CAPSTONE REPORT

PRESENTED TO

b

SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

PRESENTED BY

MARIO BEVILACQUA
CHARLES BEYRARD
MIGUEL LOPEZ ROZA
ALDEMAR PINZON
MARCOS RAY

ETHICS STATEMENT

We acknowledge the use of ChatGPT for grammatical and clarification purposes. The prompts included “please clarify the meaning of this paragraph” and “please re-write this sentence in a concise manner”.

Mario Bevilacqua

Aldemar Pinzon Rodriguez

Miguel Lopez Rosa

TABLE OF CONTENTS

INTRODUCTION

In the quickly transforming fitness sector, AI-powered exercise applications have emerged as transformative instruments, fundamentally changing how people strategy their wellbeing and physical fitness targets. These programs leverage synthetic intelligence to offer you custom exercise plans, real-time opinions, and complete wellness monitoring, generating fitness more obtainable, efficient, and engaging for end users all over the world.[1] The integration of AI and machine vision into the fitness market has not only reshaped the way in which fitness services are delivered but additionally has spawned a vibrant startup ecosystem centered on innovation and personalization. As technological innovation carries on to progress, this ecosystem is poised for further growth, presenting new possibilities for startups to revolutionize the fitness business.[2]

Amongst those increasing firms lies FITIZENS. The mission statement of this venture is always to remedy a prevalent issue while in the fitness marketplace: the shortage of the automated and correct technique for detecting and analyzing exercises. They aim to solve this problem by introducing a unique product or service that leverages AI to boost the teaching practical experience for both athletes and trainers. This product is designed to empower end users along with the power to oversee and modify the depth and efficacy of workouts, supplying a tailored approach to physical fitness that could be informed by precise detailed analytics. The challenge is constructed on a basis of cutting-edge technological know-how, like a 9-axis Inertial Measurement Unit (IMU) developed by Movesense, a Suunto spin-off, which can be a C++ programmable device. The computer software part of FITIZENS incorporates an Athlete App for end users, an Arena App for fitness facilities, cloud infrastructure for info storage and real-time communication of purposes, and proprietary AI algorithms for physical activity detection.[3]

The issue that FITIZENS aims to solve revolves around developing a scheme that can instantly identify specific exercises and precisely tally repetitions, calculate rhythm, and gauge velocity in real time. In this light, we were tasked with developing a model to detect a set of exercises. This capacity would be crucial for providing prompt, applicable feedback to athletes and instructors, allowing a more information-driven approach to fitness. By leveraging state-of-the-art AI algorithms and computer vision technologies, FITIZENS seeks to eliminate reliance on imprecise tracking methods and offer a solution that furnishes detailed insights into each workout session.

[1] Aryan Karn, ‘Applications of Artificial Intelligence in IoT and Sensor Networks: A Survey’, 2021, https://www.semanticscholar.org/paper/Applications-of-Artificial-Intelligence-in-IoT-and-Karn/93b9c0b952872e267549a5fa13b6d35bbbb80b30.

[2] Gaudenz Boesch, ‘Computer Vision in Sports - Use Cases in 2024’, viso.ai, 10 November 2023, https://viso.ai/applications/visual-ai-in-sports/.

[3] Victor Gonzalez, Dani Sierra, ‘Fitizens Capstone Project’, 2023.

METHODOLOGY

For our project, we decided to explore the use of pre-trained models. This approach is particularly beneficial for the FITIZENS project for several reasons. Training a model from scratch requires significant computational resources and time. Transfer learning, by contrast, is a more resource-efficient method, as the pre-trained model has already undergone the initial and most intensive phase of the training process. In terms of efficiency, transfer learning allows the project to leverage a model that has already learned a wide range of features from a large and diverse dataset, such as Kinetics. This means the model requires less data and potentially less computational time to achieve high performance on the specific task of detecting exercises.[4] Models pre-trained on large datasets have been exposed to a variety of features, which can help improve the accuracy and robustness of the FITIZENS model when detecting exercises, even in challenging conditions[5]. Transfer learning enables the FITIZENS team to quickly prototype and test their model, which is crucial for iterative development and timely project delivery. [6] A pre-trained model can be adapted to new tasks with relative ease, making it a flexible solution for the FITIZENS project as it evolves and potentially expands to detect a wider range of exercises or to incorporate additional features. To achieve the goal of this project, we combine several technological components to analyze and recognize human actions, specifically exercises, from video data.[7]

[4] IABAC®, ‘Transfer Learning: Leveraging Pre-Trained Models for Faster Results’, IABAC®, 2 August 2023, https://iabac.org/blog/transfer-learning-leveraging-pre-trained-models-for-faster-results.

[5] Dan Hendrycks, Kimin Lee, and Mantas Mazeika, ‘Using Pre-Training Can Improve Model Robustness and Uncertainty’ (arXiv, 20 October 2019), http://arxiv.org/abs/1901.09960.

[6] YiRan Ke et al., ‘Recognition Technology of Human Body Movement Behavior in Fitness Exercise Based on Transfer Learning’, *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 9 April 2021, 1002–6, https://doi.org/10.1109/ICSP51882.2021.9409004.

[7] Ke et al.

LITERATURE REVIEW

As we began our research for this project, we split our focus into three sections to understand the task at-hand. We first looked at how Computer Vision and Artificial Intelligence have changed the industry. We then looked at available datasets tailored for human-action recognition and used this analysis as a benchmark to guide our research over some pre-trained models we can use before choosing our final architecture.

1) The growing role of technology in the fitness industry

The amalgamation of engineering and physical prowess has substantially evolved from the straightforward exercise videotapes and mechanical treadmills of the 1980s to today's sophisticated ecosystem encompassing wearable devices, intelligent apparatus, and online fitness platforms. [8]This progression underscores the transformative impact of technology on conditioning, with computer vision playing a pivotal part in shaping the current landscape of workout and well-being. At the outset, the application of computer vision in conditioning technology zeroed in on basic motion tracking. The advent of mobile and wearable engineering broadened the horizons for fitness analytics, allowing for the detailed quantification of bodily metrics.[9] This initial stage established the framework for the integration of more complex computer vision techniques in conditioning, facilitating a deeper examination of human movement and exercise form.

