AGENT: Automated Guidance for Exercise and Therapy with Human Pose Estimation and Machine Learning

Kevin A. Gines and Arian J. Jacildo

***Abstract*— Exercising provides tons of benefits both physically and mentally. The Internet and social media have made it easier for people to find the proper routine that suits their fitness goals without having personal trainers. However, there is a heightened risk of injury when such exercises are done with poor form and improper execution. In this study a mobile application called AGENT, was developed to provide users feedback to their exercise form. For this study, a bodyweight squat dataset was used to train a Recurrent Neural Network (RNN) model to identify if a user's squat form is correct or not. The RNN utilizes the joint angles found from each frame of the video for its classification. These joint angles are extracted using human pose estimation by using OpenCV and OpenPose. The RNN had an F-score of 0.71 and an accuracy rating of 71%. Using PSSUQ, it was observed that the respondents who evaluated AGENT were overall satisfied with the application and its core feature. Further improvements for AGENT include the use of a larger dataset, and the addition of exercises other than the squat.**

***Index Terms*— human pose estimation, exercise checker, mobile application**

I. Introduction

A. Background of the Study

Exercise, regardless of age, provides many benefits to humans both physically and mentally. Some of its physical benefits include weight control, risk reduction to some heart diseases and cancers, and strengthening of bones and muscles. At the same time, it also improves mental health, mood, and cognitive function [1].

Exercise can also be used as a tool to "correct impairments, restore muscular and skeletal function, and/or maintain a state of well-being" [2]. This is termed as exercise therapy. For instance, children with autism can be prescribed a high-intensity exercise program. This was proven in a study conducted by Kozlowski et al. [3] where they concluded that there are significant improvements in the children performing sit-ups, squats, and the standing long jump. In addition, another study confirmed that children with cerebral palsy showed improvements in their motor abilities as well as white matter connectivity in their brains after a six-month exercise therapy. This study, conducted by Samsir et al. [4] , further mentioned that exercise therapy can be incorporated as a home-based training regimen for strengthening and maintaining muscles.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

The COVID-19 pandemic drove nations around the world to impose restrictions and lockdowns. In effect, business establishments such as fitness centers and gyms were forced to close. A study by Brand et al. [5] investigated the effects of these lockdowns – during the first wave of the pandemic – on exercise routines and its effects on subjective well-being. With respondents coming from 99 countries (including the Philippines), the study concluded that people who rarely exercised before the pandemic tended to increase their exercise frequency during the pandemic. Additionally. respondents who exercised five times a week reported having positive mood states. In contrast, those who reduced their exercise frequency experienced worse mood states. Kaur et al. [6] conducted a similar study where they also concluded that having regular fitness routines at home help overcome psychological issues and fitness concerns especially during the pandemic.

Having that said, exercising at home with fitness applications is becoming more prevalent. The Internet and social media have made it easier for people to find the proper routine that suits their fitness goals without having personal trainers. However, there is a heightened risk of injury when such exercises are done with poor form and improper execution. Putting things into perspective, Canadian physiotherapists reported that they received an increased number of exercise-related injuries during the pandemic as compared to previous years. The primary causes of these injuries were incorrect exercise form and execution [7].

In the Philippines, however, an online survey reported that majority of older Filipinos are less active than they were before the pandemic. The respondents further mentioned experiences of lethargy, lack of energy, joint and muscle pain, and weight gain as an effect of having a sedentary lifestyle [8]. Another online survey was conducted to determine the barriers to physical activity of college students in Manila. Adapting a quiz developed by the U.S. Centers for Disease Control and Prevention, it was observed in the results that lack of resources had the highest mean score as a barrier to students to undergo physical activity [9].

Applications that can be used to study exercises as well as identify correct exercise form and execution can aid in preventing exercise-related injuries. A computer software, Pose Trainer, was developed by Chen and Yang [10] where it was able to detect proper form as well as pinpoint form improvements in weight training exercises.

Zhu [11] developed a motion assistance evaluation system for improving sports training through computer vision and deep learning instead of using wearable sensors. In addition to that, parents of children undergoing exercise therapy can utilize these applications to teach their children the proper form and execution, especially in the absence of a therapist. Computer vision-based, marker-less, human pose estimation systems show a promising future in physiotherapy application as mentioned in a critical overview provided by Hellsten et al. [12]. They recommend that such applications should be subjected to rigorous testing, implementation, and real-world scenario accuracy testing.

However, the aforementioned systems were only limited to users that have personal computers and webcams. With mobile phones being ubiquitous, a fitness application that provides exercise feedback directly from the user's performance would be practical. This helps in learning the proper form of an exercise and lessen the risk of injuries in a convenient manner.

In this research, a mobile application was developed where users are given feedbacks (whether correct or incorrect) about their exercise form using computer vision and machine learning. Users can record videos of themselves performing the exercise through their phones. The application will be utilizing a computer vision library for human pose detection and machine learning for analyzing patterns in data and constructing inferences. These will be used to provide feedback whether the user executed the exercise correctly.

B. Statement of the Problem

Several systems and applications have been developed that aim to provide users with a personal exercise form checker. This allows users, such as people undergoing exercise therapy, to perform exercises at the comfort of their homes. However, most of these studies and systems require a personal computer, a webcam, and even a Kinect Sensor. This could be a challenge economically as well as in home space setups. With phones being ubiquitous, there is a need to develop an exercise form checker that is available on mobile phones. This is because mobile phones consume less space and are easier to set up.

C. Significance of the Study

Generally, the output of this study would improve the quality of healthy living through exercise and would lessen the risk of exercise-related injuries, especially in the absence of professional trainers and exercise experts. The mobile application would help users learn an exercise correctly by performing it themselves rather than just relying on videos, photos, and text descriptions. Furthermore, people undergoing home exercise therapies can utilize the application for learning and executing the exercise properly and safely. It can also be used as a tool by experts (e.g. physiotherapists, fitness instructors, etc.) when teaching and training exercises to their clients. Lastly, researchers in this field can use this research as a reference in constructing innovative models and developing better applications.

D. Scope and Limitations

This study utilized an open-source computer vision software library (OpenCV) for receiving input data. Moreover, the pre-trained human pose estimator, OpenPose, was used since it is included as a library in OpenCV. TensorFlow was used in training the machine learning model. YouTube, Kaggle, and Darebee.com, a non-profit free, ad-free, and product placement free global fitness resource website, was used to source video exercises for the system. Only one exercise (bodyweight squat) was used in training the model and assessing the application.

