Development of a Mobile Application for Real-Time Push-Up Progression Feedback via Angle Heuristics

John Achapero Jr*, Jose Antonio Bolado, Richelle Ann Juayong, Jozelle Addawe, and Jaime Caro

Service Science and Software Engineering Laboratory,
Department of Computer Science, College of Engineering,
University of the Philippines Diliman, 1101, Quezon City, Philippines,
jcachapero@alumni.up.edu.ph

Abstract. The push-up is an accessible, low-equipment exercise where improvement comes through changes in positioning and leverage through progressions, rather than increasing weight. To produce optimal results from performing push-ups, one requires optimal form, traditionally advised through coaching. As coaching can sometimes be inaccessible, form feedback can be received from applications instead. One such mobile application is Progress-Up, which provides real-time form feedback specifically for push-up progressions, an area of training neglected by other similar applications. Progress-Up leverages modern human pose estimation and form-feedback techniques, and was tested on five different push-up progressions. While the results show promise for specifically tackling progressions, there remain challenges in the design and current technology that make the application impractical to use. Thorough examination of the application's performance provides a detailed breakdown of design weaknesses and strengths, alongside corresponding recommendations for improvement in future studies.

Keywords: angle heuristics, human pose estimation, push-ups, progressions

1 Introduction

Calisthenics is a form of resistance training that primarily uses body weight for resistance, making it one of the most accessible forms of strength training by requiring minimal equipment. A key principle of resistance training is *progression*, where exercises are gradually made more difficult to improve one's strength and endurance. In a standard gym setting, progression is quite straightforward, simply done by loading more (or less) weight onto the equipment being used. In calisthenics, however, progressions rely more on adjustments in the body's positioning, typically to increase or decrease leverage in the exercise.

Take the **push-up**, a fundamental calisthenics exercise with a wide range of progressions that can cater to different fitness levels. For example, knee push-ups, standard push-ups, and decline push-ups represent push-ups of increasing

J. Caro et al. (eds.), *Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2025)*, Atlantis Highlights in Computer Sciences 24,
https://doi.org/10.2991/978-94-6239-638-8_12

difficulty, requiring the user to push increasing percentages of their body weight [3]. Not only do these progressions provide a clear path for improvement in one exercise, but they can also act as stepping stones to aid in performing more challenging movements. An example of such is the pseudo-planche push-up, which serves as an intermediate step toward the full planche [4, 9].

While calisthenics is a highly accessible field, obtaining expert feedback on exercise form typically requires a personal trainer. This can be quite costly, and creates a potential financial barrier to many interested individuals. To solve this, recent studies and existing applications leverage computer vision to provide real-time feedback for a variety of exercises, including the push-up.

12 form-feedback implementations were analyzed by Achapero et al. [1], where it was established that a critical research gap exists: these tools are limited to *standard movements* of exercises and *neglect their various progressions.* This omission becomes a significant technological barrier the moment a user attempts to scale the difficulty (e.g., advancing from the standard push-up to the decline push-up). Individuals pursuing self-guided training are left to attempt more complex movements without objective form correction, increasing the risk of injury. This gap demonstrates that current technology democratizes only part of the fitness journey, but fails to support the necessary process of improvement and mastery.

To address this gap, this study develops an implementation of **"Progress-Up": a mobile application designed to provide real-time form feedback for push-up progressions**, originally established by Achapero et al. [1]. This application aims to improve the accessibility of calisthenics by offering a cost-effective alternative to personal trainers. By integrating feedback on various push-up progressions, Progress-Up can support beginners and enthusiasts alike, allowing them to select and master the appropriate exercise for their current physical capabilities. This not only makes resistance training more accessible, but also helps ensure users maintain proper form, improving exercise effectiveness and reducing the risk of injury.

2 Methodology

The following sections describe the design and implementation of the system, its corresponding design limitations, and the testing procedure performed for analysis of system accuracy.

2.1 System Architecture

Progress-Up was developed following a three-stage approach to evaluate user push-up form, as identified in a review conducted by Tharatipyakul et al. [12] on computer vision applications that evaluate movement: **human pose estimation**, **movement assessment**, and **augmented feedback**. Further technical and implementation details of the system design are discussed in the corresponding study conducted by Achapero et al. [1].

