Grenze International Journal of Engineering and Technology, June Issue

Enhanced Yoga Posture Detection

Jay Firke[1], Vaishnavi Bhujbal[2], Mrudula Bodke[3] and Pranav Warke[4]
[1-4]Pimpri Chinchwad College of Engineering and Research, Computer Department, Pune, India
Email: firke0419@gmail.com, vaishnavibhujbal14@gmail.com, mrudula.bodke_comp2020@pccoer.in, pranav.warke_comp21@pccoer.in

***Abstract*— Human pose detection is a rapidly growing area in computer vision, which has various applications such as action recognition, sports analysis, and human-computer interaction. Yoga, a holistic practice that promotes physical and mental well-being, has gained widespread popularity in recent years. With the advent of technology, there is an increasing demand for innovations that can enhance the yoga experience. The objective is to develop an automated system that can detect and recognize yoga poses in real-time using a live camera. This research tackles the challenging task of real-time yoga pose detection through a live camera feed, aiming to facilitate human-computer interaction and advance the field of computer vision. Yoga poses are complex and highly variable, making their automated detection a noteworthy endeavor. This research aims to develop an iOS application for detecting yoga postures that leverages the capabilities of AV Foundation, Combine, Vision, and Core ML frameworks to construct an advanced system. Its applications extend beyond yoga, encompassing fitness monitoring, rehabilitation, and wellness tracking, further emphasizing its significance in modern society.**

***Index Terms*— Machine Learning, Create ML, CoreML, Vision Framework, AV Capture session**

I. INTRODUCTION

In recent years, yoga has become a global wellness trend, attracting people seeking a holistic approach to health and fitness. Rooted in ancient practices, yoga combines physical exercise, breath control, and mindfulness, offering a comprehensive approach to well-being. Its adaptability has made it popular across diverse demographics. Simultaneously, the use of health and fitness apps has surged, allowing individuals to personalize their wellness journeys.

Amidst this trend, there's a growing need for an efficient yoga posture detection system. While yoga has numerous benefits, correct posture is crucial for maximizing efficacy and preventing injury. Traditional methods like in-person classes or static videos often lack the precision needed for dynamic yoga poses. Introducing technology, specifically in the form of an iOS app, becomes a compelling solution.

The motivation behind this research lies in addressing the challenges faced by yoga practitioners, both beginners and experienced, in maintaining correct postures and alignment. Misalignment during yoga poses can lead to injuries, hinder the flow of energy, and diminish the therapeutic benefits of the practice. The Enhanced Yoga Posture Detection System aims to mitigate these challenges by providing instant feedback, allowing practitioners to make real-time adjustments and progress in their practice.

The research aims to develop an iOS app that not only facilitates yoga practice but enhances it with improved posture detection. Using Swift, AV Capture session, Vision Processor, and Create ML, the goal is to create a sophisticated system beyond conventional apps. This system will provide real-time, accurate feedback on yoga

Grenze ID: 01.GIJET.10.2.155

postures, promoting self-awareness and alignment. The blend of ancient yogic wisdom with cutting-edge technologies represents a progressive approach to holistic health. It caters to the growing interest in using digital solutions for personal well-being. As technology shapes the health and fitness landscape, the envisioned iOS app pioneers' efforts to enrich the yoga experience, making it more accessible, engaging, and precise for practitioners worldwide.

II. Material And Methods

A. CreateML Framework

Create ML framework is used to develop and train custom machine learning models directly. This powerful framework allows you to train models for various tasks, including image recognition, text comprehension, or discovering relationships in numerical data.

The procedure is feeding representative samples into a model to train it to identify patterns. For example, you can train a model to identify cats by exposing it to a variety of cat images. You can assess the model's performance using unseen data once it has been trained. Once you're satisfied with its accuracy, you can use Core ML to easily incorporate the model into your application.

The Create ML workflow includes gathering data, training the model, and evaluating its performance. This streamlined process ensures that your model is effectively learning from the provided data and can generalize well to new instances.

