YOGA RECOMMENDATION AND MONITORING SYSTEM

A PROJECT REPORT

Submitted by

AISHWARYA .S 20202001

DURGA DEVI.R 20202017

VIGNESH KUMAR.R 20202058

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

PAAVAI ENGINEERING COLLEGE

(AUTONOMOUS)

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"YOGA RECOMMENDATION AND MONITORING SYSTEM"** is the bonafide work **"S.AISHWARYA (20202001), R.DURGA DEVI (20202017)"**and **"R.VIGNESH KUMAR (20202058)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.B.VENKATESAN, M.E.,Ph.D.,	**Mrs.M.SARANYA ,ME.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Associate professor,	Assistant professor,
Information Technology,	Information Technology,
Paavai Engineering College,	Paavai Engineering College,
Namakkal-637018.	Namakkal-637018.

Submitted for the University Examination held on…………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We, **AISHWARYA.S (20202001)** and **DURGA DEVI.R (20202017) VIGNESH KUMAR.R (20202058),** hereby declare that the project report titled **"YOGA RECOMMENDATION NAD MONITORING SYSTEM"** done by us under the guidance of **Mrs.M.SARANYA, M.E.,** is submitted in partial fufillment of the requirements for the award of Bachelor of Technology in Information Technology. Certified further that, to the best of our knowledge, the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

1.

2.

DATE: **3.**

PLACE: **SIGNATURE OF THE CANDIDATES**

ACKNOWLEDGEMENT

A great deal of arduous work and effort has been spent in implementing this project work. Several special people have guided us and have contributed significantly to this work and so this becomes obligatory to record our thanks to them.

We express our profound gratitude to our honourable **Chairman, Shri**.**CA. N.V.NATARAJAN, B.Com., F.C.A.,** and also to our Correspondent, Smt. **MANGAINATARAJAN, M.Sc.,** for providing all necessary facilities for the successful completion of our project.

We wish to express our sincere thanks to our respected Director-Administration**, Dr. K.K.RAMASAMY, M.E., Ph.D.,** for all the blessing and help provided during the period of project work.

We would like to thank our respected **Principal Dr. M.PREM KUMAR**,**M.E., Ph.D.,** for allowing us to do this project and providing required time to complete the same.

We wish to express our deep gratitude to **Dr. B.VENKATESAN, M.E., Ph.D.,** Head of the Department and Supervisor, for the able guidance and useful suggestions, which helped us for completing project work in time.

We express our sincere thanks to **Professor Dr. G.MADASAMY RAJA, M.E., Ph.D.,** Project Coordinator for the useful suggestions, which helped us for completing the project work in time.

We would like to extend our sincere thanks to **M.SARANYA, M.E.,** supervisor for giving us this opportunity to do this project and for her inspiring guidance, generous help and support. We would like to extend our sincere thanks to **all our department staff members and to our parents** for their advice and encouragement to do the project work with full interest and enthusiasm.

ABSTRACT

This study introduces a cutting-edge solution for enhancing the practice of yoga through a live monitoring and recommendation system powered by deep learning. Leveraging convolutional neural networks (CNNs), the system provides real-time feedback and personalized recommendations to users as they engage in yoga poses using a live camera feed. By addressing the challenges associated with manual pose correction, this innovative system aims to significantly improve the effectiveness, accessibility, and engagement of yoga practice. Through the utilization of deep learning algorithms trained on extensive datasets of annotated yoga pose images, the system achieves accurate detection and estimation of yoga poses from live camera feeds. This enables real-time processing of camera feeds, allowing for instantaneous feedback and tailored recommendations for individual users. Notably, the system is robust to variations in body shape, size, clothing, and environmental conditions, adaptable to different yoga styles and skill levels, and scalable to accommodate a wide range of poses and modifications. Moreover, adaptive feedback mechanisms dynamically adjust recommendations based on user progress and performance, further enhancing the personalized nature of the yoga experience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

CNNS	-	Convolutional Neural Networks
RGB	-	Red Green Blue
DL	-	Deep Learning
HTML	-	Hyper Text Markup Language
CSS	-	Casscading Style Sheet
WSGI	-	Web Server Gate Way Interface

CHAPTER 1

INTRODUCTION

Yoga Recommendation presents a pioneering approach to revolutionize the practice of yoga through the integration of advanced technology. Rooted in the ancient tradition of yoga, which has flourished for millennia as a means of cultivating physical vitality, mental clarity, and spiritual enlightenment, this study endeavors to harness the power of modern innovations to enhance the yoga experience. As yoga continues to gain popularity worldwide, there emerges a pressing need for solutions that not only preserve its traditional essence but also leverage cutting-edge methodologies to address contemporary challenges and optimize the benefits for practitioners of all levels. At the heart of this endeavor lies the development of a deep learning-based live yoga pose estimation system, poised to redefine how yoga is practiced and experienced in the digital age. By leveraging convolutional neural networks (CNNs) and real-time processing techniques, this system aims to provide practitioners with personalized feedback and recommendations, thereby facilitating correct posture alignment and enhancing the effectiveness, accessibility, and engagement of their yoga journey. With a focus on bridging the gap between ancient wisdom and modern technology, this paper sets out to unveil a transformative paradigm in the realm of yoga practice, paving the way for a more integrated and immersive experience for practitioners worldwide.

1.1 DEEP LEARNING

Deep learning can be defined as the method of machine learning and artificial intelligence that is intended to intimidate humans and their actions based on certain human brain functions to make effective decisions. It is a very important data science element that channels its modeling based on data-driven techniques under predictive modeling and statistics. To drive such a human-like ability to adapt and learn and to function accordingly, there have to be some strong forces which we popularly called algorithms. It has gained significant attention and popularity due to its ability to automatically learn and represent complex patterns and features from large amounts of unlabeled data. Deep learning algorithms have demonstrated remarkable performance across various domains such as computer vision, natural language processing, speech recognition, and more.

1.2 CONVOLUTION WORKS

Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (i.e the channel as images generally have red, green, and blue channels).

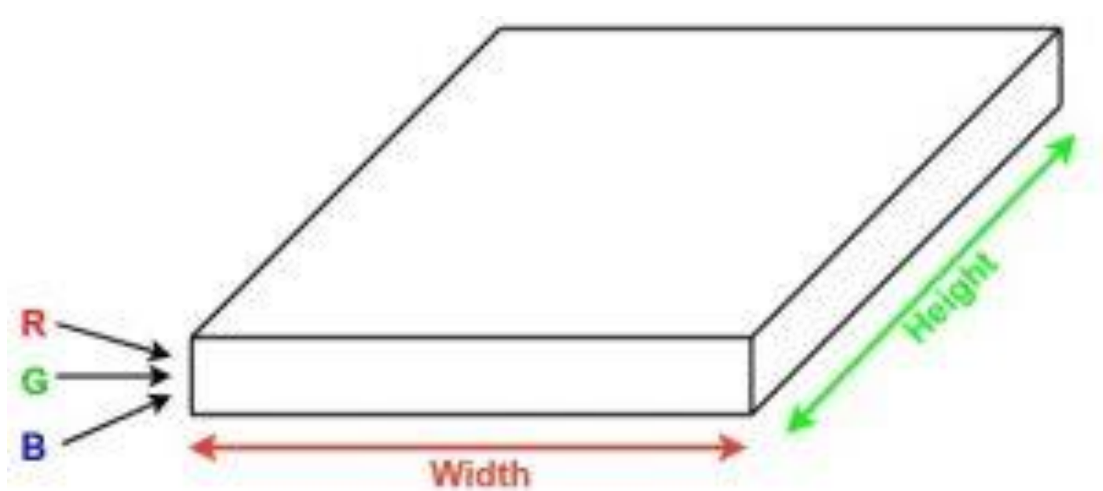


figure 1. Image source

Now imagine taking a small patch of this image and running a small neural network, called a filter or kernel on it, with say, K outputs and representing them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different widths, heights, and depths. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called **Convolution**. If the patch size is the same as that of the image it will be a regular neural network. Because of this small patch, we have fewer weights.

The data is fed into the model and output from each layer is obtained from the above step is called **feed forward**, we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. The error function measures how well the network is performing. After that, we back propagate into the model by calculating the derivatives. This step is called **Back propagation** which basically is used to minimize the loss.