Human Pose Estimation Given a video feed of a user performing push-ups, the first task of the system is to acquire usable data from these images. To do so, Progress-Up utilizes a **human pose estimation (HPE)** model, a computer vision model that generates labeled keypoint coordinates corresponding to certain predefined body parts.

Progress-Up specifically uses MoveNet Thunder, a 2-dimensional HPE model from the TensorFlow library. This is thanks to its lightweight performance on mobile devices and higher accuracy than 3-dimensional HPE counterparts [2]. MoveNet outputs 17 labeled (x,y) keypoints, as seen in Fig. 1, which are used to infer the user's positioning in the following phase.

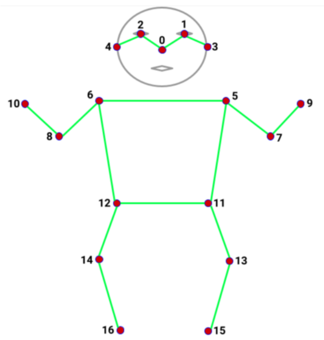

Fig. 1. Keypoints generated by MoveNet

Movement Assessment Using the keypoints obtained from the previous phase, the system now needs to evaluate the user's correctness in their push-up form. Progress-Up employs an approach using **angle heuristics**; through the use of the labeled coordinates, certain angles of and metrics of interest can be calculated and used to infer the user's position [13]. This approach shows to be the preferred method of form evaluation using HPE, as evidenced by numerous studies that tackle push-ups and other exercises [6, 5, 8, 10, 11].

In order to implement angle heuristics, however, it is first necessary to create a mapping between (a) the qualitative form cues for the various push-up progressions, and (b) the quantified metrics that can be measured by the system via HPE. The qualitative form cues for the push-up were primarily based on Low's *Overcoming Gravity* [9] and Kritz's *Movement Competency Screen* (MCS) [7], a sample of which can be seen in Table 1. These qualitative form cues were then converted to corresponding quantitative metrics (defined with the guidance of an expert on movement analysis and biomechanics) for different form cues in different progressions. A sample of the quantified metrics specific to the standard push-up can be found in Table 2.

Defining the quantified metrics per form cue allows the system to mimic the visual inspection performed by personal trainers to identify errors in user

Table 1. Standard push-up form cues from the MCS

Body Part/Aspect	Form Cue
Head	Centered
Shoulders	1) Hands under shoulders 2) Depressed 3) Moving rhythmically throughout the movement
Lumbar	Neutral
Hips	1) Aligned with trunk 2) Stable
Knees	Stable
Ankles	Aligned
Feet	Aligned
Depth	Chest touches the floor
Balance	Maintained

form *with specificity.* For example, one of the push-up's standard form cues is that the individual's back should remain straight and aligned throughout the exercise. Should the user's back greatly sag or arch at any point in the push-up motion, the specific violated form cue can be flagged, and the user can be informed appropriately.

Table 2. Quantified Metrics for the Standard Push-up

Form Cue	Quantified Metric
Head centered, lumbar neutral, and hips aligned with trunk	The angle created by the head, shoulder, and hip keypoints is approximately 180°
The hips and legs should be aligned and straight	The angle created by the shoulder, hip, and knee keypoints is approximately 180°
At the starting position, the elbows should be straight and locked out	The angle created by the shoulder, elbow, and wrist keypoints is approximately 180°
Knees should not be bent (except for knee push-up)	The angle created by the hip, knee, and foot keypoints is approximately 180°
In the middle of the movement, the chest should be close to touching the floor	The distance between the shoulders keypoints and the ground should be minimized
Hands remain directly under the shoulders	Hands and shoulders should share a similar x-coordinate
Feet and hands should be level and aligned	Hands and feet should share a similar y-coordinate

Augmented Feedback Once the user's form has been evaluated, the system's final task is to relay the necessary information back to the user to correct any errors in form. Progress-Up does this in two ways: **real-time feedback**, and **post-exercise session logs**.

Real-time feedback is provided to the user via two mediums, visual and auditory. Given any detected errors in form through the previous phase, the system displays the list of mistakes at the end of each repetition. This gives the user the opportunity to adjust and correct their form immediately as they continue with the workout session. Additionally, users may also enable optional beeping sounds, with a confirmatory beep signaling perfect form throughout a repetition, and a light buzzer to signal that one or more errors was detected.

