This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3414654

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier

YogiCombineDeep: Enhanced Yogic Posture Classification using Combined Deep Fusion of VGG16 and VGG19 Features.

R. ARUN KUMAR[1], and S. SIBI CHAKKARAVARTHY[1] (Member, IEEE)
[1]Center of Excellance, Artificial Intelligence Robotics (AIR) & School of Computer Science and Engineering, Vellore Institute of Technology – Andhra Pradesh (VIT-AP), Amaravathi, India

Corresponding authors: R. Arun Kumar (arunkumar.20phd7002@vitap.ac.in) and S. Sibi Chakkaravarthy (chakkaravarthy.sibi@vitap.ac.in)

ABSTRACT Yoga garnered significant attention during this pandemic based on its physical benefits in recent years. Regular yoga practice can enhance both physical and mental health. However, some body parts are occluded due to the significant variations in specific asanas with complicated posture formations and their backgrounds, making yogic posture detection more complex. This study establishes the classification of yoga postures in the yoga-16 dataset utilizing the combination of deep learning and machine learning approaches. A pre-trained CNN architecture VGG16 and VGG19 collects the deep features from the images of yoga postures. Then, the collected features are combined and entered into classifiers to train and assess the outcome of yoga posture classification. To classify the yogic postures from the collected yoga-16 dataset, the proposed model includes logistic regression, support vector machines, random forest, and extra tree classifiers. The proposed approach examined the yoga-16 dataset containing 16 classes and 6561 images. The proposed combined deep-fused approach utilizing Linear SVM yields better results than all the existing yogic posture classification models with outstanding scores of 99.94% precision, 99.94% recall, 100% f1-score, and 99.92% accuracy, respectively. The results show that the proposed approach is effective at attaining excellent performance in yogic posture classification. Performance comparisons with the most recent models have also been listed.

INDEX TERMS Yoga pose classification, VGG16, VGG19, Machine Learning, Deep Learning models, Combined features.

I. INTRODUCTION

Yoga has shown potential as a therapeutic exercise. However, the links underlying yogic practice and general wellness have received little attention. Yoga may be an effective intervention for improving health behaviours or health issues connected to lifestyle. How frequently someone practises at home matters more than how long they have been practising or how many lessons they attend. It matters what one practises, whether it be different physical positions, breathing exercises, or meditation, as each yoga practice style may have varied health advantages. The specific posture of yoga may benefit certain people, depending on the parts of their bodies they wish to enhance [1].

In recent years, the convergence of computer vision and yoga practice has produced promising opportunities for improving the knowledge and implementation of this ancient discipline. Yoga, well-known for its physical and mental health advantages, consists of various poses requiring efficacy and alignment. Yogic postures have the potential to be accurately classified and analyzed to improve safety, offer practitioners helpful feedback, and progress the area of human pose analysis as a whole. While natural to humans, yoga posture classification presents distinct hurdles for automated systems. Conventional pose analysis systems frequently use handcrafted features and traditional machine learning approaches.

Yoga is a psychophysiological practice which improves one's body, mind, emotions, and spiritual connection. Yoga improves both physical and mental wellness. As a result, this is often used as an additional therapy to treat Covid-19, an illness that occurs soon after Covid-19 [2].

Specific health ailments and medical conditions have been proven to benefit from yoga. Regular yogic practice promotes a healthier lifestyle, such as vegetarianism, nonsmoking, less alcohol use, more exercise, and less stress, with economic savings to the community [3].

Every physical activity has the potential to go wrong and result in several issues that will hurt an individual's health rather than improve it. When performing an Asana, for instance, placing a limb in the incorrect position or angle could result in sprains or muscle strains. Using an instructor when performing challenging exercises for the first time is advised for those who need a more prior understanding of the subject. Because instructors and classes would be pricey, this would be a problem. The demand for a system to evaluate individuals remotely is rising quickly due to emerging conditions like the threat of a global pandemic [4].

Traditional image feature collection and image analysis approaches were directed by hand-crafted methods, often profiting from conventional feature extraction methods in computer vision. These approaches were successful to some extent across various image classification models, such as ordinary photography, clinical imaging, and satellite imaging. CNNs and deep learning, however, have seen a resurgence in recent years.

Our study comes in this environment as a response to the constraints of established approaches. We acknowledge the need for a more comprehensive and flexible approach to yoga posture classification. Convolutional Neural Networks (CNNs) [5] serve as the foundation of our technique, inspired by the revolutionary potential of deep learning. CNNs have exhibited exceptional skills in picture classification tasks by automatically learning relevant characteristics from raw data, which makes them especially suitable for applications like yogic posture classification. We want to solve the problems that have prohibited automatic yogic posture detection in the past by utilizing the inherent power of deep learning.

Numerous studies on yoga posture recognition have been conducted over the last decade due to the advancements in vision and sensor technologies [6]. The current study investigates the extraction of features by utilizing two CNN architectures, VGG16 and VGG19 [7], as feature extractors and combining those features to acquire significant aspects of yogic posture images. The suggested method also uses a transfer learning approach to avoid overfitting and shorten training time. In several existing investigations, pre-trained CNN structures were favoured; this is discussed in the next section. Yoga-16 dataset was utilized to assess the suggested methodology. Transfer learning is one of the methods we use to extract features from the pre-trained CNN architectures of VGG16 and VGG19 without the requirement for comprehensive CNN training. These pre-trained models work well as feature extractors when removing top classification layers. By utilizing the advantages of both models, this method helps to reduce the risk of overfitting. The overall categorization results will be enhanced by integrating these features. This method also decreases the complexity of CNN training while improving the model's generalization capability. The proposed method outperformed numerous recent approaches and attained state-of-the-art performance.

Our study integrates multiple methodologies to achieve effective yoga pose classification using deep learning and machine learning techniques. Specifically, we leverage pre-trained CNN architectures, VGG16 and VGG19, to extract deep features from yoga pose images. These architectures have demonstrated strong performance in image recognition tasks due to their ability to capture complex visual patterns. The collected deep features from VGG16 and VGG19 are combined using an element-wise addition operation. This fusion approach allows us to integrate the complementary information these networks capture, resulting in more discriminative feature representations of yoga poses. The combined features are inputs to various classifiers, including logistic regression, support vector machine, random forest, and extra tree classifiers.

Our method is characterized by the following key steps:

- Feature Extraction: We utilize pre-trained VGG16 and VGG19 models to extract deep features from yoga pose images.
- Feature Fusion: The extracted features are integrated using element-wise addition to enhance their representational power.
- Classifier Training: We employ multiple classifiers to learn from the combined deep features and classify yoga poses accurately.
- Evaluation: The trained classifiers are evaluated on our Yoga-16 dataset and two publicly available datasets to assess classification performance.

Our approach differs from a simple combination of methods or patchwork of techniques by leveraging feature fusion to capitalize on the strengths of multiple CNN architectures. Integrating deep features from VGG16 and VGG19 through element-wise addition represents a novel contribution that enhances the discriminative power of our classification system.

A. CHALLENGES IN YOGA POSTURE CLASSIFICATION

Yoga posture classification is a sophisticated subset of image classification. Because of the subject matter and the intricacies involved in precise classification, it poses distinct difficulties and constraints. This section summarizes the main issues encountered in this field:

- Depending on the practitioner's body type, the photo's angle, and its environment, yoga poses can appear very different in images. Due to this diversity, classification models have trouble reliably identifying the same posture in various images.
- Many yoga poses are remarkably identical, with minor variations in limb position or body alignment. Distinguishing among similar postures demands a model to be extremely sensitive to small details, which makes it a difficult task in image classification.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3414654

- Compared to other image classification domains like ImageNet, yoga posture classification lacks more enormous, diversified, and well-annotated datasets. It restricts the quantity of training data and can make it more difficult to create reliable models.
- In real-world circumstances, yogic postures can have occlusions (where some body parts are not visible in the posture) or background noise, which can confuse classification techniques. Ensuring that models can address such complexities is a significant task.
- Given the nuances and variances of yoga poses, complicated models are required to classify them effectively. However, increasing model complexity can cause overfitting and decreased generalizability, posing a challenging task in finding the right balance.

This work uses a novel strategy that utilizes the combined properties of the VGG16 and VGG19 models to overcome these issues and aims to advance the classification of yoga postures. The method used in this study is intended to address some of these issues, notably those concerning model sensitivity and dataset limits.

Despite substantial advances in image classification, the specific domain of yogic posture classification remains unexplored, posing unique issues that were never fully explored in prior investigations. This research highlights a significant gap in the existing body of knowledge and presents a novel approach for bridging this gap.