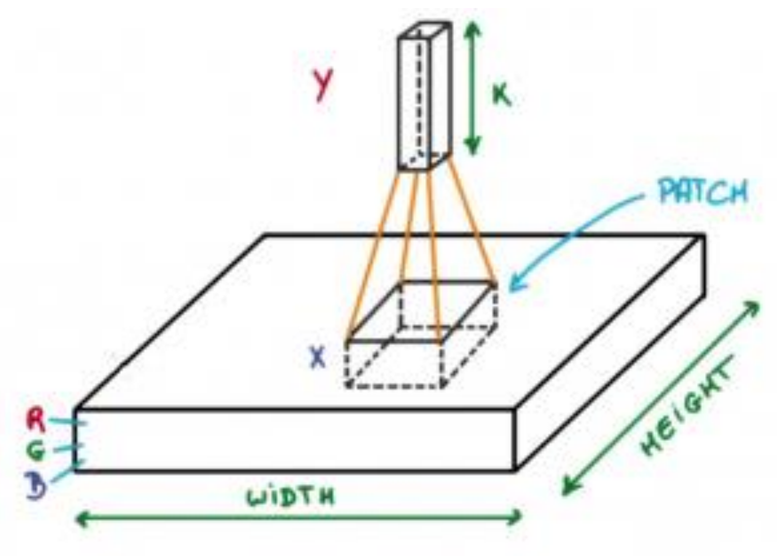


Figure 2. Image source: Deep Learning

1.3 IMAGE PROCESSING

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

- Image processing basically includes the following three steps:
- Importing the image via image acquisition tools;
- Analyzing and manipulating the image;
- Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

In this lecture we will talk about a few fundamental definitions such as image, digital image, and digital image processing. Different sources of digital images will be discussed and examples for each source will be provided. The continuum from image processing to computer vision will be covered in this lecture. Finally we will talk about image acquisition and different types of image sensors.

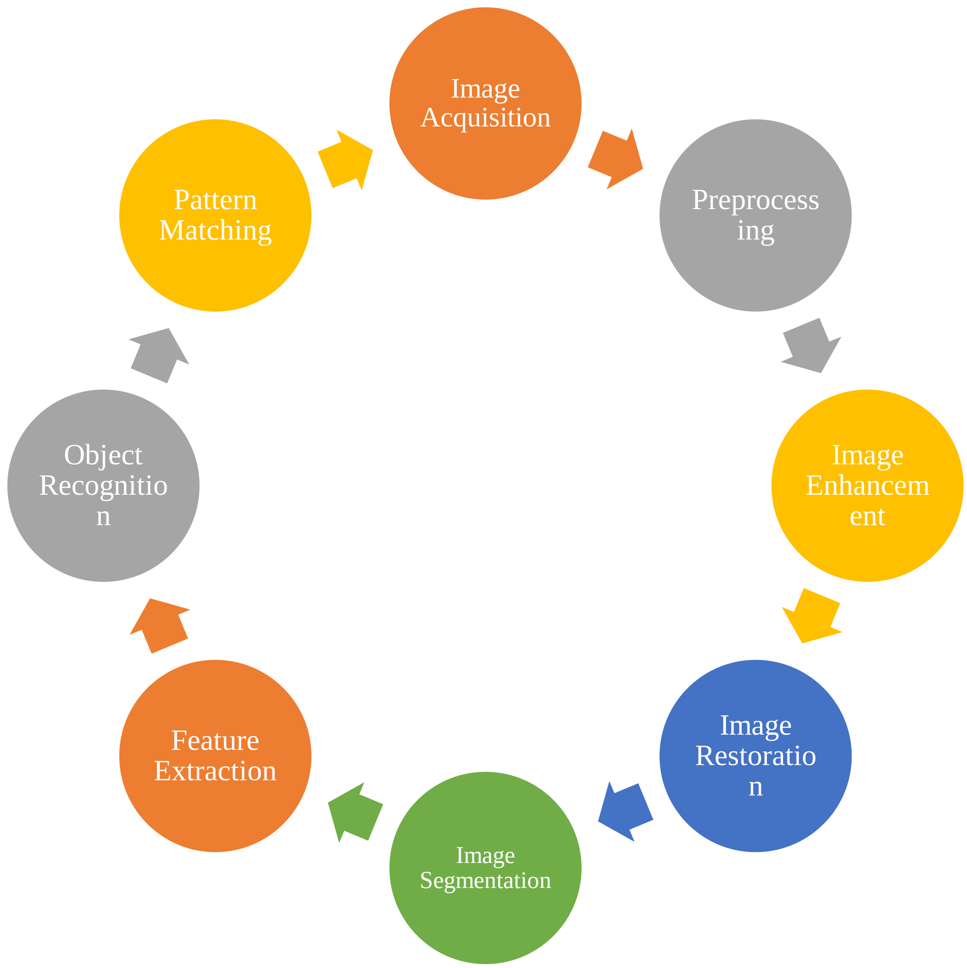


Figure 3. Image preprocessing

1.4 IMAGE TYPES

- Binary image
- Grayscale image
- Indexed image
- True color or RGB image

Binary Image:

A binary image is the simplest form of image representation, consisting of only two colors: black and white (or 0 and 1). Each pixel in the image is either completely black (0) or completely white (1). Binary images are commonly used in applications like image segmentation and object detection.

Grayscale Image:

In a grayscale image, each pixel is represented by a single intensity value ranging from 0 (black) to 255 (white) in an 8-bit depth. The shades of gray are used to represent variations in intensity. Grayscale images are commonly used in applications where color information is not necessary, such as medical imaging and certain types of image processing.

Indexed Image:

In an indexed image, the color information is stored in a separate color map or palette. Instead of representing each pixel with RGB values, each pixel contains an index that corresponds to a color in the palette. This reduces the amount of data required to represent the image. Indexed images are often used in applications where memory efficiency is crucial, such as in GIF images.

1.5 TRUE COLOR:

True color images, also known as RGB (Red, Green, Blue) images, represent each pixel with three color channels: red, green, and blue. Each channel can have intensity values ranging from 0 to 255, resulting in a wide spectrum of colors. True color images are commonly used in photography, digital displays, and most applications where detailed color information is essential.

1.6 APPLICATION

Photo Enhancement App:

Develop an application that automatically enhances the quality of photos by adjusting brightness, contrast, sharpness, and color balance.

Facial Recognition System:

Create a facial recognition system that can identify and authenticate individuals based on facial features. This can be used for security, access control, or social media tagging.

Object Detection and Recognition:

Build an application that detects and recognizes objects in images. This can be applied to assist visually impaired individuals, inventory management, or security surveillance**.**

Medical Image Analysis:

Develop a system for analyzing medical images like X-rays or MRIs. This can aid in the early detection of diseases and assist healthcare professionals in diagnosis.

Document Scanner:

Create an application that turns a smartphone into a document scanner. It can automatically detect the document edges, correct perspective, and enhance the document for better readability.

1.7 OBJECTIVE

- Address challenges associated with manual pose correction in yoga practice.
- Enhance the effectiveness, accessibility To develop a deep learning-based live yoga pose estimation system.
- In this is utilizing convolutional neural networks (CNNs) for accurate detection and estimation of yoga poses.

CHAPTER 2

LITERATURE SURVEY

2.1 TITLE: Enhanced Yoga Posture Detection using Deep Learning

AUTHOR: Shah Imran ,Abidul Islam

YEAR:2024

DESCRIPTION: Yoga is one of the best at-home exercises for maintaining our physical health. However, yoga is all about successfully performing the 82 Yoga Asanas throughout the course of six classes. Lamentably, not everyone has the knowledge or can perform yoga accurately. So to do yoga poses correctly we will have to find a yoga instructor, but it can be very hard and expensive to find yoga instructors considering all possible general situations and status. Using Deep Learning(DL) and modifying some pre-trained models to some extent can be a possible solution to detect yoga pose and class separately, which can eventually help general people. This work proposes a detailed experiment using two pre-trained CNN models along with an ensemble model to detect yoga poses accurately. The work was done on a total of 18488 images divided into 6 major yoga classes and 82 different poses. The aftermath of using ensemble modeling was instrumental as it was able to detect yoga poses with a 95% chance of assurance. Index Terms—Yoga poses, Transfer learning, Posture detection.

This dataset benefits app developers, machine learning researchers, Yoga instructors, and practitioners, who can use it to develop apps, train computer vision algorithms, and improve their practice.

2.2 TITLE:A Survey on Yogic Posture Recognition

AUTHOR: Arun Kumar RajendranAndSibiChakkaravarthySethuraman

YEAR:2023

DESCRIPTION: Yoga has been a great form of physical activity and one of the promising applications in personal health care. Several studies prove that yoga is used as one of the physical treatments for cancer, musculoskeletal disorder, depression, Parkinson's disease, and respiratory heart diseases. In yoga, the body should be mechanically aligned with some effort on the muscles, ligaments, and joints for optimal posture. Postural-based yoga increases flexibility, energy, overall brain activity and reduces stress, blood pressure, and back pain. Body Postural Alignment is a very important aspect while performing yogic asanas. Many yogic asanas including uttanasana, kurmasana, ustrasana, and dhanurasana, require bending forward or backward, and if the asanas are performed incorrectly, strain in the joints, ligaments, and backbone can result, which can cause problems with the hip joints. Hence it is vital to monitor the correct yoga poses while performing different asanas.

Yoga posture prediction and automatic movement analysis are now possible because of advancements in computer vision algorithms and sensors. This research investigates a thorough analysis of yoga posture identification systems using computer vision, machine learning, and deep learning techniques

The image dataset is organized into 10 subfolders, each with "Effective (right) Steps" and "Ineffective (wrong) Steps" folders. The video dataset has 4 videos for each posture, with 40 videos demonstrating effective (right) postures and 40 demonstrating.