The emergence of AI-powered instructors denoted a considerable advancement, leveraging computer vision, on-device AI, and dialogue systems to dissect and furnish feedback on users' workout form in real time.[10] This innovation brought customized coaching into the homes of users, rendering effective conditioning guidance more accessible.[11] Further developments led to the invention of sophisticated AI trainers that employ intricate computer vision techniques and machine learning algorithms to deliver real-time, personalized conditioning recommendations and remarks.[12] This era of conditioning technology centers around customizing the exercise experience to individual needs, significantly enhancing workout proficiency and safety. Furthermore, the fusion of computer vision with wearable and detector technology has generated a rich data ecosystem for fitness tracking, allowing for

[8] Junqing Xie et al., ‘Evaluating the Validity of Current Mainstream Wearable Devices in Fitness Tracking Under Various Physical Activities: Comparative Study’, *JMIR mHealth and uHealth* 6, no. 4 (12 April 2018): e94, https://doi.org/10.2196/mhealth.9754.

[9] Anurag Bajpai et al., ‘Quantifiable Fitness Tracking Using Wearable Devices’, *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference* 2015 (August 2015): 1633–37, https://doi.org/10.1109/EMBC.2015.7318688.

[10] Nghia Duong-Trung, Hitesh Kotte, and M. Kravcík, ‘Systems: A Case Study in Real-Time Pose Tracking to Enhance the Self-Learning of Fitness Exercises’, accessed 12 March 2024, https://www.semanticscholar.org/paper/Systems%3A-A-Case-Study-in-Real-Time-Pose-Tracking-to-Duong-Trung-Kotte/cd4636f96c45b4abc5db0579369a371853a2160a.

[11] Aryan Jagani et al., ‘CogniPoseAI: A Futuristic AI-Enhanced Personal Trainer’, *2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, 18 October 2023, 188–93, https://doi.org/10.1109/ICSSAS57918.2023.10331759.

[12] B Adibasava, Gowtham R, and Dr Asha K H, ‘AI Fitness Model Using Deep Learning’, *International Journal of Advanced Research in Science, Communication and Technology*, 7 February 2024, 459–64, https://doi.org/10.48175/IJARSCT-15361.

a comprehensive analysis of performance and health metrics. This blend offers a holistic view of an individual's conditioning journey, bridging the gap between quantitative data and qualitative exercise examination.[13]

2) Overview of available datasets

We explored utilizing numerous datasets for FITIZENS. We were first guided by our tutor to look at Kinetics 400, with its extensive sampling of human behaviors sourced from YouTube, which supplies a fundamental dataset for coaching models in computer vision, especially for movement classification. The Kinetics-400 dataset is a large-scale, high-quality dataset of video URLs which includes approximately 400 human action classes, with at least 400 video clips for each action. Each clip lasts around 10 seconds and is taken from YouTube videos. Actions cover a wide range of activities, such as "playing instruments," "dancing," "swimming," and "riding a bike," among others. The dataset was introduced by the DeepMind team and is commonly used in the field of computer vision and deep learning for tasks such as action recognition, video classification, and video understanding.[14] It provides researchers and engineers with a substantial amount of labeled video data to train and evaluate algorithms that can understand human actions in videos. Its strengths include well-annotated labels, genuine scenarios, and ready-made models, making it beneficial for preliminary model preparation and benchmarking. However, its constraints involve the generalization problems presented by its diverse material, absent context particular to physical exercise annotations, and potential issues with accessibility and privacy owing to reliance on YouTube videos. These alternatives let us look at alternative datasets.

Alternatives like HMDB51[15] and UCF101[16] furnish a broad assortment of human activities each with their perks, such as diverse motions and considerable data, advantageous for coaching general movement recognition models. In any case, they share comparable constraints with Kinetics, but lack specificity for exercise-linked behaviors and detailed annotations for exercise metrics. ExerciseNTU specifically structured for exercise recognition, lines up tightly with FITIZENS' targets, offering focused material for preparing models on workout detection and examination. Though, its limitation involves the more modest dataset size, which could influence the model's ability to generalize across diverse conditions. [17] PoseTrack stands out for its thorough pose annotations in lively scenes, invaluable for dissecting exercise form profoundly, yet it does not zero in on exercise actions singularly and lacks annotations for specific exercise measurements.

[13] 'AI Body Detection and Teaching System Based on Mediapipe Machine Learning Platform and OpenCV Computer Vision Library | Semantic Scholar', accessed 12 March 2024, https://www.semanticscholar.org/paper/AI-Body-Detection-and-Teaching-System-based-on-and-Li-Huang/de2c62af71e8a78c3a61af46a01d413d34c32986.

[14] Will Kay et al., 'The Kinetics Human Action Video Dataset' (arXiv, 19 May 2017), https://doi.org/10.48550/arXiv.1705.06950.

[15] 'Papers with Code - HMDB51 Dataset', accessed 12 March 2024, https://paperswithcode.com/dataset/hmdb51.

[16] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, 'UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild' (arXiv, 3 December 2012), https://doi.org/10.48550/arXiv.1212.0402.

[17] Amir Shahroudy et al., 'NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis' (arXiv, 11 April 2016), https://doi.org/10.48550/arXiv.1604.02808.

3) Overview of pre-trained models

There are many existing models that have been trained on these datasets and will be useful to our project's scope. The art of transferring learning is a technique in machine learning that involves using a pre-trained model and fine-tuning it to specific problems. Features like pre-training and rapid deployment are both advantages. Nonetheless, other considerations must be considered. The model size and complexity, potential data privacy breaches, and dependencies on such models in terms of designing a fitting architecture. Using such a framework will require careful tailoring of our global architecture.

For action recognition, MMAction is a comprehensive video understanding toolbox that includes both action recognition and localization frameworks based on PyTorch. MMAction is rich in pre-trained models and tools designed specifically for video action recognition, which makes it very suitable to serve as the starting point of the FITIZENS project.[18] Another option is OpenVINO, a toolkit for optimizing deep learning models on Intel hardware for instance, which supports a wide array of models from different frameworks. OpenVINO can deal with various models and frameworks, making it a highly flexible and adaptable tool for optimization purposes. However, the optimizations themselves are made for Intel hardware, which might limit deployment options if FITIZENS uses heterogeneous hardware environments. [19]

In terms of pose detection and recognition counting, RepNet could be a potentially valuable part of the project. Its focus on counting repetitions fits very well with the need for accurately tracking how many exercises are done. It can efficiently handle videos of varying lengths and repetitions, crucial when doing real-time analysis.[20] But whether it can fully meet all project requirements is uncertain, for example in terms of accurately detecting exercises and velocity measurements. Also, the model's performance is very much hinged on quality and representativeness of training data. MediaPipe is another cross-platform, customizable ML solution for live and streaming media, including pose estimation. Its computer vision platform is optimized for real time applications. Hence, with its range of pre-built models especially for pose estimation, MediaPipe can also be used for detecting exercises and assessing form. However, we would need manual rules for counting, as it only recognizes human poses and not the sequence of actions.[21]

