNet201 yields the highest accuracy of 90.4%.

Published in: 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)

Date of Conference: 16-17 February 2024

DOI: 10.1109/ICWITE59797.2024.10503024

Date Added to IEEE *Xplore*: 23 April 2024

Publisher: IEEE

Conference Location: Bangalore, India

ISBN Information:

I. Introduction
With yoga's worldwide popularity growing at an exponential rate, an auto-assisted model that can support and promote properly taught and self-guided yoga practises is urgently needed. The identification of the individual body parts moving in a certain direc... is the focus of this issue statement. Yoga positions that are incorrect or incorrect can cause more harm than good

Sign in to Continue Reading

Need Full-Text access to IEEE *Xplore* for your organization? CONTACT IEEE TO SUBSCRIBE >

More Like This

Image Edge Detection Based on Swarm Intelligence Using Memristive Networks
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems
Published: 2018

Muscle fatigue detection method based on feature extraction and deep learning
2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)
Published: 2022

Show More

Get Published in the *IEEE Open Journal of Nanotechnology*

Learn More

Acceptance from conference 1: ICWITE 2024

Acceptance Notification
1 message

Microsoft CMT <email@msr-cmt.org> Thu, 11 Jan 2024 at 14:06
Reply to: Pushpa Mala S <pushpasiddaraju@gmail.com>
To: Debashree Debalaxmi <debashreedebalaxmi@gmail.com>

Dear Debashree Debalaxmi,
Greetings from ICWITE 2024!!!
Congratulations!!!
We are pleased to inform you that your paper has been Accepted for Oral presentation at the IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE 2024), to be held during 16th - 17th February 2024, at Vemana Institute of Technology, Bengaluru.
Your paper submission details referenced are per below:

Submission ID: 871
Title: A Comparative Study of Deep Learning Architectures for 2D Image-Based Hatha-Yoga Pose Recognition
Status: Accept

The review comments will be enabled after 16th Jan 2024 on the primary authors CMT submission portal.
The conference proceedings will be submitted to the IEEE Xplore® digital library.
Please read the following guidelines carefully before submitting the final version of paper:
1. Please prepare your Camera-ready paper (as a word document only) by strictly following the guidelines mentioned below: (Articles not in format will not be submitted to IEEE Xplore)
a. The Final Camera Ready Copy should be strictly according to IEEE format specified by IEEE. Use the A4 size template at http://www.ieee.org/conferences_events/conferences/publishing/templates.html
b. Mention affiliations of all the authors & mail id of all the authors. Do not mention Salutations like "Dr./Mr./Mrs./Prof./ Research Scholar/Student"
c. Run a Spellcheck and proofread your paper thoroughly to ensure that it contains no typo and grammatical errors
d. Include good quality figures/graphics, Tables must be drawn in table format, not as an image.
e. Text description and callouts for all the figure numbers, table numbers and equation numbers must be present in the paper as per the order of their occurrence
f. Incorporate all the changes as per the reviewers' comments before submitting the camera ready manuscript.
h. Ensure that the Similarity Index (Plagiarism Report) is less than 30% Including References.
i. Include Header and Footer in your Paper for the First Page Only.
Header: 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE 2024) ----- [Align Left, Time New Roma, Font 10]
Footer: 978-1-6654-0017-6/21/$31.00 ©2024 IEEE [As per Final camera Ready template]

k. A mail will be sent for copyright transfer soon. Authors are requested complete the copyright transfer.
2. Please ensure that the camera ready copy is in the final submission form since further changes will not be possible.
4. It is mandatory for the authors to present their paper at the conference. Without presentation the accepted paper will not be considered for publication.
We look forward to your participation at the conference. If you are not the presenting author, please forward this message to your co-author who would be presenting the paper.