2.3 TITLE: Real-Time Yoga Pose Detection Using OpenCV and MediaPipe

AUTHOR: IshaChaudhary**,**NongmeikapamThoiba Singh

YEAR:2023

DESCRIPTION: In this era of advanced technology, where everything is going online, from shopping to education, there may be ways to get an automated, personalised trainer and mentor for yoga and meditation. Yoga is one of the best exercises for maintaining good physical and mental health. But what if yoga is done incorrectly? It can be harmful to our bodies, so there is a need for a personalised yoga trainer and mentor who is available 24 x 7 to assist you with the poses and provide you with immediate feedback in order to maintain a healthy and wealthy body. In this model, we are coming up with the idea of providing real-time posture detection using deep learning and computer vision that will guide the user to the correct postures after comparing them with the standard yoga postures and also help them track their gym activities.

Our research also aims to integrate all of the features from various platforms related to posture detection applications on a single platform rather than switching between multiple platforms. In order to make it more user-friendly, we are giving users the option of selecting between an AI tutor and a physical trainer based on their needs. One of the objectives of the model is to display information about the pose, such as the benefits of performing it and how to incorporate it into routine exercises that the trainee can simultaneously practise to increase its effectiveness. Keywords— yoga, MediaPipe, OpenCV, pose detection.

At the heart of this endeavor lies the development of a deep learning-based live yoga pose estimation system, poised to redefine how yoga is practiced.

2.4 TITLE:A Real Time Monitoring System for Yoga Practitioners

AUTHOR:Prasanna Mani, ArunkumarThangavelu

YEAR:2017

DESCRIPTION: Now a days a majority share of people undergoes through some health problems or mental stress. The reason for both may be many factors like unhealthy food habits, lack of doing body exercise, pressure from family and job environment etc. These factors can physically and mentally deteriorate the creativity and productivity of a person. Yoga can be considered as one of the finest solutions in this case though it refreshes both physically and mentally. Yoga transforms a person balanced with mental, physical and spiritual elements in the right composition. In this technology era, it will be good to integrate yoga with the trends of technology. This work proposes a scheme of integrating and monitoring yoga activities using the concept of Internet of Things (IoT) and a smart application.

The body sensors (Pressure, temperature, humidity) attached with the person doing yoga senses relevant data and is processed using a central processor (Smart phone or smart devices) to provide necessary suggestions or feedbacks to the user. This work will provide a platform to the yoga practice person to monitor and review their yoga activities by themselves. Our research results claims that the various parameters like blood pressure, temperature and heart beat rate is improvised a lot after practicing yoga. Our survey results with the yoga persons doing this IoTbased yoga system shows that our system is much helpful for the yoga persons to practice yoga efficiently and effectively.

2.5 TITLE: Deep Learning Models for Yoga Pose Monitoring

AUTHOR: Debabrata Swain 1ORCID,SantoshSatapathy 1,Biswaranjan Acharya 2

YEAR:2022

DESCRIPTION: Activity recognition is the process of continuously monitoring a person's activity and movement. Human posture recognition can be utilized to assemble a self-guidance practice framework that permits individuals to accurately learn and rehearse yoga postures without getting help from anyone else. With the use of deep learning algorithms, we propose an approach for the efficient detection and recognition of various yoga poses. The chosen dataset consists of 85 videos with 6 yoga postures performed by 15 participants, where the key points of users are extracted using the Media pipe library.

A combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) has been employed for yoga pose recognition through real-time monitored videos as a deep learning model. Specifically, the CNN layer is used for the extraction of features from the key points and the following LSTM layer understands the occurrence of sequence of frames for predictions to be implemented. In following, the poses are classified as correct or incorrect; if a correct pose is identified, then the system will provide user the corresponding feedback through text/speech. This paper combines machine learning foundations with data structures as the synergy between these two areas can be established in the sense that machine learning techniques and especially deep learning can efficiently recognize data schemas and make them interoper**.**

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In existing yoga-related technologies may lack adaptability to different body types, skill levels, and yoga styles, limiting their effectiveness and relevance for a diverse range of practitioners. Moreover, the lack of standardized data and evaluation metrics makes it challenging to compare and assess the performance of different systems objectively The existing system of practicing yoga is characterized by a reliance on traditional instruction methods, supplemented by a variety of digital resources and technologies. While these resources offer convenience and accessibility, they often fall short in providing real-time feedback, personalized guidance. In existing system is yoga has become part of life for many people across the world. Due to this there is the need of scientific analysis of y postures. It has been observed that pose detection techniques can be used to identify the postures and also to assist the people to perform yoga more accurately. Recognition of posture is a challenging task due to the lack availability of dataset and also to detect posture on real-time bases. To overcome this problem a large dataset has been created which contain at least 5500 images of ten different yoga pose and used a tf-pose estimation Algorithm which draws a skeleton of a human body on the real-time bases. Angles of the joints in the human body are extracted using the tf-pose skeleton and used them as a feature to implement various machine learning models

DISADVANTAGES:

- **Limited Feedback**: Traditional methods of practicing yoga often lack immediate feedback on posture, alignment, and technique. Without real-time feedback, practitioners may struggle to correct their form or maintain proper alignment, potentially leading to ineffective or even injurious practice.
- **Difficulty in Self-Correction**: Without the guidance of an experienced instructor, practitioners may find it challenging to self-correct their poses and movements.

3.2 PROPOSED SYSTEM

The proposed system constitutes a pioneering endeavor in the realm of yoga practice, aiming to redefine the way practitioners engage with and benefit from this ancient tradition through the integration of advanced technology. Central to this initiative is the implementation of a CNN-based algorithm specifically tailored for yoga pose estimation. By leveraging deep learning methodologies, the system enables real-time processing of live camera feeds, providing practitioners with instantaneous feedback and personalized recommendations to enhance their practice experience. This innovative approach not only addresses the limitations of existing yoga practice methods but also holds the potential to revolutionize the effectiveness, accessibility, and engagement of yoga practice on a global scale. An essential component of the proposed system is the meticulous curation of a diverse dataset of annotated yoga pose images. This dataset captures a broad spectrum of poses, variations, and body types, ensuring its representativeness and balance for robust model training. Each image is meticulously annotated with key points representing joints and body segments, facilitated through labeling tools or manual annotation techniques. This comprehensive dataset serves as the foundation upon which the CNN-based algorithm is trained, equipping the system with the capacity to accurately detect and estimate yoga poses from live camera feeds in real-time. Through the seamless integration of cutting-edge technology with ancient wisdom, the proposed system endeavors to usher in a new era of transformative and personalized yoga practice experiences.

The operates through a systematic and iterative process designed to facilitate real-time yoga pose recognition during yoga practice sessions. It begins with the collection of a diverse dataset comprising annotated yoga pose images, capturing a wide range of poses, variations, and body types. These images undergo preprocessing, including resizing, normalization, and noise reduction, to ensure consistency and optimal quality. Feature extraction techniques are then applied to identify key patterns and characteristics within the preprocessed images. The heart of the system lies in the training of a convolutional neural network (CNN) algorithm, utilizing the extracted features to train a model capable of accurately detecting and estimating yoga poses. Through iterative training iterations, the model's parameters are adjusted to minimize error and enhance performance. Once trained, the model is deployed to enable real-time processing of live.

video inputs captured during yoga practice sessions. As practitioners engage in various poses, the system analyzes the live video feed, recognizing human poses and providing instantaneous feedback by displaying the detected yoga pose names in real-time. This seamless integration of data collection, preprocessing, feature extraction, algorithm training, model deployment, and real-time testing culminates in a holistic system that enhances the yoga practice experience by offering practitioners personalized guidance and support.

ADVANTAGES

- The real-time detected during yoga practice sessions.
- Enhances the effectiveness and accuracy of yoga pose recognition.
- Accommodates variations in body shape, size, clothing, and environmental conditions.
- Promotes correct posture alignment and technique refinement.
- Increases accessibility to yoga practice through innovative technology.
- Fosters engagement and motivation among practitioners.
- To enables seamless integration into existing yoga practice routines.
- Improves posture alignment and promotes correct execution of yoga poses.

Once preprocessed, the input images are fed into the feature extraction module, where salient characteristics and key patterns are identified. Leveraging advanced techniques such as edge detection, texture analysis, and keypoint extraction, this module discerns relevant features that encapsulate the unique attributes of each yoga pose. These extracted features serve as the basis for subsequent stages of algorithmic processing, including training the convolution neural network (CNN) model and deploying it for real-time pose recognition. The input images provide the foundation for the system to deliver accurate and personalized feedback, ultimately enhancing the effectiveness and engagement of yoga practice sessions for practitioners.

These platforms offer a plethora of images capturing individuals performing a wide range of yoga poses in diverse settings, providing a comprehensive representation of the yoga practice landscape. Each image is meticulously annotated with its associated yoga pose name, facilitating supervised learning during subsequent stages of algorithm development.

