# WORK-OUT GUIDE AND CLASSIFIER

**A Project Report**

Submitted by

**MELVIN JAMES K** **JEC20AD030**

to

**APJ Abdul Kalam Technological University**

*in partial fulfillment of the requirements for the award of the Degree of*

**Bachelor of Technology (B.Tech)**

in

**ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

Under the guidance of

**MS. PARVATHY JYOTHI**

CREATING TECHNOLOGY
LEADERS OF TOMORROW
ESTD 2002

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

Jyothi Engineering College
NAAC Accredited College with NBA Accredited Programmes*
Approved by AICTE & affiliated to APJ Abdul Kalam Technological University
A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR
JYOTHI HILLS, VETTIKATTIRI P.O, CHERUTHURUTHY, THRISSUR. PIN-679531 PH : +91- 4884-259000, 274423 FAX : 04884-274777
NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

**June 2024**

# DECLARATION

I hereby declare that the project report “ WORKOUT GUIDE AND CLASSIFIER”, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Ms. PARVATHY JYOTHI. This submission represents the ideas in my own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in this submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously used by anybody as a basis for the award of any degree, diploma or similar title of any other University.

| **Name of Student** | **Signature** |
| --- | --- |
| MELVIN JAMES K (JEC20AD030) | |

**Place:**

**Date:**

# ACKNOWLEDGEMENT

I take this opportunity to thank everyone who helped us profusely, for the successful completion of my project work. With prayers, I thank **God Almighty** for his grace and blessings, for without his unseen guidance, this project would have remained only in our dreams.

I thank the **Management** of Jyothi Engineering College and the Principal, **Dr. JOSE P THERATTIL** for providing all the facilities to carry out this project work. I am greatful to the Head of the Department **Prof. BINEESH M** for his valuable suggestions and encouragement to carry out this project work.

I would like to express my whole hearted gratitude to the project guide **Mrs. PARVATHY JYOTHI** for his encouragement, support and guidance in the right direction during the entire project work.

I thank my Project Coordinators **Ms. PARVATHY JYOTHI** & **Ms. SEENIA FRANCIS** for their constant encouragement during the entire project work. I extend our gratefulness to all teaching and non teaching staff members who directly or indirectly involved in the successful completion of this project work.

Finally, I take this opportunity to express my gratitude to the parents for their love, care and support and also to our friends who have been constant sources of support and inspiration for completing this project work.

MELVIN JAMES K ( JEC20AD030)

## VISION OF THE INSTITUTE

Creating eminent and ethical leaders through quality professional education with emphasis on holistic excellence.

## MISSION OF THE INSTITUTE

- To emerge as an institution par excellence of global standards by imparting quality Engineering and other professional programmes with state-of-the-art facilities.
- To equip the students with appropriate skills for a meaningful career in the global scenario.
- To inculcate ethical values among students and ignite their passion for holistic excellence through social initiatives.
- To participate in the development of society through technology incubation, entrepreneurship and industry interaction.

# VISION OF THE DEPARTMENT

Creating ethical leaders in the domain of Artificial intelligence and data Science through effectual teaching and learning process to develop emerging technology solutions for the benefits of industry and society with a focus on holistic learning and excellence.

# MISSION OF THE DEPARTMENT

- Strengthening basic competencies in the domains of Artificial Intelligence and Data Science.
- Providing high-quality, value-based technical education and developing technology professionals with creative ideas and compelling leadership abilities.
- Using logical thinking to create and develop cutting-edge products in collaboration with industry stakeholders in order to meet global expectations and requirements.
- Enabling graduates to adapt to new technologies via strong fundamentals and lifetime learning.

# PROGRAMME EDUCATIONAL OBJECTIVES

**PEO 1:** To disseminate in-depth technical knowledge in the field of artificial intelligence.

**PEO 2:** To gain a broad grasp of computer science and engineering at many abstraction levels, including computer architecture and design, operating systems, database management, algorithms, and applications.

**PEO 3:** To provide students with a solid foundation in math and engineering foundations, which will enable them to examine and assess real-world engineering challenges connected to data science and artificial intelligence, as well as to further prepare them for further education and R&D.

**PEO 4:** To inspire students, a desire to learn for the rest of their lives and to make them aware of their professional and societal responsibilities.

**PEO 5:** To inculcate in students an awareness of how to use their computer engineering and mathematical theory skills to address current and future computing challenges.

# PROGRAMME SPECIFIC OUTCOMES

The students upon completion of Programme, will be able: -

**PSO 1:** Understand and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics and networking by identifying, demonstrating and analyzing the knowledge of engineering in efficient design of computer-based systems of varying complexity.

**PSO 2:** Applying algorithmic principles, innovative Computer science and engineering design and implementation skills to propose optimal solutions to complex problems by choosing a better platform for research in AI and data science.

**PSO 3:** Identify standard Software Engineering practices and strategies by applying software project development methods using open-source programming environment to design and evaluate a quality product for business success.

**PSO 4:** Demonstrate and examine basic understanding of engineering fundamentals, professional/social ethics and apply mathematical foundations to design and solve computational problems.

# PROGRAMME OUTCOMES

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

# COURSE OUTCOMES

| COs | Description |
|---|---|
| CO.1 | Identify technically and economically feasible problems of social relevance. |
| CO.2 | Identify and survey the relevant literature for getting exposed to related solutions. |
| CO.3 | Perform requirement analysis and identify design methodologies and develop adaptable and reusable solutions of minimal complexity by using modern tools and advanced programming techniques. |
| CO.4 | Prepare technical report and deliver presentation. |
| CO.5 | Apply engineering and management principles to achieve the goal of the project. |

# CO MAPPING TO POs

| | POs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **COs** | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO.1 | 3 | 3 | 3 | 3 | 0 | 3 | 3 | 2 | 3 | 3 | 3 | 3 |
| C0.2 | 2 | 1 | 3 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 2 | 2 |
| C0.3 | 3 | 1 | 2 | 1 | 1 | 1 | 2 | 3 | 3 | 3 | 3 | 3 |
| C0.4 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 1 | 2 | 1 | 2 | 2 |
| C0.5 | 3 | 2 | 3 | 2 | 2 | 3 | 3 | 2 | 3 | 0 | 2 | 2 |
| **Average** | 2.6 | 1.8 | 2.6 | 2 | 1.4 | 1.8 | 1.6 | 2 | 2.6 | 1.8 | 2.4 | 2.4 |

# CO MAPPING TO PSOs

| | PSOs | | | |
|---|---|---|---|---|
| **COs** | PSO1 | PSO2 | PSO3 | PSO4 |
| CO.1 | 3 | 3 | 3 | 3 |
| C0.2 | 3 | 3 | 2 | 3 |
| CO.3 | 2 | 2 | 1 | 3 |
| CO.4 | 3 | 2 | 3 | 1 |
| CO.5 | 3 | 2 | 3 | 1 |
| **Average** | 2.8 | 2.4 | 2.4 | 2.2 |

# ABSTRACT

This study presents a CNN model based on computer vision and deep learning techniques for exercise classification, which is part of a proposed project aimed at assisting individuals in achieving a healthy lifestyle. The project utilizes user-provided information such as height, weight, and age to calculate the Body Mass Index (BMI). Based on the calculated BMI, the system suggests four exercises: Bicep curls, squats, push-ups, and sit-ups, each with varying numbers of repetitions.

To improve the accuracy and effectiveness of the workout routine, the project integrates the OpenCV and Mediapipe libraries in Python. These libraries enable pose estimation by tracking landmarks on the human body. By analyzing the angles between these landmarks, the system can determine if the exercise poses are correct or incorrect. During the workout, the proposed model monitors the user's execution of each exercise. If the pose is correct, the system increments the rep count accordingly. This feedback mechanism ensures that users maintain the correct form and maximize the benefits of each exercise.

By leveraging the power of computer vision and machine learning techniques, this workout guide offers personalized exercise recommendations and real-time pose analysis. Users can utilize the system to maintain a healthy lifestyle, gain insight into their exercise routines, and track their progress over time. The CNN model, incorporating the Inception V3 architecture, accurately classifies input images into one of four exercise categories, namely squats, push-ups, sit-ups, and curls, with high classification accuracy. The integration of OpenCV and Mediapipe libraries further enhances the system's capabilities by providing real-time pose analysis and feedback during the workout. Overall, this project combines computer vision, deep learning, and user-specific information to create a comprehensive workout guide with potential applications in exercise tracking, fitness monitoring, and personalized workout guidance.

# CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

The project aims to develop a workout guide that assists individuals in achieving a healthy lifestyle by providing personalized exercise recommendations and real-time pose analysis. It utilizes computer vision and deep learning techniques to classify exercises and ensure proper execution. The system begins by calculating the user's Body Mass Index (BMI) based on their height, weight, and age. Using this information, it suggests four exercises: Bicep curls, squats, push-ups, and sit-ups, with varying repetitions. To enhance accuracy, the project integrates the OpenCV and Mediapipe libraries, enabling pose estimation by tracking landmarks on the human body. By analyzing the angles between these landmarks, the system can determine if the exercise poses are correct. During the workout, the model monitors the user's execution and provides feedback on their form. This feedback mechanism helps users maintain the correct posture and maximize exercise benefits. Overall, the project combines computer vision, deep learning, and user-specific information to create a comprehensive workout guide with potential applications in exercise tracking, fitness monitoring, and personalized workout guidance.

## 1.2 Objectives

The objective of the model is to create a comprehensive workout guide that combines pose estimation using MediaPipe and OpenCV with an image classifier model based on Inception v3. The model aims to provide a range of functionalities to assist users during their workout sessions. Firstly, the model utilizes MediaPipe and OpenCV to accurately estimate the poses of individuals performing various workouts. By analyzing key body joint positions, it can provide real-time feedback on posture and form during exercises. This feedback helps users maintain proper alignment and reduce the risk of injury while performing different movements. In addition to pose estimation, the model incorporates an image classifier model based on Inception v3. This enables the model to identify and classify different workout exercises. By analyzing input images or video frames, the model can recognize a wide range of workout movements and provide information about the specific exercise being performed. This allows users to receive specific guidance and instructions tailored to their current workout routine. The objective of this model is to combine pose estimation and image classification techniques to create an interactive and informative workout guide. By providing real-time feedback, personalized guidance, and a comprehensive exercise database, the model aims to help users perform exercises correctly, reduce the risk of injury, and achieve their fitness goals effectively.

## 1.3 Organization of the Project

The report is organised as follows:

• Chapter 1: Introduction- Gives an introduction to ”Work out guide”

• Chapter 2: Literature Survey- Summarizes the various existing techniques that helped us in achieving the desired result.

• Chapter 3: Methodology- Methods which are used in this project.

• Chapter 4: Results and Discussion- The results of work and discussion

• Chapter 5: Conclusion & Future Scope- The chapter gives a conclusion of the overall work along with the future scope of implementation.

• Chapter 6: References- Includes the references for the project.

# CHAPTER 2

# LITERATURE SURVEY

### 2.0.1 Pose Estimation and Correcting Exercise Posture

Published by: Rahul Ravikant Kanase et al., UG Student, Department of Electronics Engineering, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai, 400 706, India, in ITM Web of Conferences 40, 03031 (2021) ICACC-2021.

Various exercises such as dead lifts, squats, and shoulder presses are favorable to human body fitness but can also be very harmful if performed improperly. Injuries to the muscles or ligaments can be caused due to the involvement of heavy weights in these exercises. Lack of training or knowledge often leads to incorrect posture while performing these exercises regularly, resulting in muscle fatigue and strain. In this course project, utilizing the latest techniques in pose estimation, we assist individuals in performing exercises with the correct posture. Our project detects the user's pose while exercising, provides feedback, and suggests improvements if necessary. The objective of this project is to prevent injuries and enhance the form of human workout using just a computer and a camera. Initially, human pose estimation, a highly applicable domain of computer vision, is employed. A trained model identifies a person's joints as a list of skeletal key points from the provided data, which can be an RGB image or a depth map. Pose estimation plays a crucial role in solving problems related to human detection and activity recognition, as well as complex problems involving movement detection. OpenPose, which utilizes neural networks for inference, is employed in this project. The latter part of our project involves assessing the quality of human pose for a given exercise, employing heuristic-based and machine learning models. The complete application comprises two main components, which can capture a video of an exercise and provide feedback to the user.

### 2.0.2 DeepPose: Human Pose Estimation via Deep Neural Networks

Published by: Alexander Toshev et al., Google, 1600 Amphitheatre Pkwy, Mountain View, in CVPR 2014G, 20 Aug 2014.

The abstract introduces a proposed method for human pose estimation using Deep Neural Networks (DNNs). The primary objective of this research is to address the challenge of accurately estimating human poses by formulating it as a regression problem solved by DNNs. By leveraging recent advancements in Deep Learning, the authors aim to develop a powerful and efficient approach that provides high precision pose estimates.

The proposed method employs a cascade of DNN regressors, which allows for a holistic understanding of human poses. This cascade architecture enhances the precision of pose estimation by sequentially refining the initial estimates. The formulation of the approach is designed to take advantage of the capabilities of Deep Learning, thereby capitalizing on its ability to learn intricate patterns and representations from data.

To evaluate the effectiveness of the proposed method, the authors conducted a comprehensive empirical analysis. The evaluation involved four academic benchmarks comprising diverse real-world images. The results obtained from these benchmarks demonstrate the state-of-the-art performance achieved by the proposed approach, surpassing or at least equaling the performance of existing methods.

The significance of this research lies in its contribution to advancing the field of human pose estimation. By leveraging Deep Neural Networks and developing a cascade of DNN regressors, the proposed method offers a robust and efficient solution for accurately estimating human poses. The empirical analysis validates the effectiveness of the approach, reinforcing its potential to outperform existing methods. The findings of this research have implications for various applications, including human-computer interaction, motion analysis, and augmented reality, where precise pose estimation is crucial.

### 2.0.3 Deep High-Resolution Representation Learning for Human Pose Estimation

Published by: Ke Sun et al., University of Science and Technology of China, Microsoft Research Asia.

This paper focuses on addressing the human pose estimation problem by emphasizing the learning of reliable high-resolution representations. Unlike existing methods that recover high-resolution representations from low-resolution ones generated by a high-to-low resolution network, our proposed network maintains high-resolution representations throughout the entire process.The approach starts with a high-resolution subnetwork as the initial stage and gradually incorporates high-to-low resolution subnetworks, forming multiple stages. These multi-resolution subnetworks are connected in parallel. To enhance the richness of high-resolution representations, the network employs repeated multi-scale fusions. This iterative fusion process ensures that each high-to-low resolution representation receives information from other parallel representations, leading to more accurate and spatially precise predicted keypoint heatmaps.

The effectiveness of the proposed network is empirically demonstrated through superior pose estimation results on two benchmark datasets: the COCO keypoint detection dataset and the MPII Human Pose dataset. The experimental evaluations showcase the advantages of our network in terms of accuracy and spatial precision compared to other existing methods. Furthermore, the paper showcases the superiority of our network in the context of pose tracking using the PoseTrack dataset. This highlights the versatility and robustness of the proposed approach in handling dynamic scenarios and tracking human poses over time.

Overall, the research presented in this paper introduces a novel network architecture that maintains high-resolution representations throughout the pose estimation process. The empirical evaluations on benchmark datasets demonstrate the effectiveness and superiority of our approach, paving the way for advancements in human pose estimation techniques.

### 2.0.4 Comparative Study Of Human Pose Estimation Models

Published by: Rahul Pradhan et al., Dept. of Computer Engineering and Applications, GLA University Mathura, India, in International Conference on Innovative Advancement in Engineering and Technology (IAET-2020).

Human pose estimation is the process of localizing human joints to depict the configuration of a person's body parts in an image or video. It can be performed in two-dimensional (2D) or three-dimensional (3D) form. In 2D pose estimation, (x, y) coordinates are estimated for each joint in an RGB image, while in 3D pose estimation, (x, y, z) coordinates are estimated. Additionally, there are single person pose estimation and multi-person pose estimation, depending on the number of individuals in a frame.

Various applications have emerged based on human pose recognition, such as activity recognition, motion capture and augmented reality, training robots, and motion tracking for consoles, among others. However, researchers have faced several challenges that have been partially addressed but high efficiency has not yet been achieved. Some significant challenges include diversification in the appearance of human visuals, alterations in illumination, variability in human configurations, partial obstructions, complexity of human skeletal structure, high dimensionality of the pose, and loss of information when observing 3D images from 2D planar image projections.

Numerous models have been developed to handle annotated images, which may not be easily visualized or understood. Despite the challenges in this technology, many researchers have made progress and obtained favorable results.

### 2.0.5 Human activity recognition by using convolutional neural network

Published by: Hankil Kim et al., Department of Computer Engineering, PaiChai University, South Korea, in International Journal of Electrical and Computer Engineering (IJECE) Vol. 9, No. 6, December 2019.