II. Objectives

This study aims to develop a mobile application that incorporates computer vision and machine learning in providing feedback to users when they are performing an exercise. Specifically, the objectives of this research are:

1) To employ OpenCV & OpenPose for human pose estimation on a video captured through a single camera
2) To train a machine learning model that identifies whether an exercise is performed correctly or not; and
3) To develop a fitness learning mobile application

III. Review of Related Literature

Several studies have been performed for the development of various software and systems that utilize computer vision (CV) for detecting physical activity. These systems are applied in sports training, exercise, rehabilitation, and physical therapy [10], [11], [13], [14], [15], [16], [17], [18], [19]. These studies are centered on human body detection which can be achieved through the integration of a depth camera. An example of this is Microsoft's Kinect which was built for this particular capability.

A. Kinect-based CV Systems

ArthriKin, a Kinect-based system, was developed by Dorado et al [13]. for Rheumatoid Arthritis (RA) patients. Without the use of reference markers for body detection, it was able to provide supervision and feedback to patients as they perform their prescribed exercise routines at home. Overall, respondents found the system easy to use and a majority of them was satisfied with the system. ArthriKin's movement tracking accuracy was compared with a high-accuracy 3D measurement system called Krypton K400. Results showed that Kinect had high accuracy in detecting elbows and wrists, mixed results for the shoulder, and less detection accuracy for hips. The system provides feedback by comparing the patient's video with a master video (a video performing the correct form of the exercise) that was provided by the therapist. In some exercises, the angle of the limb with respect to a reference point is considered. This method of detecting proper form through limb angles was adopted in this study.

A similar Kinect-based system as an exercise form checker was developed by Conner and Poor [14] for weight training exercises. The study stated that the software made it easier for the respondents to realize that their actions had incorrect form and was able to correct them using their system. Additionally,

the squat exercise was used in testing the application since it has minimal variants, and it also utilizes majority of the muscles in the body. For this reason, this study also used the squat exercise for training the model and testing the application.

A hybrid desktop and web application for remote physical therapy was developed and deployed by Pandit et al [15]. The system, ExerciseCheck, provides a cloud-based platform that makes it easier for users to install and interact with. Generally, the users found it easier to understand the mechanics of the exercises to be performed with the help of ExerciseCheck and they were largely satisfied. This cloud-based implementation will also be adopted in this study. This is to allow heavy calculations to be performed in the cloud rather than in the mobile device itself.

The aforementioned studies have shown that computer vision has potential in expanding their respective fields of study by providing additional perspectives for gathering and extracting information. It also has the capability to be used for other rehabilitation routines that can be performed at home [13]. Furthermore, users can freely perform exercises and receive feedback without having to wear any motion tracking device [14]. However, using Kinect for applications such as these have its downsides. Prototypes using Kinect are not suitable for multi-site use and deployment. It also does not provide physical therapists the ability to "personalize and update exercise parameters and configuration details at any time" [15]. This limitation was addressed in the implementation of ExerciseCheck. The hybrid implementation of ExerciseCheck shows the capability of such applications to be used in different platforms.

B. Human Pose Estimation

Another method for human body detection is through human pose estimation, an application of computer vision where it detects, associates, and tracks semantic key points in an image or video in order to predict or estimate the human pose exhibited therein [20].

OpenPose is a popular human pose estimation tool that is able to detect multiple persons in a video, video stream, or image using Part Affinity Fields (PAF). In the study documenting its development, OpenPose was compared to other pose estimation libraries. The results revealed that OpenPose had superior inference time as compared to the other libraries [21]. OpenCV, an open-source computer vision library, has included OpenPose in their Deep Neural Networks module. Because of that, OpenCV along with OpenPose were used in this study.

C. OpenPose-based Systems

The quantification of postural control evaluations was implemented using OpenPose by Hagihara et al. Preschool children were recorded performing the One Arm and One Leg Balance test ("bird dog posture"). The recordings were subjected to OpenPose to detect posture and collect certain measurements. These measurements or key points will be used for the algorithms that will generate quantitative indices. In terms of effectivity, the calculated indices were compared to two other metrics: 1) the traditional metric for the bird dog posture test: Duration Time (DT); and 2) Therapists' Quantitative Clinical Evaluations (TQCE), a 7-pt Likert Scale evaluation accomplished by actual therapists. The results show that the generated indices from the CV-based system had a positive correlation with the evaluations made by the therapists (TQCE). Also, the CV-based indices reflected the TQCE evaluations to a greater degree over the traditional metric, DT. Further regression and correlation analyses showed that CV-based indices provide more detailed quantitative information [16].

Since the study of Hagihara et al [16]. centered on postural control and existing assessment tests, a mathematical formula or algorithm along with computer-vision was used to devise a technique that further improved postural control tests. The values entered in the algorithms are from key points extracted from the recordings. This is a common theme in these studies where key points in a frame are used to determine distances as well as angles to estimate poses. These would also serve as the basis for identifying correct exercise form in this study.

Nagarkoti et al. developed a real-time indoor workout analysis using the same notion. In this system, a reference video (a video showing the correct form) uploaded by a trainer will be overlaid with the video uploaded by a user. This is achieved with the use of Dynamic Time Warping (DTW) to synchronize the two videos. On top of that, Affine Transformations was also used to "normalize" the user's body ratios. Transforming the user's body ratio is done so that it accurately corresponds with the body ratio of the trainer when both are overlaid with each other. Additionally, this process maps the limb pairs of the trainer with those of the user. These, along with a threshold angle, will be used to calculate the errors made by the user as they perform the exercise [17].

In contrast, Zhu [11] built a system where the user's body ratio was not transformed to comply with standard data. Instead, the 'close Angle' concept was used to eliminate the need to compensate for the errors caused by inconsistent limb lengths and differences in the camera angle used in a recording. In the system, a standard action is defined with all the necessary data and is saved in the standard motion database. These data will be used as the basis for the athlete that is training. When the athlete's video (test motion) is uploaded, the system will extract all the joint angles from the video and compare it with similar data found in the standard motion database. The difference between the test motions and the standard motions will be displayed to the athlete. The close Angle concept introduced here will be adapted in this study in order to eliminate the need for normalizing the key points as well as resolving the concerns in video acquisition from different camera angles or distances. That said, the equations used to calculate the close Angle will be used as well.