CHAPTER 4

SYSTEM REQUIREMENT

4.1 HARDWARE SYSTEM CONFIGURATION:-

- processor - Pentium – IV
- RAM - 4 GB (min)
- Hard Disk - 20 GB

4.2 SOFTWARE SYSTEM CONFIGURATION:-

- Operating System : Windows 7 or 8
- Frontend :HTML,CSS
- Backend :python

4.3 FRONT END

4.3.1 HTML

Figure 4.Html

As we all know HTML is a language of the web. It’s used to design the web pages or we can say structure the page layouts of a website. HTML stands for HYPERTEXT MARKUP LANGUAGE, as its full form suggests it’s not any programming language, a markup language. So, while the execution of HTML code we can’t face any such error. In real HTML code wasn’t compiled or interpreted because HTML code was rendered by the browser. which is similar to the compilation of a program. Html content is parched through the browser to display the content of HTML.

4.3.2 CASCADING STYLE SHEETS (CSS)

Figure 5.css

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, the development of various parts of CSS specification was done synchronously, which allowed the versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, or even CSS3. There will never be a CSS3 or a CSS4; rather, everything is now CSS without a version number. After CSS 2.1, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to develop and release recommendations separately per module. Instead of versioning the CSS specification, W3C now periodically takes a snapshot of the latest stable state of the CSS specification.

4.4 BACKEND

4.4.1 PYTHON INTRODUCTION

Figure 6. python

Python is a general purpose programming language. It is very easy to learn, easy syntax and readability is one of the reasons why developers are switching to python from other programming languages. We can use python as object oriented and procedure oriented language as well. It is open source and has tons of libraries for various implementations.Python is a high level interpreted language, which is best suited for writing python scripts for automation and code re-usability. It was created in 1991 by Guido Van Rossum. The origin of its name is inspired by the comedy series called 'Monty python'.

Python is a high-level scripting language, interpreted, interactive, and object-oriented. it is designed to be highly readable. It often uses keywords in English where other languages use punctuation and has less syntactic constructions than other languages.

- It is a powerful and easy to learn programming language and efficient high-level data structures and a simple but effective approach to object-oriented programming.

4.4.2 FLASK FRAMWORK

Figure 7. flask framework

Flask is a lightweight and flexible web framework for Python, designed to make web development simple and scalable. It provides developers with the tools they need to build web applications quickly and efficiently, without imposing too many constraints or dependencies. At its core, Flask is based on the WSGI (Web Server Gateway Interface) standard, making it compatible with a wide range of web servers and deployment options. One of Flask's key features is its simplicity. It follows the "microframework" philosophy, which means it focuses on doing one thing well – handling HTTP requests and responses – while leaving other tasks like database management and authentication to extensions or third-party libraries. This minimalistic approach gives developers the freedom to choose the components they need for their projects, resulting in lean and efficient web applications. Additionally, Flask offers a rich ecosystem of extensions and plugins that extend its functionality and provide additional features such as form validation, user authentication, and database integration. These extensions, combined with Flask's intuitive API and extensive documentation, make it easy for developers to customize and scale their applications according to their specific requirements.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

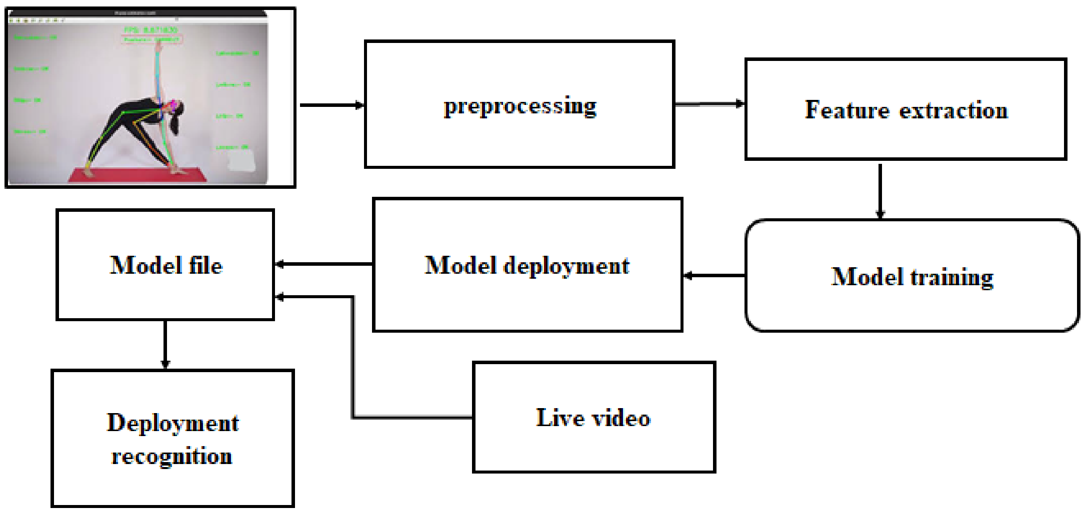


Figure 8. System Architecture

5.2 MODULE LIST

- Data collection module
- Image preprocessing module
- Feature extraction module
- CNN algorithm training module
- Model deployment module
- Train module build module
- Testing input module
- Yoga pose recognition module

5.3 MODULE DESCRIPTION

5.3.1 INPUT IMAGE

In the proposed system, the input image plays a crucial role in the process of yoga pose recognition and feedback delivery. Each input image represents a snapshot of a practitioner performing a yoga pose, captured in real-time through a live camera feed. These images serve as the raw data that the system analyzes to provide instantaneous feedback and personalized recommendations during yoga practice sessions. Before processing, the input images undergo a series of preprocessing steps to ensure consistency and optimal quality. Techniques such as resizing, normalization, and noise reduction are applied to standardize the images and mitigate any factors that could hinder subsequent algorithmic processing. Resizing ensures that all images have uniform dimensions, facilitating consistent input sizes for the deep learning model. Normalization techniques adjust pixel values across the images, enhancing comparability and reducing the influence of variations in brightness and contrast that may arise from different lighting conditions or camera settings. Additionally, noise reduction methods are employed to diminish unwanted artifacts or distortions present in the images, ensuring that the algorithm can focus on extracting meaningful features relevant to yoga pose recognition without being impeded by extraneous visual clutter or interference. Once preprocessed, the input images are fed into the feature extraction module, where salient characteristics and key patterns are identified. Leveraging advanced techniques such as edge detection, texture analysis, and keypoint extraction, this module discerns relevant features that encapsulate the unique attributes of each yoga pose. These extracted features serve as the basis for subsequent stages of algorithmic processing, including training the convolutional neural network (CNN) model and deploying it for real-time pose recognition.

5.3.2 DATA COLLECTION

The Data Collection module serves as the foundation of this paper's methodology, playing a pivotal role in acquiring a diverse and comprehensive dataset of yoga pose images alongside their corresponding names. To ensure the dataset's richness and relevance, a meticulous process is undertaken to source images from various repositories such as Kaggle and other reputable websites dedicated to yoga-related content. These platforms offer a plethora of images capturing individuals performing a wide range of yoga poses in diverse settings, providing a comprehensive representation of the yoga practice landscape. Each image is meticulously annotated with its associated yoga pose name, facilitating supervised learning during subsequent stages of algorithm development. This rigorous data collection process ensures the dataset's suitability for training the deep learning model, enabling it to accurately recognize and classify yoga poses from live video inputs. Moreover, the data collection module prioritizes inclusivity and diversity, striving to encompass a wide array of yoga poses performed by individuals of different ages, genders, and body types. This emphasis on diversity ensures that the trained model is robust and capable of accurately recognizing yoga poses across various demographics and contexts. By gathering a comprehensive dataset that reflects the breadth and depth of the yoga practice spectrum, the data collection module lays the groundwork for subsequent stages of the algorithm's development, ultimately contributing to the system's effectiveness.

5.3.3 IMAGE PROCESSING

The Image Preprocessing module is a crucial step in preparing the collected yoga pose images for effective analysis and accurate pose estimation within this paper's methodology. Through a series of techniques including resizing, normalization, and noise reduction, the module aims to standardize the images and mitigate any factors that could hinder the subsequent stages of algorithmic processing. Resizing ensures uniformity in image dimensions, facilitating consistentinput sizes for the deep learning model and optimizing computational efficiency during training and inference. Additionally, normalization techniques are applied to adjust pixel values across the images, enhancing their comparability and reducing the influence of variations in brightness and contrast that may arise from different lighting conditions or camera settings. Furthermore, noise reduction methods are employed to diminish any unwanted artifacts or distortions present in the images, ensuring that the algorithm can focus on extracting meaningful features relevant to yoga pose recognition without being impeded by extraneous visual clutter or interference. By implementing these preprocessing techniques, the Image Preprocessing module aims to enhance the quality and uniformity of the yoga pose images, thereby improving the accuracy and reliability of subsequent algorithmic processing stages. This standardized preprocessing approach not only streamlines the workflow but also contributes to the overall effectiveness and robustness of the system in providing real-time pose estimation and feedback during yoga practice sessions. As a foundational component of the methodology, the Image Preprocessing module plays a crucial role in ensuring that the collected data is optimally prepared for feature extraction and model training, ultimately facilitating the system's ability to deliver accurate and meaningful insights to practitioners engaged in yoga practice.