The HAR system, a widely used pattern recognition system [1-3], consists of several modules such as sensing, feature extraction, classification, segmentation, and post-processing [4]. HAR systems can be categorized into two types: time-based and acceleration-based. Acceleration-based methods require the use of multiple accelerometers for data collection, while time-based methods typically rely on one or more cameras. However, the acceleration method can cause discomfort during activities such as walking, running, and lying down. In this study, various human activities, including hand waving, punching, kicking, lying down, walking, running, and standing, are monitored. A vision-based system offers the advantage of not requiring sensors to be attached to the body. However, recognition performance is influenced by lighting conditions, viewing angles, and other factors. To address these challenges, this paper proposes a system that utilizes a time-based dataset [5-8] captured by a thermal camera [9,10] and a CNN structure [11-14]. This approach reduces the manual processing steps and improves accuracy.

### 2.0.6 Deep Recurrent Neural Networks for Human Activity Recognition

Published by: Abdulmajid Murad et al., Department of Information, Chosun University, Korea, in November 2017.

Adopting deep learning methods for human activity recognition has been effective in extracting discriminative features from raw input sequences acquired from body-worn sensors. However, typical machine learning methods do not fully exploit the temporal correlations present in the sequential nature of human movements. Convolutional neural networks (CNNs) address this limitation by using convolutions across a one-dimensional temporal sequence to capture dependencies between input data samples. Nevertheless, the size of convolutional kernels restricts the range of dependencies captured, making typical models inflexible for various activity recognition configurations and requiring fixed-length input windows.

In this paper, the authors propose the use of deep recurrent neural networks (DRNNs) to build recognition models capable of capturing long-range dependencies in variable-length input sequences. They present unidirectional, bidirectional, and cascaded architectures based on long short-term memory (LSTM) DRNNs and evaluate their effectiveness using diverse benchmark datasets. The experimental results demonstrate that the proposed models outperform conventional machine learning methods such as support vector machines (SVM) and k-nearest neighbors (KNN). Furthermore, the proposed models exhibit superior performance compared to other deep learning techniques like deep belief networks (DBNs) and CNNs.

### 2.0.7 Human Activity Recognition Using Tools of Convolutional Neural Networks

Published by: Md. Milon Islam et al., Centre for Pattern Analysis and Machine Intelligence, Department of Electrical and Computer Engineering, University of Waterloo, Canada; Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates; Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; Center of Smart Robotics Research, King Saud University, Riyadh, Saudi Arabia.

Human Activity Recognition (HAR) plays a significant role in the everyday life of people because of its ability to learn extensive high-level information about human activity from wearable or stationary devices. A substantial amount of research has been conducted on HAR, and the research community has explored numerous approaches based on deep learning and machine learning for classifying human activities. The main goal of this review, conducted by Md. Milon Islam et al., is to summarize recent works based on a wide range of deep neural network architectures, particularly convolutional neural networks (CNNs), for human activity recognition.

The reviewed systems are clustered into four categories based on the use of input devices: multimodal sensing devices, smartphones, radar, and vision devices. The review provides descriptions of the performances, strengths, weaknesses, and the hyperparameters used in CNN architectures for each reviewed system, along with an overview of available public data sources. Additionally, the review discusses the current challenges faced by CNN-based HAR systems. Finally, potential future directions in this field are presented to assist researchers who wish to contribute to this area of study.

### 2.0.8 Sports Recognition using Convolutional Neural Network

Published by: Shakil Ahmed Reja et al., Faculty of Science  Engineering, International Islamic University Chittagong (IIUC), Kumira, Chittagong-4318, Bangladesh.

This research paper addresses the challenging task of automated image and video recognition of sports, specifically classifying images and video clips into specific sports categories. The authors propose a sports detection system that utilizes a deeper Convolutional Neural Network (CNN) model with a combination of a fully connected layer and fine-tuning. The system aims to classify five distinct sports groups using images and videos.

To construct the sports detection system, the authors employ two pre-trained deep CNN models, namely Extended ResNet50 and VGG16, which are adapted for video classification based on images. The authors utilize three optimizers (RMSProp, ADAM, and SGD) to train the extended CNN models. The models are trained for five epochs using a carefully curated dataset consisting of thousands of sports images collected from the internet.

The experimental results demonstrate promising outcomes. The ResNet50 model, trained with the SGD optimizer, achieves an accuracy of approximately 83 percent for classifying five sports classes, and an impressive accuracy of 95 percent for three sports classes. These results highlight the effectiveness of the proposed sports detection system in accurately categorizing different types of sports based on visual content.

In summary, this paper presents a sports detection system that utilizes a deeper CNN model combined with a fully connected layer and fine-tuning. The system achieves significant accuracy in classifying sports images and videos, showcasing its potential for automated sports recognition. The use of pre-trained models, careful dataset selection, and optimizer evaluation contribute to the successful classification performance of the proposed system.

### 2.0.9 Two-Stream Convolutional Networks for Action Recognition in Videos

Published by: Karen Simonyan et al., Visual Geometry Group, University of Oxford.

In this research study, the authors investigate the architectures of deep Convolutional Networks (ConvNets) for action recognition in videos. The main objective is to effectively capture both appearance information from still frames and motion information between frames, while leveraging a data-driven learning approach to improve performance compared to hand-crafted features.

The paper presents three key contributions. Firstly, the authors propose a two-stream ConvNet architecture that combines spatial and temporal networks. This architecture enables the model to leverage both visual appearance and motion cues, capturing complementary information. Secondly, the authors demonstrate the effectiveness of training a ConvNet on multi-frame dense optical flow, even with limited training data. This approach allows the model to capture temporal dynamics efficiently. Finally, the authors employ multi-task learning by utilizing two different action classification datasets. This technique increases the amount of available training data and enhances performance on both datasets.

To evaluate the proposed architecture, the authors conduct experiments on well-established video action recognition benchmarks, including UCF-101 and HMDB-51. The results demonstrate that their architecture achieves competitive performance compared to state-of-the-art methods and significantly outperforms previous attempts to use deep nets for video classification.

Overall, this research study introduces a novel two-stream ConvNet architecture for action recognition in videos. By incorporating spatial and temporal networks, leveraging multi-frame dense optical flow, and employing multi-task learning, the proposed approach demonstrates strong performance even with limited training data. The experimental evaluation on standard benchmarks validates the effectiveness of the proposed architecture, surpassing previous deep net-based approaches in video classification.

### 2.0.10 Continuous Human Action Recognition for Human-Machine Interaction

Published by : Harshala Gammulle, David Ahmedt-Aristizabal, Simon Denman, Lachlan Tychsen-Smith, Lars Petersson, Clinton Fookes, Cornell university, Ithaca, Newyork. This paper addresses the challenges of action recognition and action transition detection in video streams, which are crucial tasks for real-time human-machine interaction applications. With the advancements in data-driven machine learning research, various models have been proposed to capture spatio-temporal features for video analysis. The authors conduct a comprehensive review of recent literature in the field, examining a wide range of action segmentation methods. They thoroughly analyze, explain, and compare these methods while providing insights into the feature extraction and learning strategies employed in state-of-the-art approaches. The paper also discusses the influence of object detection and tracking techniques on human action segmentation methodologies, highlighting their impact on performance. Furthermore, the authors investigate the application of these models in real-world scenarios, considering the practical implications and challenges involved. They delve into the limitations of existing approaches and identify key research directions for enhancing interpretability, generalization, optimization, and deployment of action segmentation models. By thoroughly analyzing the existing literature and discussing various aspects related to action segmentation in video streams, this paper provides a comprehensive overview of the field. It highlights the importance of feature extraction and learning strategies, as well as the impact of object detection and tracking techniques. Moreover, the paper sheds light on the practical application of these models and outlines future research directions to overcome limitations and improve the interpretability and performance of action segmentation methods.

# CHAPTER 3

# METHODOLOGY

## 3.1 Existing Systems

- The proposed model is a workout guide that utilizes computer vision and deep learning techniques for exercise classification. It offers personalized exercise recommendations and real-time pose analysis to assist individuals in achieving a healthy lifestyle. The model incorporates user-provided information such as height, weight, and age to calculate the Body Mass Index (BMI) and suggests exercises accordingly. By integrating the OpenCV and Mediapipe libraries, the system enables pose estimation and analyzes the correctness of exercise poses. During the workout, the model monitors the user's execution, provides real-time feedback, and adjusts the rep count for correct poses. This comprehensive approach combines computer vision, deep learning, and user-specific data to create an efficient workout guide. Users can track their progress and gain insights into their exercise routines, promoting a healthy lifestyle. The model's accuracy is enhanced by the Inception V3 architecture, which accurately classifies input images into four exercise categories: squats, push-ups, sit-ups, and bicep curls.

