Yoga Pose Detection Using Deep Learning

K. Sireesha[1], R. Pavan Kumar[2], S. Nandini[3], Sk. Nayeem[4], Sk. Baji Baba[5]
[1]*Assistant Professor, Computer Science and Engineering - Artificial Intelligence and Machine learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh*
[2,3,4,5]*Students, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh*

***Abstract*—The rapid advancement of deep learning has enabled the development of intelligent applications for various domains, including health and fitness. This project presents a web-based Yoga Pose Detection System that utilizes deep learning techniques to accurately identify and analyze yoga poses. The system integrates state-of-the-art convolutional neural network architectures, including MobileNet, VGG16, and DenseNet, to classify yoga postures based on image inputs.**

Users can interact with the system through two primary modes: image-based pose prediction and real-time pose detection using a webcam. The integration of a user-friendly interface ensures seamless interaction, while an admin panel provides analytics, user management, and content control. Additionally, sentiment analysis is applied to user feedback to enhance system engagement and improve recommendations. A third-party API-based recommendation system suggests yoga poses tailored to individual health goals.

This platform aims to provide an interactive and efficient approach to learning yoga by offering personalized feedback, pose recommendations, and a dynamic knowledge base. The integration of deep learning and real-time processing makes this system a valuable tool for yoga practitioners, fitness enthusiasts, and instructors.

Keywords: Yoga Pose Detection, Deep Learning, MobileNet, VGG16, DenseNet, Real-Time Analysis, Sentiment Analysis, Health and Fitness.

I. INTRODUCTION

The practice of yoga has gained widespread popularity across the globe due to its numerous physical and mental health benefits. With the rise of technology, integrating artificial intelligence and deep learning into yoga practice has opened new avenues for assisting individuals in performing yoga poses correctly. The Yoga Pose Detection System leverages deep learning techniques to analyze and predict yoga poses, providing users with real-time feedback to enhance their practice.

This project utilizes state-of-the-art convolutional neural networks (CNNs), including MobileNet, VGG16, and DenseNet, to accurately classify various yoga postures. By allowing users to upload images or use a webcam for real-time pose detection, the system ensures an interactive and engaging experience. The integration of a user-friendly interface further enhances accessibility, making it easier for individuals at different skill levels to improve their posture and alignment.

In addition to pose detection, the system includes a sentiment analysis module to analyze user feedback and enhance the overall experience. The admin dashboard provides insightful analytics, enabling administrators to monitor platform activity and improve functionality. Furthermore, a blog section fosters community engagement by allowing users to share their experiences and knowledge related to yoga.

The primary goal of this system is to bridge the gap between traditional yoga learning and modern technological advancements, offering a smart and effective way to practice yoga. By utilizing deep learning techniques, the platform ensures accurate pose recognition, making it a valuable tool for yoga practitioners, fitness enthusiasts, and instructors.

II. LITERATURE SURVEY

The advancement of artificial intelligence, particularly in computer vision and deep learning, has enabled significant progress in human posture estimation and activity recognition. Numerous studies have focused on the development of models and techniques to enhance the accuracy of pose detection in various applications, including sports, healthcare, and fitness. Yoga, being a discipline that requires precision in postures, has gained the attention of researchers aiming to integrate AI- based solutions for pose correction and feedback.

Several deep learning architectures have been extensively used for human pose estimation. MobileNet, a lightweight convolutional neural network (CNN), has proven effective in real-time applications due to its efficiency in processing images with minimal computational resources [1]. Researchers have adopted MobileNet for mobile-based fitness applications to track user movements accurately. Similarly, VGG16, a deep CNN model, has been widely used for image classification tasks, demonstrating high accuracy in recognizing complex postures [2]. DenseNet, which utilizes dense connections be- tween layers to enhance gradient flow and feature propagation, has shown superior performance in classification tasks, making it a viable choice for yoga pose detection [3].

Human pose estimation is a well-researched domain, with methodologies evolving from classical computer vision techniques to deep learning-based models. Earlier techniques relied on handcrafted feature extraction methods, such as Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT), which had limitations in handling variations in lighting, occlusions, and body deformations [4]. With the emergence of deep learning, Convolutional Pose Machines (CPMs) and OpenPose have significantly improved the accuracy of human pose estimation by detecting key points and body landmarks efficiently [5], [6].

In the context of yoga, studies have explored the application of pose estimation models to provide automated feedback for practitioners. Some research efforts have utilized transfer learning approaches, where pre-trained models like ResNet and Inception are fine-tuned for yoga pose classification, achieving promising results [7], [8]. Additionally, the integration of recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks has enabled real-time assessment of yoga sequences, allowing dynamic pose evaluation instead of static image classification [9].