The close Angle concept can be visualized in this study as pairs of key points in the human body. For instance, the nose-neck line segment (AB) and the neck-right shoulder line segment (BC) form two-line segments that intersect at the point of the neck. The angle made by the two segments is the joint angle. Equations 1 to 4 calculate the values needed

in determining the cosine of the joint angle B shown in Equation 5. From there, the joint angle can be calculated.

$$\overrightarrow{AB} = (x_2 - x_1, y_2 - y_1) \tag{1}$$

$$|AB| = \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2} \tag{2}$$

$$\overrightarrow{BC} = (x_3 - x_2, y_3 - y_2) \tag{3}$$

$$|AB| = \sqrt{(x_3 - x_2)^2 - (y_3 - y_2)^2} \tag{4}$$

$$cos\angle B = \frac{\overrightarrow{AB} \times \overrightarrow{BC}}{|AB||AC|} \tag{5}$$

Another approach in determining the correct form of an exercise or an activity is by utilizing machine learning. Chen and Yang [10] developed an application that employed both approaches: geometric (measuring angles) and machine learning. This gives the software application, PoseTrainer, the capability to provide insightful feedback with these two approaches. Four weight training exercises, each with their own datasets for machine learning, were properly detected by the system. It was also able to provide specific points of improvement when certain portions of the exercises were labelled as incorrect. For example, the system can detect if the user is not performing the full range of motion of the exercise and prompts the user to do so. It can also suggest lowering the weight if necessary. In contrast, unlike Zhu's study [11], the key points collected from the input data were normalized as ratios of torso length. For example, the upper arm is normalized by getting the ratio of its length with the torso length.

D. Human Pose Estimation in Mobile Applications

One of the challenges in developing mobile applications with human pose estimation is the difficulty in deploying models in "resource-constrained devices such as smartphones and embedded systems, which have low-powered processing units and small memory" [19]. Several studies have developed lightweight 2D [19], [22] and 3D [23] human pose estimation for mobile devices. The following studies have shown the potential of integrating lightweight human pose estimation in mobile applications.

A running form checker mobile application was developed by Takeichi et al [18]. The application incorporated Convolutional Pose Machines, a pose estimation technique, which was implemented using Keras. A motion capture system was used to evaluate the application's accuracy. Despite the videos only having 30 frames per second (FPS), the application was able to detect and evaluate the user's running form. It also showed excellent correlation with the motion capture system's metrics. Additionally, the evaluation was consistent with the actual coaches' evaluations on the user's running form.

On the other hand, Jeon et al. [19] developed a real-time 2D human pose estimation model with Knowledge Distillation (KD) learning that is suitable for the smartphone environment. The model also used Simple Baseline as the basic architecture along with a MobileNet V2 as the backbone architecture to further reduce the model's computation cost. Putting things into perspective, the model had an average inference time of 13 milliseconds for a Samsung Galaxy S10. The machine learning model was trained with an exercise dataset sourced from YouTube. With this trained model, a fitness coaching application was constructed to evaluate a user's motion.

BlazePose [22] is a lightweight human pose estimation tool that is capable of providing real-time use cases such as fitness tracking. Unlike OpenPose and Kinect, BlazePose uses 33 key points on the human body. These key points are located by the system through a combination of heatmap, offset, and regression approach. To test its quality, BlazePose's performance was compared with OpenPose in detecting human poses on two datasets: 1) various human poses; and 2) yoga/fitness poses. While it performed worse on the dataset containing various human poses, BlazePose was able to outperform OpenPose on the yoga/fitness dataset. This implies that BlazePose can be used for real-time fitness use cases on mobile applications. This pose estimation tool can be used as an alternative in developing this study's application.

IV. Materials and Methods

The following subsections would be defining the sequence of methods that were done for the development of AGENT. First, an overview of the development process would be outlined. Next, the setup, which includes the devices used for developing the machine learning models, web servers, and mobile application would be specified. Lastly, the user flow and application wireframes will be discussed along with the usability testing conducted after.

A. Development Process

As summarized in Figure 1, this study involved the training of machine learning models that would help in the classification whether the input video shows a correct or incorrect exercise form. The models were developed using TensorFlow with Keras on Python. Moreover, these models were integrated to the mobile application by deploying it in a remote server using FastAPI on Python. The mobile application employed React Native for the user interface. Lastly, the feasibility of the application was evaluated through a survey answered by invited and willing respondents.

B. Development Setup

1) Machine Specifications: The machine used in developing the model and application was an Acer Aspire 3 A315-A41G-R4BW with Windows 10 Home Single as the operating system. The system consisted of an AMD Ryzen 5 2500U Processor (with Radeon Vega Mobile Gfx clocking at 2.00 GHz and eight logical processors) along with an AMD Radeon Vega 8 Graphical Processing Unit, a 12-gigabyte (GB) Random Access Memory (RAM) with only 10.9 GB that is usable, and an L1 cache with a size of 384 kilobytes (KB).

An iPhone 7 Plus with 3 GB RAM and iOS Software Version 15.1 was mainly used for running the application during development. Occasionally, a Redmi 9A with 2 GB RAM, Octa-core Max 2.00 GHz CPU, and Android version 10 QP1A was also used to check if the application and its components are compatible with an Android operating system.

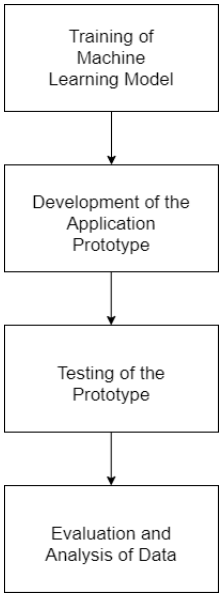


Fig. 1: The Development Process

2) Dataset & Video Preprocessing using OpenCV and OpenPose for the RNN: The bodyweight squat was the exercise selected for building the RNN model. The squat was selected because it utilizes most of the muscles in the body in executing the movement. It is also the exercise that has little variability in how it can be performed [14]. The dataset for training was sourced from DareBee, Kaggle, and YouTube. The videos from those sites were thoroughly inspected in order to build a machine learning model that learned the bodyweight squat form from reliable sources. The short video clips contained one repetition of the correct or incorrect form of the exercise.