### 3.1.1 Disadvantages of existing systems

- They lack accurate real-time feedback and posture correction, which can compromise exercise effectiveness and increase the risk of injury.
- They don't offer personalized training plans and they often fail to consider individual preferences, goals, and limitations. This lack of customization can hinder motivation and hinder long-term adherence.
- They lack a system to identify and classify the exercises.

## 3.2 Problem Statement

- To Develop a Work Out guide that implement posture detection technol- ogy to improve the user's posture during the workout and achieve per- sonal fitness goals based on BMI
- To create an image classifier which identifies which exercise is being performed and classifies it into squats, push-ups, sit-ups, and curls respectively

## 3.3 Proposed System

The proposed workout guide system utilizes Mediapipe and OpenCV libraries to estimate the user's posture during exercise routines. By inputting height, weight, and age, the system calculates the Body Mass Index (BMI) and suggests personalized exercises accordingly. The BMI is calculated using the equation:

$$BMI = \frac{weight(kg)}{height^2(m^2)}$$

During workouts, the system tracks and analyzes body landmarks in real-time to determine the accuracy of exercise execution. If the posture is correct, the system provides feedback and increments the rep count. The user-friendly interface offers personalized training plans based on BMI and preferences, enhancing the overall experience. By incorporating posture estimation, the system aims to promote proper form, reduce injury risks, and optimize exercise effectiveness. Users can improve their fitness levels and achieve wellness goals with this comprehensive workout guide.

## 3.4 Modules

### 3.4.1 For Workout guide

**User Information Collection**

The User Information Module serves as the initial step of the workout guide, gathering essential data from users, including their height, weight, and age. This information is crucial for creating a personalized fitness experience. Once the user provides the necessary details, the module proceeds to calculate their Body Mass Index (BMI). By employing the collected data, this module employs a specific formula to determine the BMI, which is a valuable metric for evaluating an individual's body composition and overall health. The calculated BMI serves as a foundation for tailoring exercise recommendations and designing workout routines that align with the user's specific needs and goals..

**Exercise Recommendation**

The Exercise Recommendation Module utilizes the calculated BMI to generate personalized exercise suggestions that cater to the user's specific needs. Drawing upon this information, the module offers tailored recommendations, including exercises such as bicep curls, squats, push-ups, and sit-ups. To optimize the workout routine, the module takes into account the user's BMI and assigns varying numbers of repetitions for each exercise. By customizing the workout plan in this manner, users can effectively target their desired fitness goals while ensuring that the exercise intensity aligns with their individual BMI and capabilities.

#### Pose Estimation

The Pose Estimation Module combines the power of OpenCV and Mediapipe libraries to enable accurate pose estimation during workouts. By tracking specific landmarks on the user's body, such as joints and body segments, this module analyzes exercise poses in real-time. Utilizing the data obtained from these landmarks, the module examines the angles and alignment to determine whether the exercise poses are correct or incorrect. This ensures that users maintain proper form and posture, maximizing the effectiveness and safety of their workout sessions.

#### Real-time Feedback

The Real-time Feedback Module actively monitors the user's performance during each exercise throughout the workout session. It offers valuable real-time feedback on exercise form and the correctness of poses, enabling users to make immediate adjustments and corrections. Additionally, this module incorporates a rep count mechanism that dynamically increments the count when the user maintains the correct form, ensuring accuracy in tracking progress and promoting adherence to proper exercise techniques.

### 3.4.2 For Workout Classifier

#### Data Collection

The dataset comprises four sets of images representing distinct exercises: squats, push-ups, sit-ups, and curls. These images are carefully gathered and organized to provide a diverse representation of each exercise category. The model is trained on this comprehensive dataset to accurately classify input images into one of the four exercise categories. The data collection process ensures the model's effectiveness in identifying the performed exercise or the exercise depicted in the input images.

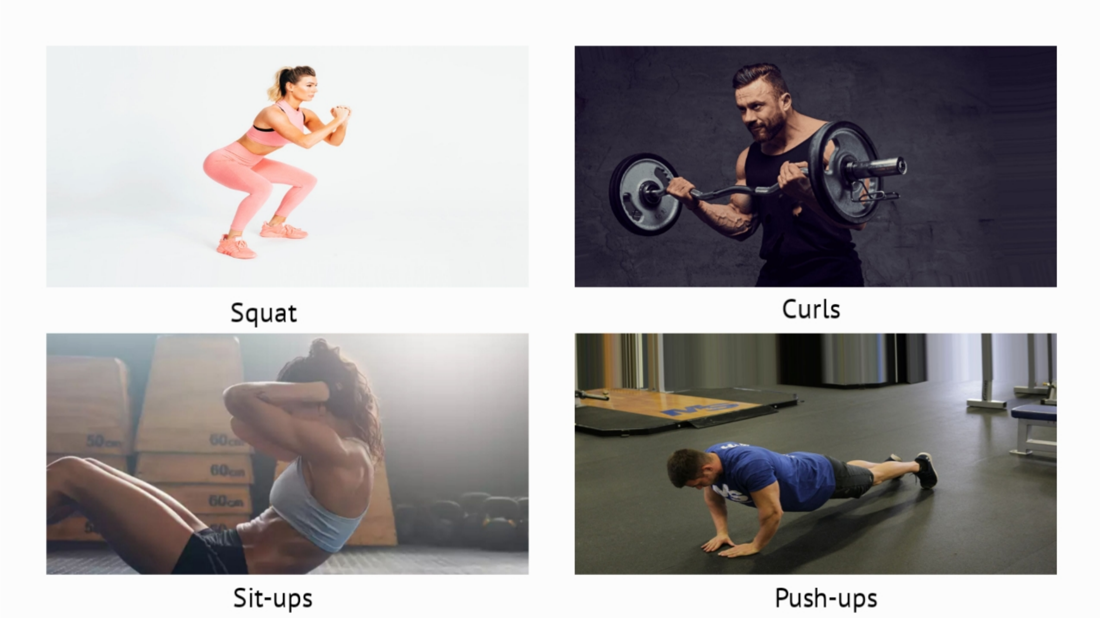



Figure 3.1: Dataset Images

**Data Preprocessing**

The VGG16 model assists in tasks such as resizing, normalization, and feature extraction, enabling the extraction of meaningful features from the images. By utilizing this preprocessing technique, the model can effectively learn and classify input images into their respective exercise categories with high accuracy

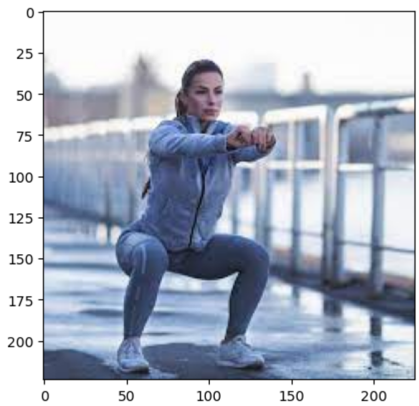



Figure 3.2: Data Preprocessing

**Data Augmentation**

The model employs geometrical augmentation to transform the data. Geometrical augmentation refers to the process of applying various transformations to an image, such as rotation, scaling, translation, and flipping, to create new training data for machine learning models. In Python, you can use libraries like OpenCV or scikit-image to perform geometrical augmentation. This augmentation process aids in enhancing the model's ability to handle variations in lighting, orientation, and other real-world factors. The augmented dataset ensures improved generalization and robustness of the CNN model, leading to accurate classification of input images into the four exercise categories: squats, push-ups, sit-ups, and curls.
The equation for geometric augmentation is:

$$x' = x * r + t$$

where,

- x is the original input
- x' is the augmented input
- r is the geometric scaling factor
- t is the translation vector

The Geometrical image augmentation techniques used in the model includes:

- Image Rotation:
  - Rotate the image by a specified angle in degrees.
  - Functions: cv2.rotate(), PIL.Image.rotate(), skimage.transform.rotate().
- Image Translation:
  - Shift the image horizontally and/or vertically.
  - Functions: cv2.warpAffine(), PIL.Image.transform(), skimage.transform.warp().
- Image Scaling:
  - Rotate the image by a specified angle in degrees.
  - Functions: cv2.resize(), PIL.Image.resize(), skimage.transform.resize().
- Image Flipping:
  - Flip the image horizontally and/or vertically.

  - Functions: cv2.flip(), PIL.Image.transpose(), skimage.transform.flip().