Brochure of conference 1: ICWITE 2024

IEEE ICWITE 2024

16th & 17th February 2024 | Bangalore | India

Venue: Vemana Institue of Technology, Bangalore

ICWITE 2024
Engineering Sustainable Futures Responsibly

International Conference for Women in Innovation, Technology and Entrepreneurship, ICWITE is a flagship conference of the IEEE WIE AG, Bangalore Section, India. The second edition of ICWITE will be held at Bangalore, India between 16th – 17th February 2024. The conference will feature plenary talks, tutorials, workshops and invited papers by distinguished researchers and technologists as well as contributed papers from academics and industry professionals. ICWITE 2024 is a platform for women to showcase their expertise in Innovation, as technologists, researchers, entrepreneurship and industry leadership. It is an International Conference that aims to celebrate the achievements and accomplishments from across the globe. The conference enables them to share their ideas on emerging technologies and innovative solutions that can guide and lead towards a safer, sustainable and better tomorrow.

IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE 2024) has been created to provide opportunities to foster an ecosystem for innovation and entrepreneurship among the Women. This international conference attempts to celebrate and showcase women who are leaders in Innovation, Entrepreneurship, and Future Technology. The theme of the second edition of ICWITE 2024 will be "Engineering Sustainable Futures Responsibly"

The IEEE ICWITE 2024 Summit will be held at Bangalore, India between 16th – 17th February 2024. This is the flagship conference of IEEE WIE Bangalore Section AG. It will be held under the aegis of IEEE Bangalore Section.

IEEE Women in Engineering, IEEE Bangalore Section

IEEE BANGALORE SECTION

General Chair

Deepa Shenoy,UVCE, Past chair IEEE Bangalore Section & WiE Bangalore Section AG

General Co-Chair

Vijayalakshmi K, BMSCE, Past Chair IEEE WIE Bangalore Section AG

Ashwini V R, Canara Engineering College, Chair IEEE WIE Bangalore Section AG 2024

Organizing Committee Chair:

Ashwini Appaji, BMSCE, IEEE WIE Bangalore Section AG 2024

Ankitha A, WiE Faculty Advisor Vemana Institute of Technology, Ground Operations IEEE ICWITE 2024

Aruna Reddy, Vemana Institute of Technology, Ground operations IEEE ICWITE 2024

Vijayasree K, USRC-ISRO, IEEE WIE Bangalore Section AG 2024

Supriya Kamthania, HPE, Vice Chair IEEE WIE Bangalore Section AG 2024

Chief Guest:

Ceena Sunil, Deputy Director, Scientist 'G' Mechanical System Area, ISTRAC ISRO, Bangalore

Papers/ posters/ dissertation will be made in the following areas:

- AI/ ML, Blockchain,
- Computer Networks & Cybersecurity
- Biomedical Engineering, Bioinformatics & Biomimicry
- Cloud, Big Data & 5G Communications
- Quantum Computing
- Devices, Materials & Processing
- IOT, Sensor Technologies & Wearables
- Digital Reality – AR/VR, Digital Twins
- Smart Grids & Renewable Energy
- Robotics, Control & Automation
- Smart Cities & Infrastructure Technologies
- HealthTech, FinTech, EduTech, AgriTech - Applications for Humanitarian Technology

IMPORTANT DATES

Paper Submission starts on:	1st Aug 2023
Deadline for submission of Paper	10th Dec 2023
Notification of acceptance:	25th Dec 2023
Accepted Paper Author Registration	31st Jan 2024
Submission of Camera-ready papers:	3rd Feb 2024

Conference Awards

Academic Awards

1. ICWITE Best Doctoral Thesis
2. ICWITE Best Paper Award
3. ICWITE Best Poster Award

Professional Awards

1. Impactful Innovation Awards
2. Technologist of the Year
3. Humanitarian of the Year
4. Entrepreneur of the Year
5. HE 4 HER HERO Award

IBM TVS NITTE EDUCATION TRUST IEEE COMPUTER SOCIETY Bangalore Chapter IEEE Signal Processing Society BANGALORE CHAPTER IEEE Photonics Society Bangalore Chapter IEEE TEMS Bangalore IEEE young professionals Bangalore Section