Integration of Create ML with models for tasks such as image classification and natural language processing significantly reduces training time while maintaining high performance. This integration allows for the development of smaller models that require less time to train. The goal is to make machine learning more accessible to developers in the ecosystem, enabling them to seamlessly integrate custom models into their applications using Core ML.

B. CoreML

Core ML stands out as a highly potent tool, boasting optimization for on-device performance. This versatile framework has the capability to employ models for a range of tasks, including image and sound classification, as well as the analysis of actions and drawings. Its functionality extends to working with both audio and text, harnessing the power of neural networks. Moreover, Core ML seamlessly integrates with external GPUs and takes advantage of the robust Metal Performance Shaders within the Metal framework, enhancing its overall efficiency and enabling advanced processing capabilities on-device.

These specialized Machine Learning APIs streamline the utilization of these capabilities, enabling to leverage advanced functionalities with just a few lines of code:

- The *Vision Framework* facilitates the development of image and video processing features using computer vision technologies. This includes image classification, object detection, and action classification, making it a versatile tool for applications requiring advanced visual analysis.
- The *Natural Language Framework* is designed for the analysis of text, providing segmentation into paragraphs, sentences, and words. It also supports tagging information about text segments, enabling developers to extract meaningful insights and enhancing language-related features in various applications.
- The *Speech Framework* empowers developers to integrate speech recognition capabilities into applications, supporting both live and prerecorded audio in multiple languages. This is essential for creating voice-controlled interfaces, transcription services, and applications requiring spoken language conversion.
- The *Sound Analysis Framework* specializes in audio analysis, allowing developers to classify sounds based on their characteristics. This is valuable for applications focused on environmental monitoring, audio event detection, and personalized audio experiences based on identified sound patterns.

C. Working of CoreML

- In your application, you input data into Core ML, which can take various forms such as text, images, or a video feed.
- Core ML processes this data input by running it through the model, executing the algorithm it has been trained on.
- Subsequently, the system provides the inferred labels along with their confidence levels as predictions.
- This entire sequence is referred to as the Inference process.
- The accuracy and reliability of these predictions depend on the type and quality of the model, with the goal of achieving high-confidence outcomes aligned with the intended use case.

D. Using and Creating Model

- Core ML serves as a robust toolchain for enhancing your applications with intelligent features, and the beauty lies in the fact that you don't have to navigate this process alone. Existing models are readily available for direct integration into Core ML. Additionally, you have the flexibility to convert compatible models from other machine learning toolchains, or alternatively, create your own custom machine learning models.
- Numerous preconfigured machine learning models tailored for Core ML are accessible from the Apple Website, covering diverse functionalities such as image classification, object detection, pose detection, drawing classification, and text analysis. The online developer community also offers an extensive array of additional models for various applications.
- In cases where your specific use case is not addressed by Apple's existing models, you can seamlessly convert models from alternative toolchains like TensorFlow or Turi. This opens up access to an abundant variety of machine learning models available within the broader developer community.
- When none of the available options align with your use case, you have the option to craft your own custom machine learning models using Create ML. This tool empowers you to build and train machine learning models on your Mac, eliminating the complexities associated with the model training process.

III. Proposed System

Step 1: Start the Video Capture Session

In the iOS application for yoga posture detection, the process begins by initializing a video capture session using Apple's AV Foundation framework. This allows the app to access the device's camera and continuously capture video frames for subsequent analysis**.**

Step 2. Create a Frame Publisher

Utilizing the Combine framework, a frame publisher is created to handle the stream of video frames obtained from the capture session. Combine facilitates the handling of asynchronous and event-driven code, enabling seamless integration with other frameworks like Vision Processor.