In addition to real-time feedback, Progress-Up also provides a summary of the user's completed workout sessions, as can be seen in Fig. 2. This allows the user to view their most common mistakes throughout the session, as well as an itemized breakdown of all repetitions performed. Having both methods of feedback equips the user with a better understanding of their own personal rooms for improvement in performing the different push-up progressions.

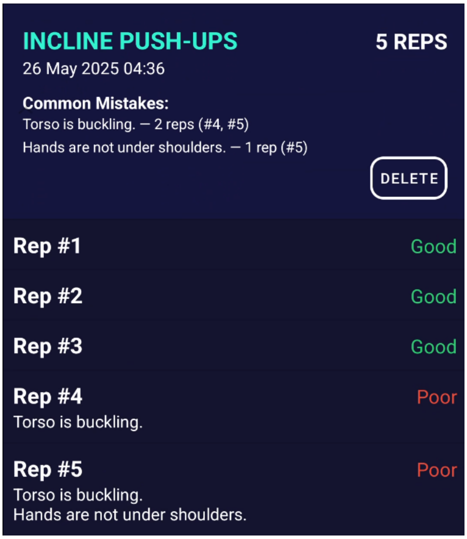

Fig. 2. Workout summary for a tracked session

2.2 Design Limitations

Chosen Progressions To cover a broad variety of exercise difficulty, a limited but representative selection of push-up progressions has been chosen, consistent with typical strength training regimens. These specific progressions, enumerated in the following list and visually represented in Fig. 3, were selected for their feasibility in participant testing and amenability to biomechanical analysis via angle-based heuristics [1].

1. Wall push-up
2. Incline push-up
3. Knee push-up
4. Standard push-up
5. Decline push-up
6. Pseudo-planche push-up

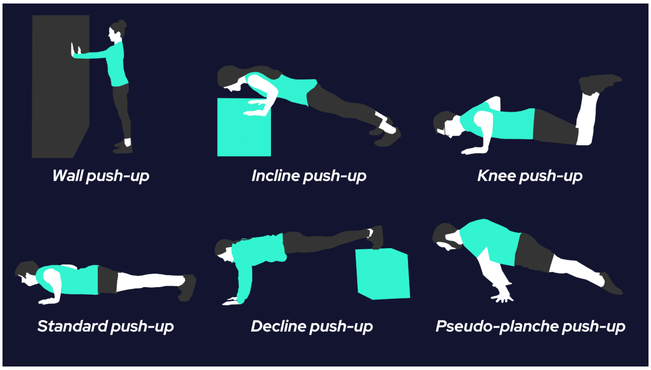

Fig. 3. Chosen push-up progressions for Progress-Up

Usage To ensure application consistency during operation, the application was created with the intention that its usage would not go beyond the following assumptions:

1. **User count:** The application assumes that a single user will be interacting with the application throughout its usage. Thus, only one person should be in frame, and no functionality supports the differentiation of different users (either through form checking or database logging).
2. **Positioning:** The user's whole body (hands, feet, and head) should be in frame, with no parts cut out, either fully or partially.
3. **Distance:** The user should be close enough to the camera such that they fill up the majority of the frame while still remaining fully in frame.
4. **Angle:** The application assumes that the frame records the user in a specific, lateral/side-view angle, where the body is parallel to both the ground and the camera lens. Examples can be seen in Section 3.
5. **Movement:** The camera, while being used by the application, should remain still and unmoving; that is, only the user within the frame is allowed to move, and the frame itself should not move.

Form Cues While the benefits of constraining usage are obvious, it also comes with inherent drawbacks, similar to the choice of using 2D HPE over other techniques. When it comes to usage, the use of the side camera angle, while chosen for its large coverage of push-up form cues, cannot cover them all. As an example, observe the "head" form cue in Table 1, which informs that the head be centered. While the side camera angle can see if the head is centered forward and back, it becomes much more difficult to tell if the head is centered side-to-side. Thus, the side-to-side centering of the head and other similar form cues difficult to distinguish from the side are not under consideration for the application's form checking system.

Changes to the body which are obscured by other parts of the body are also impossible to account for. For example, the archer push-up involves keeping one arm straight and away from the body while using the other arm to push the body up. When the arm further from the camera is kept straight and the body approaches the ground, the arm becomes obscured entirely, leaving the application to guess its position, resulting in faulty and incorrect form feedback.