Registration for conference 1: ICWITE 2024

CONECCT

eXPLARA

IEEE ICWITE 2024

16 Feb - 17 Feb 2024 (Asia/Kolkata), Bengaluru.

Order No. - KT302KS-EFBJICJ-240124-151323-634 Attendees : **1** AMOUNT : ₹ 2588.5

Name	Debashree Debalaxmi	
Registration No.	2673	
Registration Type	Early Bird Non-IEEE Student (Indian)	
Registration Price	INR 2500	

Merchant Transaction No : KT302KS-EFBJICJ-240124-151834-V496
Transaction Id : order_NSPDUGIBV6A8nu

Time & Venue

Friday, Feb 16 2024 09:00 AM to Saturday, Feb 17 2024 06:00 PM (Asia/Kolkata)

At Vemana Institute of Technology, 3rd Block, No. 1, Mahakavi Vemana Rd, Koramangala 3 Block, Koramangala, Bengaluru, Karnataka, India, 560034

ORGANIZER

IEEE ICWITE 2024

Note : For any further clarification, you can reach out to **IEEE Bangalore Section** at **ieeeconecct@gmail.com** .

Terms and Conditions

Your Order No - KT302KS-EFBJICJ-240124-151323-634

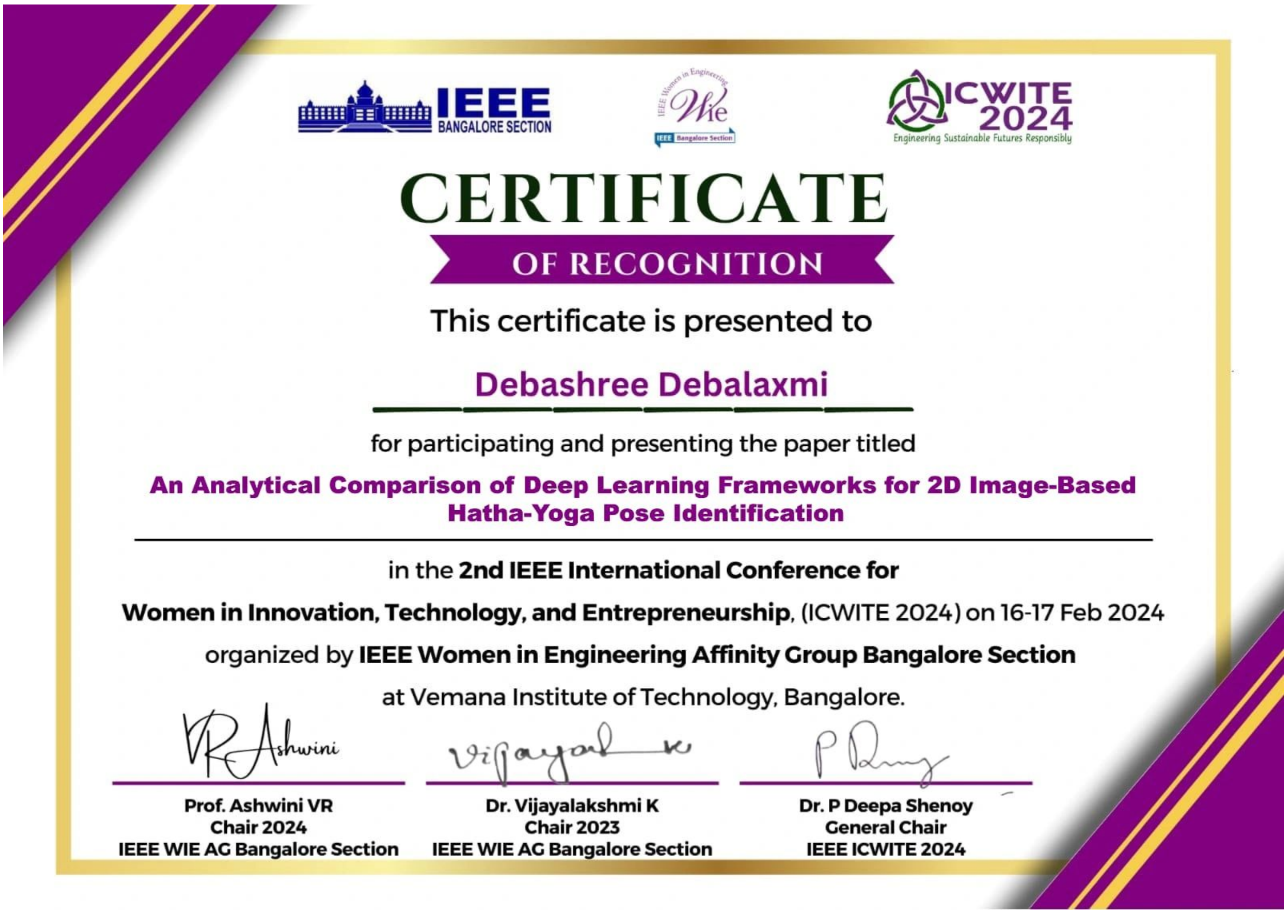
If the event gets cancelled, the event organizer will be responsible for taking decision about refund. Explara is not responsible for your cancellation and refund.