5.3.4 MODEL DEPLOYMENT MODULE

The Model Deployment module marks the transition from development to practical application, as it involves integrating the trained Convolutional Neural Network (CNN) model into a software framework or application capable of real-time processing. This integration is crucial for enabling the system to analyze live video streams captured during yoga practice sessions, providing practitioners with instantaneous feedback and pose recognition as they perform various yoga poses. By deploying the trained CNN model, the system becomes equipped to process incoming video data in real-time, enabling it to identify and classify yoga poses as they are performed, facilitating accurate posture alignment and enhancing the overall yoga practice experience. Moreover, the Model Deployment module serves as the bridge between the algorithmic components of the system and the end-user interface, ensuring seamless interaction and usability. The integrated CNN model operates within the software framework or application, where it continuously analyzes incoming video streams, detects yoga poses, and provides real-time feedback to practitioners. This deployment enables the system to operate efficiently and effectively in real-world scenarios, where practitioners can engage in yoga practice sessions with the assurance of receiving immediate guidance and support. Ultimately, by enabling the analysis of live video streams and facilitating real-time processing, the Model Deployment module empowers practitioners to enhance their yoga practice experience and achieve optimal posture alignment with the assistance of advanced technology.

5.3.5 TESTING INPUT MODULE

The Testing Input module is an integral part of the system's functionality, designed to seamlessly integrate live video streams as input data for real-time processing. Its primary function is to facilitate the acquisition of live video feeds captured.

5.4 ALGORITHM CONVOLUTIONAL NEURAL NETWORKS (CNNS)

Convolutional Neural Networks (CNNs) stand as a cornerstone in modern deep learning architectures, particularly renowned for their remarkable capabilities in image recognition and analysis. Developed inspired by the structure and functioning of the visual cortex in the human brain, CNNs have revolutionized various fields, including computer vision, pattern recognition, and natural language processing. The CNNs lies a hierarchical arrangement of convolutional layers, pooling layers, and fully connected layers, meticulously designed to extract and learn intricate features from input data. These layers operate through a process of convolution, where small filters or kernels are systematically applied across the input image to detect spatial patterns and features. Subsequent pooling layers serve to down sample the spatial dimensions of the feature maps, thereby reducing computational complexity while preserving essential information.

Through iterative training processes involving forward and backward propagation of errors, CNNs autonomously learn to discern and classify complex patterns within images, enabling tasks such as object detection, segmentation, and pose estimation with unprecedented accuracy and efficiency. In essence, CNNs represent a paradigm shift in machine learning, empowering systems to perceive, understand, and interpret visual information with human-like precision, thereby unlocking a myriad of applications across diverse domains.

The Image Preprocessing module aims to enhance the quality and uniformity of the yoga pose images, thereby improving the accuracy and reliability of subsequent algorithmic processing stages. This standardized preprocessing approach not only streamlines the workflow but also contributes to the overall effectiveness and robustness.

In essence, CNNs represent a paradigm shift in machine learning, empowering systems to perceive, understand, and interpret visual information with human-like precision, thereby unlocking a myriad of applications across diverse domains.

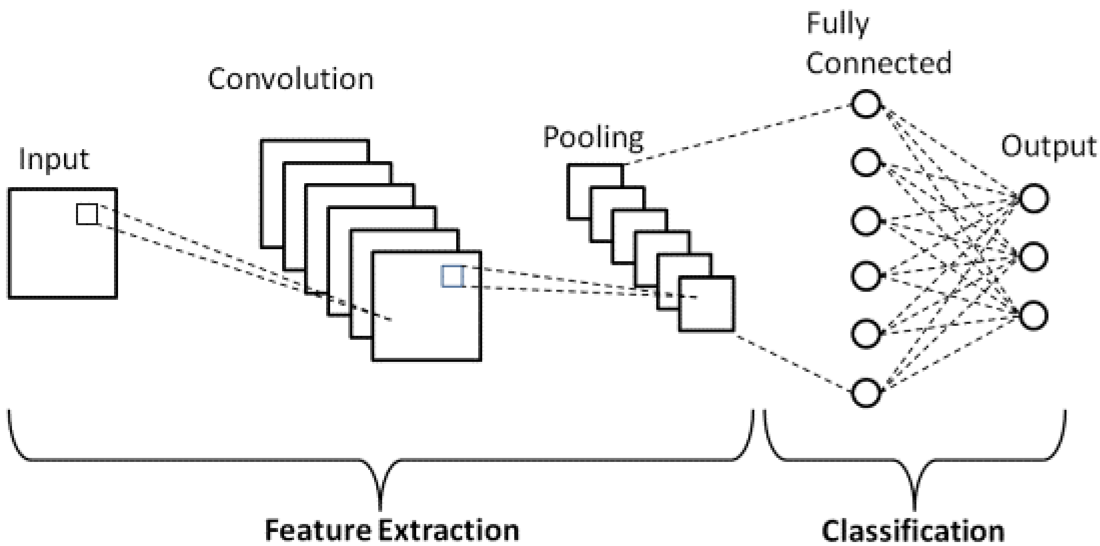


Figure 9. Algorithm Convolutional CNNS

- ❖ Input Layer: The algorithm begins by receiving input data, which consists of images captured from live video feeds depicting practitioners in various yoga poses.
- ❖ Convolutional Layers: These layers form the backbone of the algorithm, where a series of convolutional filters are applied to the input images. Each filter
- ❖ Activation Function: Following each convolutional operation, an activation function, commonly ReLU (Rectified Linear Unit), is applied element-wise to introduce non-linearity into the network.

CHAPTER 6

SYSTEM IMPLEMENTATION

The implementation of the proposed deep learning-based live yoga pose estimation system yielded promising results, demonstrating its effectiveness in real-time pose recognition and feedback provision during yoga practice sessions. Through rigorous data collection efforts, a diverse dataset of annotated yoga pose images was compiled, enabling the system to accurately recognize and classify yoga poses from live video inputs. The utilization of image preprocessing techniques ensured standardized and enhanced image quality, while feature extraction methods facilitated the extraction of discriminative features essential for accurate pose estimation. The training of the Convolutional Neural Network (CNN) model on the extracted features and corresponding labels resulted in a highly optimized model capable of robust performance in real-world scenarios. Furthermore, the deployment of the trained CNN model within a software framework or application enabled seamless integration and analysis of live video streams captured during yoga practice sessions.

This integration facilitated real-time processing of video inputs, empowering practitioners with instantaneous feedback and personalized recommendations tailored to their individual needs. The system's human pose recognition module successfully detected and recognized yoga poses in real-time, outputting the detected pose names to practitioners and enhancing their practice experience. Overall, the results obtained highlight the potential of the proposed system to revolutionize the way yoga is practiced, offering practitioners a transformative and personalized yoga experience that promotes correct posture alignment, technique refinement, and overall well-being.