The use of 2D HPE also reduces the user to a limited number of keypoints. This lack of resolution will affect the form cues that the application can handle. For example, as MoveNet does not distinguish the position of the shoulder blades (only having keypoints for the shoulders themselves), form cues involving scapular depression, retraction, and protraction cannot be accounted for (see the "shoulders" aspect in Table 1).

Occlusion Even with form cues tackled by the application, 2D HPE still struggles with camera occlusion. When the body is obscured by the camera (for example, when the arm further from the camera is obstructed by the arm closer to the camera in the push-up), then the model is left to assume where certain parts of the body are, leading to undefined behavior. Attempts can be made to remedy this, such as increasing the degrees of freedom assigned to further body parts, but these methods are not foolproof. Occlusion is a well-understood limitation of 2D HPE; for example, the same errors are noted by Rahmadani et al. [11] in reference to bicep curls when seen from the side.

Environment According to Gamra and Akhloufi [2], in uncontrolled environments it is still preferable to use 2D HPE versus 3D HPE. However, there are still environmental conditions in which a 2D HPE model like MoveNet Thunder will struggle to perform. For example, when the color of the user's clothes are similar to the color of the background, or when the application is used in locations where lighting is minimal. While many assumptions on the position of the body and the camera can be made during development, it is less practical to make assumptions on the body's clothing and the camera's lighting conditions, thus this remains a weak point of the application and its usage.

2.3 Testing Procedure

Participant Recruitment Participants were chosen through a sign-up form such that they meet the following criteria:

- Should have at least six (6) months of experience in resistance training

- Should be familiar with performing at least 3 consecutive repetitions of any subset of the relevant progressions: wall, incline, knee, standard, decline, pseudo-planche
- Should have no recent history of physical injuries, especially in the wrists, elbows, or shoulders
- Must willingly consent to the testing procedure through the consent form
- Can attend a face-to-face testing session at a designated location

The objective of the functional testing is to obtain at least 30 repetitions per progression, for a total of 180 correct-form reps. As not every participant can perform multiple reps and not every participant can perform every exercise, a specific number of participants will not be set.

Testing Proper The testing procedure proper occurs as follows:

1. Select a progression (listed in Subsection 2.2) where the participant is able to perform at least three consecutive repetitions.
2. Set up the testing apparatus (application, camera) and position the participant accordingly.
3. Perform a warm-up procedure with the participant.
4. For every chosen progression, perform at least three repetitions of the corresponding push-ups. The participant is free to perform the repetitions at a comfortable pace.
5. If performing multiple progressions, rest for at least three minutes between each set.
6. Once finished, perform light cool-down stretches with the participant.

Participants were given minimal cues about the correct form to prevent bias. As the screening requires that they have previous experience with the progression, this allows a fair evaluation of the application functionality without great risk of injury.

During testing, the testing device running the Progress-Up application was also recording its screen. While this slightly impacts application performance, it is worth the trade-off of being able to investigate the application's behavior in greater detail.

Domain Expert Validation In order to validate the application's functionality, domain experts with knowledge on the push-up motion were contacted to review the footage taken from the secondary camera. They were provided a per-repetition checklist of form cues, and tasked to provide feedback on the form found in the secondary camera footage according to the form cues listed on the checklist.

In order to prevent bias, the recorded application footage was not shown to the domain experts. They were only provided with the raw camera footage.

3 Results & Discussion

The testing procedure was implemented succeeding the application's development. The following sections discuss information relevant to the testing proper and its associated findings.

3.1 Testing Limitations

Worth discussing are inherent limitations in the testing design which may result in biases with regards to the results being analyzed. These biases may skew the discussion or analyses one way or another, and should be acknowledged throughout the discussion of the application.

- **Lack of negative rep testing**: Based off consultations with domain experts, testing procedures seen from other studies, as well as general safety precautions, the testing procedure indicated in Subsection 2.3 was designed such that no tester is *forced* to perform poor-form repetitions. This fails to rigorously test poor-form related functionalities of the application, and may skew the testing data accordingly.
- **Testing conditions only emulate ideal-case scenario**: Testing was performed in a laboratory, thus lighting is well controlled and backgrounds are close to white. While these are not factors specified in Subsection 2.2, they are nonetheless the testing conditions where functional testing took place in, therefore are worth mentioning as a potential source of bias.
- **Limited tester variety**: While consultations did not raise any questions on the testing metric of counting reps over counting testers, results reveal that errors tend to be set or user dependent rather than rep-dependent. That is, errors present in one rep tended to be consistent throughout the rep's entire set or for all sets by that tester. Thus, certain form errors may be unintentionally favored throughout the testing procedure.