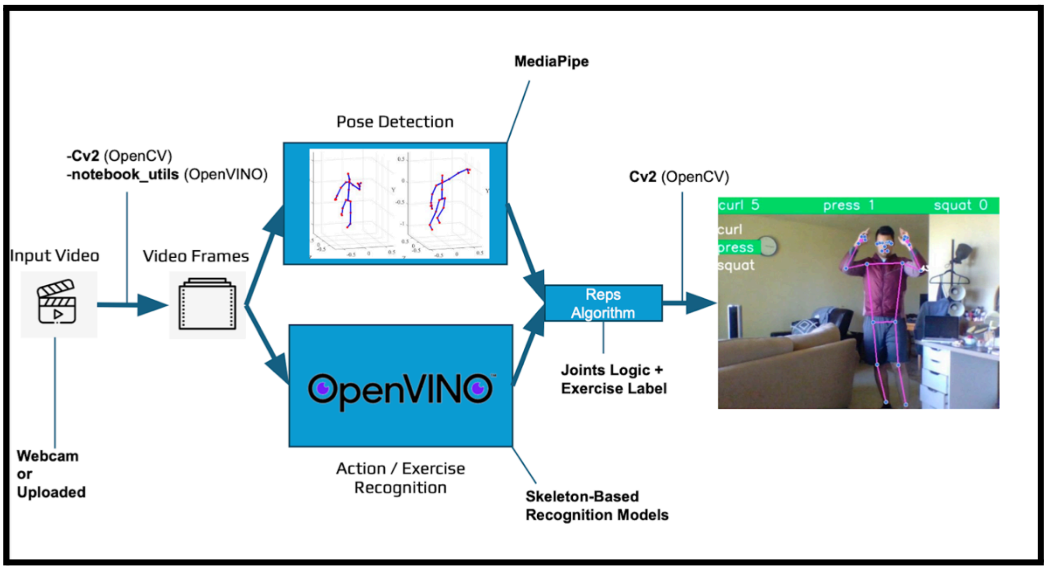
[18] 'Welcome to MMAction2's Documentation! — MMAction2 1.2.0 Documentation', accessed 12 March 2024, https://mmaction2.readthedocs.io/en/latest/.

[19] Alexander Demidovskij et al., 'OpenVINO Deep Learning Workbench: A Platform for Model Optimization, Analysis and Deployment', *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, November 2020, 661–68, https://doi.org/10.1109/ICTAI50040.2020.00106.

[20] Xinjing Zhang and Qixun Zhou, 'RepNet: A Lightweight Human Pose Regression Network Based on Re-Parameterization', *Applied Sciences* 13, no. 16 (21 August 2023): 9475, https://doi.org/10.3390/app13169475.

[21] 'AI Body Detection and Teaching System Based on Mediapipe Machine Learning Platform and OpenCV Computer Vision Library | Semantic Scholar'.

GLOBAL ARCHITECTURE



Initially, the project captures video input either through a webcam or from an uploaded file. It employs the cv2 module from OpenCV for the extraction of individual frames from the video, while utilizing notebook_utils from OpenVINO for additional video processing or model loading tasks. For pose detection, the system leverages MediaPipe, a sophisticated framework developed by Google, which accurately identifies the positions of various body parts across the video frames.

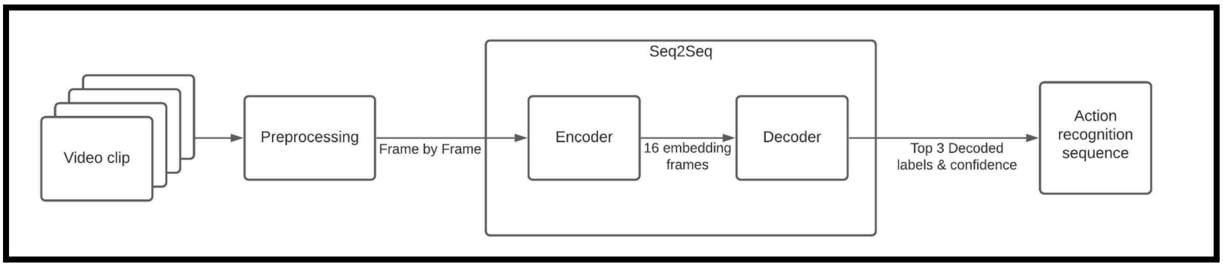
Following pose detection, the project uses the OpenVINO toolkit, specifically employing the action-recognition-0001 model. This model, which is adeptly trained on the extensive Kinetics-400 dataset, comprises an encoder with 21.2 million parameters and a decoder with 4.4 million parameters.

A custom algorithm is then applied to count repetitions of exercises, further distinguishing the type of exercise performed based on the detailed pose detection data.

1) Action Recognition

a. Action_recognition_0001 model

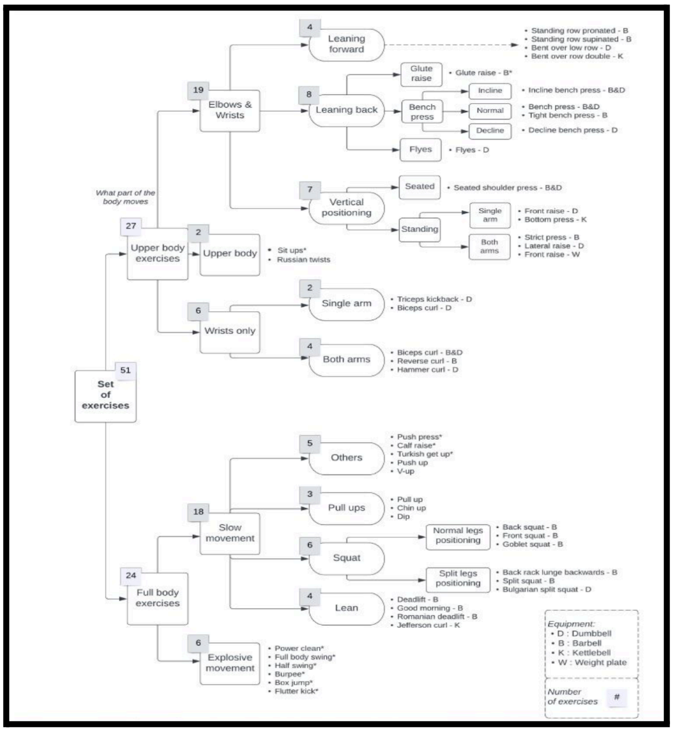
The action_recognition_0001 model from OpenVINO's Open Model Zoo is an advanced action recognition model that leverages a two-stream architecture combining a Video Transformer approach with a ResNet34 backbone for the encoder, and a separate decoder for processing video frames. Specifically designed for recognizing actions in videos, it is trained on the Kinetics-400 dataset, a comprehensive collection of video clips spanning 400 action categories. The encoder part of the model processes video frames to produce embeddings with an input shape of 1, 3, 224, 224 (B, C, H, W) and an output tensor shape of 1, 512, 1, 1, indicating the embedding of the processed frame. The decoder, on the other hand, takes a stack of these frame embeddings (with an input shape of 1, 16, 512) and outputs a logits vector of 400 actions, each corresponding to a different human activity. This model architecture, combining the depth and efficiency of ResNet34 within a Video Transformer framework, and trained on a dataset as diverse as Kinetics-400, is optimized for robust and accurate action recognition across a wide range of scenarios.[22]