With OpenCV and OpenPose, the data extracted from these videos consisted of the coordinates of all 18 key locations in the human body. To illustrate, Figure 2 shows a video where lines are overlaid on the human pose. These lines have endpoints that represent the key locations in the human body detected. Further, these key locations also come with its respective prediction confidence [10]. With these lines, the joint angles of the body were derived. This was done in order to eliminate the problems caused by inconsistent limb lengths and camera acquisition equipment [11]. This also eliminated the need to normalize the joint key points which was the case for PoseTrainer [10]. These joint angles served as the input data for the RNN model that will be trained. Further, the joint angles were calculated from the different joint angle pairs in the body which are namely:

1) Neck and Right Shoulder – Neck and Nose
2) Neck and Right Shoulder – Neck and Right Hip
3) Neck and Right Shoulder – Right Shoulder and Right Elbow
4) Neck and Left Shoulder – Neck and Nose
5) Neck and Left Shoulder – Neck and Left Hip
6) Neck and Left Shoulder – Left Shoulder and Left Elbow
7) Right Shoulder and Right Elbow – Right Elbow and Right Wrist

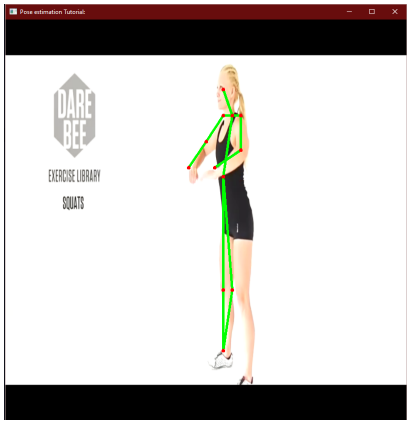


Fig. 2: Detecting Human Pose in a video using OpenPose and OpenCV

8) Left Shoulder and Left Elbow – Left Elbow and Left Wrist
9) Neck and Right Hip – Right Hip and Right Knee
10) Neck and Left Hip – Left Hip and Left Knee
11) Right Hip and Right Knee – Right Knee and Right Ankle
12) Left Hip and Left Knee – Left Knee and Left Ankle

3) Machine Learning Models: The machine learning models were developed using TensorFlow with Keras on Python. In this study, two models were developed:

1) Convolutional Neural Network-Recurrent Neural Network (CNN-RNN): For this model, the documentation presented by Sayak Paul entitled 'Video Classification with a CNN-RNN Architecture" [24] was used. In this tutorial, a general action recognition classifier was constructed with the UCF 101 dataset. The dataset is composed of "realistic action videos, collected from Youtube, having 101 action categories" [25]. The CNN portion of the model is responsible for processing spatial information for each frame in the video. On the other hand, the RNN is in charge of processing temporal information which, in this case, is the sequence of frames [24]. For this study, only three exercise classifications of the 101 action classes were selected, these are the Bodyweight Squats, Pushups, and Pullups. Hence, the dataset for this study's CNN-RNN only consisted of those exercises.

   The purpose of this CNN-RNN model is for identifying and verifying whether the uploaded video in the mobile application is the appropriate exercise for evaluation. For instance, since the application will only be evaluating squats, the CNN-RNN should return a classification of 'BodyWeight Squats' before the app proceeds to the evaluation performed by the next model. This model does not necessarily identify whether the exercise video exhibits the right form. Conversely, the RNN was trained to evaluate bodyweight squats however, it cannot determine if the video it is working on is actually a squat.

2) Recurrent Neural Network (RNN): Compared with the CNN-RNN model, the RNN does not need to make use of a CNN since the data it receives is just a sequence of numbers. In this study, the sequence of numbers are the joint angles of each body part for each frame of a video. The generated sequences of joint angles were then padded with zeros in order to make the sequence length of each frame to be uniform across all other sequences of frames. In this study, the length of each sequence is at 325 since it was the maximum number of frames generated from the sourced squat videos.

   After preprocessing, the joint angles dataset was fed to the RNN model for training. The model consisted of an input layer, a Gated Recurrent Units (GRU) cell, which is the RNN cell, followed by five hidden layers, and an output layer. This model was trained to classify whether the input joint angles yielded a squat that is CORRECT or WRONG. This RNN model was based on the tutorial for Recurrent Neural Networks (RNN) with Keras provided by TensorFlow [26].

After training and testing, both of the models were exported and loaded onto the server/API that interacts with the mobile application.

C. Mobile Application Server/Application Program Interface (API)

A simple web server was built using FastAPI, a high-performance Python-based API building framework [27]. Along with the trained models, the server also contained the different functions needed for preprocessing the videos. The video preprocessing and predicting were all performed server-side since these are computation heavy. The preprocessed video and the result of the prediction are all returned to the mobile application. The API was dockerized and deployed in Linode, an infrastructure-as-a-service platform.

D. Mobile Application

The user interface of the application was developed using React Native with TypeScript. This framework was used since it allows the development of applications that can run on different platforms whether mobile (Android or iOS) or desktop [28].

Illustrated in Figure 3 is the flowchart for this study. This flowchart was based on the application pipeline used by Pose Trainer [10]. After uploading the input video, the video is sent to the web server for the CNN-RNN classification. If the video is classified as a bodyweight squat, it will proceed to the next step which is the evaluation of the exercise. In this step, the joint angles for each joint angle pair are calculated for each video frame. The generated joint angles are sent to the RNN model for classification (Correct or Wrong). The output of the RNN model will be provided as feedback to the user.

Shown in Figures 4 to 6 are the low-fidelity wireframes or the skeletal structures of the graphical user interfaces of the application. Figure 4a shows the Home Screen of the

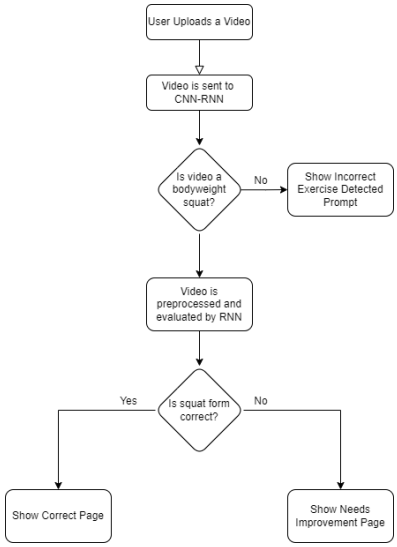
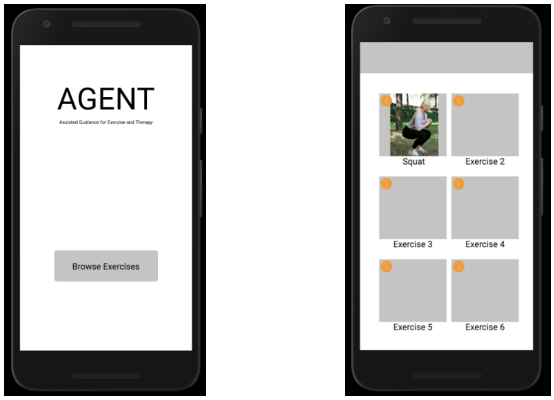


Fig. 3: Application Flowchart



(a) Home Screen (b) Browse Screen

Fig. 4: Main Page Wireframes

application where users only have the option to browse the exercises available in the application (see Figure 4b).