Image Shearing:

- Distort the image by tilting or slanting it along a particular axis.
- Functions: PIL.Image.transform(), skimage.transform.AffineTransform().

**Data Splitting**

The dataset, comprising four sets of images representing squats, push-ups, sit-ups, and curls, is divided into training and validation sets. The training set is utilized to optimize the model's parameters through supervised learning techniques, including the construction of a deep learning architecture. The validation set aids in assessing the model's performance and making necessary adjustments to enhance classification accuracy. Experimental evaluations highlight the effectiveness of the CNN model in accurately identifying the exercise category, making it a valuable tool for exercise tracking, fitness monitoring, and personalized workout guidance.

## 3.5 System requirements and specifications

### 3.5.1 Python 3.10.0

Python is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality, more maintainable code.

### 3.5.2 Jupyter Notebook v4.11

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It is widely used in the field of data science and machine learning for developing and presenting models, conducting experiments, and documenting the entire workflow.

In the context of a model, Jupyter Notebook provides an interactive environment where you can write, execute, and modify code in real-time. It allows you to seamlessly integrate code snippets with explanatory text, visualizations, and mathematical equations, making it easier to understand and communicate the model's concepts and results.

## 3.6 Implementation

1. User Input Collection:
   Collect user-provided information such as height, weight, and age. This data is used to calculate the Body Mass Index (BMI) and personalize the exercise recommendations.

2. BMI Calculation:
   Use the user's height, weight, and age to calculate the BMI. This metric is important for determining the appropriate exercises and their intensities.

3. Pose Estimation:
   Utilize the OpenCV and Mediapipe libraries to perform pose estimation. These libraries track landmarks on the human body by analyzing video frames or images.

4. Pose Analysis:
   Analyze the angles between the tracked landmarks to determine if the exercise poses are correct or incorrect. Compare the detected poses with predefined correct poses for each exercise category.

5. CNN Model Execution:
   Apply the trained CNN model, based on the Inception V3 architecture, to classify input images or video frames into one of the four exercise categories: bicep curls, squats, push-ups, and sit-ups.

6. Repetition Counting:
   Generate a form including the major content of the garbage, location, date, time and user data.
   Submit the form to the respective local authority.

7. Provides Real-time Feedback:
   Provide real-time feedback to the user during the workout. The system does not add reps if their pose is incorrect, allowing the users to adjust and maintain proper form.

8. Personalized Exercise Recommendations
   Based on the calculated BMI, suggest the appropriate exercises and the number of repetitions to the user. These recommendations are tailored to each individual's specific needs and goals.

### 3.6.1 Architecture Diagram

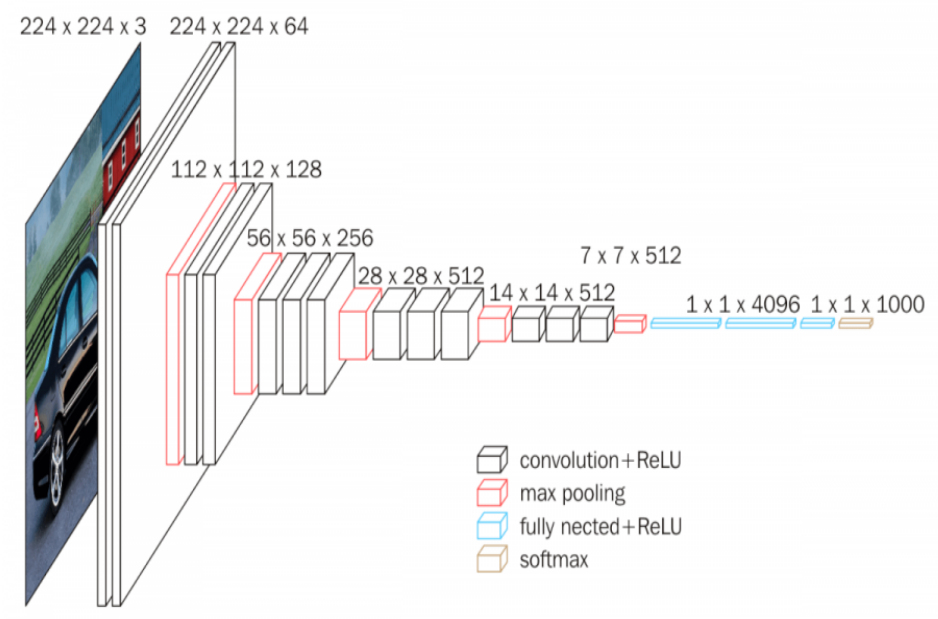



Figure 3.3: Architecture diagram

VGG16 is composed of 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers. Therefore, the number of layers having tunable parameters is 16 (13 convolutional layers and 3 fully connected layers). That is the reason why the model name is VGG16. The number of filters in the first block is 64, then this number is doubled in the later blocks until it reaches 512. This model is finished by two fully connected hidden layers and one output layer. The two fully connected layers have the same neuron numbers which are 4096. The output layer consists of 1000 neurons corresponding to the number of categories of the Imagenet dataset. In the next section, we are going to implement this architecture on Keras. One of the notable contributions of VGG16 is its simplicity and uniformity in architecture. The consistent use of 3x3 filters with stride 1 and padding 1 in the convolutional layers, followed by max-pooling layers, helps to maintain the spatial resolution of the input while reducing the spatial dimensions. The pooling layers progressively downsample the feature maps, capturing higher-level information while retaining important features.

The VGG16 model has been widely used as a feature extractor or a base model in various computer vision tasks. By removing the fully connected layers and using the convolutional layers as feature extractors, VGG16 can generate rich and meaningful representations of images. These feature maps can then be used as input to other models, such as a classifier or a regression network, to solve specific tasks.

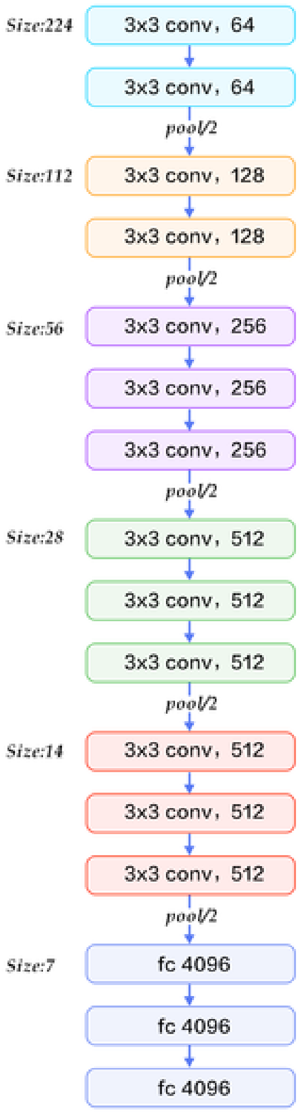



Figure 3.4: Architecture diagram

### 3.6.2 Importing Libraries

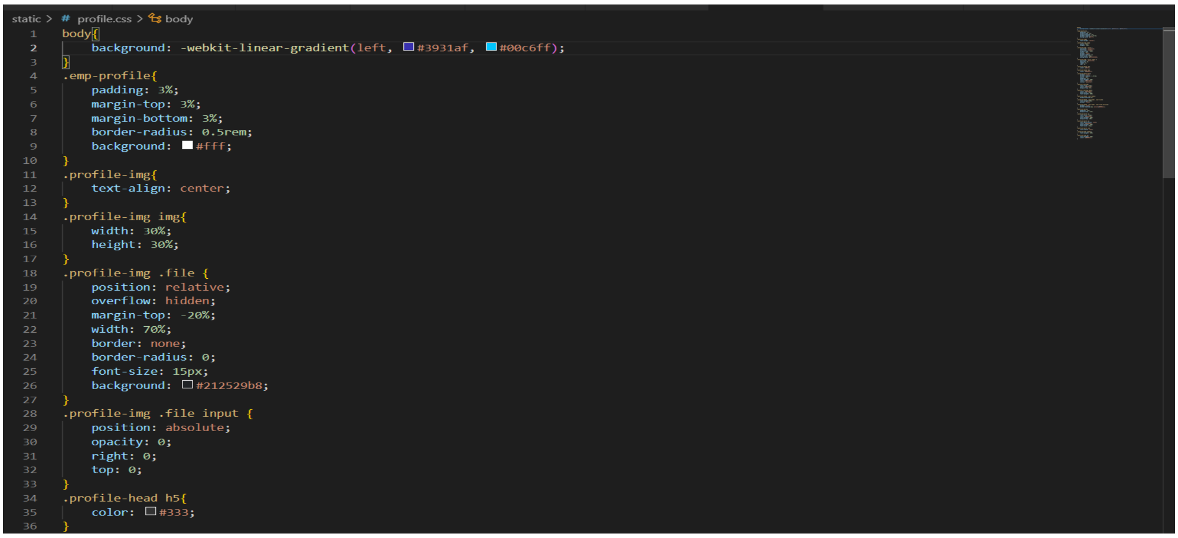
- OS
  - The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The os and os.path modules include many functions to interact with the file system.