Please carry ticket printout for entry to the event.

Please carry valid identity proof to the venue.

Cancellation and Rescheduling of tickets is not allowed.

Certificate from conference 1: ICWITE 2024



IEEE BANGALORE SECTION

IEEE Women in Engineering WIE IEEE Bangalore Section

ICWITE 2024 Engineering Sustainable Futures Responsibly

CERTIFICATE

OF RECOGNITION

This certificate is presented to

Debashree Debalaxmi

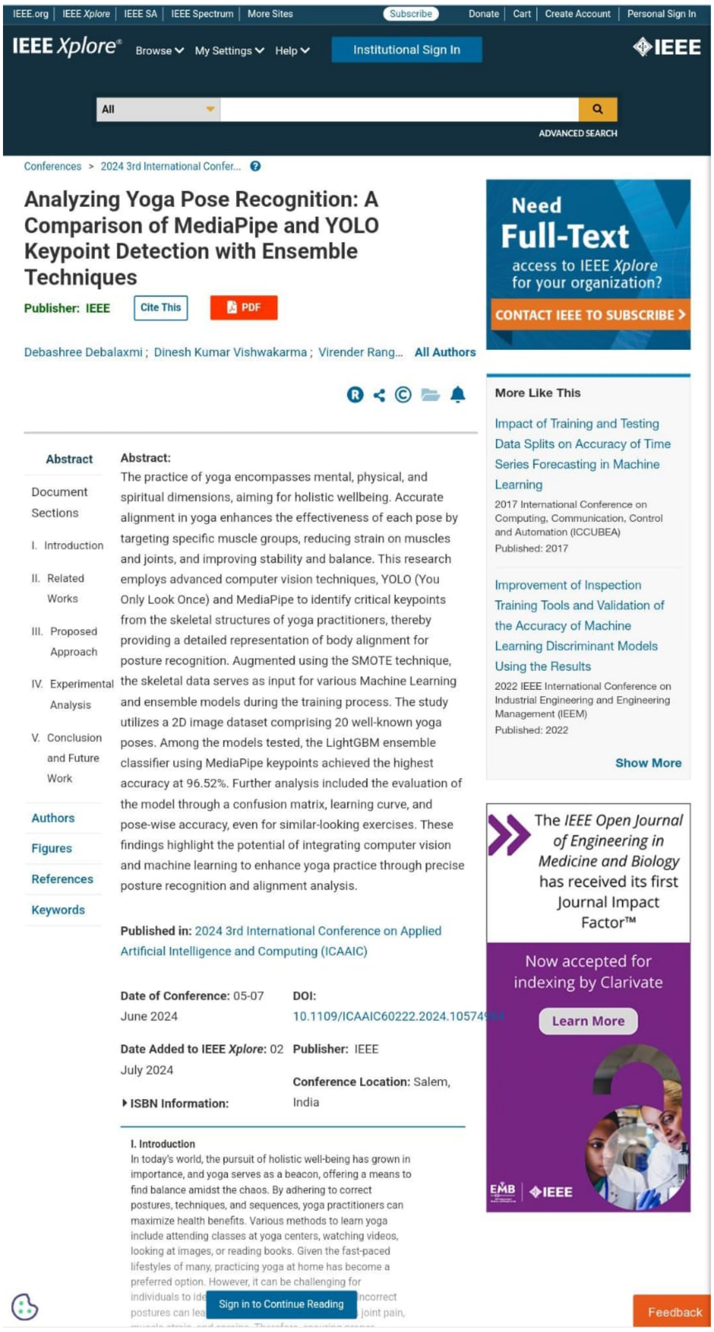
for participating and presenting the paper titled