Figure 5 features a modal that appears when the user taps on the ”Check your form!” button. For this study, the only available option for uploading a video to the application is by accessing the mobile device’s gallery. Figures 6a and 6b shows the feedback pages of the application. Both pages would include the user’s uploaded video. As for the incorrect form screen, a ’Needs Improvement’ text will be shown rather than directly stating that the user’s form is wrong.

E. User Testing

Random respondents, regardless of their exercise background, were invited for testing the prototype. A demo video was provided to the respondents that served as a guide for using the application and the testing process. In testing the core feature of the application, the respondents were instructed to provide at least one video of them performing the squat as correctly as possible and another video intentionally performing the incorrect form. Performing the incorrect form will not pose any risk of injuries to the participants.

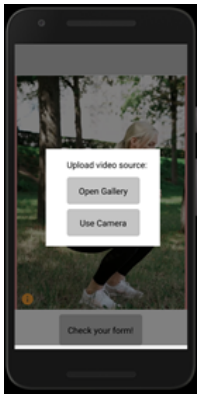
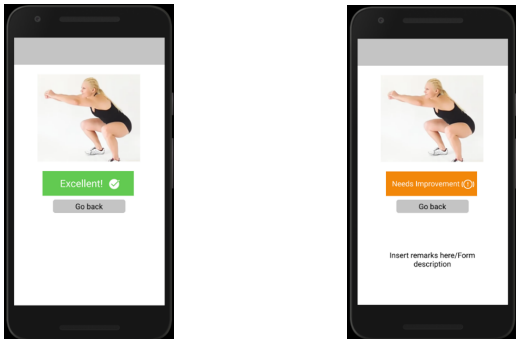


Fig. 5: Modal asking for video source



(a) Correct Form (b) Incorrect Form Screen

Fig. 6: Feedback pages

They were required to use their mobile devices or tablets for testing. After the application testing process, the respondents were interviewed for general feedback regarding the application. They were also requested to answer a Post-Study System Usability Questionnaire (PSSUQ). PSSUQ was developed in 1992 by IBM to conduct thorough system quality and user satisfaction research. It is composed of 16 questions that evaluate the system on three crucial metrics: System usefulness, Information quality, and Interface quality [29]. Each question in PSSUQ can be answered with a score ranging from 1 to 7 where 1 represents 'strongly agree', 4 is neutral, and 7 is 'strongly disagree'. This implies that the lower the score, the better the feedback is received from the system [30]. An additional option is also added which represents NA. In this study, the NA option was represented by the user skipping on a survey question. Additionally, open-ended questions were included in the survey which aims to gather their personal feedback regarding the application. The survey form questions are enumerated in Appendix II

V. Results and Discussion

A. Dataset & Video Preprocessing using OpenCV and OpenPose for the RNN

Sourcing videos that showed the incorrect form of a bodyweight squat was difficult. There were little to no videos that showed at least one complete repetition of an incorrect bodyweight squat form in real-time (meaning the video is not in slow motion, paused, or altered in any way). For this reason, the researcher provided a self-recording of one possible incorrect bodyweight squat form. This incorrect form includes the lower body rising first during the concentric movement of the squat instead of the entire body moving as one unit.

For the most part, OpenCV and OpenPose were able to detect the human pose in the videos collected online. However, the human pose was incorrectly estimated for the videos sourced from YouTube that contained the incorrect form. Shown in Figure 7 is a video taken from YouTube (video by Jeremy Ethier) which best illustrates this issue. The figure shows several factors affecting the accuracy of human pose estimation. One of these factors could be the surrounding materials found in the environment. For instance, the main subject of the video (the person performing the incorrect squat form) was surrounded by workout equipment making OpenPose to incorrectly detect those equipment as a body part (see Figure 8). Other possible causes would be the graphics shown on the screen that obscured the human pose from the video, and lastly, the human body not being completely shown on the screen (e.g., the head is not in the frame as shown in Figure 7). For this reason, the small set of incorrect form videos were removed from the dataset and only the self-recorded incorrect form videos were left for training.

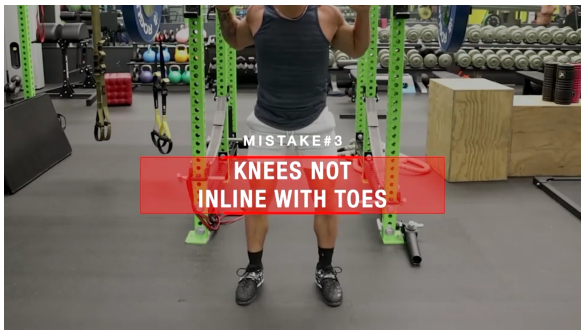


Fig. 7: Factors affecting human pose estimation accuracy

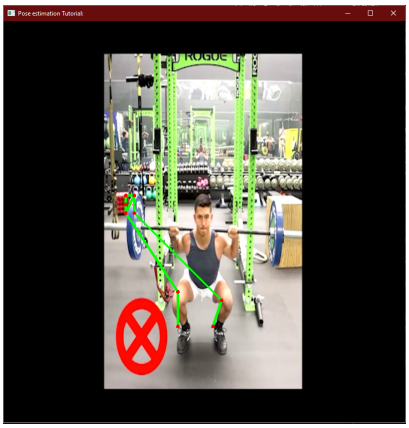


Fig. 8: Incorrect Human Pose Estimation

B. Machine Learning Models

Before the models were loaded onto the API, both were subjected to another set of videos to check their prediction capabilities. The CNN-RNN model was not able to classify exercises correctly during the first testing. The hyperparameters were adjusted in order to make the model more accurate. The parameters adjusted included the image size and the maximum number of frames it will be processing. For uniformity, the number of frames was set to 325. As for the RNN, it was observed that the model is strict when it comes to the video's length and the human's activity on each frame. Extra movements in the video and dead airs (i.e. no human movement occurring on subsequent frames) greatly affects the classification returned by the model. For this reason, during user testing, the requested videos from the respondents will be inspected in order to remove unnecessary frames that the model might process. The format of the videos should look similar to the training set wherein the human immediately performs the squat as soon as the clip starts and immediately ends after it has been performed. However, for future studies, it is recommended that the model would be lenient when it comes to the videos it will be classifying. Minor additional movements and brief pauses/dead-airs should not greatly affect the model's classification accuracy. This could be achieved by training the model with a larger video dataset.