- Mediapipe
  - Mediapipe is python library that enables the building of real-time multimedia processing pipelines. It provides a set of components and algorithms which performs video processing, object detection and tracking, facial recognition, and gesture analysis.MediaPipe simplifies the development of complex computer vision and machine learning applications.
  - Media pipe is intended for tracking movements made by a person. On each limb displayed points connected by lines. This line will connect the coordinate points and form a person's frame. The library must be executed in conjunction with the library for the video from the tracking image to be displayed. The angles between these limb points are calculated to find the accuracy of the workout.
- OpenCV
  - Open-Source Computer Vision (OpenCV)is an open-source library widely used for computer vision and image processing tasks. The point is that computers have capabilities similar to visual processing in humans. OpenCV has provided many basic computer vision algorithms and object detection modules using computer vision.zees. OpenCV already has many features, including face recognition, face tracking, face detection, Kalman filtering, and various AI (Artificial Intelligence) methods. OpenCV also provides a variety of simple algorithms related to Computer Vision.
- VGG16
  - The VGG library, by providing VGG models, can play a vital role in classifying images into the four exercise categories mentioned: bicep curls, squats, push-ups, and sit-ups.. It plays a crucial role in feature extraction and classification tasks. By using the learned representations from a large dataset, VGG16 enables the model to extract meaningful features from exercise images. These features are then used to accurately classify the images into different exercise categories. The inclusion of VGG16 eliminates the need for manual feature engineering and enhances the overall performance and accuracy of the exercise classification system.

- ReduceLROnPlateau
  - The ReduceLROnPlateau library is employed in the program to dynamically adjust the learning rate of the deep learning model during training. This library monitors a specified metric, such as the validation loss or accuracy, and reduces the learning rate when the metric plateaus or stops improving. By reducing the learning rate, the model can fine-tune its parameters more effectively and avoid getting stuck in local minima. This adaptive learning rate strategy improves the model's convergence and can lead to better overall performance. The ReduceLROnPlateau library helps optimize the training process and enhances the model's ability to accurately classify exercises.
- EarlyStopping
  - The EarlyStopping library is utilized in the program to prevent overfitting and optimize the training process of the deep learning model. It monitors a specified metric, such as the validation loss, and stops the training if the metric fails to improve for a certain number of consecutive epochs. By stopping the training early, the model avoids unnecessary computation and prevents it from memorizing the training data too well. This technique helps generalize the model's performance on unseen data and improves its ability to accurately classify exercises. The EarlyStopping library contributes to efficient training and ensures the model achieves optimal performance without overfitting.

### 3.6.3 UI

UI is created using Tkinter library for Python. It provides a set of tools and widgets for building desktop applications with graphical interfaces.
User interface simplifies the entire model for the user. Helps the user to use the entire system without knowing how it works.
In the UI there is a space where the data related to the user can be entered and Based on the values of the user's data the BMI is calculated and displayed. And then the camera pop-ups appear which is associated with the 4 exercises namely sit-ups,curls,push-ups,squat.



static > # profile.css > body

```css
body{
    background: -webkit-linear-gradient(left, #3931af, #00c6ff);
}
.emp-profile{
    padding: 3%;
    margin-top: 3%;
    margin-bottom: 3%;
    border-radius: 0.5rem;
    background: #fff;
}
.profile-img{
    text-align: center;
}
.profile-img img{
    width: 30%;
    height: 30%;
}
.profile-img .file {
    position: relative;
    overflow: hidden;
    margin-top: -20%;
    width: 70%;
    border: none;
    border-radius: 0;
    font-size: 15px;
    background: #212529b8;
}
.profile-img .file input {
    position: absolute;
    opacity: 0;
    right: 0;
    top: 0;
}
.profile-head h5{
    color: #333;
}
```

Figure 3.5: CSS code

```css
.profile-head .nav-tabs{
    margin-bottom:5%;
}
.profile-head .nav-tabs .nav-link{
    font-weight:600;
    border: none;
}
.profile-head .nav-tabs .nav-link.active{
    border: none;
    border-bottom:2px solid #0062cc;
}
.profile-work{
    padding: 14%;
    margin-top: -15%;
}
.profile-work p{
    font-size: 12px;
    color: #818182;
    font-weight: 600;
    margin-top: 10%;
}
.profile-work a{
    text-decoration: none;
    color: #495057;
    font-weight: 600;
    font-size: 14px;
}
.profile-work ul{
    list-style: none;
}
.profile-tab label{
    font-weight: 600;
}
.profile-tab p{
    font-weight: 600;
    color: #0062cc;
}
```

Figure 3.6: CSS code

```html
plates > base.html
{%  load static %}
<!DOCTYPE html>

<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
        <meta name="description" content="" />
        <meta name="author" content="" />
        <title>Page Title - SB Admin</title>
        {% comment %} <link rel="stylesheet" type="text/css" href="{% static 'styles.css' %}"> {% endcomment %}
        <link rel="stylesheet" type="text/css" href="{% static 'login1.css' %}">
        <link rel="preconnect" href="https://fonts.gstatic.com">
        <link href="https://fonts.googleapis.com/css2?family=Big+Shoulders+Stencil+Text:wght@700&display=swap" rel="stylesheet">
        <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/js/all.min.js" crossorigin="anonymous"></script>
    </head>
    <body>
    <div class="messages" id="alert-message">
  {% for message in messages %}
  <div class="alert alert-{{ message.tags }} mb-0 alert-dismissible fade show" role="alert">
    <strong>Message :</strong>{{ message }}
    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  {% endfor %}
</div>
 {% block body %}
  {% endblock body %}
  <div id="layoutAuthentication_footer">
                <footer class="py-4 bg-light mt-auto">
                    <div class="container-fluid">
                        <div class="d-flex align-items-center justify-content-between small">
                         <div class="text-muted"></div>
                            <div>
                                <a href="#"></a>
```

Figure 3.7: HTML code

```html
            <span aria-hidden="true">&times;</span>
          </button>
      </div>
      {% endfor %}
   </div>
     {% block body %}
      {% endblock body %}
      <div id="layoutAuthentication_footer">
                    <footer class="py-4 bg-light mt-auto">
                        <div class="container-fluid">
                            <div class="d-flex align-items-center justify-content-between small">
                             <div class="text-muted"></div>
                                <div>
                                    <a href="#"></a>
                                    &middot;
                                    <a href="#">&amp;</a>
                                </div>
                            </div>
                        </div>
                    </footer>
                </div>
            </div>
            <script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>
        <script type="text/javascript">
        $(function(){
        setTimeout(function(){
            $("#alert-message").hide();
            }, 2000);
          });
        </script>

            <script src="https://code.jquery.com/jquery-3.5.1.min.js" crossorigin="anonymous"></script>
            <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js" crossorigin="anonymous"></script>
            <script src="js/scripts.js"></script>
        </body>
    </html>
```

Figure 3.8: HTML code

```javascript
'use strict';
const fs = require('fs');
const upath = require('upath');
const pug = require('pug');
const sh = require('shelljs');
const prettier = require('prettier');

module.exports = function renderPug(filePath) {
    const destPath = filePath.replace(/src\/pug\/\pages/, 'dist').replace(/\.pug$/, '.html');
    const srcPath = upath.resolve(upath.dirname(__filename), '../src');

    console.log(`### INFO: Rendering ${filePath} to ${destPath}`);
    const html = pug.renderFile(filePath, {
        doctype: 'html',
        filename: filePath,
        basedir: srcPath
    });

    const destPathDirname = upath.dirname(destPath);
    if (!sh.test('-e', destPathDirname)) {
        sh.mkdir('-p', destPathDirname);
    }

    const prettified = prettier.format(html, {
        printWidth: 1000,
        tabWidth: 4,
        singleQuote: true,
        proseWrap: 'preserve',
        endOfLine: 'lf',
        parser: 'html',
        htmlWhitespaceSensitivity: 'ignore'
    });

    fs.writeFileSync(destPath, prettified);
};
```

Figure 3.9: Javascript code

# CHAPTER 4

# RESULTS & DISCUSSION

### 4.0.1 Results

#### Accuracy and Loss

The training process aimed to optimize the model parameters to minimize the loss function and maximize the accuracy of the dataset. After performing the training with different numbers of epochs 12 was obtained to be the ideal number of epochs for the model and 32 was found to be the optimal batch size. In this model Testing loss is 0.2872, Testing accuracy is 91.5% Based on the value of training accuracy and validation loss it is evident that the model could classify the data well into four exercises namely situps, pushups, squats, and curls and perform well in generalizing the unseen data. The accuracy above 91% and the least value of the loss suggest that the model is successful in obtaining patterns from the datasets mounted into the model. Making it most suitable for a number of real-world applications.