1) Single-model architectures are primarily used in existing studies on the classification of yoga postures for feature extraction and classification. However, these methods often fail to capture the efficient features required to distinguish between similar yoga poses. This study aims to improve the model's ability to identify and distinguish minute differences in yoga poses by integrating the feature sets of two potent convolutional neural networks, VGG16 and VGG19.
2) Earlier investigations have focused on the models' end-to-end classification performance. There needs to be more emphasis on how robust and diversified feature extraction can considerably enhance classification accuracy, particularly in a field with minor distinctions across classes. Our technique emphasizes the relevance of efficient feature extraction for improving the accuracy of yoga posture classification.
3) A significant limitation in the existing research is the homogenous and small amounts of yogic posture image datasets. This often produces improperly generalized models that fail in real-world scenarios. Our work tackles this by building a more resilient classifier that can handle a variety of complicated and diverse images by combining distinctive features from several models.
4) Present solutions need help to balance efficiency and classification accuracy. Finding the ideal balance is the goal of the study's suggested methodology, which offers a computationally attainable and precise solution for real-world applications.

Our work provides insightful analysis and a solid platform for further study. We do not intend to revolutionize but rather to contribute another brick to the knowledge superstructure, emphasizing the practical applications of artificial intelligence in healthcare. Every small step forward, like ours, counts when applying technology to combine the vital features of images extracted from pre-trained models in posture classification. As a result, our emphasis on classifying yogic postures and using TL approaches offers insightful information and creates a baseline for future study in this field.

This research substantially contributes to the yogic posture classification by filling in those specific gaps. The approaches and conclusions reported in this study offer a fresh perspective and pave the way for additional research and breakthroughs in this field.

The current study has five main contributions, which are listed below.

- We propose a robust approach entitled YogiCombineDeep model by extracting and combining he deep features of VGG16 & VGG19 and a custom Yoga-16 dataset comprising sixteen distinct classes of visual appearance yogic posture based on its English and Sanskrit names.
- We present the modifications of VGG16 and VGG19 architectures to combine the rich features of our dataset for achieving better yogic posture classification. Our work highlights their capability to capture intricate details and delicate features inherent in yoga postures, showcasing the rich information these models can provide.
- In this model, the combined features classified using Linear SVM show high accuracy and outperform all other utilized classifiers.
- Our proposed model achieves exceptional precision and recall rates, ensuring reliable classification and significantly reducing the occurrence of false positives and negatives in identifying yoga postures on our dataset.
- Through our work, we contribute to establishing a crucial standard for automated yoga posture analysis. It sets the stage for future research at the intersection of computer vision and yoga practice, opening avenues for further exploration and innovation.

The sole objective of this study is to investigate the potential improvements of combining features acquired from both models and assess their impact on classification performance.

The remaining portion of this work is arranged into the following sections: Section 2 introduces the background of this work. Section 3 provides a detailed description of the recommended approach. The experimental findings of the suggested approach appear in Section 4. Section 5 concludes this study.

II. BACKGROUND

Classification of yogic postures is an emerging area of deep learning. Yoga posture classification challenges have been identified, and researchers have worked hard to address them. This section addresses existing studies of yoga posture classification and detecting the human body posture challenges [8].

Authors of [9] were able to construct an accurate real-time DL approach utilizing OpenPose for extracting the key points from the dataset of yogic posture video clips consisting of six classes of posture. This approach achieves 99.04% accuracy in framewise input and 99.38% in 45 frames.

Another deep learning method was used in [10] to train the model to recognize yogic positions. After the features extracted with the Keras multi-purpose pose estimate, a classification was employed to classify the six classes of yogic posture. Multilayer Perceptron is used to implement this procedure, with an accuracy of 99.58%.

The YOGI dataset with roughly 5500 images of ten distinct yoga poses was utilized in [11]. Joints and angles from the input yogic postures are acquired utilizing the tf-pose estimation algorithm. Extracted features are fed to different ML models, including LR, RF, SVM, DT, NB, and KNN. RF classifier outperforms all others with an accuracy of 99.04%.

This study uses transfer learning to identify 136 critical points distributed over the body from human posture estimation models to train RF classifiers to estimate yoga asanas. An extensive library of in-house yoga video datasets featuring 51 yogis' clips from four distinct angles is used to analyze the results. They are using standard 10-fold cross-validation to assess the proposed model. The RF classifier attained 99.70% mean accuracy [12].

The authors of [13] employed pre-trained CNN architectures like vgg16, vgg19, and InceptionV3. They used various supervised machine learning classification techniques, including SVM, Random Forest, Logistic Regression, and Neural Networks. The Yoga Pose Image categorization dataset was gathered via Kaggle, and it's a multi-class dataset of 5975 images divided into 107 classes of distinct yoga poses. They yield the best results in VGG16, VGG19, and InceptionV3 using LR with 98%, 97.8%, and 97.1% accuracy.

In this study, 12 yoga asanas were categorized using transfer learning algorithms that use neural networks. In addition, four TL models are used: VGG16, VGG19, MobileNet, Xception and it achieves 72.1%, 82.3%, 83.4%, 89.6%, and 91.2% accuracy. With an accuracy of 91.2%, Mobilenet outperformed all other TL models utilized in this work. This approach yields the most excellent F1 score of 98.1% in Bhairavasana, and the least is in Surya Namaskara, with an F1 score of 67.4% [14].

The authors developed a yogic posture guidance system using transfer learning in this work. They collected 14 classes of yogic postures. They suggested a transfer learning-based system for coaching yogic posture. It employed an interactive display to detect a yogic stance in real time using weights pre-trained using the CNN architecture model. Mobilenet with data augmentation approach yields the best results compared to other utilized TL models with a 98.43% accuracy. Additionally, when the user practises yoga in front of the system, their yoga posture tutoring system gives feedback on postural guidance [15].

The authors of [16] successfully extracted the skeleton information from the yoga posture images. They utilized MobileNet, VGG16, and VGG19 models to categorize the original and skeletonized yogic posture data sets. Using the skeletal image data set, the customized CNN model attains the best test set accuracy of 94.29%. MobileNet achieves the best test set accuracy using the original yogic posture images. However, it only reaches 80.71% accuracy, significantly less than the customized CNN.

In this study, the authors correct the user's yogic stance using the multi-person 2D pose prediction algorithm. OpenPose and deep learning approaches using CNN and TL techniques for pose recognition. The system is built for 18 various asanas that users can practise. Given that the user may face the camera in one of three possible directions when performing yoga—left side, right side, or front view—the algorithm can forecast an asana with a precision of over 87.6%. The system guides the user in the right path to correct them [17].

Table 1 summarizes the details of the related works on yogic posture classification using deep features.

Several yoga posture classification methodologies are summarized in this literature review section recent years.

III. PROPOSED METHODOLOGY

The VGG16 and VGG19 designs were chosen for feature extraction in our work for the following reasons.

- VGG16 and VGG19 have proved their ability to collect hierarchical information in images.
- Their extensive use and performance in numerous image classification applications make them dependable feature extraction options.
- The knowledge embodied in these networks is harnessed using large-scale picture datasets like ImageNet with the utilization of pre-trained VGG16 and VGG19 models.
- This transfer learning strategy allows our model to acquire valuable features learned from different visual data, which is useful when handling a relatively small yoga posture dataset.
- These designs are excellent choices for extracting discriminative features from yogic posture images, contributing to our method's overall effectiveness.
- The exact dimensions of features emerge from the base convolutional layers of VGG16 and VGG19 when their top layers aren't utilized for feature extraction.
- Unlike other models that require feature transformation during feature fusion, this uniformity simplifies the integration of VGG16 and VGG19 features.
- Compared to other CNN pre-trained models requiring reshaping or conversion for varying feature dimensions,

TABLE 1: Comparison of existing TL-based yogic posture classification models.

Author	Year	Features	Classification	#Asanas	#Images	Accuracy%
Jose J and Shailesh S [18]	2021	VGG16	DNN	10	700	85
Long et.al [15]	2022	VGG16, VGG19, InceptionV3, DenseNet201, MobileNet, and MobileNetV2.	DNN	14	-	94.90, 94.90, 91.76, 98.04, 98.43, 92.16.
Goel et.al [13]	2022	VGG16, VGG19, and InceptionV3	DT, SVM, RF, NN, LR, AB, and KNN	107	5975	Achieved highest accuracy 98, 97.8, 97.1 in LR.
Ananth G and Anuradha R [19]	2022	CNN, VGG16, VGG19, and MobileNet.	DNN	5	950	88.89, 99.24, 99.49, 35.
Palanimeera J and Ponmozhi K [20]	2022	AlexNet	SVM KNN SVM-KNN	7	-	89.91, 94.83, 98.15.
Kinger et al. [21]	2022	VGG19	-	6	2498	93
Rathikarani et al. [22]	2022	MobileNetV2, and DenseNet201.	SVM, RF	5	1551	71.99, 86.37, 69.86, 83.46.
Saluja J and Singh K K [14]	2023	Baseline CNN, VGG16, VGG19, Xception, and MobileNet.	-	12	709	72.1, 82.3, 83.4, 89.6, 91.2.
Proposed Work	2023	VGG16, VGG19, VGG16+VGG19	LR, SVM, RF, ET	16	6561	Achieved highest accuracy 99.84, 99.77, 99.92 in SVM.

this proposed approach is efficient and reduces extra operations and memory overhead.