During prototype usability testing, the 31 videos requested from the respondents were only squat videos. For this reason, evaluating the performance of the CNN-RNN model is only a matter of observing if it correctly classified the videos as a bodyweight squat or not. Interestingly, there were instances where the respondents attempted to upload random videos that did not contain squats. In all of those cases, the model did not respond with a classification of a Bodyweight Squat and thus it prompted a "Wrong Exercise Uploaded" error on the app. For all squat videos uploaded by the respondents, regardless of it being in correct or wrong form, the CNN-RNN model was able to classify them as Bodyweight Squats.

As for the Recurrrent Neural Network, Table I shows the resulting confusion matrix from 31 squat videos classified during user testing. It can be observed that there was a total of 17 videos that were classified as wrong where six of them were actually correct (false negatives) and 11 were actually wrong (true negatives). On the other hand, there was a total of 14 videos that were classified as correct despite three of those videos were actually wrong (false positives) and 11 were actually correct (true positives).

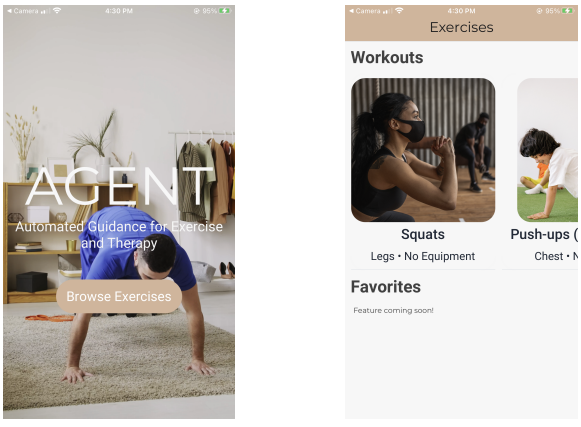
Furthermore, several rates can be derived such as the model's accuracy, true positive rate (or recall), precision, and F-score. The model's accuracy was calculated by dividing the sum of the true positives and true negatives with the total number of videos. This yields an accuracy of **0.71** which meant that the model's classification was accurate **71%** of the time. Next, we have the true positive rate or recall which was calculated by dividing the True Positive with the total number of videos that were actually correct. This gives a recall of **0.65** which meant that the model was able to classify videos as 'CORRECT' when those videos are actually 'CORRECT' **65%** of the time. As for the model's precision, it was calculated by dividing the true positive by the total number of videos that were classified as correct. This resulted to a value of **0.79**. This means that the model was able to correctly classify a video as "CORRECT" **79%** of the time. Generally, these values show that the model has better precision over its recall (true positive rate). Lastly, the RNN model had an F-score of **0.71**, which was calculated by getting the ratio of the product of the model's recall and precision with its respective sum. The resulting value was then multiplied by two. The F-score shows the weighted average of the precision and recall.

TABLE I: Confusion Matrix for the RNN model

	Predicted 'Wrong'	Predicted 'Correct'
Actual 'Wrong'	11	3
Actual 'Correct'	6	11

C. Mobile Application

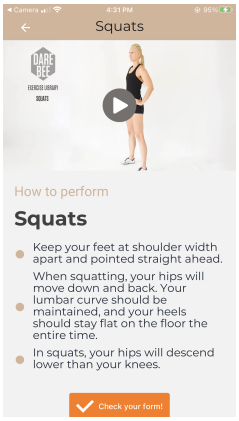
Shown in Figures 9a to 9b are the final Home Screen and Browse Exercises Screen, respectively. Compared to the browse screen illustrated in Figure 4b, the final Browse Screen features a horizontally scrollable collection of exercises for browsing. This is to make the screen less cluttered and to make room for other possible features (e.g. Favorited exercises). Regarding this subject, a possible feature for future development is to allow users to mark some exercises as favorites for easier access. Since this feature is outside the scope of this study, this was not included in the development and was only placed as a filler.



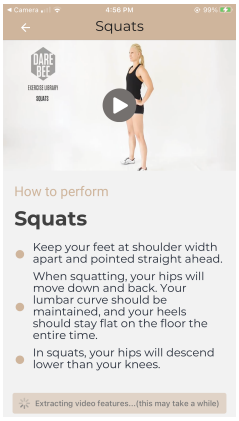
(a) Home Screen (b) Browse Screen

Fig. 9

When an exercise is selected, in this case, the bodyweight squat, the Exercise Screen would be displayed. In this screen, as shown in Figure 10a, the user would see a sample video for the exercise along with some guidelines for performing it. The user can tap on the 'Check your form!' button which will

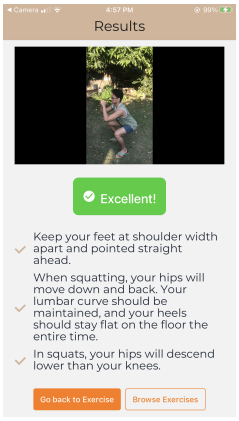


(a) Exercise Screen

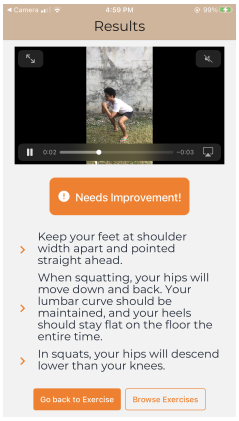


(b) Joint Angle extraction prompt

Fig. 10



(a) Correct Screen



(b) Needs Improvement Screen

Fig. 11

display an upload video modal/pop-up. The modal can be seen in the Appendix section I Figure 13a.

Using the application requires internet connection because the video will be uploaded and sent to the API for preprocessing and classification. During this time, the application would display a message prompting the user about what task the application is currently running as shown in Figure 10b. In this figure, the application is currently extracting the joint angles in the video they uploaded. After evaluation, the application displays the appropriate screen depending on the classification made by the RNN model. Figures 11a and 11b shows the 'Correct' and 'Needs Improvement' screens, respectively.

D. User Testing Results

Majority of the 15 respondents were around the age of 20-23 years old where 47% of them are physically active and are exercising regularly. All of them were shown video samples of the correct and incorrect squat form before asking for a pre-recorded video. A synchronous session was conducted to guide them through the testing process from start to finish.