CHAPTER 7

APPENDIX

7.1 SOURCE CODE

```
from __future__ import division, print_function

import sys

import os

import glob

import re

import numpy as np

import tensorflow as tf

import time

import serial

#from mail import report_send_mail

#from mail import*



from tensorflow.compat.v1 import ConfigProto

from tensorflow.compat.v1 import InteractiveSession


config = ConfigProto()
```

```
config.gpu_options.per_process_gpu_memory_fraction = 0.5
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
'''
from werkzeug.utils import secure_filename
ser = serial.Serial(
    port='COM13',
    baudrate=9600,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    bytesize=serial.EIGHTBITS,
    timeout=.1,
    rtscts=0
)'''

MODEL_PATH = 'keras_model.h5'
```

```
model = load_model(MODEL_PATH)



def model_predict(img_path, model):
    print(img_path)
    img = image.load_img(img_path, target_size=(224, 224))
    x = image.img_to_array(img)

    x = x / 255
    x = np.expand_dims(x, axis=0)

    preds = model.predict(x)
    preds = np.argmax(preds, axis=1)
    if preds == 0:
        preds = "Gomukasana"
        #ser.write(b'1')
       # print(ser.write(b'1'))
        #report_send_mail(preds, 'image.jpg')
        time.sleep(3)
    elif preds==1:
```

```
        preds = "dandasana"
        time.sleep(3)
    elif preds==2:
         preds = "balasana"
         time.sleep(3)
    elif preds==3:
        preds = "Halasana"


       # print(ser.write(b'2'))
            elif preds==7:
        preds = "Ustrasana"
        #ser.write(b'2')
        #report_send_mail(preds, 'image.jpg')
       # print(ser.write(b'2'))
        time.sleep(3)


img_path = filedialog.askopenfilename(title="Select Image", filetypes=[("Image files", "*.jpg; *.jpeg; *.png")])
    img = cv2.imread(img_path)
    cv2.imshow("Selected Image",img)
```

```
    time.sleep(3)
elif preds==6:
    preds = "Salabhasana"
    #ser.write(b'2')
    #report_send_mail(preds, 'image.jpg')
   # print(ser.write(b'2'))
    time.sleep(3)

elif preds==7:
    preds = "Ustrasana"
    #ser.write(b'2')
    #report_send_mail(preds, 'image.jpg')
   # print(ser.write(b'2'))
    time.sleep(3)

elif preds==8:
    preds = "Virabharasana"
    #ser.write(b'2')
    #report_send_mail(preds, 'image.jpg')
   # print(ser.write(b'2'))
```

```
    time.sleep(3)
elif preds==2:
     preds = "balasana"
     time.sleep(3)
elif preds==3:
    preds = "Halasana"
    #ser.write(b'2')
    #report_send_mail(preds, 'image.jpg')
   # print(ser.write(b'2'))
    time.sleep(3)
elif preds==4:
    preds = "Malasana"
    #ser.write(b'2')
    #report_send_mail(preds, 'image.jpg')
   # print(ser.write(b'2'))
    time.sleep(3)
elif preds==5:
    preds = "Padmasana"
    #ser.write(b'2')
    #report_send_mail(preds, 'image.jpg')
```

```
    # print(ser.write(b'2'))
     time.sleep(3)
elif preds==6:
     preds = "Salabhasana"
     #ser.write(b'2')
     #report_send_mail(preds, 'image.jpg')
    # print(ser.write(b'2'))
     time.sleep(3)


elif preds==7:
     preds = "Ustrasana"
     #ser.write(b'2')
     #report_send_mail(preds, 'image.jpg')
    # print(ser.write(b'2'))
     time.sleep(3)


elif preds==8:
     preds = "Virabharasana"
     #ser.write(b'2')
     #report_send_mail(preds, 'image.jpg')
```

```
        # print(ser.write(b'2'))
         time.sleep(3)


    elif preds==9:
         preds = "Savasana"
         #ser.write(b'2')
         #report_send_mail(preds, 'image.jpg')
        # print(ser.write(b'2'))
         time.sleep(3)
    return preds
```

```
import cv2


# Open the video capture
cap = cv2.VideoCapture(0)  # Use 0 for the default camera, or specify the video file path
```

```
# Check if the camera/video is opened successfully
if not cap.isOpened():
    print("Error opening video capture.")
    exit()

# Set the video capture duration (in seconds)
capture_duration = 5

# Set the frame rate of the video capture
frame_rate = 30  # Adjust as per your requirement

# Calculate the number of frames to capture
num_frames = int(capture_duration * frame_rate)

# Capture the frames
for i in range(num_frames):
    ret, frame = cap.read()  # Read a frame from the video capture

    if not ret:
        print("Error reading frame.")
```

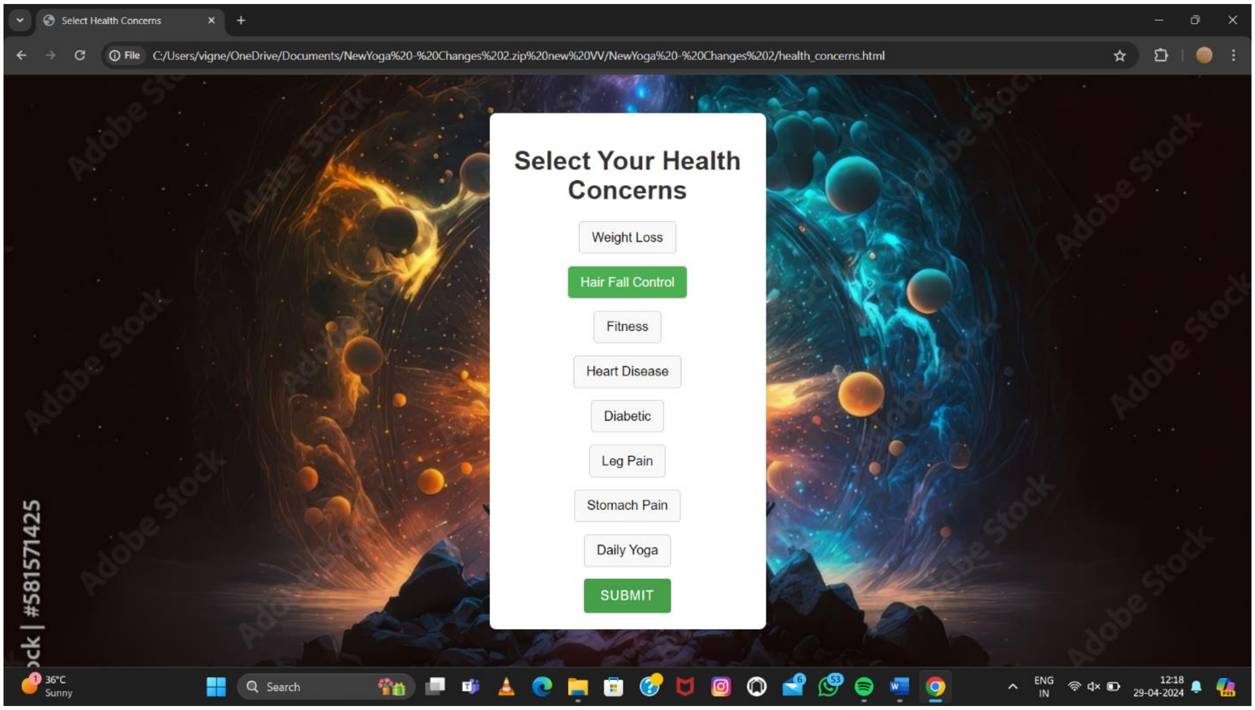
```
        break

    cv2.imshow("Video Capture", frame)  # Display the frame

    # Wait for 1ms and check if the user pressed the 'q' key to exit
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Save the last captured frame as an image
image_path = "image.jpg"  # Specify the path and filename for the image
cv2.imwrite(image_path, frame)

# Release the video capture and close any open windows
cap.release()
cv2.destroyAllWindows()

print("Image saved successfully.")



        time.sleep(3)
```

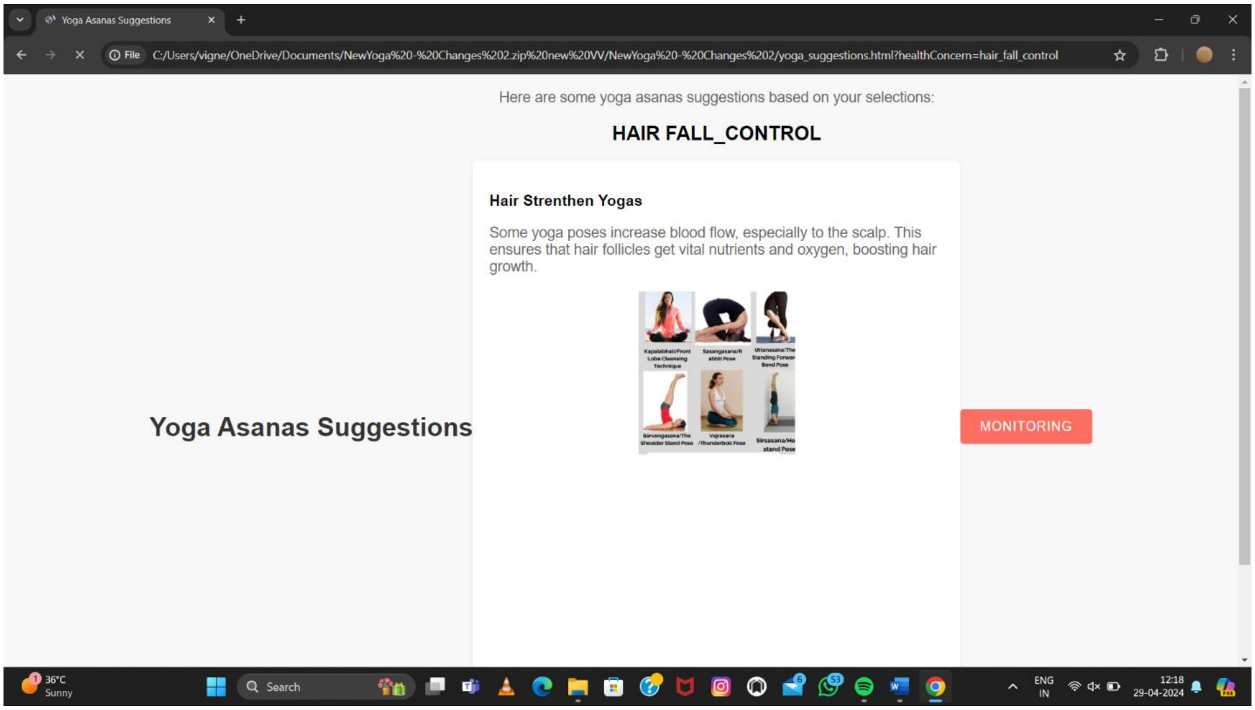
```
    elif preds==9:
        preds = "Savasana"
        #ser.write(b'2')
        #report_send_mail(preds, 'image.jpg')
       # print(ser.write(b'2'))
        time.sleep(3)
    return preds


import cv2


# Open the video capture
cap = cv2.VideoCapture(0)  # Use 0 for the default camera, or specify the video file path


# Check if the camera/video is opened successfully
if not cap.isOpened():
    print("Error opening video capture.")
    exit()
```