As previously indicated, the suggested methodology combines the deep features collected from yoga posture images for accurate classification. The following Figure 1 depicts the entire structure of the suggested approach.

A. DATA COLLECTION AND PRE-PROCESSING

We collected sixteen types of yogic postures for classification using a Mozilla Firefox add-on Image Downloader from Google Image Search Engine. Images were obtained using the original Sanskrit and English names of yogic positions, and each image was cleaned and labelled individually. Every picture shows one or more people performing a similar yogic stance. Surprisingly, collected yoga positions were captured from varied camera perspectives. In addition, there are several pictures with backdrops that are entirely random. Since yoga postures are more concerned with the overall structure of the human body than with the material of clothing and skin, some photographs show the silhouette, sketches, layouts, and painting genre of yoga positions. These images were kept in the collection. To make this article easier to read and understand, we only use the names of yoga postures in English [23].

Data cleaning includes deleting duplicates, rectifying errors, and assuring data consistency. It contributes to the image dataset's quality and integrity. Labelling images entails associating relevant descriptions or names of classes with each image. This phase is essential for supervised learning applications, where computer models are trained to identify and categorize items or content inside the photos. It is critical to ensure labelling accuracy and uniformity. Quality assurance checks could be performed to ensure that labels appropriately depict the content of the images.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3414654

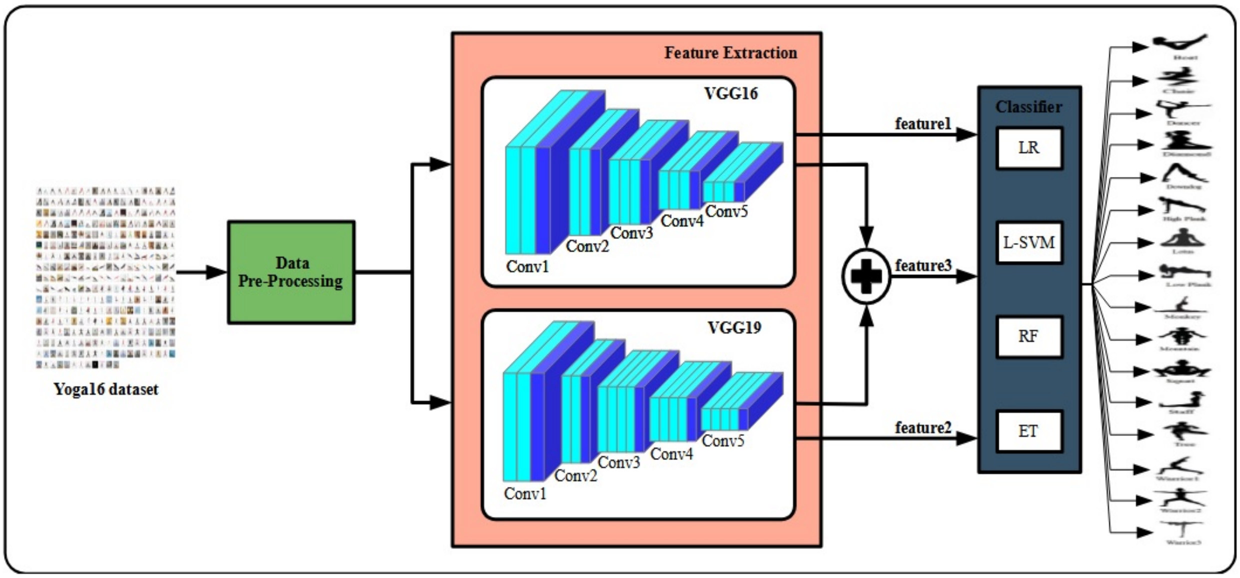

FIGURE 1: Proposed YogiCombineDeep Network

The yogic posture dataset (yoga-16) has been split 80:20 for training and testing the proposed model. Figure 2 shows the collected yogic posture dataset samples.

Boat Pose, also known as Navasana, is a yoga pose that strengthens the core and tones the abdominal muscles while enhancing balance and digestion. Regular practice can also help with stress reduction and mental clarity. Chair Pose, commonly known as Utkatasana, stimulates leg muscles and tones the lower body. It helps to improve circulation by strengthening the spine and promoting the heart. This pose can be used to increase stability and endurance. Dancer Pose, or the Natarajasana, helps to stretch the legs, shoulders, and chest while enhancing balance and attention. Additionally, it widens the heart and fosters elegance and grace. Diamond Pose, or the Vajrasana, supports digestive health and primarily benefits knee pain patients. It helps with grounding and centering, promoting calm and attention. Downward-Facing Dog, also known as Adho Mukha Svanasana, is a full-body stretch that eases tension in the back, shoulders, and hamstrings. Additionally, it gives the body energy, improves circulation, and relaxes the mind. High Plank pose, this posture improves posture while strengthening the arms, wrists, and core. It increases awareness of one's body and general upper body strength. Padmasana, also known as the Lotus Pose, is a traditional meditation pose that helps to increase hip and knee flexibility. It promotes calmness of mind, relaxation, and deep breathing. Low Plank Pose, or the Chaturanga Dandasana, strengthens and stabilizes the upper body. It also supports the abdominal muscles and prepares the body for complex arm balances. The wide-legged forward bend-like pose known as the monkey pose stretches the hamstrings and inner thighs. It improves flexibility and eases lower back stress. Mountain Pose, or the Tadasana, helps with alignment and posture. It improves concentration and awareness while calming the body and psyche. Squat Pose, or Malasana, improves flexibility by stretching the hips and groins. Both digestive help and lower back pain relief are possible with it. The Staff Pose, or Dandasana, strengthens the spine and enhances posture. It develops stability and gets the body ready for sitting meditation. Tree Pose, also known as Vrikshasana, improves equilibrium and focus. It strengthens the legs and promotes a sense of roots and inner serenity. The Warrior Pose, known as Virabhadrasana I, strengthens the legs and expands the chest. It improves balance and resolve while instilling a sensation of power. Warrior 2 Pose, or Virabhadrasana II, stretches and strengthens the hips and legs. It increases inner strength and a sense of presence. Warrior 3 Pose, often known as the Virabhadrasana III, enhances balance and core strength. It improves attention and self-confidence while toning the physique.

We chose to take a comprehensive strategy to avoid any biases and inconsistencies that may have been induced by utilizing just one search engine. We gathered images from various online resources, including dependable yoga websites, instruction manuals, and publicly accessible image archives. This multifaceted method enables us to capture a broad range of yoga poses while minimizing the impact of any given source's biases. We understood that the resolution, clarity, and background clutter of images gathered from diverse sources could vary. As a result, we used substantial data preparation to organize the images uniformly. This preprocessing included techniques including scaling photographs to a constant resolution, removing backgrounds, and reducing noise. These steps are intended to improve the dataset's consistency and remove undesirable deviations.

We created strict image selection rules to ensure rele-

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3414654

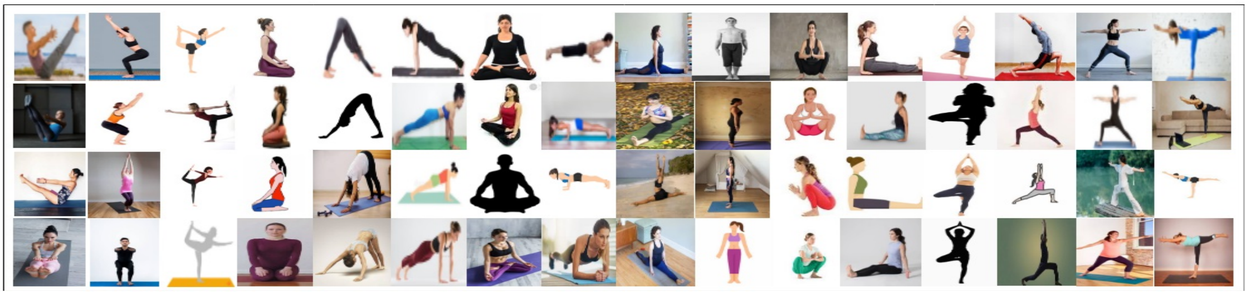

FIGURE 2: Sample yoga posture images from the sixteen categories, 'boat', 'chair', 'dancer', 'diamond', 'downdog', 'highplank', 'lotus', 'lowplank', 'monkey', 'mountain', 'squat', 'staff', 'tree', 'warrior1', 'warrior2', and 'warrior3' of the Yoga-16 dataset. Each column represents a specific class.

vancy and consistency. Only images that portrayed yogic positions and adhered to the prescribed postures for yoga were included in the collection. Each image in the dataset was meticulously annotated and labeled with the matching yoga stance. The following actions are performed after collecting the yogic posture images. At first, we carefully chose the images from various sources that depicted a wide range of yogic postures. The second step was data preprocessing, which involved standardizing image sizes and eliminating inefficient or artifact information. These two actions were implemented to ensure the collected data's quality and remove the unwanted postures.