[23]

b. Classification tree

As we are only interested in a small subset of the 400 physical activities being performed, we apply a custom logic using classification to group irrelevant activities together such as "walking dog" or "knitting" as "No exercise." For relevant activities, we start applying a grouping logic to improve results. For example, there isn't explicitly an activity "Burpee," but when a burpee is performed, we see that the probability of Push Up, Exercising with a Ball, and Squat increase to similar levels, so we group these activities as “burpee”.



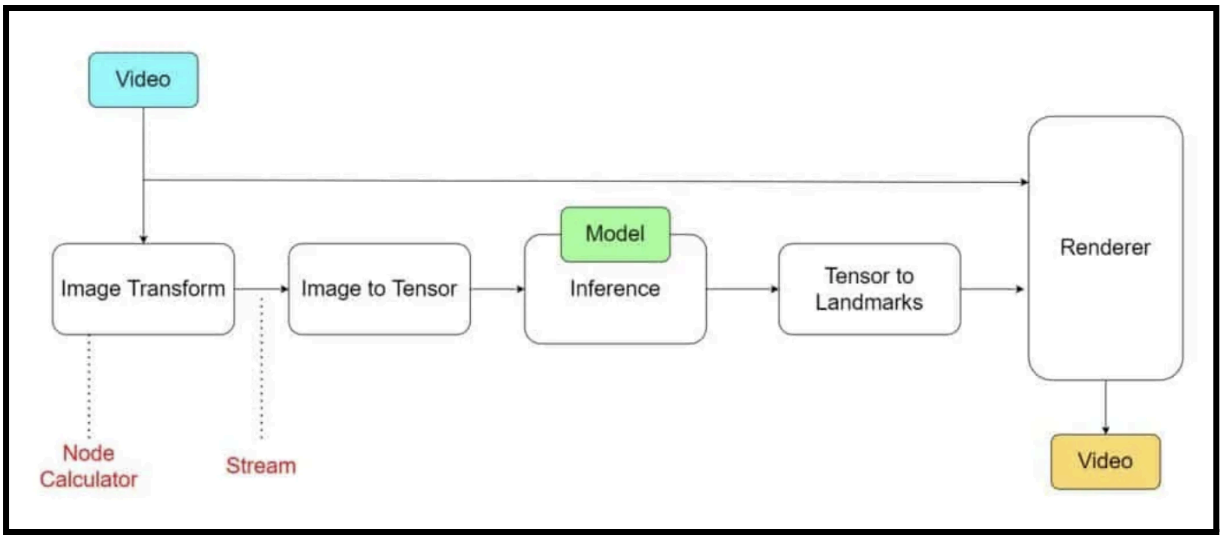
[22] “AI Body Detection and Teaching System Based on Mediapipe Machine Learning Platform and OpenCV Computer Vision Library | Semantic Scholar’.
[23] 'Openvino_notebooks/Notebooks/403-Action-Recognition-Webcam/403-Action-Recognition-Webcam.Ipynb at Main · Openvinotoolkit/Openvino_notebooks · GitHub’, accessed 12 March 2024, https://github.com/openvinotoolkit/openvino_notebooks/blob/main/notebooks/403-action-recognition-webcam/403-action-recognition-webcam.ipynb.

2) Pose detection

The architecture of the MediaPipe Pose model is based on a convolutional neural network and is optimized for on-device, real-time applications, particularly in the fitness domain. The model operates in two stages: pose detection and pose landmarking.[24]

In the first stage, the pose detection model identifies the presence of human bodies within an image frame and provides a few key pose landmarks. In the second stage, the pose landmarker model builds upon the initial detection to map out a complete set of 33 3-dimensional pose landmarks. This detailed mapping is achieved using a variant of the BlazePose model, which employs GHUM, a 3D human shape modeling pipeline, to estimate the full 3D body pose of an individual in images or videos.

We use this model's functionality in two ways. We use its ability to detect human bodies to center the frame around the exercise to optimize the features being fed into the encoder. Once the exercise is detected, we go back to the original frame and use this framework to track the cycle of a physical activity to count the repetitions.



25

3) Algorithm to count repetitions and exercises

For counting repetitions we apply a two-step logic on the model. In most of the cases, an exercise has an Extension and a Contraction of certain body parts, so we assume that a repetition is completed when both an Extension and a Contraction are identified. In addition, not all body parts are relevant for every exercise, so depending on the exercise, we will focus on certain zones.

The first step to count repetitions is to verify if there is a recognized exercise being performed. If that is the case, the code will retrieve a list containing the name of the relevant body-landmarks for the counting, and the angles required to calculate the Extension and

[24] Pose Landmark Detection Guide | MediaPipe', Google for Developers, accessed 12 March 2024, https://developers.google.com/mediapipe/solutions/vision/pose_landmarker.

[25] Saaisri, 'MediaPipe: Pose Dectection in Images and Videos', *Featurepreneur* (blog), 31 July 2022, https://medium.com/featurepreneur/mediapipe-pose-dectection-in-images-and-videos-31a583d5a7fb.

Contraction stages. The landmarks come in groups of three (i.e. Wrist, Elbow, and Shoulder) and there are two of these groups for every exercise (left and right side of the body). Next, we calculate the angles of the 2 groups of body-landmarks and assign in which part of the repetition they are (Extension, Contraction, None). Finally, we check if the 2 steps have already been completed, we add one repetition to the exercise counter and we restart the stages for next iterations. By monitoring changes in calculated angles over time, the system can count repetitions of certain actions, such as exercises. This functionality is achieved by detecting cyclic patterns within the pose data, marking transitions between distinct action phases.