An Analytical Comparison of Deep Learning Frameworks for 2D Image-Based Hatha-Yoga Pose Identification

in the **2nd IEEE International Conference for Women in Innovation, Technology, and Entrepreneurship**, (ICWITE 2024) on 16-17 Feb 2024 organized by **IEEE Women in Engineering Affinity Group Bangalore Section** at Vemana Institute of Technology, Bangalore.

Prof. Ashwini VR **Chair 2024** **IEEE WIE AG Bangalore Section**	**Dr. Vijayalakshmi K** **Chair 2023** **IEEE WIE AG Bangalore Section**	**Dr. P Deepa Shenoy** **General Chair** **IEEE ICWITE 2024**

Publication in conference 2: ICAAIC 2024



IEEE.org | IEEE Xplore | IEEE SA | IEEE Spectrum | More Sites | Subscribe | Donate | Cart | Create Account | Personal Sign In

IEEE Xplore® Browse My Settings Help Institutional Sign In IEEE

All | ADVANCED SEARCH

Conferences > 2024 3rd International Confer...

Analyzing Yoga Pose Recognition: A Comparison of MediaPipe and YOLO Keypoint Detection with Ensemble Techniques

Publisher: IEEE | Cite This | PDF

Debashree Debalaxmi ; Dinesh Kumar Vishwakarma ; Virender Rang... All Authors

Abstract | Document Sections: I. Introduction; II. Related Works; III. Proposed Approach; IV. Experimental Analysis; V. Conclusion and Future Work | Authors | Figures | References | Keywords

Abstract:
The practice of yoga encompasses mental, physical, and spiritual dimensions, aiming for holistic wellbeing. Accurate alignment in yoga enhances the effectiveness of each pose by targeting specific muscle groups, reducing strain on muscles and joints, and improving stability and balance. This research employs advanced computer vision techniques, YOLO (You Only Look Once) and MediaPipe to identify critical keypoints from the skeletal structures of yoga practitioners, thereby providing a detailed representation of body alignment for posture recognition. Augmented using the SMOTE technique, the skeletal data serves as input for various Machine Learning and ensemble models during the training process. The study utilizes a 2D image dataset comprising 20 well-known yoga poses. Among the models tested, the LightGBM ensemble classifier using MediaPipe keypoints achieved the highest accuracy at 96.52%. Further analysis included the evaluation of the model through a confusion matrix, learning curve, and pose-wise accuracy, even for similar-looking exercises. These findings highlight the potential of integrating computer vision and machine learning to enhance yoga practice through precise posture recognition and alignment analysis.

Published in: 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)

Date of Conference: 05-07 June 2024

Date Added to IEEE Xplore: 02 July 2024

ISBN Information:

DOI: 10.1109/ICAAIC60222.2024.10574

Publisher: IEEE

Conference Location: Salem, India

I. Introduction
In today's world, the pursuit of holistic well-being has grown in importance, and yoga serves as a beacon, offering a means to find balance amidst the chaos. By adhering to correct postures, techniques, and sequences, yoga practitioners can maximize health benefits. Various methods to learn yoga include attending classes at yoga centers, watching videos, looking at images, or reading books. Given the fast-paced lifestyles of many, practicing yoga at home has become a preferred option. However, it can be challenging for individuals to ide... incorrect postures can lea... joint pain,

Sign in to Continue Reading

Acceptance from conference 2: ICAAIC 2024

ICAAIC

3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC-2024)

5-7, June 2024 | Salem, India

icaaic.contact@gmail.com | http://icaaic.com/2024/

Letter of Acceptance

Author Name: Debashree Debalaxmi, Dinesh Kumar Vishwakarma, Virender Ranga

Affiliation Details: Delhi Technological University, India.