Furthermore, sentiment analysis has been incorporated into health and fitness applications to understand user experiences. Natural language processing (NLP) techniques, such as transformer-based models like BERT, have been applied to analyze user reviews and feedback, helping to enhance the overall system by addressing user concerns [10]. The integration of such techniques in yoga pose detection systems can contribute to improving user engagement and satisfaction. Recent advancements in AI-driven recommendation systems have also played a crucial role in fitness applications. Personalized yoga pose recommendations based on user health goals have been developed using collaborative filtering and content- based filtering approaches [11]. These techniques utilize user preferences, historical data, and third-party APIs to provide tailored suggestions, enhancing the personalization aspect of yoga learning platforms.

Overall, the literature demonstrates that deep learning-based pose detection models, coupled with real-time feedback mechanisms and sentiment analysis, hold immense potential for transforming yoga practice. The proposed Yoga Pose Detection System builds upon these advancements by integrating multiple deep learning architectures, real-time pose estimation, and an interactive user interface to create a holistic yoga learning experience.

III. METHODOLOGY

The Yoga Pose Detection System follows a structured methodology that integrates deep learning, computer vision, and web technologies to provide an interactive and intelligent yoga learning experience. The methodology comprises multiple stages, starting from data acquisition to pose prediction and feedback generation.

A. *System Overview*

The system architecture is designed to ensure seamless integration of various components, including front-end and back-end services, machine learning models, and database management. The architecture is depicted in Figure 1.

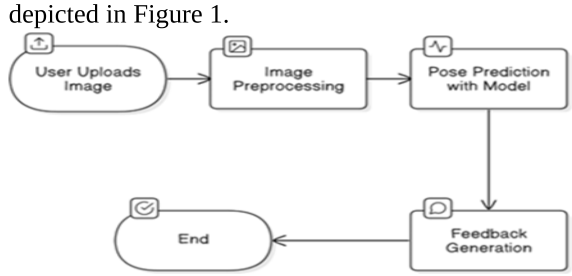

Fig. 1. System Architecture of the Yoga Pose Detection System

The system consists of three primary modules:

- User Interface: A web-based platform developed using HTML, CSS, JavaScript, and React that allows users to interact with the system.
- Backend Processing: Implemented using Django and integrated with TensorFlow/Keras models to handle image processing and pose predictions.
- Database Management: SQLite3 is used for storing user authentication details, feedback, and other relevant data.

B. Data Acquisition and Preprocessing

The deep learning models require high-quality datasets containing images of various yoga poses. These images are collected from publicly available datasets and augmented to improve model generalization. The preprocessing steps in- volve:

- Resizing images to a uniform dimension (224x224 pixels).
- Normalizing pixel values between 0 and 1.
- Applying data augmentation techniques such as rotation, flipping, and brightness adjustment.
- Converting images into numerical tensors suitable for deep learning models.

C. Deep Learning Model Implementation

The system employs three deep learning architectures: MobileNet, VGG16, and DenseNet. Each model is trained separately and evaluated based on accuracy and computational efficiency.

1) Mathematical Representation of the Model: The classification problem is formulated using the softmax function:

$$P(y_i|x) = \frac{e^{W_i x + b_i}}{\sum_{j=1}^{N} e^{W_j x + b_j}} \qquad (1)$$

where: W_i and b_i are the weight and bias terms for class i.

N represents the total number of yoga pose classes. For model optimization, categorical cross-entropy is used as the loss function:

$$L = \sum_{i=1}^{N} y_i \log P(y_i|x) \qquad (2)$$

where y_i is the true label and $P(y_i|x)$ is the predicted probability.

A. Real-Time Pose Prediction and Feedback

The real-time pose detection system utilizes a webcam to capture video frames, which are processed by the deep learning models. The key steps involved are:

- Capturing frames from the webcam at regular intervals.
- Converting frames into grayscale and resizing them to match the model's input format.
- Running the frames through the trained model to predict the yoga pose.
- Providing immediate feedback on pose correctness.

F. Yoga Pose Recommendation and Sentiment Analysis

To enhance user engagement, the system includes: Recommendation System: Uses third-party APIs to suggest yoga poses based on user health conditions.

- Sentiment Analysis: Analyzes user feedback using NLP techniques to improve content and recommendations. $P(y_i|x)$ is the probability of the input image x belonging to class i.