PRACTICAL IMPLEMENTATION

1) Initial setup and configuration

a. Environment preparation

The notebook initializes by setting up the necessary libraries, including OpenVINO for deep learning inference optimization and MediaPipe for pose estimation, enhancing computational efficiency and accuracy in pose detection.

b. Action recognition model acquisition

We then import the action-recognition-0001 model from OpenVINO's Model Zoo. As stated, the model is specifically optimized for recognizing human actions by processing sequences of video frames.

2) Video Input and Pose Estimation

c. Flexible Video Input

The system accepts video input from various sources, adjusting dynamically to either live webcam feeds or pre-recorded videos. It strategically processes the last 30 frames of the input, skipping every second frame to maintain a balance between capturing the essence of actions and computational efficiency. The precision format of the model. "FP16" is chosen, which stands for 16-bit floating point numbers. This is often used for inference on devices with limited computational power as it provides a good balance between precision and performance.

d. Real-Time Pose Detection

Leveraging MediaPipe Pose, the pipeline accurately detects human poses within these frames, identifying 33 key landmarks that outline the human body's posture and movements. This detailed spatial information is pivotal for guiding the preprocessing steps and ensuring focus on relevant action areas within the frames.

3) Preprocessing and feature extraction

e. Frame Preprocessing

The frame is duplicated (the original remaining intact), and this duplicate is preprocessed for the action-recognition-001 encoder. Utilizing pose estimation data, frames are cropped and resized around the person performing the exercise to meet the encoder's input requirements. This step ensures that the frames are optimized for feature extraction, focusing on regions of interest highlighted by pose landmarks.

f. Encoder Inference

The encoder processes the preprocessed frames to extract dense feature vectors, effectively summarizing the visual information pertinent to action recognition. This transformation facilitates a compact representation of the video segment's content, capturing the dynamics of human movements.

4) Action Recognition: Vector Pack Assembly and Decoding

g. Vector Pack Assembly

Once 16 processed vectors are collected, the entire bag is sent to the Decoder, which returns the 400 activities with the probability of each one, but we do not need all of them. Using the custom logic we establish using the classification tree, we group the exercises to derive enhanced probabilities of the exercise being performed.

Therefore, we apply this logic: classify irrelevant activities such as "walking dog" or "knitting" as "No exercise." For relevant activities, we start applying a grouping logic to improve results. For example, there isn't explicitly an activity "Burpee," but when a burpee is performed, we see that the probability of Push Up, Exercising with a Ball, and Squat increase to similar levels, so we group these activities as burpees.

h. Decoder Analysis

The decoder model interprets these vector packs, outputting a (1x400) tensor that reflects the model's confidence across 400 potential actions. The action corresponding to the highest confidence score is identified as the performed action. We then group the classifications according to our classification tree.

5) Angle Calculation and Repetition Counting

i. Angle Calculation and Repetition Counting

Once the exercise is defined, we use MediaPipe’s landmark functionality to extract the evolution of the movement of the human body. Using our custom logic, we are then able to define a cycle of the exercise commencing and ending.

6) Output Visualization and Analysis

j. Output and Visualization

The function returns a dictionary of exercises detected in the video with their respective repetition counts and an array of preprocessed frames tagged as 'no exercise' when no specific action is recognized. The recognized actions, along with pose estimations, can be visualized on the video frames. This dual-overlay approach offers intuitive insights into how the model perceives and classifies human actions.

ALTERNATIVE ARCHITECTURE

In our exploratory research, we started working with MMaction2 as a baseline to understand the project's requirements for action detection. For pose recognition, we also explored the usage of RepNet for repetition counting and attempted to work with these models before transitioning to different solutions due to their shortcomings in implementation.

1) Reasons for moving away from MMaction 2

The decision to transition away from MMAction2 for our project stemmed from several critical issues that directly impacted its suitability. First and foremost, MMAction2's installation process was excessively complex, demanding the installation of numerous dependencies. This complexity significantly curtailed system flexibility, a pivotal aspect for our project's requirements. Moreover, MMAction2 is primarily tailored for processing pre-recorded video files, rendering it incompatible with our project's imperative for real-time data analysis. Adapting its architecture to support real-time processing proved arduous and convoluted. Furthermore, MMAction2's architecture constraints it to labeling a single action per video, which falls short of our project's need for recognizing multiple actions or nuanced activities within the same video stream.

2) Reasons for moving away Repnet

Similarly, our decision to pivot away from RepNet was influenced by several significant challenges. RepNet posed similar obstacles to MMAction2 with its complex and time-consuming installation process, thereby impeding system performance and hindering efficient deployment. Through rigorous testing, particularly with videos provided by our colleague Marcos, RepNet's performance and accuracy were found to be unsatisfactory for our project's goals, failing to meet the desired level of precision and reliability. Moreover, RepNet's focus on pre-recorded video inputs and its intricate architecture made it difficult to customize for real-time processing. This limitation significantly impeded our ability to adapt the framework to meet the dynamic demands of our project. Therefore, considering the challenges encountered with both MMAction2 and RepNet, it became evident that exploring alternative solutions was necessary to effectively address our project's technical needs and objectives.

DISCUSSION

1) Accuracy and efficiency

Our model displays remarkable skills in distinguishing a diverse spectrum of physical movements, underscoring its dependability and potency. It adeptly identifies prevalent actions like push-ups and sit-ups with exactness, highlighting its competence in discerning standard motions. However, some exercises, notably burpees, that are absent from the Kinetics 400 dataset and are a fusion of energetic motions, struggle to be identified accurately and are misclassified. Our attempts to overcome this limitation by combining postures from multiple physical activities to identify burpees exactly does not match the detection accuracy of other exercises. Thus, while our model remains highly compelling for the activities it was prepared on, it is vital to acknowledge its constraints when experiencing undertakings outside its preparation extent, such as burpees.

Should the model be adjusted to evaluate every single frame, its complexity and robustness would indeed increase, potentially improving its evaluative precision. However, such an adjustment would demand higher computational resources and could slow down the analysis process. This trade-off between model complexity and processing speed is pivotal in maintaining a balance that ensures the model remains both effective in its analytical capabilities and efficient in its operation. Striking the right balance is essential for achieving a system that not only accurately identifies exercises but also does so in a timely manner, thereby preserving the practicality of real-time exercise monitoring and repetition counting.