In this research, every collected yogic posture image varied in size because they were all acquired on distinct websites. It may impact the classifier's performance, which will be trained using this data, and the accuracy of the classifier may be lowered. All yogic posture images are downsized to 224 * 224 pixels before being fed into the pre-trained convolutional networks during the pre-processing stage, and both the training and testing yogic posture images need to be divided by 255 to normalize the pixel values ranging from 0 to 1.

B. DEEP FEATURES COLLECTION

CNNs popularity has increased dramatically in all studies and industrial fields in recent years. The collection of sufficient size of datasets for training the CNN is one of the significant issues. Therefore, the lack of training data affects the overall performance due to overfitting. Furthermore, the backpropagation process is employed in CNNs training, and it is prolonged and needs several hyperparameter tunings too. To overcome these limitations, this study proposes a novel approach for collecting efficient deep features from pre-trained CNN architectures VGG16 and VGG19.

Combining rich features from VGG16 and VGG19 was achieved through a simple conjunction of their feature vectors. Notably, the structural commonalities between VGG16 and VGG19 allow both models to provide precisely the exact final feature dimensions for the given input images. For each input image, features were extracted from VGG16 and VGG19, and then these feature vectors were concatenated to provide a composite feature representation. The generated feature vector was next fed into our classifiers as input. The idea behind this fusion was to take advantage of both models' strengths in capturing diverse features of yoga postures, hence improving our model's discriminative capacity.

While designs such as ResNet, DenseNet, and EfficientNet are compelling and efficient in many circumstances, their architectural peculiarities may offer issues when combining their features directly. On the other hand, the structural consistency of VGG16 and VGG19 allowed us to include these models in our fusion technique efficiently.

Deep-learned features are generated from two phases in the feature extraction process. At first, we begin with each model's pre-trained weights and freeze each model's top layers to extract the features of training and testing images of the yogic posture dataset. After the model's training, learned features are extracted from the yogic postures. After that, deep-learned features extracted from VGG16, VGG19 and their combined features are fed to the classifiers.

1) VGG16 and VGG19 features

VGG16, an abbreviation for Visual Geometry Group 16, are a CNN structure developed by the University of Oxford's Visual Geometry Group [7]. It became a groundbreaking computer vision and deep learning model and attracted much attention. VGG16's primary goal was to improve the performance of image classification tasks to build deeper neural network models.

VGG19 is an expansion of the VGG16 architecture. Like VGG16, VGG19 was created at the University of Oxford's Visual Geometry Group and has significantly impacted computer vision and deep learning [7]. By expanding the model's depth and capability, the goal was to improve performance further.

VGG16 displayed excellent performance in various applications for computer vision, which comprises identifying objects, localization, & classification of images. VGG16 is a foundational model that has influenced the creation of subsequent CNN architectures, despite being quite significant in terms of the parameters and the computation compared to more recent architectures. It is also frequently used as a benchmark for assessing new models and methods in computer vision.

Our research primarily focused on extracting features from pre-trained models using these features and then combining those to train and test the chosen classifiers with rich features. Specific hyperparameters such as learning rates, weight decays, and epochs were not used because our strategy relied on pre-trained models to extract features without fine-tuning. Instead, we utilized the pre-trained models' default configurations. The convolution base of VGG16 and VGG19 and its utilized parameters are specified in tables 2 and 3.

TABLE 2: Utilized parameters of VGG16 and VGG19.

Parameters	VGG16	VGG19
Convolutional Layers	13	16
Pooling	Max	Max
Filter Sizes	3x3	3x3
Input Image Size	224x224	224x224
Output Dimension	7x7x512	7x7x512
Weight Initialization	Pre-trained on ImageNet	Pre-trained on ImageNet
Activation Function	ReLU	ReLU
Dropout Rate	N/A	N/A
Fully Connected Layers	3 (Removed)	3 (Removed)

TABLE 3: Feature Extraction base of VGG16 and VGG19.

Layer Name	VGG16	VGG19
Input	(224, 224, 3)	(224, 224, 3)
Conv2D (64 filters, 3x3)	(224, 224, 64)	(224, 224, 64)
Conv2D (64 filters, 3x3)	(224, 224, 64)	(224, 224, 64)
MaxPooling (2x2)	(112, 112, 64)	(112, 112, 64)
Conv2D (128 filters, 3x3)	(112, 112, 128)	(112, 112, 128)
Conv2D (128 filters, 3x3)	(112, 112, 128)	(112, 112, 128)
MaxPooling (2x2)	(56, 56, 128)	(56, 56, 128)
Conv2D (256 filters, 3x3)	(56, 56, 256)	(56, 56, 256)
Conv2D (256 filters, 3x3)	(56, 56, 256)	(56, 56, 256)
Conv2D (256 filters, 3x3)	(56, 56, 256)	(56, 56, 256)
Conv2D (256 filters, 3x3)	NA	(56, 56, 256)
MaxPooling (2x2)	(28, 28, 256)	(28, 28, 256)
Conv2D (512 filters, 3x3)	(28, 28, 512)	(28, 28, 512)
Conv2D (512 filters, 3x3)	(28, 28, 512)	(28, 28, 512)
Conv2D (512 filters, 3x3)	(28, 28, 512)	(28, 28, 512)
Conv2D (512 filters, 3x3)	NA	(28, 28, 512)
MaxPooling (2x2)	(14, 14, 512)	(14, 14, 512)
Conv2D (512 filters, 3x3)	(14, 14, 512)	(14, 14, 512)
Conv2D (512 filters, 3x3)	(14, 14, 512)	(14, 14, 512)
Conv2D (512 filters, 3x3)	(14, 14, 512)	(14, 14, 512)
Conv2D (512 filters, 3x3)	NA	(14, 14, 512)
MaxPooling (2x2)	(7, 7, 512)	(7, 7, 512)

The final convolutional layer, which comes right before the fully linked layers, is where the 512 features for the VGG16 architecture are extracted. The VGG19 architecture, deeper than VGG16, collects 512 features from the final convolutional layer, as does VGG16. These features indicate the network's learning of spatial and abstract features from the given yogic posture images. These features record the many textures, visual patterns, and other properties of the yogic posture images that the models have learned to represent at a high level.

VGG16 and VGG19 can capture a wide range of visual elements at various levels of abstraction, including edges, textures, and object sections. Each model's distinctive features arise from divergences in the depth and width of the architectural design. Because of its slightly shallower design, VGG16 can capture more fundamental elements like corners, edges, and simple textures. It has the potential to recognize basic structures and patterns in images. In general, VGG19 improves upon VGG16's success by providing an even more powerful and expressive DL framework for various computer vision applications. VGG19 can capture more abstract and complex features due to its improved depth. It has the potential to recognize higher-level structures, object portions, and intricate textures.

Our main goal of this research is to use the capabilities of Convolutional Neural Networks (CNNs) that have already been trained to extract detailed information from images of yoga postures. We use two well-known CNN designs, VGG16 and VGG19, which are famous for their efficiency in image identification applications. From enormous datasets like ImageNet, these pre-trained models have already acquired rich hierarchical representations of the feature data. Utilizing the convolutional layers of the VGG16 and VGG19, we want to use this information by extracting deep features. These features capture the most important visual information from the yoga pose images, which can then be used to classify the images using a variety of machine-learning classifiers. With this approach, we may use these pre-trained CNNs' strength and generalization skills while modifying them for yoga posture categorization.

2) Combined features

In this section, we added the elements of each feature map of yogic posture to produce a new feature map of the same size. Furthermore, combining features from both models could capitalize on the strengths of both designs, perhaps resulting in enhanced classification performance by exploiting a broader set of features. Each feature map comprises a grid of values representing distinct spatial places in the yoga posture image.

Let us represent the final convolutional layer output feature maps of VGG16 and VGG19 as follows: $F_i[i,j]$ describes both features' shapes. These feature maps are the same size for all yogic postures in the collection.

In this case, i denotes the total number of training and testing images utilized, and j represents the dimensions of each image feature.

The expression for calculating the dimension of the feature map is depicted in Equation 1.

$$j = N \times H \times W \tag{1}$$

Here, N is the number of channels and H and W are the height and width of the feature maps. The $F_1[i,j]$ and $F_2[i,j]$ represent the output feature maps of VGG16's and VGG19's final convolutional layer. In this work, the number of VGG16 and VGG19 channels is 512, and the height and width of the feature maps are 7.

Mathematically, for each image i and feature index j, the combined feature map $F_{Combined}[i,j]$ of training and

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3414654

testing images are calculated as the sum of the corresponding elements from $F_1[i,j]$ and $F_2[i,j]$ is depicted in Equation 2.

$$F_{Combined}[i,j] = F_1[i,j] + F_2[i,j] \tag{2}$$

This approach ensures that the combined feature map's dimensions remain the same as those of its component feature maps. Each element of the feature maps is added separately, producing a new feature map with the same height, breadth, and number of channels as the original.