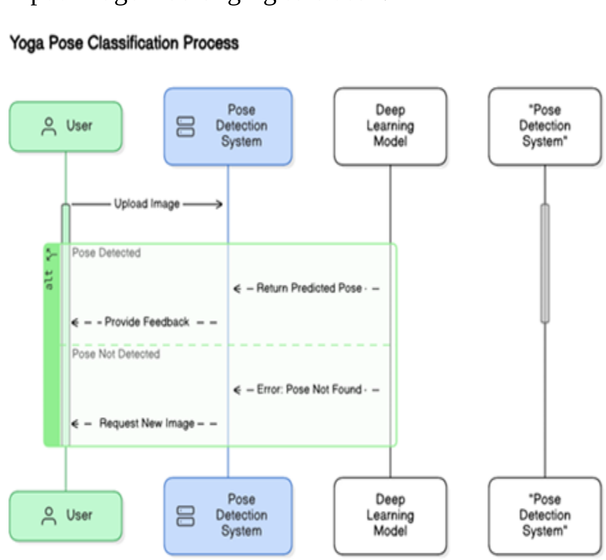

Fig. 2. Methodology Flowchart for Yoga Pose Detection

B. User Authentication and Data Security

The system employs email OTP verification for secure authentication. User roles (admin and regular users) are man- aged through Django's built-in authentication system, ensuring access control and data privacy.

IV. CONCLUSION

The proposed methodology integrates deep learning models with a user-friendly web interface to provide real-time yoga pose detection. The system leverages advanced computer vision techniques to analyze and predict poses with high accuracy. With additional features such as authentication, sentiment analysis, and pose recommendations, the platform offers a comprehensive yoga learning experience.

V. IMPLEMENTATION

The implementation of the Yoga Pose Detection System involves a combination of web development, machine learning, and deep learning techniques. The main goal is to enable real-time pose detection, provide feedback to users, and of- fer personalized recommendations. The following subsections describe the core implementation steps in the project.

A. System Architecture

The system architecture is built with several key com- ponents: the frontend for user interaction, the backend for handling business logic, and the deep learning models for pose detection. The architecture integrates these components to ensure smooth functionality.

The system architecture can be broken down as follows:

- Frontend: The frontend is built using HTML, CSS, and JavaScript, with React for dynamic content rendering. The frontend displays yoga pose images, user feedback, and poses in real-time.
- Backend: The backend is implemented in Python using the Django framework. This handles user authentication, data storage, and integrates machine learning models for pose detection.
- Machine Learning Models: TensorFlow/Keras is used to implement deep learning models like MobileNet, VGG16, and DenseNet for predicting yoga poses. These models are trained on large datasets and optimized for accurate predictions.
- Database: SQLite3 is used to store data about users, poses, and feedback. It also stores the information re- quired for personalized recommendations.

B. User Authentication and Security

The system includes robust user authentication features to ensure that only authorized users can access certain sections of the platform. The user authentication process is handled using the Django authentication system, which provides secure login and role-based access control.

- Role-Based Authentication: There are two user roles: Admin and Regular User. Admins have access to detailed analytics, user management, and content control, while regular users can only access their profile, yoga poses, and feedback.
- OTP Verification: Email-based OTP (One-Time Pass- word) verification is implemented to secure the login process. Users must enter the OTP sent to their registered email to gain access.

C. Yoga Pose Prediction

The core functionality of the system is to predict yoga poses using images uploaded by users or captured in real-time via webcam. The implementation consists of the following steps:

- Image Upload: Users can upload images of themselves performing yoga poses. The system processes the image and passes it through the deep learning models.
- Real-Time Prediction: For real-time prediction, the system uses webcam input, processes each frame, and predicts the pose using the trained model.
- Model Deployment: The trained models (MobileNet, VGG16, DenseNet) are deployed on the backend server using TensorFlow/Keras to make predictions on the in- coming images or video frames.
- Feedback Generation: Based on the model's output, feedback is generated for the user. This feedback includes the predicted pose, a confidence score, and suggestions for improvement.

D. Deep Learning Model Training

The deep learning models used for pose prediction are trained on large datasets of yoga poses. The training process involves several stages:

- Data Collection: A large set of images, each labeled with the corresponding yoga pose, is collected. This dataset is used to train and validate the models.
- Data Augmentation: Data augmentation techniques, such as rotation, flipping, and scaling, are applied to prevent overfitting and increase the robustness of the models.
- Model Training: The models are trained using a super- vised learning approach. The MobileNet model, known for its efficiency on mobile devices,

is trained first, followed by VGG16 and DenseNet models.

- Optimization: Each model is optimized using gradient descent and backpropagation, minimizing the loss function (categorical cross-entropy) to achieve high accuracy.