2) Feasibility and applicability

When initiated on a desktop computer with 12 gigabytes of RAM, the performance of our algorithm operated at a pace of 11 frames per second. This notably influenced actual-time identification and frequency enumeration. These lagging responses can substantially affect the speed at which the system perceives movements, specifically in scenarios involving hastily executed or sophisticated motions. Therefore, while accomplishing eleven frames for each second may meet certain program prerequisites, it is crucial to consider the potential postponements and their consequences for actual-time output. Augmenting the frame rates and computational power would generally relieve these tests and furnish more reactive and exact activity recognition and repetition counting functionalities. It should be noted that our action detection model is optimized for Intel equipment. FITIZENS has not stated to be using computers from that brand and while this is not ultimately a hurdle, it should be considered when choosing the appropriate framework for this project.

Another feature of our model is that it is designed specifically for detecting and analyzing the motions of a single person. Attempting to utilize it for the simultaneous detection of two individuals could compromise its integrity and accuracy. When the camera identifies multiple people in the video feed, it introduces complexities that the model is not equipped to handle, potentially leading to misinterpretation of actions or confusion in tracking movements. Therefore, it's crucial to ensure that the model operates under conditions where only one person is the focus of analysis to maintain its effectiveness and reliability in motion detection tasks.

3) Application to FITIZENS

The model developed by FITIZENS, which automatically detects exercises and counts repetitions, cadence, and velocity in real-time, offers a significant competitive advantage for the startup. By providing trainers with the ability to monitor and control the effort, intensity, fatigue, and enjoyment of their athletes, FITIZENS enhances the training experience and enables personalized workout adjustments. This level of detail in workout analytics is a strong selling point for fitness centers, as it can lead to increased average ticket sizes and athlete retention by offering a premium, data-driven service.

For athletes, the benefits of using FITIZENS' model are clear. The real-time feedback on performance allows for immediate adjustments, which can accelerate progress and improve the overall effectiveness of workouts. This increased engagement and satisfaction with the training process can motivate athletes to train more consistently and with greater intensity. Additionally, the ability to share progress and achievements helps build a stronger, more united community within fitness centers, fostering a sense of camaraderie and shared goals.

From a business perspective, the data collected by FITIZENS' model is invaluable. It enables fitness centers to make data-driven decisions regarding class scheduling, equipment purchasing, and staffing. Furthermore, the continuous feedback loop from the system's usage can guide product enhancements and the development of new features, ensuring that FITIZENS remains at the forefront of fitness technology. These features may include collection of data for each athlete considering their form/style of their exercises, similarities between repetitions in a set, as well as advice and recommendations to improve posture in the exercise. The potential for market expansion, such as remote training capabilities and partnerships with health organizations or corporate wellness programs, opens new avenues for growth and revenue for the startup.

Additionally, the ongoing data generated by the system's operation could serve as a cornerstone for continual product improvement and innovation, solidifying FITIZENS' position as a leader in fitness technology. Potential features might involve gathering detailed data on each athlete, factoring in the nuances of their exercise form and technique, providing customized guidance and recommendations to enhance exercise form and posture. This proactive approach to leveraging user feedback and data analytics underscores FITIZENS' commitment to delivering cutting-edge solutions to the Fitness community.

CONCLUSION

The FITIZENS endeavor embarked upon an ambitious quest to harness the powers of artificial intelligence and machine vision in transforming the fitness realm by providing a nuanced, real-time examination of exercises. By leveraging cutting-edge technologies and methodologies, including pre-trained models, transfer learning, and sophisticated pose detection algorithms, we established a system capable of accurately detecting a broad assortment of physical activities and counting repetitions instantly. This system represents a significant step ahead in generating a more interactive, personalized fitness experience for users and a potent tool for trainers and fitness facilities.

Throughout the duration of this project, we successfully exhibited the feasibility of employing AI to enhance the quality and effectiveness of exercise monitoring. The employment of pre-trained models substantially reduced the development time and computational resources necessary, while still achieving high levels of precision and efficiency in exercise recognition. Our methodology, which combined the strengths of various technologies such as MediaPipe for pose detection and OpenVINO for action recognition, proved to be a robust solution to the challenges presented by real-time exercise analysis.

Despite these successes, the project also encountered restrictions, particularly in recognizing complex exercises like burpees, which are not well-represented in existing data sets. This highlights an area for further research and progress, emphasizing the need for more comprehensive and specialized data sets in the fitness domain.

The practical implications of FITIZENS for the fitness industry are profound. By offering detailed, real-time analytics on exercises, FITIZENS empowers users to train more effectively and safely, while providing trainers and fitness centers with valuable insights to customize their services to individual needs. This level of customization and feedback can enhance user involvement, improve training outcomes, and foster a more connected and motivated fitness community.

Looking ahead, the scalability of this project for FITIZENS is promising. The modular nature of our architecture allows for the integration of additional exercises and enhanced features, such as form correction and fatigue monitoring, without significant alterations to the core system. As the project evolves, it has the potential to not only serve individual users and fitness centers but also to integrate with health and wellness programs on a larger scale, offering insights into overall health and encouraging a more active lifestyle across diverse populations.

In conclusion, the FITIZENS project achieved its objective of developing an AI-powered system that advances the way exercises are monitored and analyzed. While acknowledging the challenges and limitations encountered, the project lays a solid foundation for future innovations in the fitness industry. As technology continues to evolve, so too will the opportunities to enrich the fitness experience, making it more accessible, engaging, and effective for everyone.

FURTHER RECOMMENDATIONS

In addressing the constraints identified during the FITIZENS project, a set of recommendations emerges. One of the primary limitations is the current need for one person per video, which can be resource intensive. To tackle this, parallel executions leveraging significant computational power can be implemented, allowing the system to process multiple individuals simultaneously. This enhancement would significantly scale up the model's applicability in varied and dynamic fitness environments.

Furthermore, the challenge of integrating new exercises into the system, such as the plank, traditionally requires collecting numerous labeled videos and retraining the model—a process that can be both time-consuming and costly. An innovative solution lies in adopting a Few-Shot Learning approach, which dramatically reduces the need for extensive retraining. With this method, only a handful of videos for the new exercise are needed to generate a representative vector using a highly capable model that can encode the essence of the exercise. The development of such an advanced model remains aspirational, pushing the boundaries of what's currently achievable in machine learning and action recognition.

The pursuit of efficiency in action recognition suggests a move toward a machine learning classification approach for identifying exercise classes. This could alleviate the need for creating intricate decision trees and instead, rely on a model's learned patterns to classify exercises, necessitating robust label data. For the issue of videos with lower frame rates, adopting an adaptive frame iteration strategy can enhance the model's ability to process such inputs effectively. By enabling the model to consume all frames for action recognition rather than every second one, accuracy can be improved, but at the expense of greater computational demand.