Dear Author:

It is with great pleasure that we extend our warmest congratulations to you on the acceptance of the paper titled **"Analyzing Yoga Pose Recognition: A Comparison of MediaPipe and YOLO Keypoint Detection with Ensemble Techniques "- PAPER ID: ICAAIC 673** for presentation at the 3rd International Conference on Applied Artificial Intelligence and Computing, scheduled to be held in R P Sarathy Institute of Technology, Salem, India from June 5th to June 7th, 2024.

Your submission was subjected to a rigorous review process, and the result that your paper has been selected for inclusion in our conference program. We believe that your contribution will greatly enrich the discussions and knowledge exchange at our event.

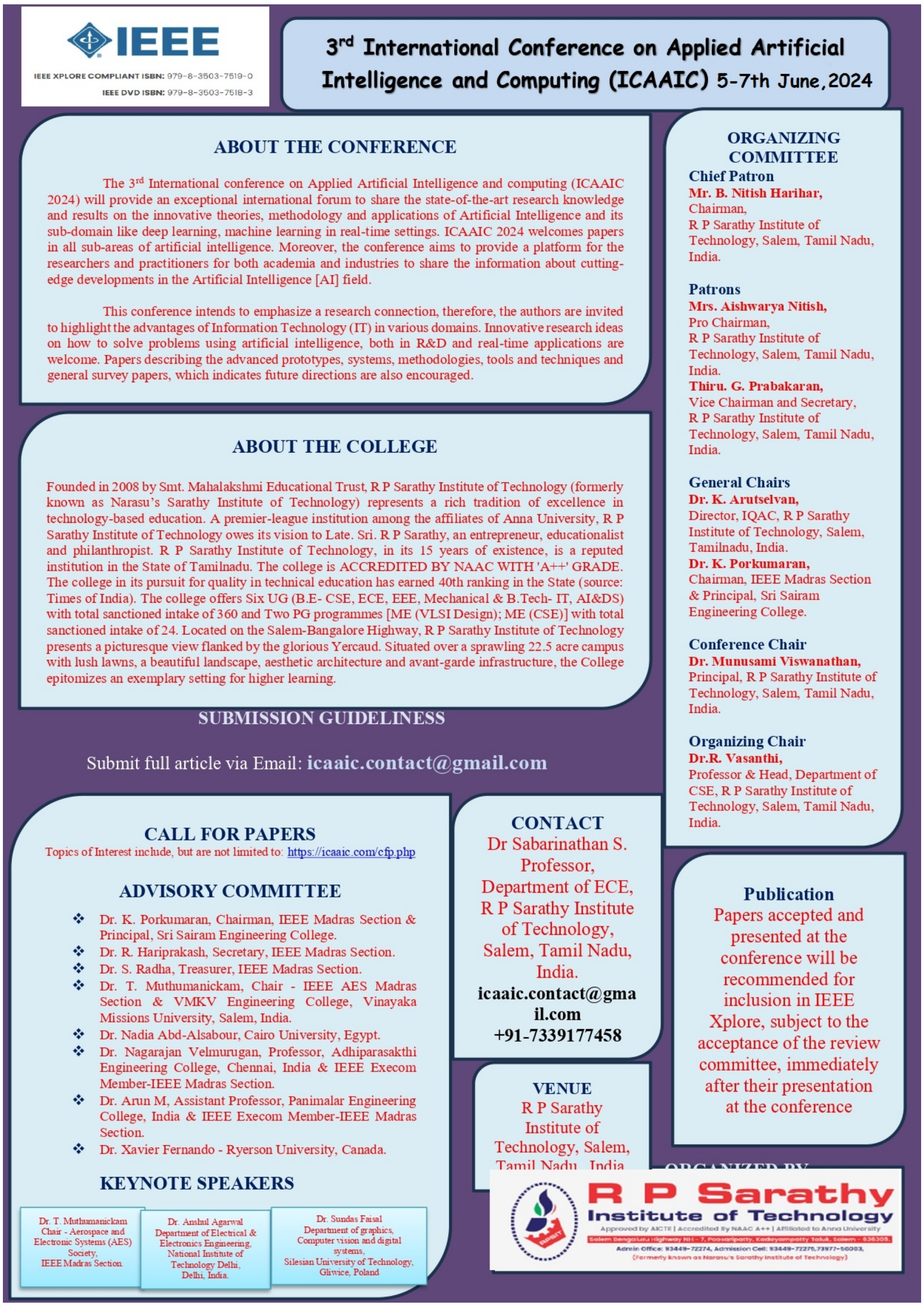
Your participation will undoubtedly contribute to the success of the 3rd International Conference on Applied Artificial Intelligence and Computing.