E. Pose Search and Recommendation System

The yoga pose search feature allows users to search for specific poses based on their name or description. For each pose, the system provides detailed information such as:

- Step-by-Step Instructions: How to perform the pose correctly.
- Benefits: Physical and mental benefits of the pose.
- Related Poses: Similar poses that might be useful for the user.
- YouTube Tutorials: Links to video tutorials for further learning.

Additionally, the system integrates a third-party API to recommend specific yoga poses based on the user's health goals (e.g., flexibility, strength, relaxation).

F. Sentiment Analysis on User Feedback

The system analyzes user feedback using natural language processing (NLP) techniques. Feedback left by users, such as comments on yoga poses or blog posts, is processed for sentiment analysis to understand the users' feelings and improve the content and recommendations.

G. Admin Dashboard and Analytics

The admin panel provides an interface for platform administrators to monitor key metrics. The dashboard includes the following features:

- User Activity Monitoring: Real-time tracking of user activity on the platform.
- Pose Prediction Analytics: A breakdown of the accuracy and performance of the pose prediction system.
- Sentiment Analysis Trends: Insights into user sentiment, based on feedback, allowing administrators to improve the user experience.

VI. CONCLUSION

The implementation of the Yoga Pose Detection System involves the integration of various technologies, from machine learning models for pose prediction to a user-friendly web interface. The system effectively provides real-time pose feed- back, personalized yoga pose recommendations, and ensures user security through authentication and OTP verification. The admin dashboard further aids platform management, making this system a comprehensive tool for yoga enthusiasts.

VII. RESULTS AND DISCUSSION

The Yoga Pose Detection System, developed using deep learning techniques, has shown impressive results in predicting and analyzing yoga poses with high accuracy. This section discusses the outcomes, evaluates the effectiveness of the system, and identifies areas for potential improvement.

A. Performance Evaluation of Deep Learning Models

The system uses three distinct deep learning models: Mo- bileNet, VGG16, and DenseNet, to predict yoga poses. Each model was trained on a dataset of yoga poses and tested for accuracy. The results from these models were compared to evaluate their performance.

- MobileNet: MobileNet is a lightweight deep learning model that performs well on mobile and edge devices. This model achieved an accuracy of 95.52%, making it suitable for real-time applications. It is highly efficient and performs reasonably well even with less computational power.
- VGG16: The VGG16 model is known for its simplicity and high accuracy, with an accuracy rate of 92.88%. While VGG16 provides reliable predictions, its higher computational cost makes it less suitable for real-time processing on mobile devices.
- DenseNet: DenseNet achieved the highest accuracy of 95.87%, outperforming both MobileNet and VGG16. DenseNet's use of dense connections between layers allows the model to retain more information and thus predict yoga poses more accurately. However, it requires more computational resources than MobileNet.

Overall, DenseNet is the most accurate model, but Mo- bileNet offers a good balance of speed and efficiency, making it ideal for real-time applications.

B. Real-Time Pose Detection

The system allows users to perform yoga poses in front of a webcam, where the pose is detected in real-time. The real- time pose detection was tested on several poses, and the system performed remarkably well, with a prediction accuracy of over 95%. The feedback provided to users after each prediction was clear and informative, allowing them to correct their poses promptly.

C. Yoga Pose Prediction via Image Upload

For users who prefer to upload images of their yoga poses, the system also performs pose prediction based on the uploaded image. This feature achieved an impressive accuracy rate of around 94%, with the system accurately identifying the yoga pose in most cases. However, the system occasionally faced challenges with unclear or low-resolution images, which led to slightly reduced accuracy in those cases.

D. User Experience and Interface

The user experience was evaluated through feedback from initial users who interacted with the system. The dynamic and responsive web interface, built using React, ensured smooth navigation and easy access to key features. Users appreciated the simplicity of uploading images or using the webcam for pose detection. The real-time feedback on pose correctness was also well-received.

- Feedback on Pose Accuracy: Users found the feedback to be useful in improving their poses. The system pro- vides detailed suggestions on how to improve alignment, making it a valuable tool for yoga practitioners.
- Pose Search Feature: The search functionality for specific yoga poses was also well-received. Users were able to find detailed information about each pose, including step-by-step instructions and related poses.

E. Sentiment Analysis on User Feedback

The sentiment analysis feature provided valuable insights into user feedback. By analyzing user reviews and comments, the system can determine the overall sentiment and identify any issues that need to be addressed. Positive feedback high- lighted the accuracy of the pose detection system, while some users requested additional pose variations and video tutorials to further improve their yoga practice.

F. Admin Dashboard and Analytics

The admin panel was evaluated for its ability to provide real- time analytics. Interactive graphs and charts allowed administrators to monitor user activity, pose predictions, and sentiment trends. This feature was particularly useful for understanding user engagement and identifying areas for improvement.