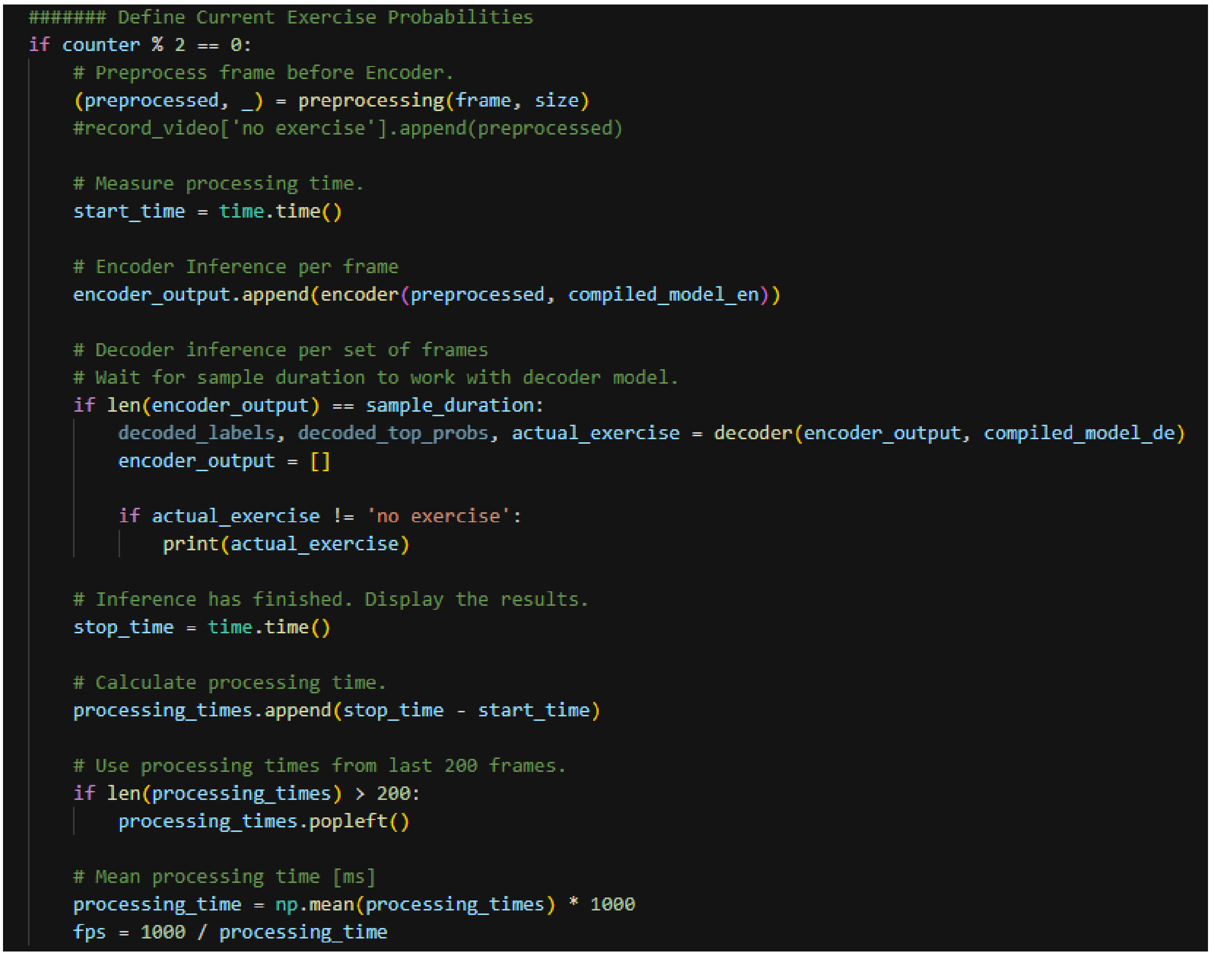
As for repetition cycle detection, shifting from fixed angle measurements to a machine learning model that learns and determines cycles could offer more nuanced recognition capabilities, adapting to a variety of exercises and individual forms. Lastly, the ambition of seamlessly incorporating new exercises without exhaustive model retraining could potentially be met with an advanced, pre-trained general encoder model. Such a model would be a leap forward, representing a versatile solution capable of learning from minimal data to recognize new exercises, thus embodying the cutting-edge vision of FITIZENS in the domain of fitness technology.

APPENDICES

Our full code can be found in this repository:

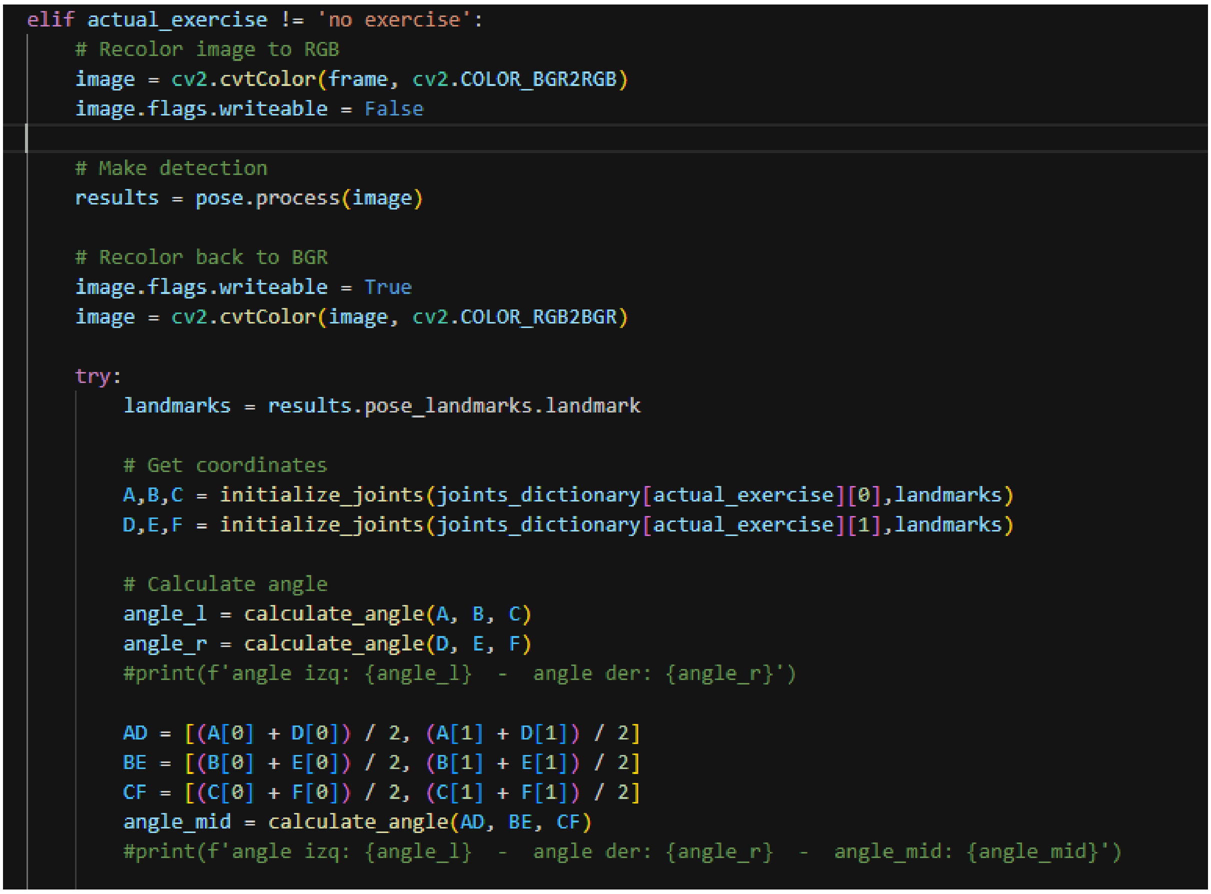
https://github.com/mariobevi/Fitizens_CorporateProject