Step 3. Build a Publisher Chain

To build a publisher chain in the iOS application for yoga posture detection, Combine is employed to establish a sequential flow of data processing stages for video frames. The chain integrates the Vision Processor, a component of the Vision framework, to conduct real-time analysis on each frame. Specifically, body pose detection is emphasized, utilizing pre-trained machine learning models crafted with Create ML. Within this chain, each new publisher is created using Combine's powerful publisher methods, such as map, compactMap, and scan. These methods enable the transformation, filtering, and accumulation of data, respectively, as it progresses through the various stages of the chain. The usage of these methods within the publisher chain enhances the application's flexibility, allowing for efficient manipulation and refinement of the data at each processing step, contributing to the overall accuracy of the yoga posture detection system

Step 4. Analyze Each Frame for Body Poses

Vision Processor processes each video frame using Core ML, the machine learning framework on iOS. The machine learning model, trained using Create ML, is designed for body pose detection, providing insights into the positions of body joints, critical for accurate yoga posture assessment.

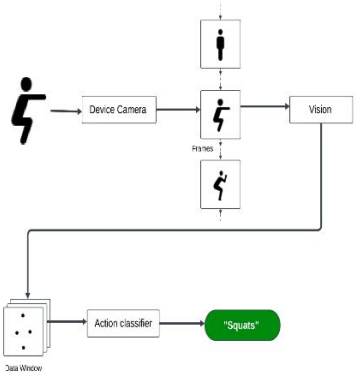

Figure 1: Proposed System

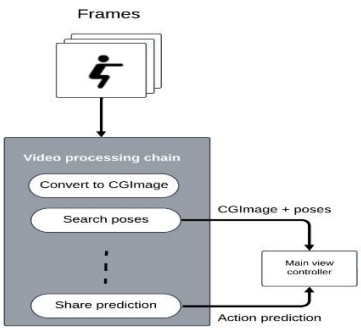

Figure 2: Building Publisher Chain

Step 5. Isolate a Body Pose

After analyzing the frame, the identified body pose is isolated, and relevant data is extracted. Core ML, along with the model created using Create ML, contributes to the accuracy of the pose detection, ensuring precise identification of yoga postures. When the app recognizes one or more people in the frame, it overlays a wireframe body pose on each person. At the same time, the app predicts the prominent person's current action; typically, this is the person closest to the camera.

Step 6. Retrieve the Multiarray

The Core ML model outputs a multiarray containing information about the detected body pose. This multiarray, representing key points on the body, is retrieved for further processing and analysis.

Step 7. Gather a Window of Multiarray

A window of consecutive multiarray is gathered, representing a temporal sequence of body poses. This approach, facilitated by Combine, allows the application to consider the context of yoga movements over time, contributing to improved accuracy and understanding of dynamic postures.

Step 8. Monitor the Window Size

The window size, indicating the number of consecutive frames considered, is monitored dynamically. This ensures flexibility in adapting to varying yoga poses and movement speeds, enhancing the model's capability to generalize effectively.

Step 9. Predict the Person's Action

To predict the user's ongoing yoga action or posture in the iOS application, an Action Classifier is employed. This classifier utilizes machine learning algorithms, specifically implemented using Core ML and created with Create ML, to analyze the gathered window of multiarray. The Action Classifier plays a pivotal role in enhancing the accuracy of understanding complex yoga movements, enabling the application to provide real-time predictions regarding the user's current yoga actions. This integration of machine learning models focuses on capturing the intricacies of yoga poses, contributing to a more comprehensive and accurate yoga posture detection system.

Step 10. Present the Prediction to the User

The predicted yoga action is presented to the user in real-time, enhancing the user experience. Combine helps manage the asynchronous nature of these predictions, ensuring smooth interactions with the app.

Step 11. Present the Poses to the User

The identified body poses, along with corresponding predictions, are visually presented to the user. The Vision framework aids in rendering and overlaying the detected poses on the video feed, contributing to a comprehensive and informative user interface.