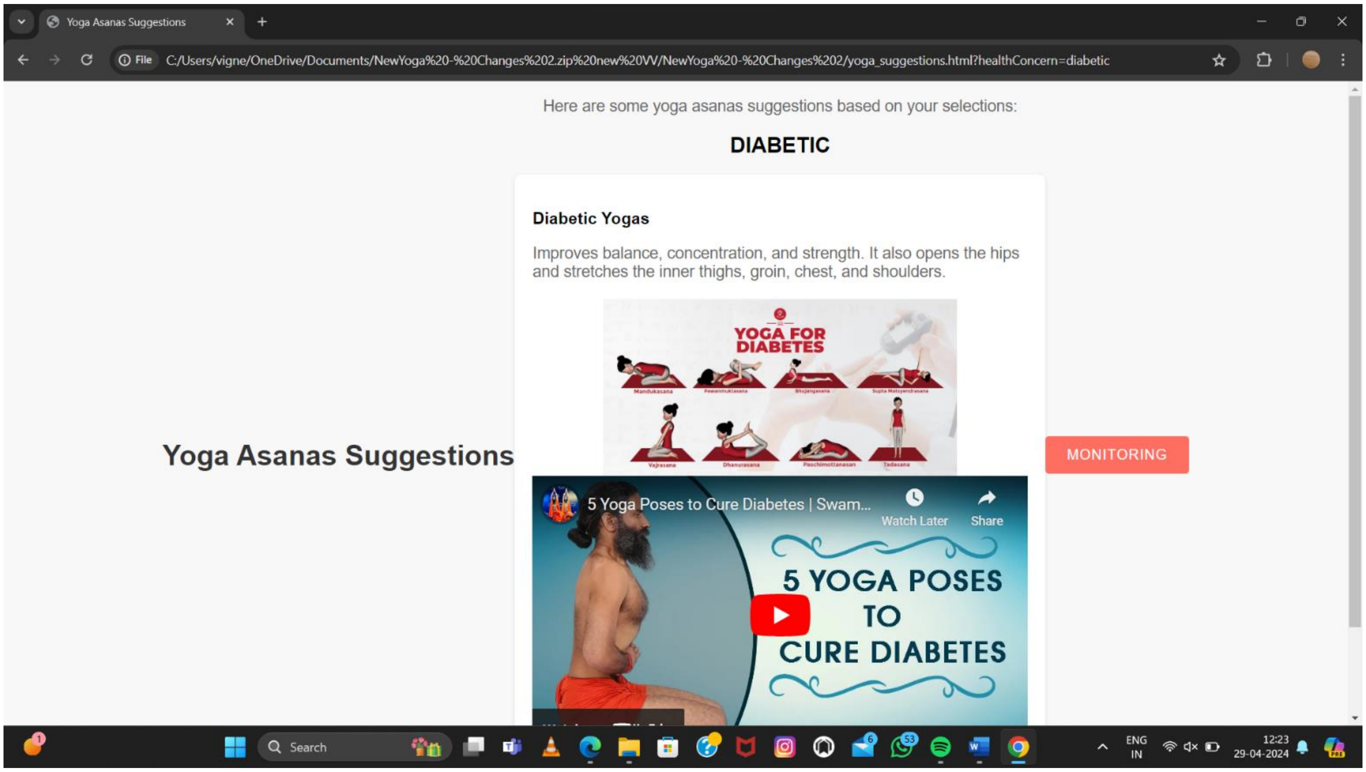
```
# Set the video capture duration (in seconds)
capture_duration = 5

# Set the frame rate of the video capture
frame_rate = 30  # Adjust as per your requirement

# Calculate the number of frames to capture
num_frames = int(capture_duration * frame_rate)

# Capture the frames
for i in range(num_frames):
    ret, frame = cap.read()  # Read a frame from the video capture

    if not ret:
        print("Error reading frame.")
        break

    cv2.imshow("Video Capture", frame)  # Display the frame

    # Wait for 1ms and check if the user pressed the 'q' key to exit
```

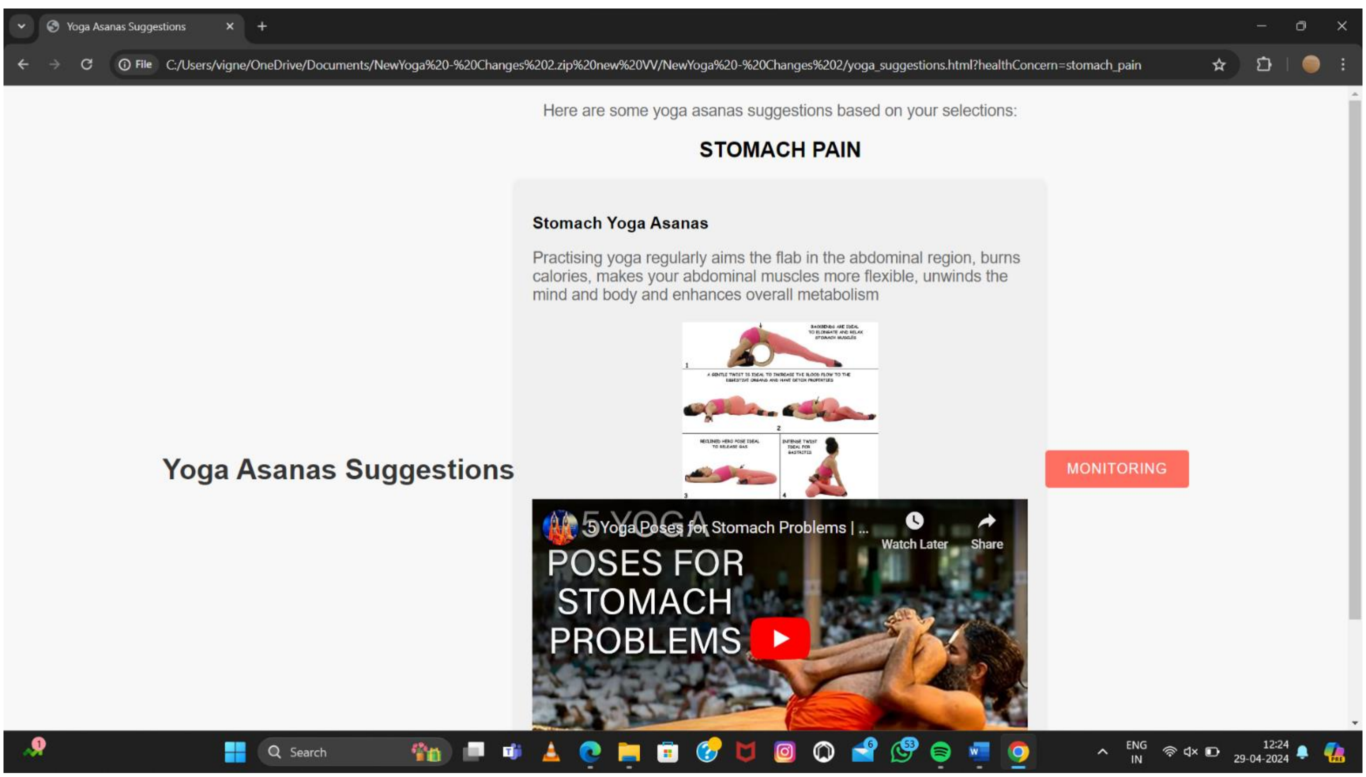
```
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Save the last captured frame as an image
image_path = "image.jpg"  # Specify the path and filename for the image
cv2.imwrite(image_path, frame)

    cv2.imwrite(image_path, img)

if not img_path:
    print("No image selected. Exiting.")
    sys.exit()

# Call the model prediction function
result = model_predict(img_path, model)# Display the result
print("Predicted Result:", result)
```

7.2 0UTPUT SCREENSHOTS






Yoga Asanas Suggestions
C:/Users/vigne/OneDrive/Documents/NewYoga%20-%20Changes%202.zip%20new%20VV/NewYoga%20-%20Changes%202/yoga_suggestions.html?healthConcern=diabetic
Here are some yoga asanas suggestions based on your selections:
DIABETIC
Yoga Asanas Suggestions
Diabetic Yogas
Improves balance, concentration, and strength. It also opens the hips and stretches the inner thighs, groin, chest, and shoulders.
YOGA FOR DIABETES
MONITORING
5 Yoga Poses to Cure Diabetes | Swam...
Watch Later
Share
5 YOGA POSES TO CURE DIABETES
Search
ENG IN
12:23
29-04-2024


Yoga Asanas Suggestions
C:/Users/vigne/OneDrive/Documents/NewYoga%20-%20Changes%202.zip%20new%20VV/NewYoga%20-%20Changes%202/yoga_suggestions.html?healthConcern=stomach_pain
Here are some yoga asanas suggestions based on your selections:
STOMACH PAIN
Yoga Asanas Suggestions
Stomach Yoga Asanas
Practising yoga regularly aims the flab in the abdominal region, burns calories, makes your abdominal muscles more flexible, unwinds the mind and body and enhances overall metabolism
MONITORING
5 Yoga Poses for Stomach Problems | ...
Watch Later
Share
5 YOGA POSES FOR STOMACH PROBLEMS
Search
ENG IN
12:24
29-04-2024