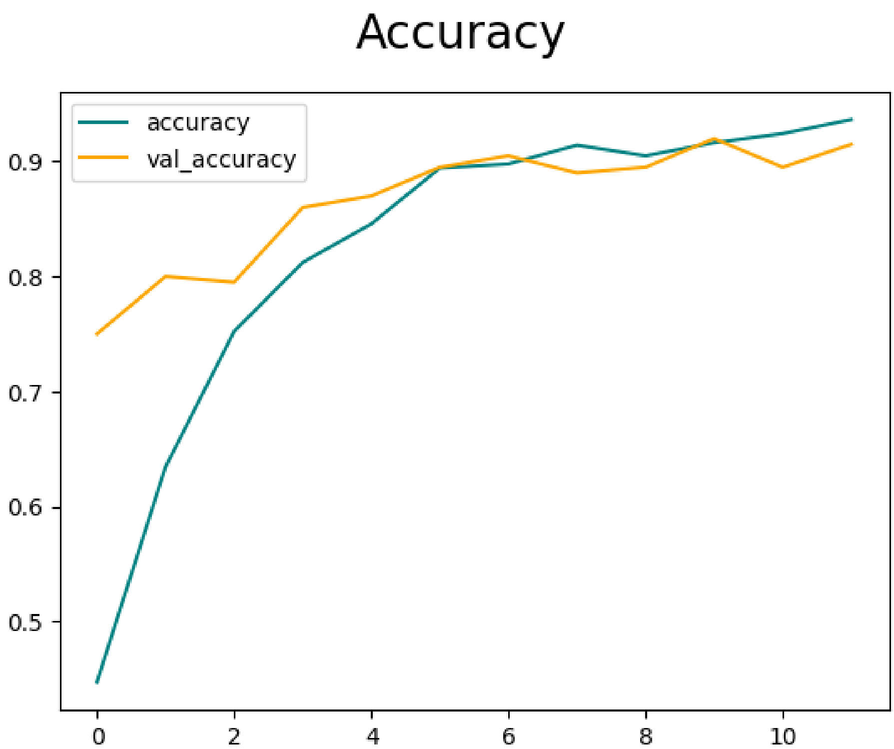



Figure 4.1: Accuracy Graph

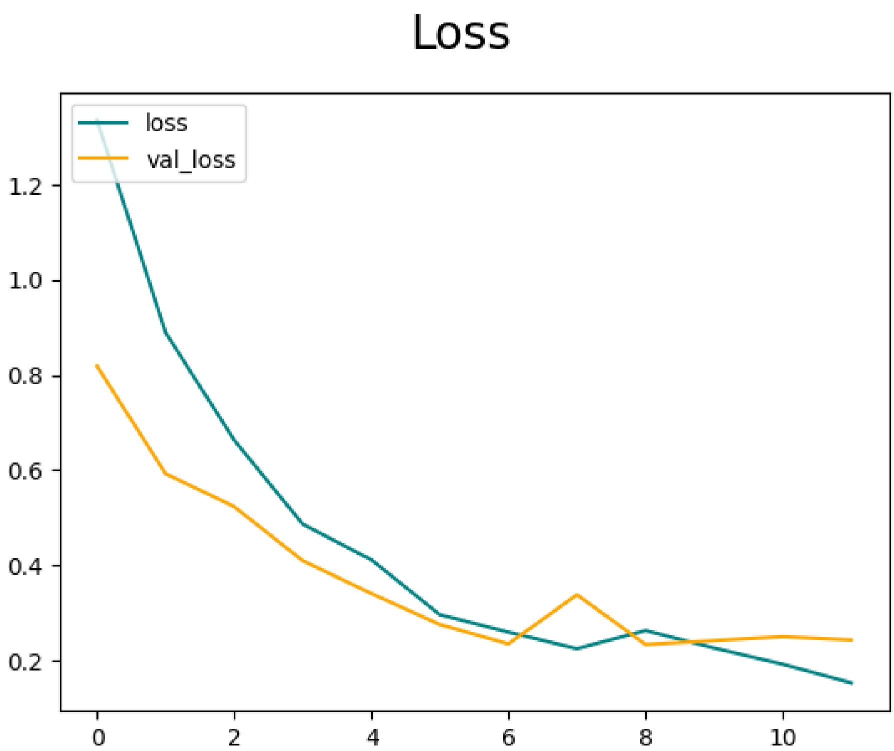



Figure 4.2: Loss Graph

The optimizer used in the model is ADAM.Adam (Adaptive Moment Estimation), is known for its effectiveness in training deep neural networks. Adam combines the benefits of two other optimization algorithms: AdaGrad, which adapts the learning rate individually for each parameter, and RMSProp, which uses exponential moving averages of gradients to adjust the learning rate. The key idea behind Adam is to maintain separate learning rates for each parameter and adapt them based on the magnitude of gradients and the past gradients. The following is the equation for the ADAM optimizer:

$$m_t = beta_1 * m_{t-1} + (1 - beta_1) * g_t$$

$$v_t = beta_2 * v_{t-1} + (1 - beta_2) * g_t^2$$

$$m_{th}at = m_t / (1 - beta_1^t)$$

$$v_{th}at = v_t / (1 - beta_2^t)$$

$$w_t = w_{t-1} - lr * m_{th}at / (sqrt(v_{th}at) +)$$

$where :$

- $m_t istheexponentiallyweightedaverageofthegradientsv_t istheexponentiallyweig$
- $m_{th}atandv_{th}atarebias - correctedversionsofm_t$ and $v_t$
- $w_t istheupdatedweightisthelearningrateg_t isthegradientattimet$
- $beta_1 andbeta_2 arehyperparametersthatcontrolthedecayratesofm_t$ and $v_t$
- (epsilon) is a small constant to prevent division by zero

**Classification Report**

The classification Report is to obtain an overview of the system's performance in classifying the exercises under four classes namely situps, pushups, curls and squats. The report displays the precision, recall,support and F1-score for these four classes

- Precision measures the proportion of correctly predicted samples for each class out of all the samples predicted as that class. In the given model the weighted average precision across all classes is 0.92.

  Precision = True Positives / (True Positives + False Positives)

- Recall measures the proportion of correctly predicted samples for each class out of all the samples that actually belong to that class. In the given model the weighted average recall across all classes is 0.92.

  Recall = True Positives / (True Positives + False Negatives)

- F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's accuracy. In the given model the weighted average F1 score across all classes is 0.91.

  F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

- Support represents the number of samples in each class in the test data set. The support ranges from 99 to 109 instances per class.

  Support = True Positives + False Negatives

| Classes | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| curl | 0.98 | 0.86 | 0.92 | 109 |
| pushup | 0.92 | 0.94 | 0.93 | 99 |
| situp | 0.93 | 0.87 | 0.9 | 86 |
| squats | 0.85 | 0.98 | 0.91 | 106 |
| | | | | |
| accuracy | | | 0.92 | 400 |
| macro avg | 0.92 | 0.91 | 0.91 | 400 |
| weighted avg | 0.92 | 0.92 | 0.91 | 400 |

Figure 4.3: Classification Report

### Confusion Matrix

The Confusion Matrix summarizes the performance of the classification model by showing the number of instances that were correctly or incorrectly classified for each class. In this case, the confusion matrix would be a 4x4 matrix representing the four classes namely curls,situps,pushups,squats.It calculates the number of true positive, false positive, true negative, and false negative predictions for each classes.This visual representation of the model is used to obtain the quick overview of the model's performance. And this representation helps to obtain the region of confusuion in the dataset and to understand the region where the model tends to struggle.