This strategy takes advantage of each model's strengths and diversity to increase overall accuracy and robustness in the classification process. VGG16 and VGG19 are two separate CNN designs, each with its own set of learned features and representations. We effectively exploit a more varied and extensive range of features by integrating the features retrieved from these two models. Because of this variety, our YogiCombineDeep model could capture a broader range of patterns, features, and nuances within yoga posture images.

Combining features makes a classifier more reliable and accurate since it decreases the chance of biases brought on by the representations of a single model. Furthermore, feature fusion reduces the risk of overfitting depending entirely on a single model.

This approach takes advantage of the combined features of VGG16 and VGG19, leading to better classification results than utilizing either network alone. The fusion of features improves accuracy, resilience, and capability to handle diverse and complex yogic posture images, improving classification performance.

C. CLASSIFICATION

This study uses LR, L-SVM, RF, and ET classifiers in the classification stage instead of SoftMax.

Logistic Regression (LR) is a supervised ML approach mainly utilized in classification issues for forecasting which instance belongs to that specific class. It's a type of statistics which investigates the relationship between the collection of independent variables and a collection of variables with binary values dependent on one another. It is an efficient method for making decisions. It is called regression because it takes the outcome of a linear regression model function as an input [24]. A sigmoid function is used in logistic regression to forecast the outcome. The sigmoid function yields a result that ranges from 0 to 1. In most cases, everyone chooses a threshold like 0.5. It is assumed to be 1 when this function produces a number larger than or equivalent to 0.5, which is considered 0 otherwise [25].

Support Vector Machine (SVM) is a supervised ML method whose core principle is discovering an effective hyperplane margin that properly separates data. SVM is built on employing kernel functions, which provide the best possible separation of data [26]. SVM is a strong supervised method that performs best with smaller datasets, however it can also handle more complicated ones. Everyone can employ a linear SVM if the data can be separated precisely linearly. If the data points can be divided into two groups using just one straight line, it is said that they are perfectly linearly separable [27]. One of the most widely used kernel functions is Linear.

Random Forest (RF) comprises many independent decision trees which collaborate to produce an ensemble. For each tree, the random forest generates a class prediction, and the forecast for the model is determined by the class that receives the highest number of votes. The random forest is based on the wisdom of the crowds, which is a simple but effective concept. Many reasonably uncorrelated trees working as a committee will surpass any constituent models' performance [22].

Extra Tree (ET) algorithm creates a collection of decision or regression trees without pruning under the conventional top-down approach. It randomly divides nodes by choosing cut-points and builds the trees using the complete learning sample, two key differences from earlier tree-based ensemble techniques [28].

The choice of classifiers is crucial to the success of our strategy. A more thorough explanation of our reasoning behind these decisions is as follows:

- We added logistic regression as a classifier because of its simplicity and interpretability. It acts as a foundational model and provides a simple linear decision boundary. With this decision, we can compare the performance of relatively straightforward models to more intricate ones and learn whether a linear separation works well for our feature space.
- SVMs were chosen for their ability to handle high-dimensional feature spaces. In the present scenario, the combined deep features of VGG16 and VGG19 produce a high-dimensional space. SVM is a desirable option for taking advantage of the extensive representations of features due to its capacity to identify ideal hyperplanes for classification.
- The random forest is a type of ensemble learning noticed for its resilience and capacity to deal with complicated data sets. We incorporated it to capture non-linear interactions in the feature space. By combining the results of numerous decision trees, it is possible to give a more flexible decision boundary, potentially boosting classification accuracy.
- An ensemble learning technique, extra tree classifiers are similar to random forests. They were chosen for their capacity for handling high-dimensional data and capturing complex patterns in a feature space. The 'additional' randomness introduced during tree building makes trees less prone to overfitting, which can be useful in difficult classification applications.

The selection of these classifiers was influenced by several factors, including their aptitude for handling high-dimensional feature spaces, their capacity to capture non-linear correlations and their interpretability. We wanted to examine the potential of different classifiers for our specific job of yoga posture categorization by using a mix of simple and more complicated models. Because of the variety of

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3414654

classifiers available, we were able to investigate the trade-offs between model complexity and performance, which allowed for a more thorough evaluation of the efficacy of our strategy. The following table 4 lists the utilized parameters of the chosen classifiers.

TABLE 4: Utilized parameters of the classifiers.

Classifier	Parameters
Logistic Regression	Penalty: L2 (Ridge) C (Regularization): 1.0 Solver: lbfgs
Support Vector Machine	Kernel Type: Linear C (Regularization): 1.0 Gamma (Kernel Coefficient): 'scale' (Auto)
Random Forest	Number of Estimators: 200 Maximum Depth of Trees: None Minimum Samples per Leaf: 1 Maximum Features: 'auto'
Extra Tree	Number of Estimators: 100 Maximum Depth of Trees: None Minimum Samples per Leaf: 1 Maximum Features: 'auto'

It is essential to empirically examine the performances of classifiers trained on extracted features from VGG16 and VGG19 and their combined features on yogic posture classification to forecast which yields the most effective outcomes.

Let X denotes the input feature matrix containing the combined deep features of training and testing images from VGG16 and VGG19. Then, let Y denote the target vector representing the ground truth labels for the yoga pose images. We use a standard machine learning algorithm to train a classifier using these features, such as logistic regression, support vector machine, random forest, or extra tree classifiers. The classifier unit is trained and validated based on the extracted and combined deep features. The process of training the classifier is mathematically represented in Equation 3:

$$\theta = TrainClassifier(F_{Combined_train}[i, j], Y) \quad (3)$$

The process of testing the classifier is mathematically represented in Equation 4:

$$\hat{Y} = Predict(F_{Combined_train}[i, j], Y) \quad (4)$$

The process of predicting a single test image is mathematically represented in Equation 5:

$$\hat{y} = argmax\left(\hat{Y}_{test}\right) \quad (5)$$

The integrated deep features yield better results in the yogic posture classification than the actual ones.

D. PERFORMANCE METRICS

The initial classifier performance measures are computed by comparing the classifier's predictions to the dataset's ground truth labels. These metrics are necessary to assess how well the classifier produces accurate or inaccurate predictions.

True-Positive (TP): A True-Positive is when the classifier accurately predicts a positive (or 'yes') class sample as positive. This occurs when the classifier accurately recognizes a yoga pose as belonging to a given class in the context of yoga pose classification.

True-Negative (TN): A True-Negative happens when the classifier correctly predicts a negative (or 'no') class sample to be negative. This would be when the classifier accurately recognizes that a sample does not belong to a given class when it does not.

False-Positive (FP): A False-Positive happens when the classifier wrongly predicts a negative class sample as positive. In other words, it incorrectly classifies a sample as belonging to a given class when it does not. It would be considered a misclassification regarding yoga posture classification, where the classifier places a position in the wrong class.

False-Negative (FN): False-Negative results from the classifier misclassifying a positive class sample as hostile. In this instance, despite the sample's actual membership in a specific class, the classifier misclassifies it. It occurs when the classifier fails to recognize a posture belonging to a particular class in the classification of yoga postures.

These measures are essential for evaluating the performance of the classification in terms of recall, precision, F1-score, and accuracy, amongst other metrics. They provide insights regarding the classifier's capacity to categorize data correctly and indicate areas where the classification process could be improved.

The proposed model's performance is examined through the ML classifiers LR, L-SVM, RF, and ET. At the start of the analysis, the initial metrics of the classifier's performance True-Positive, True-Negative, False-positive, and False-Negative values are estimated. Essential performance metrics, like Precision, Recall, F1-score, and Accuracy, are measured based on these initial values.

IV. RESULTS AND DISCUSSION

This section summarizes our experimental outcomes using the Yoga-16 dataset. The proposed work is performed in a workstation that has the following specifications: Intel Xeon(R) W-2125 CPU @ 4 GHz, 4008 MHz, four cores, eight logical processors, and 64GB of RAM with equipped Jupyter notebook and tensorflow keras libraries.

According to the limited yogic posture data currently available publicly, researchers have gathered their yogic posture data. As a result, we gathered our Yoga-16 dataset online. The yoga-16 dataset was systematically divided into training and testing sets using the split_folders library. This library facilitated a random split with a ratio of 80:20, allocating 80% of the yogic posture images to the training set and 20% to the testing set. Figure 3 illustrates the Yoga-16 dataset's number of images per class in training and testing. This strategy ensured a representative distribution of yoga postures in both sets, allowing for robust model training and validation.