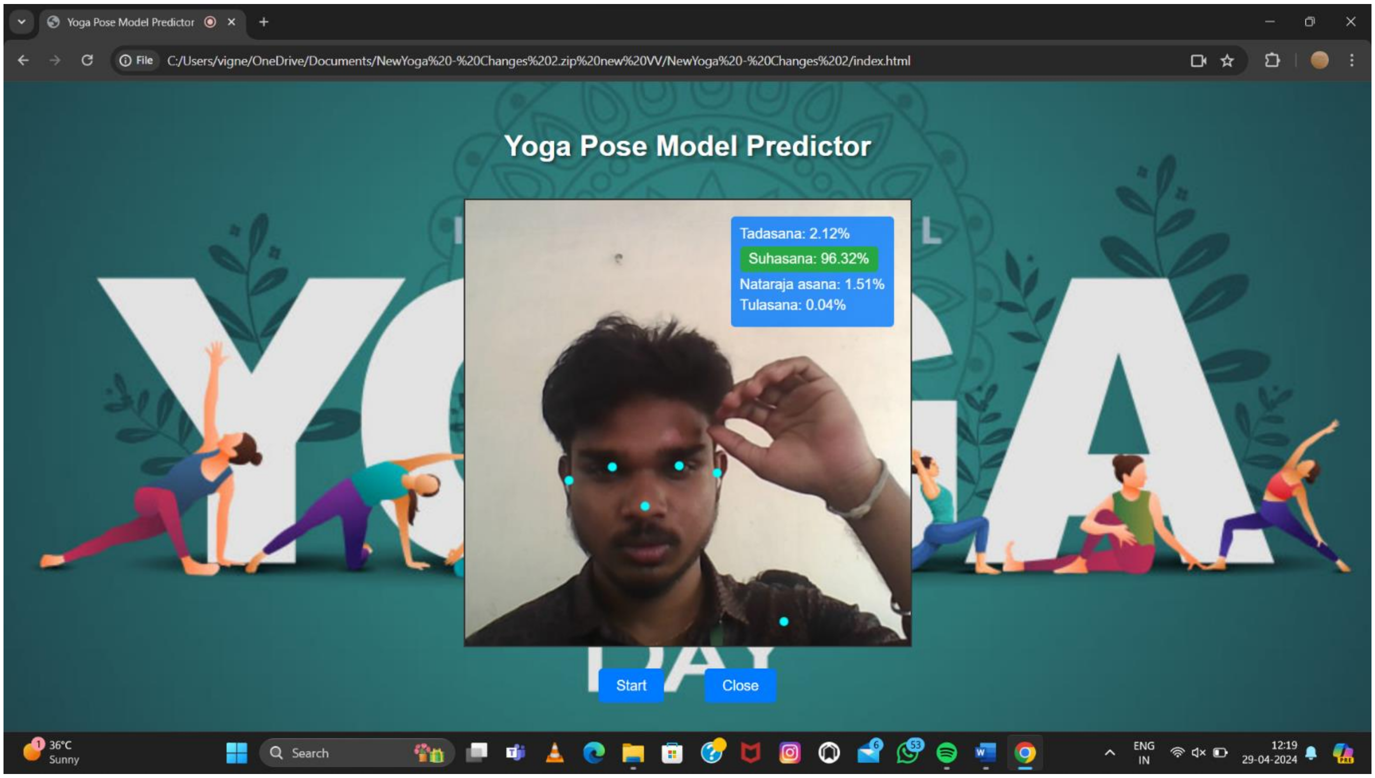


Figure 10. Output Screenshots

CHAPTER 8

CONCLUSION

In conclusion, the proposed deep learning-based live yoga pose estimation system presents a pioneering approach to enhancing the effectiveness, accessibility, and engagement of yoga practice. By leveraging Convolutional Neural Networks (CNNs) trained on comprehensive datasets of annotated yoga pose images, the system demonstrates remarkable capabilities in real-time pose recognition and feedback provision during yoga practice sessions. Through meticulous data collection, image preprocessing, feature extraction, and CNN algorithm training, the system achieves robustness to variations in body shape, size, clothing, and environmental conditions, catering to practitioners of diverse backgrounds and skill levels. Moreover, the integration of edge computing and hardware acceleration techniques enables the system to process live video inputs in real-time, providing instantaneous feedback and personalized recommendations tailored to individual users. The deployment of the trained CNN model within a software framework or application facilitates seamless analysis of live video streams captured during yoga practice sessions, empowering practitioners with immediate guidance and support. By recognizing yoga poses in real-time and outputting detected pose names to practitioners, the system enhances the yoga practice experience, promoting correct posture alignment, technique refinement, and overall well-being. In essence, the proposed system represents a convergence of ancient wisdom and modern technology, bridging the gap between traditional yoga practice methods and contemporary innovations. By offering practitioners real-time feedback and guidance, the system fosters a deeper connection between mind and body, facilitating a more immersive and rewarding yoga practice experience. Moving forward, continued advancements in deep learning

CHAPTER 9

REFERENCE

1. H. Cramer, J. Langhorst, G. Dobos, and R. Lauche, 2016 “Yoga for Metabolic Syndrome: A Systematic Review and Meta-Analysis”, European Journal of Preventive Cardiology, Vol.23, No.18, pp.1982-1993.

2. M.SivaSindhuri1 ,N. Anusha2” 2016 Text Separation in Document Images through Otsu’s Method”, IEEE WiSPNET conference.

3. L. Miranda, J. Viterbo, and F. Bernardini Apr. 2022., ‘‘A survey on the use of machine learning methods in context-aware middlewares for human activity recognition,’’ Artif. Intell. Rev., vol. 55, no. 4, pp. 3369–3400.

4. R. Liu, A. A. Ramli, H. Zhang, E. Henricson, and X. Liu 2021, ‘‘An overview of human activity recognition using wearable sensors: Healthcare and artificial intelligence,’’ in Proc. Int. Conf. Internet Things. Cham, Switzerland: Springer, , pp. 1–14.

5. S. Aftab, S. F. Ali, A. Mahmood, and U. Suleman2022, ‘‘A boosting framework for human posture recognition using spatio-temporal features along with radon transform,’’ Multimedia Tools Appl., vol. 8, pp. 42325–42351, Aug..

6. Tlili, F., Haddad, R., Bouallegue, R(2021).. et al. A Real-time Posture Monitoring System Towards Bad Posture Detection. Wireless PersCommun 120, 1207a1227 https://doi.org/10.1007/s11277-021-08511-2 .

7. G. Flutur, B. Movileanu, L. KAiroly, I. Danci, D. Cosovanu and O. ˜ P. Stan, ”Smart Chair System for Posture Correction,” 2019 22nd Euromicro Conference on Digital System Design (DSD), Kallithea, Greece, 2019, pp. 436-441, doi: 10.1109/DSD.2019.00069.

8. K. Bramhapurikar, A. Prabhune, S. Chavan, G. C. Ghivela and J. Sengupta, ”A Wearable Posture Corrector Device,” 2018 9th International Conference on Computing, Communication and

9. S. Chopra, M. Kumar and S. Sood,’Wearable posture detection and alert system,’ 2016 International Conference System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2016, pp. 130- 134, doi: 10.1109/SYSMART.2016.7894504.

10. Thoutam, V.A.; Srivastava, A.; Badal, T.; Mishra, V.K.; Sinha, G.R.; Sakalle, A.; Bhardwaj, H.; Raj, M. Yoga Pose Estimation and Feedback Generation Using Deep Learning. *Comput. Intell. Neurosci.* **2022**. Availableonline: **https://www.hindawi.com/journals/cin/2022/4311350/** (accessed on 10 September 2022).

11. S. Liaqat, K. Dashtipour, K. Arshad, K. Assaleh, and N. Ramzan, “A hybrid posture detection framework:Integrating machine learning and deep neural networks,”IEEE Sensors Journal, vol. 21, no. 7, pp. 9515–9522,2021.

12. S. Kothari, “Yoga pose classification using deep learning,” 2020.

13. T. K. K. Maddala, P. Kishore, K. K. Eepuri, and A. K. Dande, “Yoganet: 3-d yoga asana recognition using joint angular displacement maps with convnets,”IEEE Transactions on Multimedia, vol. 21, no. 10,pp. 2492–2503, 2019.

14. G. G. Chiddarwar, A. Ranjane, M. Chindhe, R. Deodhar, and P. Gangamwar, “Ai-based yoga pose estimation for android application,” Int J Inn Scien Res Tech, vol.5,pp. 1070–1073, 2020

15. xiang, X. Wang, R. Li, G. Zhang, J. Lai, and Q.Hu, “Fruit image classification based on mobilenetv2 with transfer learning technique”.