Once again, congratulations on your acceptance, and we anticipate your valuable contribution to our conference.

Yours Sincerely,

Dr. Munusami Viswanathan,
Principal,
R P Sarathy Institute of Technology,
Salem, Tamil Nadu, India.

Brochure of conference 2: ICAAIC 2024



Registration for conference 2: ICAAIC 2024

RECEIPT

Organizer
intlconf.publications
intlconf.publications@gmail.com

Receipt #: 11402912
Created: Apr 30, 2024

Bill To
Debashree Debalaxmi
debashreedebalaxmi@gmail.com

Event Name:3rd International Conference on Applied Artificial Intelligence and Computing

Order Id: 374511714452500322

Item & Description	Qty	Rate	Amount
Student Non IEEE Member	1	7750.00	7750.00

Sub Total :	7750.00
Booking fee :	182.90
Total :	**INR 7932.90**

Powered by Townscript

Certificate from conference 2: ICAAIC 2024

IEEE

XPLORE COMPLIANT ISBN
979-8-3503-7519-0

ICAAIC

R P Sarathy Institute of Technology

Certificate of Presentation

This is to certify that

Debashree Debalaxmi

has successfully presented the paper entitled

Analyzing Yoga Pose Recognition: A Comparison of MediaPipe and YOLO Keypoint Detection with Ensemble Techniques

at the

3rd International Conference on

Applied Artificial Intelligence and Computing (ICAAIC 2024) organized by

R P Sarathy Institute of Technology, Salem, India

on 5-7, June 2024.

Session Chair

Organizing Chair
Dr. R. Vasanthi

Conference Chair
Dr. Munusami Viswanathan

ANNEXURE-IV

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis: **A Comparative Study of Deep Learning and Ensemble Methodologies with Keypoint Detection for 2D Image-Based Yoga Pose Recognition**

Total Pages: **60**

Name of the Scholar: **Debashree Debalaxmi**

Supervisors:

(1) **Prof. Dinesh Kumar Vishwakarma**
(2) **Dr. Virender Ranga**

Department: **Department of Information Technology**

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: **Turnitin** Similarity Index: **6%** Total Word Count: **14578**

Date: 29/05/2024

Candidate's Signature **Signature of Supervisor 1** **Signature of Supervisor 2**

PAPER NAME

(1).pdf

WORD COUNT	CHARACTER COUNT
14578 Words	82234 Characters
PAGE COUNT	FILE SIZE
60 Pages	5.7MB
SUBMISSION DATE	REPORT DATE
May 29, 2024 7:58 PM GMT+5:30	May 29, 2024 7:59 PM GMT+5:30

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 6% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database

Excluded from Similarity Report

- Submitted Works database
- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 8 words)