Table II shows the average rating of the respondents from each of the prompts from the PSSUQ. (The order of the questions are in correspondence to the questions enumerated in Appendix II) It can be observed that the respondents are generally satisfied with the application's usefulness. With average ratings ranging from 1.10 to 1.71, the respondents agree that the mobile application is simple and easy to learn. They also felt comfortable navigating through the application during testing.

As for the quality of the information provided by the application, the ratings ranged from 1.13 to 2.43. This implies that the users are able to understand the information presented in the application whether it was the video, text, icons, and whatnot. However, the questions concerning the prompting of error messages and recovery from mistakes were relatively lower as compared to the others. This can possibly be attributed to some bugs that were discovered during user testing. One of these bugs is when a user selected the incorrect video to be uploaded and once they click on the 'Cancel' button, the 'Check your form!' button disappears. The users would have to redo everything from the beginning in order to re-upload their video. Another bug is when Android users uploaded videos that are longer than five seconds. The application allowed the users to still upload the video and the error only occurs during video analysis which should not be the case. The application should prompt the user right away that the video they selected were longer than five seconds which was the case for iOS devices.

The interface quality garnered average ratings that ranged from 1.33 to 1.80. This suggests that the users found the user interface of the application pleasant and are overall satisfied with how the different interactive components are positioned.

In general, the respondents are satisfied with the application. They especially liked the idea of them having their exercise form checked without having to go to the gym or hiring a professional trainer.

TABLE II: Average Scores from the PSSUQ

System Usefulness	
a	1.40
b	1.10
c	1.40
d	1.38
e	1.10
f	1.71
Information Quality	
a	2.43
b	1.93
c	1.40
d	1.21
e	1.14
f	1.13
Interface Quality	
a	1.40
b	1.33
c	1.80
d	1.47

Most of the respondents commented on the time it took for their videos to be analyzed. On average, the video analysis took around 20-35 seconds before they received a feedback. A possible solution to resolve this issue is by implementing multithreading during the extraction and calculation of the different joint angles in each frame. However, this may come

at a cost for the API's resource usage in the cloud. Another possible solution is by optimizing the preprocessing algorithms (since this was the most time consuming) without necessarily multithreading it.

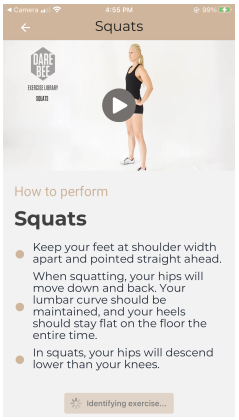
As for the mobile application, the respondents suggested a record feature wherein they can record themselves through the application itself. On top of that, they also wanted to have a video editor incorporated in the application. This is to cover cases wherein they need to trim the video length or crop frames. These features are highly recommended because it allows all the necessary tasks to be performed in one place instead of the user switching from one app to another. Additionally, the videos can be partially preprocessed on the device itself and the resulting output will be sent to the server for further processing. This is to cover some concerns on the user's privacy when uploading personal videos to the Internet. The model's can be exported into a format that is compatible with JavaScript/TypeScript so that the prediction can take place in the mobile phone instead.

VI. Conclusion and Future Work

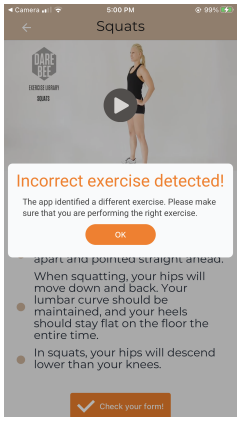
Overall, this study was able to make use of computer vision and machine learning to build a mobile application that provides feedback on users' exercise videos, specifically, the bodyweight squats. The application employed OpenCV and OpenPose for human pose estimation and joint angle calculation on each video frame. The videos that these libraries processed were recorded with a phone having a single camera. A machine learning model (RNN) was trained to identify whether a bodyweight squat was performed correctly or not. During testing, this model achieved an F-score of **0.71**. Lastly, based on the results of the PSSUQ, respondents were generally satisfied in using the application to learn whether their exercise form is done properly or not. They liked the idea of having their exercise form checked without having to go to the gym or hiring a professional trainer. Also, the application can be used by exercise experts to help them evaluate and teach their clients about the proper exercise form. However, the application still needs further tuning of the model's accuracy as well as improving its video preprocessing runtime.

For future work, the RNN can be improved by using a larger dataset that contains an ample amount of correct and incorrect forms for training. The possibility of using a CNN-RNN for correct and incorrect form can also be used instead of calculating the close angles. Multithreading can be employed to improve video preprocessing especially for joint angle calculation. However, resource usage must be taken into account. Finally, other exercises and yoga poses can be included in the application.

Appendix I

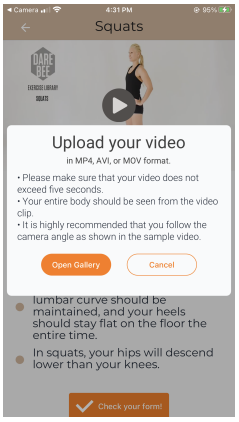


(a) Identifying the exercise

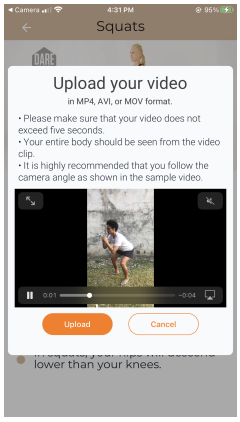


(b) Incorrect Exercise detected modal

Fig. 12



(a) Upload video modal



(b) Video preview before uploading

Fig. 13

Appendix II
AGENT: Automated Guidance for Exercise & Therapy Survey Form Questions

PSSUQ Questions:

1) System Usefulness
   a) Overall, I am satisfied with how easy it is to use this system
   b) It was simple to use this system
   c) I was able to complete the tasks and scenarios quickly using this system
   d) I felt comfortable using this system
   e) It was easy to learn to use this system
   f) I believe I could become productive quickly using this system
2) Information Quality
   a) The system gave error messages that clearly told me how to fix problems
   b) Whenever I made a mistake using the system, I could recover easily and quickly

c) The information (such as online help, on-screen messages, and other documentation) provided with this system was clear
d) It was easy to find the information I needed
e) The information was effective in helping me complete the tasks and scenarios
f) The organization of information on the system screens was clear

3) Interface Quality
a) The interface of this system was pleasant
b) I liked using the interface of this system
c) This system has all the functions and capabilities I expect it to have
d) Overall, I am satisfied with this system

General Feedback Questions:

1) What did you like THE MOST about AGENT?
2) What did you like THE LEAST about AGENT?
3) What improvements would you suggest for AGENT?