This approach ensured a representative distribution of yoga poses in both sets, supporting robust model training and evaluation. The primary data source for this study is the yoga-16 dataset, obtained from various online platforms. This

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3414654

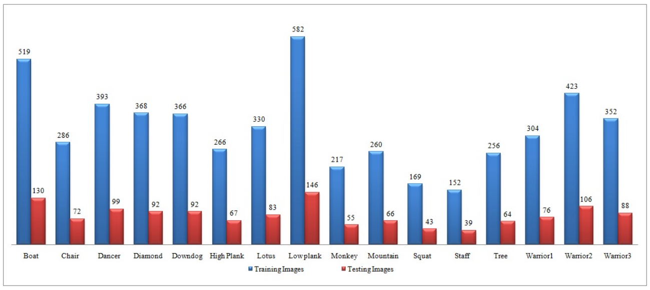

FIGURE 3: Yoga-16 dataset in detail.

dataset was curated to encompass a diverse range of yoga postures, contributing to the richness and variability of the training and testing sets. Preprocessing steps were conducted to enhance the quality of the input yogic posture images. Resizing the images to a standard dimension of VGG16 and VGG19 and normalizing pixel values are involved in this phase. These preprocessing steps contribute to a more consistent and reliable input for the classification model.

We employed a distinct approach to classifying the yoga postures and utilized transfer learning methodologies with a pre-trained architecture for extracting and combining the features. Afterwards, selected classifiers are used to classify and forecast our yoga postures.

The following figure 4 shows that yogic posture image data collection is compiled from diverse sources at the first stage. The dataset underwent preprocessing at the second to ensure uniformity and enhance model performance. All images were resized to a standardized resolution in this stage, and their pixel values were normalized to a specific range. In the third, image features are independently extracted from the VGG16 and VGG19 architectures for a given yoga posture image. Then, element-wise addition is utilized at the concatenation to enhance the combined feature vector. At last, the trained and tested yogic posture fused feature vectors are fed into the ML classifiers for training and testing.

This work uses the Yoga-16 dataset, contributing to our proposed model's comprehensive evaluation. The chosen classifiers are trained and tested using the combined deep features derived from the pre-trained VGG16 and VGG19 models. This integrated feature set is intended to improve the model's ability to distinguish subtle variations in yogic postures. We tested a variety of classifiers, including logistic regression, support vector machine, random forest, and extra tree classifiers, to evaluate their performance in yogic posture classification. Each classifier was trained and evaluated using the combined deep features. The chosen classifiers went through a thorough training process, with the integrated deep features serving as input for learning the dataset's underlying patterns. The trained models were then evaluated for generalization on a separate test set. During the validation phase, we used grid search to determine the ideal hyperparameters for each classifier. Grid search was used to investigate a variety of hyperparameter combinations systematically, and we used this strategy to identify the optimal hyperparameters for each classifier, ensuring that the models were optimized for maximum accuracy.

We conduct a thorough assessment process for our suggested approach and provide a fair comparison to existing research. In addition, we conducted 10-fold cross-validation studies to confirm the model's resilience and lower the chance of overfitting. In this approach, the dataset is divided into ten folds, each serving as a validation and test set, with the remaining folds being utilized for training. The outcomes are averaged throughout the ten iterations to produce a more complete and trustworthy measure of the model's performance. With good performance scores, cross-validation outcomes repeatedly showed our approach's resilience and efficacy. These extra validation processes strengthen the validity of our results by ensuring that our model operates consistently and can effectively generalize to new data.

When comparing VGG16 and VGG19 to other networks such as ResNet, DenseNet, and EfficientNet, one noticeable distinction is the structure of their final convolutional layers and the feature maps they generate. While these architectures generate deep feature representations, the dimensions of their output feature maps can differ. For instance, the ResNet, DenseNet, and EfficientNet architectures frequently use skip connections, dense connectivity, or efficient network block compositions to produce varying-size feature maps. Before executing any fusion operations, the output feature maps must be scaled to a consistent size to merge features from these architectures efficiently. This resizing procedure complicates the fusion method and increases computing overhead.

In contrast, VGG16 and VGG19 designs generate output feature maps of the same size for each input image. This consistency simplifies the fusion process by eliminating the need for scaling or additional preprocessing procedures before element-wise addition. The uniform size of feature maps from VGG16 and VGG19 makes integration more accessible and allows for seamless combining of deep features.

As a result, one of the key reasons for selecting VGG16 and VGG19 in this study is their capacity to generate output feature maps of uniform size, streamlining the fusion process and reducing computational burden compared to designs with varied output sizes. This choice consistently improves efficiency and efficacy in feature fusion of yogic posture classification.

Feature outcomes of our proposed scheme are depicted in Figure 5. For the selected test image, figures 5a, 5b, 5c, and 5d show the results of the first convolution layer of VGG16, VGG19, and the combined features.

Different layers of the CNN contain distinct activation regions, which results in diverse image features being retrieved by different levels of the network. When the amount of layers increases, the total number of detailed features reduces while the number of abstract advanced features increases [29].

Linear SVM is renowned for being reliable and straightforward, making it appropriate for high-dimensional feature

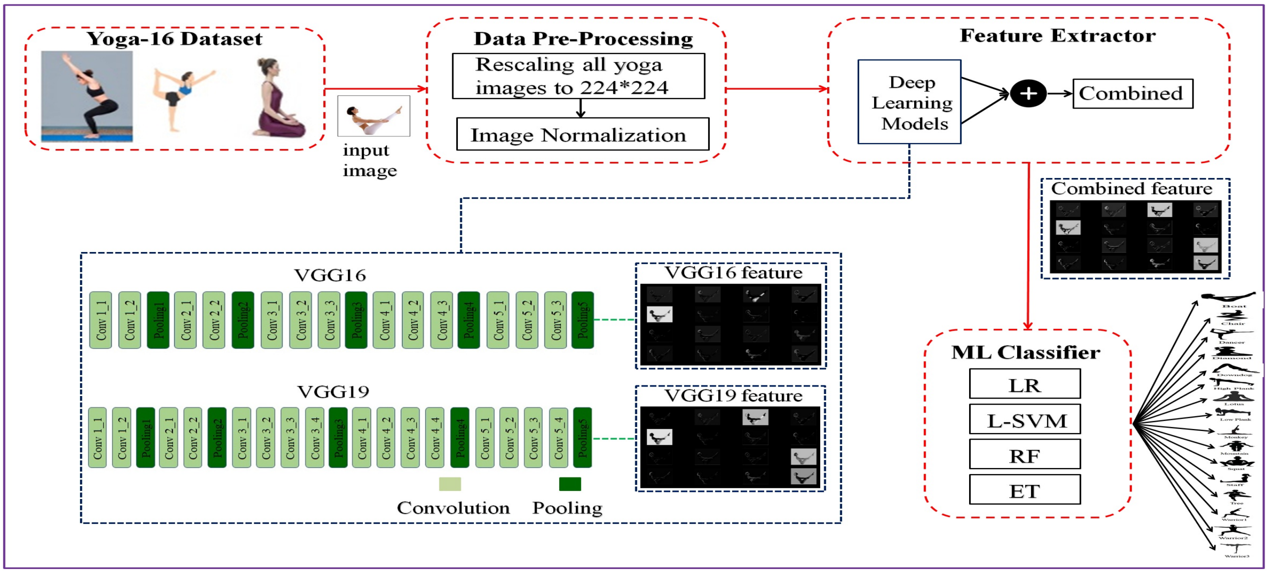

FIGURE 4: Schematic representation of YogCombineDeep approach.

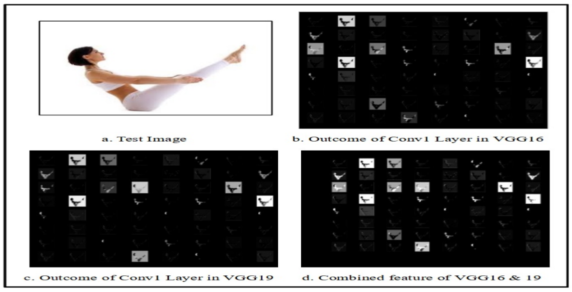

FIGURE 5: Feature visualization of yogic posture image.

spaces. Our investigation efficiently takes advantage of the detailed feature representations derived from the combined VGG16 and VGG19 features. The traits of our combined deep feature vectors are well suited to linear separation, providing discriminative power, which the Linear SVM classifier efficiently utilizes for accurate classification. The following tables illustrate Linear SVM's better performances than other utilized classifiers.

The following section presents a detailed table displaying the classification results of a rigorous study of numerous classifiers. This table is an essential point of reference for evaluating the effectiveness of each classifier in the particular area of yoga posture classification. Precision, recall, F1-score, and accuracy are among the metrics given, offering a comprehensive assessment of the classifiers' ability to identify and accurately classify yoga postures. The comparative analysis provided in this table allows users to identify the strengths and drawbacks of each classifier, assisting in selecting the best model for yogic posture categorization tasks. It also provides valuable insight into the possible real-world use of these classifiers in yoga posture detection scenarios.

The tables 5, 6, 7, and 8 illustrate the achieved precision, recall, f1-score, and accuracy of the proposed approach.