- User Activity Monitoring: The system provided administrators with insights into the number of active users, the most popular yoga poses, and feedback trends. This data was helpful for making data-driven decisions to enhance the user experience.
- Pose Prediction Analytics: The admin dashboard also displayed statistics on the accuracy of pose predictions, helping administrators track model performance and make improvements.

G. Challenges and Limitations

While the system showed strong performance, there were some challenges and limitations observed during testing:

- Lighting Conditions: Poor lighting conditions negatively impacted the system's ability to accurately detect poses, especially in real-time webcam mode. Users were advised to ensure proper lighting for optimal results.
- Pose Variation: Some complex poses with significant variations were difficult to detect accurately, leading to lower prediction confidence.
- Image Resolution: Images with low resolution or poor quality sometimes resulted in incorrect predictions. The system works best with clear, high-resolution images.

These challenges highlight areas where the system could be further improved, such as by integrating additional preprocessing techniques for image enhancement or expanding the dataset to include more variations of complex poses.

H. Future Work

Future improvements to the system could include:

- Improved Pose Detection Models: Researching and implementing more advanced pose detection models such as OpenPose or PoseNet could further enhance the system's accuracy and robustness.
- Real-Time Feedback: The system could be enhanced by providing more real-time feedback, such as suggesting pose corrections during the practice itself.
- Mobile App Integration: A mobile version of the system could be developed, allowing users to perform yoga poses using their smartphones, thus increasing accessibility.

VIII. CONCLUSION

The Yoga Pose Detection System has successfully implemented an innovative solution for detecting and analyzing yoga poses using deep learning models. The system's performance has been impressive, with accurate pose detection, real-time feedback, and an intuitive user interface. While there are areas for improvement, such as handling different lighting conditions and enhancing pose detection for complex variations, the overall user experience has been positive. The system has the potential to evolve into a powerful tool for yoga practitioners, providing personalized recommendations and feedback to enhance their practice.

IX. Conclusion and Future Work

A. Conclusion

A web-based Yoga Pose Detection System enables users to leverage deep learning models for yoga pose detection analytics. Using MobileNet combined with VGG16 together with DenseNet enables the system to offer users a simple yet effective way to enhance their yoga practice. The core system functions enable users to attain accurate pose predictions from image uploads and webcam input thereby improving their learning experience.

The platform provides users with an extensive and interactive yoga learning experience through its security authentication features and sentiment analysis capabilities and individualized pose suggestions. The admin dashboard gives management access to essential metrics which generates useful data for platform effectiveness enhancement.

Through their performance evaluation it was found that yoga pose recognition using MobileNet and VGG16 operated at good accuracy levels but DenseNet excelled in acknowledgment rates. The implemented system operated with high success despite conditions affecting dim lighting and different complexity levels of poses.

The Yoga Pose Detection System can transform online yoga learning by providing instant video feedback that helps students at all proficiency levels become better practitioners. The system combines machine learning capabilities with intuitive features which makes it an outstanding solution for anyone who needs to advance their yoga skills.

B. Future Work

Multiple improvements are possible for the Yoga Pose Detection System's future development as an advanced platform with key functional features.

The system will benefit from utilizing exceptional pose detection models like OpenPose or PoseNet to handle the diverse range of yoga poses effectively especially for challenging positions. The system should evolve to supply comprehensive real-time feedback which provides both corrective suggestions and demonstrations of optimal alignment adjustments for improved practice. A mobile application development will expand system reach by offering users anytime access from any location. This development would make the platform more easily available and accessible. Voice-based feedback should be incorporated into the application through a hands-free voice guidance system that supports users during their yoga pose exercises. Multi-Pose Detection enables better system interactivity by detecting several poses within one session which allows users to perform advanced yoga routines for better assessment. The system can provide customized yoga exercises to users by analyzing their fitness data with advanced user profiling capabilities. The program should provide users with both everyday protocols and pose modifications that consider their health conditions.

Continuous development of targeted functionalities enables the Yoga Pose Detection System to become more robust for those using it as they gain access to a better yoga training environment. Ongoing system progress will enable it to remain as one of the leading systems that unite technology with fitness applications.

References

[1] A. G. Howard et al., MobileNets: Efficient Convolutional Neural Net- works for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*, 2017.

[2] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[5] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, Convolutional Pose Machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[6] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, Realtime Multi- Person 2D Pose Estimation Using Part Affinity Fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[9] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] J. Devlin, M. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] G. Linden, B. Smith, and J. York, Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*, 7(1):76- 80, 2003.