REFERENCES

[1] "Benefits of exercise," Sep 2021. [Online]. Available: https://medlineplus.gov/benefitsofexercise.html

[2] J. E. Bielecki and P. Tadi, "Therapeutic exercise," Sep 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK555914/

[3] K. F. Kozlowski, C. Lopata, J. P. Donnelly, M. L. Thomeer, J. D. Rodgers, and C. Seymour, "Feasibility and associated physical performance outcomes of a high-intensity exercise program for children with autism," *Research Quarterly for Exercise and Sport*, vol. 92, no. 3, p. 289–300, 2020.

[4] M. S. Samsir, R. Zakaria, S. Abdul Razak, M. S. Ismail, M. Z. Abdul Rahim, C.-S. Lin, N. M. Nik Osman, M. A. Asri, N. H. Mohd, A. H. Ahmad, and et al., "Six months guided exercise therapy improves motor abilities and white matter connectivity in children with cerebral palsy," *Malaysian Journal of Medical Sciences*, vol. 27, no. 5, p. 90–100, 2020.

[5] R. Brand, S. Timme, and S. Nosrat, "When pandemic hits: Exercise frequency and subjective well-being during covid-19 pandemic," *Frontiers in Psychology*, vol. 11, 2020.

[6] H. Kaur, T. Singh, Y. K. Arya, and S. Mittal, "Physical fitness and exercise during the covid-19 pandemic: A qualitative enquiry," *Frontiers in Psychology*, vol. 11, 2020.

[7] A. Chauhan, "The rise of exercise-related injuries during the pandemic," Aug 2021. [Online]. Available: https://tdotcommunity.ca/project/the-rise-of-exercise-related-injuries-during-the-pandemic/

[8] Rappler, "Survey says 67% of filipinos move less. here's why you should be concerned," Oct 2021. [Online]. Available: https://www.rappler.com/brandrap/health-beauty-and-wellness/less-exercise-filipinos-covid-19-pandemic-anlene-survey

[9] D. A. Y. Puen, R. A. Camarador, H. C. Dimarucot, and A. G. C. Cobar, "Perceived barriers to physical activity of college students in manila, philippines during the covid-19 community quarantine: An online survey," *Sport Mont*, vol. 19, no. 2, p. 101–106, 2021.

[10] S. Chen and R. R. Yang, "Pose trainer: Correcting exercise posture using pose estimation," 2020.

[11] L. Zhu, "Computer vision-driven evaluation system for assisted decision-making in sports training," *Wireless Communications and Mobile Computing*, vol. 2021, p. 1–7, 2021.

[12] T. Hellsten, J. Karlsson, M. Shamsuzzaman, and G. Pulkkis, "The potential of computer vision-based marker-less human motion analysis for rehabilitation," *Rehabilitation Process and Outcome*, vol. 10, p. 1–12, 2021.

[13] J. Dorado, X. del Toro, M. J. Santofimia, A. Parreño, R. Cantarero, A. Rubio, and J. C. Lopez, "A computer-vision-based system for at-home rheumatoid arthritis rehabilitation," *International Journal of Distributed Sensor Networks*, vol. 15, no. 9, p. 155014771987564, 2019.

[14] C. Conner and G. M. Poor, "Correcting exercise form using body tracking," *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2016.

[15] S. Pandit, S. Tran, Y. Gu, E. Saraee, F. Jansen, S. Singh, S. Cao, A. Sadeghi, E. Shandelman, T. Ellis, *et al.*, "Exercisecheck: A scalable platform for remote physical therapy deployed as a hybrid desktop and web application," *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2019.

[16] H. Hagihara, N. Ienaga, D. Enomoto, S. Takahata, H. Ishihara, H. Noda, K. Tsuda, and K. Terayama, "Computer vision-based approach for quantifying occupational therapists' qualitative evaluations of postural control," *Occupational Therapy International*, vol. 2020, p. 1–9, 2020.

[17] A. Nagarkoti, R. Teotia, A. K. Mahale, and P. K. Das, "Realtime indoor workout analysis using machine learning & computer vision," *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019.

[18] K. Takeichi, M. Ichikawa, R. Shinayama, and T. Tagawa, "A mobile application for running form analysis based on pose estimation technique," *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2018.

[19] H. Jeon, Y. Yoon, and D. Kim, "Lightweight 2d human pose estimation for fitness coaching system," *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, 2021.

[20] E. Odemakinde, "Human pose estimation with deep learning - ultimate overview in 2021," Jan 2021. [Online]. Available: https://viso.ai/deep-learning/pose-estimation-ultimate-overview

[21] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, p. 172–186, 2021.

[22] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," 2020.

[23] S. Choi, S. Choi, and C. Kim, "Mobilehumanpose: Toward real-time 3d human pose estimation in mobile devices," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021.

[24] S. Paul, "Keras documentation: Video classification with a cnnrnn architecture," May 2021. [Online]. Available: https://keras.io/examples/vision/video_classification

[25] U. C. for Research in Computer Vision. [Online]. Available: https://www.crcv.ucf.edu/data/UCF101.php

[26] "Recurrent neural networks (rnn) with keras & tensorflow core." [Online]. Available: https://www.tensorflow.org/guide/keras/rnn

[27] Z. Ahmed, "Build high-performing apps with python – a fastapi tutorial," Sep 2020. [Online]. Available: https://www.toptal.com/python/build-high-performing-apps-with-the-python-fastapi-framework

[28] M. Budziński, "What is react native? complex guide for 2021." [Online]. Available: https://www.netguru.com/glossary/react-native

[29] TryMyUI, "Pssuq psychometric — trymyui usability testing." [Online]. Available: https://www.trymyui.com/pssuq

[30] W. T, "Pssuq (post-study system usability questionnaire)." [Online]. Available: https://uiuxtrend.com/pssuq-post-study-system-usability-questionnaire/#: :text=PSSUQ%20score%20starts%20with%201,product%20have%20p

Kevin A. Gines is a 4th year BS Computer Science student in the University of the Philippines Los Baños. He is interested in Software Engineering, Artificial Intelligence, and Financial Technology. His hobbies include reading books, playing online video games, exercising, and coding. His favorite Netflix series are My Name, Peaky Blinders, and Suits.