TABLE 5: Achieved accuracy utilizing the deep features of DL models.

feature#	LR%	L-SVM%	RF%	ET%
VGG16	99.77	99.84	99.24	99.54
VGG19	99.77	99.77	99.31	99.31
Combined	99.84	**99.92**	99.54	99.62

TABLE 6: Achieved precision utilizing the deep features of DL models.

feature#	LR%	L-SVM%	RF%	ET%
VGG16	99.81	99.88	99.50	99.69
VGG19	99.81	99.81	99.50	99.50
Combined	99.88	**99.94**	99.69	99.75

TABLE 7: Achieved recall utilizing the deep features of DL models.

feature#	LR%	L-SVM%	RF%	ET%
VGG16	99.75	99.81	99.00	99.31
VGG19	99.75	99.75	98.94	99.06
Combined	99.81	**99.94**	99.38	99.44

The table shows that the L-SVM classifier is the best-performing model among the assessed classifiers. L-SVM exhibits remarkable precision, recall, accuracy, and F1-score, displaying its strength in accurately recognizing and categorizing yoga postures. In our comparative analysis of classifiers for yoga pose classification using a fusion of features,

TABLE 8: Achieved f1-score utilizing the deep features of DL models.

feature#	LR%	L-SVM%	RF%	ET%
VGG16	99.75	99.88	99.19	99.44
VGG19	99.75	99.75	99.19	99.25
Combined	99.81	**100**	99.44	99.50

the Linear SVM stands up as the better option for numerous salient reasons:

Robustness in High-Dimensional Spaces: Linear SVM can effectively handle the complicated and rich feature sets associated with images of yoga postures because it is ideally suited for rich features. The robustness of its performance is primarily due to its ability to locate the best linear decision boundaries in high-dimensional spaces.

Efficient Separation of Classes: When classes are divided, like in the categorization of yoga postures, linear SVM performs well. It increases the margin between classes, lowering the possibility of misclassification.

Resilience to Overfitting: The linear SVM demonstrates resistance to overfitting, guaranteeing that it generalizes well to new, raw data. It is incredibly beneficial when working with different yoga postures and varied datasets.

Effective Handling of Noisy Data: Linear SVM's margin-based classification method makes it less vulnerable to the influence of noisy data points or outliers. This feature increases its dependability in real-world circumstances when data quality varies.

Figures 6 and 7 depicts the confusion matrix and classification report of the best outcome of combined features using L-SVM.

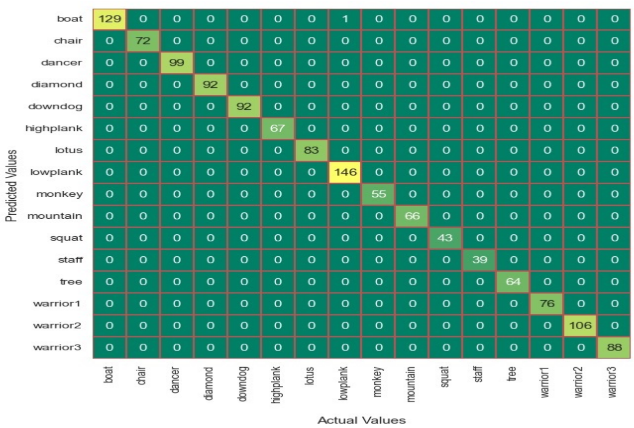

FIGURE 6: Confusion matrix of combined deep feature using L-SVM.

	precision	recall	f1-score	support
boat	1.00	0.99	1.00	130
chair	0.99	1.00	0.99	72
dancer	1.00	1.00	1.00	99
diamond	1.00	1.00	1.00	92
downdog	1.00	1.00	1.00	92
highplank	1.00	1.00	1.00	67
lotus	1.00	1.00	1.00	83
lowplank	0.99	1.00	1.00	146
monkey	1.00	1.00	1.00	55
mountain	1.00	0.98	0.99	66
squat	1.00	1.00	1.00	43
staff	1.00	1.00	1.00	39
tree	1.00	1.00	1.00	64
warrior1	1.00	1.00	1.00	76
warrior2	1.00	1.00	1.00	106
warrior3	1.00	1.00	1.00	88
accuracy			1.00	1318
macro avg	1.00	1.00	1.00	1318
weighted avg	1.00	1.00	1.00	1318

FIGURE 7: Classification report of combined deep feature using L-SVM.

The table 9 summarizes the detailed classification reports of the proposed approach.

We have conducted additional experiments using two publicly available yoga pose datasets from Kaggle [30], [31]. The first dataset contains 1081 training images and 470 testing images, while the second dataset consists of 790 training images and 198 testing images. We split the datasets in the ratio of 80:20 for training and testing. Our fusion approach, which integrates the deep features extracted from VGG16 and VGG19 through element-wise addition, was applied to these datasets. We used a linear Support Vector Machine (SVM) for classification, as it demonstrated outstanding performance compared to other classifiers such as logistic regression (LR), random forest (RF), and extra tree (ET) classifiers in our initial experiments.

The results of these experiments are promising, with our method achieving 97.38% accuracy on the first dataset and 99.86% accuracy on the second dataset. These results are detailed in the following tables 10, 11 which compare the performance of our approach on the Yoga-16 dataset and the two public datasets:

These additional results substantiate the generalizability of our proposed method. The high accuracy rates across different datasets demonstrate that the element-wise addition of deep features effectively enhances the quality of the feature representation, leading to improved classification performance. This enhancement is a significant contribution to the field of yoga posture classification, as it shows that our approach is robust and effective across various data sources.

Table 12 and Figure 8 compares the outcomes of our study with the results of some existing yogic posture classification methods which employ extracted deep features.

The proposed model is evaluated based on the combined features of VGG16 and VGG19. The table shows the classified yogic posture images for several classifiers. This table specifies the results obtained with the L-SVM classifier outperforming other classifiers. The suggested scheme attains the highest classification accuracy of 99.92%, utilizing combined features with L-SVM.

In addition to these four classifiers, we investigated several other classifiers in the machine learning literature. In preliminary studies, classifiers such as Decision Trees, k-nearest Neighbours (k-NN), Naive Bayes, and Neural Networks were evaluated. However, according to our thorough analysis, the

TABLE 9: Proposed models detailed classification report.

	VGG16			VGG19			Combined			
Name of Asana	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Support
boat	1.00	0.99	1.00	1.00	1.00	1.00	1.00	0.99	1.00	130
chair	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	72
dancer	1.00	1.00	1.00	0.99	1.00	0.99	1.00	1.00	1.00	99
diamond	1.00	1.00	1.00	0.99	1.00	0.99	1.00	1.00	1.00	92
downdog	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	92
highplank	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00	1.00	67
lotus	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	83
lowplank	0.99	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	146
monkey	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	55
mountain	1.00	0.98	0.99	1.00	0.98	0.99	1.00	1.00	1.00	66
squat	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	43
staff	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	39
tree	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	64
warrior1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	76
warrior2	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	106
warrior3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	88
macro avg	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1318
weighted avg	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1318

TABLE 10: Classification performance analysis of yogic postures in our dataset with two other public datasets using L-SVM

	Yoga-16				Public Dataset1 [30]				Public Dataset2 [31]			
Methods	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score	Accuracy
VGG16	99.88	99.81	99.88	99.84	97.60	97.40	97.40	96.61	99.40	99.40	99.60	99.49
VGG19	99.81	99.75	99.75	99.77	96.20	95.80	96.00	95.95	99.60	99.40	99.60	99.49
Combined	99.94	99.94	100	99.92	97.40	97.20	97.40	97.38	99.80	99.60	99.80	99.86

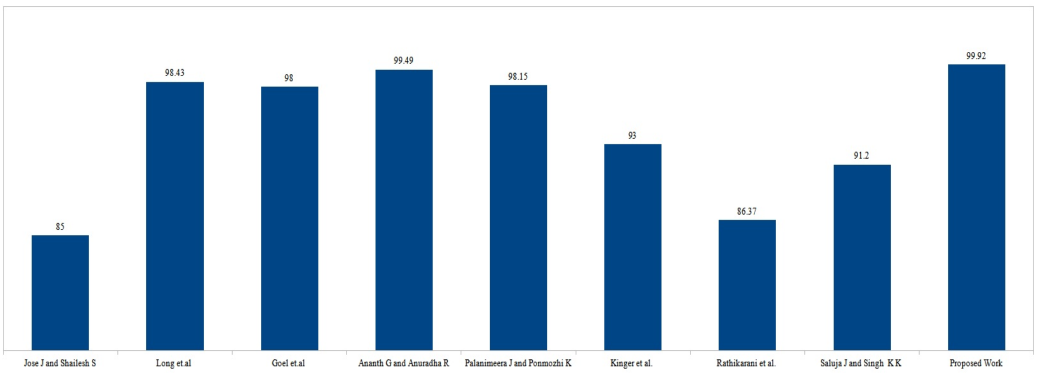

FIGURE 8: Comparison chart with existing approaches.