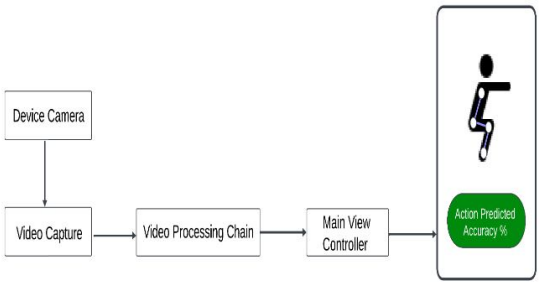

Figure 3: Overview of Proposed System

IV. RESULT

The result of this yoga posture detection project is a user-friendly iOS application that seamlessly integrates Core ML, Combine, Vision Processor, and Create ML to provide real-time feedback on users' yoga poses. The machine learning models are trained using a dataset composed of videos, allowing the system to understand a diverse range of yoga movements and achieve a high level of accuracy. The application is designed with simplicity in mind,

ensuring ease of use for practitioners of all skill levels. A minimum confidence value is set to maintain a balance between accuracy and responsiveness. If the confidence value falls below this threshold, the application displays a low confidence indication, encouraging users to adjust their posture for a more precise assessment. This ensures that users receive reliable feedback during their yoga practice, fostering a positive and effective experience.

V. Conclusions

In summary, the proposed iOS application for yoga posture detection leverages the power of AV Foundation, Combine, Vision, and Core ML frameworks to create a sophisticated system. By seamlessly integrating computer vision and machine learning, the application achieves real-time**,** accurate analysis of yoga postures. The dynamic publisher chain, the incorporation of an Action Classifier, and the consideration of temporal sequences contribute to its flexibility and overall precision. The system not only predicts users' ongoing yoga actions but also enhances their understanding and engagement in real time, showcasing the potential of advanced technologies to enrich health and wellness applications.

References

[1] Diogo C Luvizon, David Picard, Hedi Tabia, “Multi-task Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43 (8), pp.2752–2764.10.1109/TPAMI.2020.2976014.hal-02558843

[2] Neumark-Sztainer D, Watts AW, Rydell S. Yoga and body image: How do young adults practicing yoga describe its impact on their body image? Body Image. 2018 Dec;27:156-168. DOI: 10.1016/j.bodyim.2018.09.001. Epub 2018 Oct 3. PMID: 30292082; PMCID: PMC6691728.

[3] Stenum J, Cherry-Allen KM, Pyles CO, Reetzke RD, Vignos MF, Roemmich RT. “Applications of Pose Estimation in Human Health and Performance across the Lifespan.” Sensors (Basel). 2021 Nov 3;21(21):7315. DOI: 10.3390/s21217315.PMID:34770620;PMCID:PMC8588262

[4] Gupta, Ayush and Ashok Jangid. “Yoga Pose Detection and Validation.” 2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA) (2021): 319-324.

[5] Yadav, Santosh & Singh, Amitojdeep & Gupta, Abhishek & Raheja, Jagdish. (2019). Real-time Yoga recognition using deep learning. Neural Computing and Applications. 31. https://link.springer.com/article/10.1007/s00521-019. 10.1007/s00521-019-04232-7.

[6] Y. He, R. Yan, K. Fragkiadaki and S. -I. Yu, "Epipolar Transformer for Multi-view Human Pose Estimation," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 4466-4471, DOI: 10.1109/CVPRW50498.2020.00526.

[7] Huang, Chen & He, Yu-Zhen & Hsu, Chun-Chieh & Chou, Chien-Li & Lee, Suh-Yin & Lin, Bao-Shuh. (2014). “Yoga Posture Recognition for Self-training”. 496-505.10.1007/978-3-319-04114-8_42.

[8] Mane, D.., Upadhye, G. ., Gite, V .., Sarwade, G. ., Kamble, G. ., & Pawar, A. . (2023). “Smart Yoga Assistant: SVM-based Real-time Pose Detection and Correction System.” International Journal on Recent and Innovation Trends in Computing and Communication, 11(7s), 251–262.

[9] Sharma, Abhishek & Shah, Yash & Agrawal, Yash & Jain, Prateek. (2022). “Real-time Recognition of Yoga Poses using computer Vision for Smart Health Care.”