Code for Exercise Model



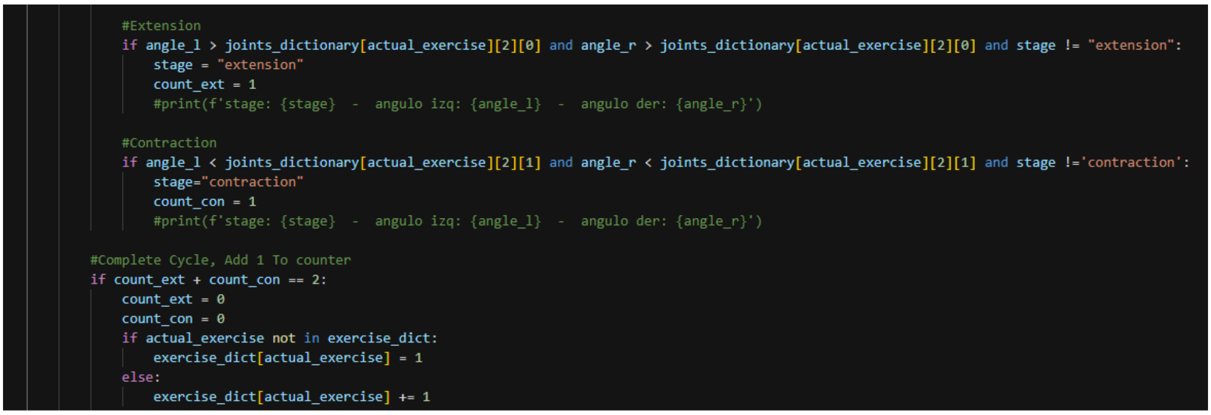
```
####### Define Current Exercise Probabilities
if counter % 2 == 0:
    # Preprocess frame before Encoder.
    (preprocessed, _) = preprocessing(frame, size)
    #record_video['no exercise'].append(preprocessed)

    # Measure processing time.
    start_time = time.time()

    # Encoder Inference per frame
    encoder_output.append(encoder(preprocessed, compiled_model_en))

    # Decoder inference per set of frames
    # Wait for sample duration to work with decoder model.
    if len(encoder_output) == sample_duration:
        decoded_labels, decoded_top_probs, actual_exercise = decoder(encoder_output, compiled_model_de)
        encoder_output = []

        if actual_exercise != 'no exercise':
            print(actual_exercise)

    # Inference has finished. Display the results.
    stop_time = time.time()

    # Calculate processing time.
    processing_times.append(stop_time - start_time)

    # Use processing times from last 200 frames.
    if len(processing_times) > 200:
        processing_times.popleft()

    # Mean processing time [ms]
    processing_time = np.mean(processing_times) * 1000
    fps = 1000 / processing_time
```

Code for Angle Extraction



```python
elif actual_exercise != 'no exercise':
    # Recolor image to RGB
    image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    image.flags.writeable = False

    # Make detection
    results = pose.process(image)

    # Recolor back to BGR
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

    try:
        landmarks = results.pose_landmarks.landmark

        # Get coordinates
        A,B,C = initialize_joints(joints_dictionary[actual_exercise][0],landmarks)
        D,E,F = initialize_joints(joints_dictionary[actual_exercise][1],landmarks)

        # Calculate angle
        angle_l = calculate_angle(A, B, C)
        angle_r = calculate_angle(D, E, F)
        #print(f'angle izq: {angle_l}  -  angle der: {angle_r}')

        AD = [(A[0] + D[0]) / 2, (A[1] + D[1]) / 2]
        BE = [(B[0] + E[0]) / 2, (B[1] + E[1]) / 2]
        CF = [(C[0] + F[0]) / 2, (C[1] + F[1]) / 2]
        angle_mid = calculate_angle(AD, BE, CF)
        #print(f'angle izq: {angle_l}  -  angle der: {angle_r}  -  angle_mid: {angle_mid}')
```

Code for Extension/Contraction/Repetition



```python
                #Extension
                if angle_l > joints_dictionary[actual_exercise][2][0] and angle_r > joints_dictionary[actual_exercise][2][0] and stage != "extension":
                    stage = "extension"
                    count_ext = 1
                    #print(f'stage: {stage}  -  angulo izq: {angle_l}  -  angulo der: {angle_r}')

                #Contraction
                if angle_l < joints_dictionary[actual_exercise][2][1] and angle_r < joints_dictionary[actual_exercise][2][1] and stage !='contraction':
                    stage="contraction"
                    count_con = 1
                    #print(f'stage: {stage}  -  angulo izq: {angle_l}  -  angulo der: {angle_r}')

            #Complete Cycle, Add 1 To counter
            if count_ext + count_con == 2:
                count_ext = 0
                count_con = 0
                if actual_exercise not in exercise_dict:
                    exercise_dict[actual_exercise] = 1
                else:
                    exercise_dict[actual_exercise] += 1
```

ACKNOWLEDGMENTS

We would like to thank our tutor Hind Azergrouz for her timely advice throughout the project.

We would also like to thank the founders of Fitizens Dani Sierra and Victor Gonzalez for giving us the opportunity to work on this project for them and hope they can extract some usage from it.