TABLE 11: Achieved accuracy using yogic posture datasets in detail using the fusion of features

Dataset	Total Images	Training Images	Testing Images	Accuracy%
Yoga-16	6561	5243	1318	99.92
Public Daaset1 [30]	1551	1081	470	97.38
Public Daaset2 [30]	988	790	198	99.86

TABLE 12: Comparative analysis of yogic posture classification using deep features.

Reference	Year	#Asanas	Accuracy%
Jose J and Shailesh S [18]	2021	10	85
Long et.al [15]	2022	14	98.43
Goel et.al [13]	2022	107	98
Ananth G and Anuradha R [19]	2022	5	99.49
Palanimeera J and Ponmozhi K [20]	2022	7	98.15
Kinger et al. [21]	2022	6	93
Rathikarani et al. [22]	2022	5	86.37
Saluja J and Singh K K [14]	2023	12	91.2
YogiCombineDeep	2023	16	99.92

best classifiers for identifying yogic postures were Logistic Regression, SVM, Random Forest, and Extra Tree Classifiers, which consistently beat other approaches regarding accuracy and robustness. As a result, we report findings for these four classifiers in this paper to show their superior performance in dealing with the nuances and intricacies of

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3414654

yogic posture classification.

The slight benefit can also become more noticeable when used with more extensive and complicated yogic posture datasets. It's crucial to note that even minor improvements can be beneficial compared with existing works, mainly when performing yogic posture classification tasks where improved accuracy is challenging. This combination of features from distinct models improves the classification performance in yogic postures by capturing a broader range of information gathered by both models.

V. CONCLUSION

In this work, we introduced a novel method for classifying yoga poses that combines machine and deep learning methods. Our approach involved exploiting the deep features derived from the VGG16 and VGG19 convolutional neural network designs combined via element-wise addition. This feature fusion enabled us to collect more detailed representations of yoga posture images, which improved classification performance. We proved the efficacy of our suggested method by thoroughly investigating the Yoga-16 dataset, which consists of 16 courses of yoga poses. We achieved higher classification accuracy than separate networks by training classifiers on the combined deep features, such as logistic regression, support vector machine, random forest, and extra tree classifiers. Specifically, we achieved 99.92% accuracy on the Yoga-16 dataset, 97.38% on the first public dataset, and 99.86% on the second public dataset. These results demonstrate our method's robustness and generalizability across multiple datasets. While our findings suggest promising advances in yoga posture classification, there is still a need for more investigation and improvement. Future research efforts could concentrate on improving the feature fusion method, investigating alternative deep learning architectures, and conducting more thorough experiments on varied datasets. Furthermore, improving the classification models' interpretability and explainability would help increase the general understanding and acceptability of automated yoga posture classification systems. Our study adds to the expanding knowledge about the intersection of computer vision and yoga posture classification. We aim to further progress automated yoga posture detection systems for fitness tracking, health monitoring, and customized wellness initiatives by bringing novel approaches and attaining significant advances in performance.

REFERENCES

[1] A. Ross, E. Friedmann, M. Bevans, and S. Thomas, "Frequency of yoga practice predicts health: results of a national survey of yoga practitioners," Evidence-Based Complementary and Alternative Medicine, pp. 1–10, 2012.

[2] D. C. Santos, S. Jaconiano, S. Macedo, F. Ribeiro, S. Ponte, P. Soares, and P. Boaventura, "Yoga for covid-19: An ancient practice for a new condition–a literature review," Complementary Therapies in Clinical Practice, p. 101717, 2022.

[3] S. Penman, M. Cohen, P. Stevens, and S. Jackson, "Yoga in australia: Results of a national survey," International journal of yoga, vol. 5, no. 2, pp. 92–101, 2012.

[4] S. Anwarul and M. Mohan, "Deep learning-based yoga pose recognition system using hyperparameter tuning," in 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO). IEEE, 2022, pp. 1–6.

[5] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," Procedia computer science, vol. 132, pp. 377–384, 2018.

[6] A. K. Rajendran and S. C. Sethuraman, "A survey on yogic posture recognition," IEEE Access, vol. 11, pp. 11 183–11 223, 2023.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[8] 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, pp. 1–12, 2022.

[9] 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.

[10] V. Anand Thoutam, A. Srivastava, T. Badal, V. Kumar 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.

[11] Y. Agrawal, Y. Shah, and A. Sharma, "Implementation of machine learning technique for identification of yoga poses," in 2020 IEEE 9th international conference on communication systems and network technologies (CSNT). IEEE, 2020, pp. 40–43.

[12] 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.

[13] S. Goel, S. Mohanty, and S. Markanday, "Classification of yoga pose using pre-trained cnn models and machine learning classifiers," in 2022 IEEE International Conference on Current Development in Engineering and Technology (CCET). IEEE, 2022, pp. 1–6.

[14] J. Saluja and K. K. Singh, "Empirical study of deep learning based yoga asana identification and its alternatives," in 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS). IEEE, 2023, pp. 40–46.

[15] 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.

[16] R. Yingdong, "Research on different convolutional neural networks in the classification scenes of yoga poses based on openpose extraction," in 2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA). IEEE, 2022, pp. 1532–1535.

[17] G. Pavitra and C. Anamika, "Deep learning-based yoga learning application," in Computer Vision and Robotics: Proceedings of CVR 2021. Springer, 2022, pp. 365–380.

[18] J. Jose and S. Shailesh, "Yoga asana identification: a deep learning approach," in IOP Conference Series: Materials Science and Engineering, vol. 1110, no. 1. IOP Publishing, 2021, p. 012002.

[19] G. Ananth and R. Anuradha, "Yoga posture classification using deep learning," in 2022 International Conference on Futuristic Technologies (INCOFT). IEEE, 2022, pp. 1–6.

[20] J. Palanimeera and K. Ponmozhi, "Transfer learning with deep representations is used to recognition yoga postures," in 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT). IEEE, 2022, pp. 1–7.

[21] S. Kinger, A. Desai, S. Patil, H. Sinalkar, and N. Deore, "Deep learning based yoga pose classification," in 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), vol. 1. IEEE, 2022, pp. 682–691.

[22] V. Rathikarani, S. Abarna, and K. Vijayakumar, "Classification of yoga pose using pretrained convolutional neural networks," Journal of Pharmaceutical Negative Results, pp. 3798–3805, 2022.

[23] 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.

[24] GeeksforGeeks, "Logistic regression in machine learning," (2023). [Online]. Available: https://www.geeksforgeeks.org/understanding-logistic-regression/

[25] R. N. Sucky, "Multiclass classification using logistic regression from scratch in python: Step by step guide," (2020). [Online]. Available: https://towardsdatascience.com/

multiclass-classification-algorithm-from-scratch-with-a-project-in-python/-step-by-step-guide-485a83c79992
[26] Y. Choubik and A. Mahmoudi, "Machine learning for real time poses classification using kinect skeleton data," in 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV). IEEE, 2016, pp. 307–311.
[27] A. Saini, "Support vector machine(svm): A complete guide for beginners," (2023). [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/
[28] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine learning, vol. 63, pp. 3–42, 2006.
[29] P. Tang, H. Wang, and S. Kwong, "G-ms2f: Googlenet based multi-stage feature fusion of deep cnn for scene recognition," Neurocomputing, vol. 225, pp. 188–197, 2017.
[30] N. Pandit, "Yoga poses dataset," (2021). [Online]. Available: https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset
[31] U. CHOWDHURY, "Yoga pose classification," (2022). [Online]. Available: https://www.kaggle.com/datasets/ujjwalchowdhury/yoga-pose-classification

ARUN KUMAR RAJENDRAN received his B.E. degree in Computer Science and Engineering, and M.E. degree in Computer and Communication from the Anna University in 2006 and 2008, respectively. He is currently a Ph.D candidate in the School of Computer Science and Engineering at Vellore Institute of Technology – Andhra Pradesh (VIT-AP) University. His research interests include Computer Vision, Machine Learning and Deep Learning. His Ph.D research focuses on developing a computer vision model for yogic posture correction and classification to assist users in better understanding the health benefits of yoga.

SIBI CHAKKARAVARTHY SETHURAMAN (M'18) received his Ph.D from Anna University in the year 2018, and he is currently working as an Associate Professor in the School of Computer Science and Engineering at Vellore Institute of Technology – Andhra Pradesh (VIT-AP) University since September 2018. He is the coordinator for Artificial Intelligence and Robotics (AIR) Research Center at VIT-AP University. He is the lead engineer for the project VISU; an Advanced 3D printed Humanoid Robot developed by VIT-AP. He is an active contributor to the open source community and a lead writer in top security magazines such as Pentestmag and eforensics. He is an active reviewer in many reputed journals, including IEEE, Springer, IET, IGI Global and Hindawi. He is a recipient of the DST fellowship.