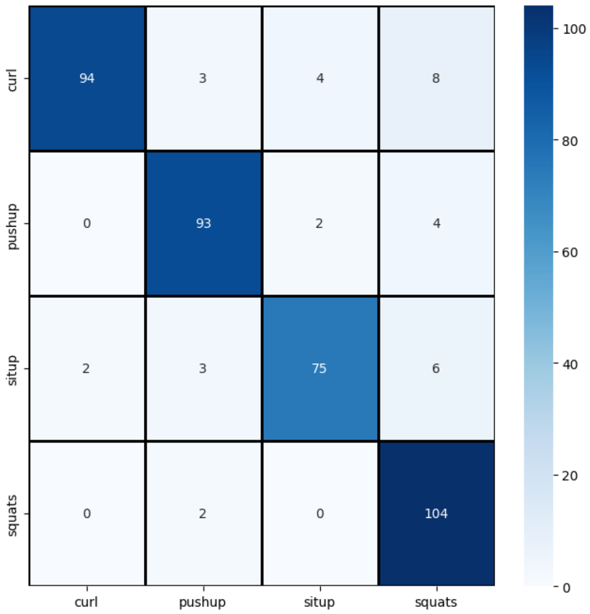



Figure 4.4: Confusion Matrix

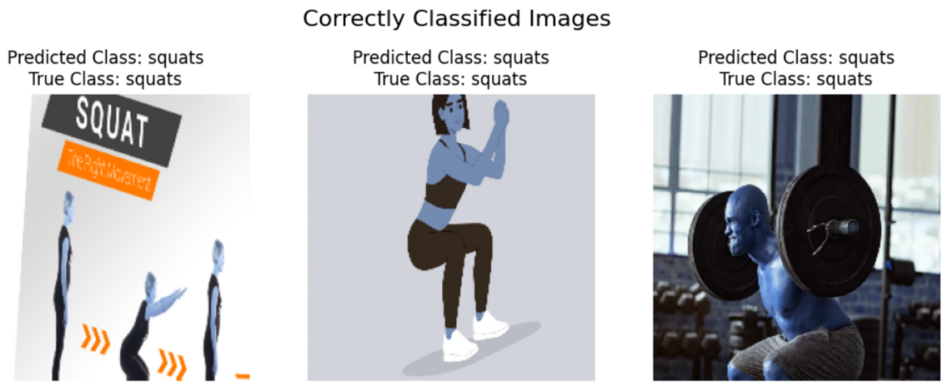



Figure 4.5: Correctly Classiified


Figure 4.6: Incorrectly Classified

### 4.0.2 Discussion

On performing an analysis on the model we could observe that the model classifies the distinct exercises correctly and also performs the estimation of poses on the human body. On comparing the results using the accuracy and loss of the system, classification report and confusion matrix,it is very clear that the model could be employed for performing this classification and estimation of poses accurately and it also gives feedback so that the posture of the person could be corrected. The precision, recall, and F1-score metrics indicate the system's ability to correctly classify the exercises into 4 states which include situps,pushups,curls and squat with higher degree of accuracy and consistency. The balanced macro average further validates the system's performance across both classes.

The UI of the model is very basic and user friendly and helps in avoiding unnecessary complexities and providing accurate results . The integration of real-time monitoring with immediate feedback enhances the effectiveness of the exercises, avoids the risks and chances of accidents related to these exercises and helps the users in maintaining a healthy lifestyle by focusing on their long term goals.

Even though the system is high on accuracy and has least chance of inaccurate results,the models leaves behind rooms for improvement which includes addition of more exercises to the model,and inclusion of other health related aspects including diet,medical history etc makes the model more personalised and complete. And training of the model with more diverse data sets helps the model to generalize better under all the situation which leads to the increased accuracy in whatever results it produces. The successful implementation of this model hence reduced the distance to various applications in the future which includes wearable technology and sensors, sports training and coaching, therapies and rehabilitation. And in a long run it helps in maintaining the quality of human life.

# CHAPTER 5

# CONCLUSION & FUTURE SCOPE

## 5.1 Conclusion

In conclusion, this project presents a CNN model that utilizes computer vision and deep learning techniques to create a comprehensive workout guide aimed at assisting individuals in achieving a healthy lifestyle. By integrating user-provided information, such as height, weight, and age, the system calculates the Body Mass Index (BMI) and suggests personalized exercises, including bicep curls, squats, push-ups, and sit-ups, with varying repetitions.

To enhance the accuracy and effectiveness of the workout routine, the project incorporates the OpenCV and Mediapipe libraries for pose estimation. This enables real-time analysis of exercise poses by tracking landmarks on the human body and analyzing their angles. The model actively monitors the user's execution of each exercise, providing valuable feedback and incrementing the rep count when the pose is correct, ensuring proper form and maximizing the benefits of each exercise.

Through the integration of computer vision, deep learning, and user-specific information, the workout guide offers personalized exercise recommendations and real-time pose analysis. Users can rely on the system to maintain a healthy lifestyle, gain insights into their exercise routines, and track their progress over time. With the CNN model leveraging the Inception V3 architecture, accurate exercise classification is achieved, while the integration of OpenCV and Mediapipe libraries enhances the system's capabilities. Overall, this project presents a promising approach with potential applications in exercise tracking, fitness monitoring, and personalized workout guidance..

## 5.2 Future Scope

In the future, the application of the proposed model and project could extend beyond exercise tracking, fitness monitoring, and personalized workout guidance. The integration of computer vision and deep learning techniques, along with user-specific information, opens up possibilities for broader applications. For instance, the model could be utilized in physical therapy and rehabilitation settings to assess and monitor patient movements and exercise adherence. By providing real-time pose analysis and feedback, the system can assist individuals in maintaining correct form and preventing injuries during rehabilitation exercises. Additionally, the model could be integrated into smart gym equipment or home fitness systems, offering real-time guidance and monitoring to users during their workouts. This would enhance the overall workout experience and ensure optimal performance.Integrating wearable technology or sensors could provide real-time feedback on additional parameters such as heart rate, calorie burn, or muscle activation, further personalizing the workout experience. Furthermore, the model's ability to accurately classify exercise categories could be harnessed in sports training and coaching, where athletes' movements and techniques can be analyzed for improvement. Overall, the future applications of the proposed model hold potential in various fields related to health, wellness, rehabilitation, and sports performance enhancement.

# REFERENCES

[1] https://www.geeksforgeeks.org/python-opencv-pose-estimation/

[2] https://developers.google.com/mediapipe/

[3].https://docs.python.org/3/library/tkinter.html/

[4] https://pub.towardsai.net/keras-earlystopping-callback-to-
-the-neural-networks-perfectly-2a3f865148f7/

[5]https://keras.io/api/callbacks/reduce$_l r_o n_p$lateau/

[6]https : //medium.com/mlearning − ai/an − overview − of − vgg16
− and − nin − models − 96e4bf398484/

[7]https : //www.researchgate.net/figure/Fig − A1 − The − standard −
− 16 − network − architecture − as − proposed − in − 32 − Note − that − only$_f$ig3$_3$22512435/

[8]RahulRavikantKanase, AkashNarayanKumavat, RohitDattaSinalkar, SakshiSoman, \Po

[9]Title = DeepPose : HumanPoseEstimationviaDeepNeuralNetworksPublishedby :
AlexanderToshev, ChristianSzegedy, Year = 2014/

[10]DeepHigh − ResolutionRepresentationLearningforHumanPoseEstimationPublishedby
KeSun, BinXiao, DongLiu, JingdongWangUniversityofScienceandTechnologyofChinaMicr
2019/

[11]ComparativeStudyOfHumanPoseEstimationModelsPublishedby : RahulPradhan, Mah
2020/

[12]HumanactivityrecognitionbyusingconvolutionalneuralnetworkHankilKim, SungockLe

[13]2.0.6DeepRecurrentNeuralNetworksforHumanActivityRecognitionPublishedby :
AbdulmajidMurad, Jae − YoungPyun.DepartmentofInformation, ChosunUniversity, 375Su

dong, Dong − gu, Gwangju501 − 759, Korea.November2017./

[14]HumanActivityRecognitionUsingToolsofConvolutionalNeuralNetworksPublishedby : Md.MilonIslam, SheikhNooruddin, FakhriKarray, GhulamMuhammad./

[15]SportsRecognitionusingConvolutionalNeuralNetworkPublishedby : ShakilAhmedReja
4318, Bangladesh./

[16]Two − StreamConvolutionalNetworksforActionRecognitioninVideosPublishedby : KarenSimonyan, AndrewZisserman, VisualGeometryGroup, UniversityofOxford./

[17]ContinuousHumanActionRecognitionforHuman − MachineInteractionPublishedby : HarshalaGammulle, DavidAhmedt − Aristizabal, SimonDenman, LachlanTychsen − Smith, LarsPetersson, ClintonFookes, Cornelluniversity, Ithaca, Newyork./