3.2 Results Overview

A total of 209 repetitions were performed and recorded as data for analysis, spanning 5 of the 6 progressions. Due to safety concerns and strict tester requirements, as informed by domain experts, no testers for the pseudo-planche push-up were found. While the functionality for the pseudo-planche push-up already exists in the application, it will not be evaluated.

Using the application screen recording, each repetition was manually logged into a spreadsheet, as well as each error and repetition counted and logged by the application. The raw camera footage was shown to domain experts and analyzed manually. Lastly, the feedback generated by the application and provided by the domain experts were compared and tallied for similar results.

A summary table of the results can be found in Table 3. For the purposes of the summary table, a rep will be considered "accurate" if the application aligned with the expert's feedback *for form cues accounted by the application*. Feedback provided by experts for form cues which have been intentionally excluded from the application's design as indicated in Subsection 2.2 will not be considered under accuracy, however they will still be noted for analysis during discussion. This style of results analysis is based on the work of Rahmadani et al. [11]

Table 3. Summary table of results from functional testing

Summary	Wall	Incline	Knee	Standard	Decline	Total
Total reps	31	39	38	52	49	209
Uncounted reps	5	0	3	10	4	22
Total counted reps	26	39	35	42	45	187
Rep count accuracy	83.87%	100.00%	92.11%	80.77%	91.84%	89.47%
Accurate reps	19	34	14	16	33	116
Accuracy vs real*	61.29%	87.18%	36.84%	30.77%	67.35%	55.50%
Accuracy vs counted**	73.08%	87.18%	40.00%	38.10%	73.33%	62.03%

**accuracy when compared with the total number of repetitions*

***accuracy when compared with only the counted number of repetitions*

As is visible in Table 3, the application was only capable of counting 89.47% of all repetitions and only accounted for 55.5% of all real feedback cues, implying that for a significant amount of repetitions, the application feedback and the domain expert feedback were not aligned. To further analyze the discrepancy between the two, the screen recording footage and raw camera footage was once again scoured in order to create a plausible line of reasoning for the inaccuracies.

Each inaccuracy was manually denoted with its corresponding potential reason. Potential reasons could be categorized according to root cause, with analysis giving rise to the following large categories:

- **Errors due to the model**
  - Model detection issues caused by baggy clothing
  - Model detection issues caused by occlusion
- **Errors due to application design**
  - Problems due to repetition detection mechanics
  - Issues due to the position of the camera
  - Errors due to the application being too lenient or too strict
- External errors outside of the application, such as user error

3.3 Errors Due to Application Design

An overwhelming majority of inaccurate repetitions can be attributed to issues with the design (totaling 72.04% of all repetitions), with 68.8% being due to issues with leniency.

The human body struggles to maintain completely straight lines. It is reasonable for body angles to have a degree of leniency when being analyzed, particularly due to movement as the progression is performed. However, the application struggles with leniency as repetitions with good form are sometimes incorrectly tagged with form errors which are due to these slight variations in angle. Likely attributing to the large proportion of design-related errors are the existence of multiple possible reasons for these issues:

1. Degrees of freedom are not correctly contextualized. The body will naturally find it easier to bend in certain ways (for example, the body is far better at bending forward than bending backward). Currently, degrees of freedom are applied universally to bending both ways, when bends opposite the natural bend of the body should be made much stricter comparatively. An example can be seen in Fig. 4 (left), where the slight back bend should have been caught as an error.
2. The application fails to consider different proportions. For example, the lateral view of the body can have the camera anywhere along the body's side. For most users, placing the camera at the median point (equidistant from head and feet) results in expected behavior. However, for long body proportions, placing the camera closer to the midsection rather than closer to the shoulder results in "fisheyeing" of the lens, resulting in straight angles of the arms not being seen as straight as they are no longer in line with the camera. An visual example can be seen in Fig. 4 (right), where this effect is visible in the tester's right arm.
3. In order to analyze the form of a user, Progress-Up creates a specific order of operations for counting a repetition: start, middle, end, return to start. Thus, the application must always wait for these to occur *in order*. When a user tends towards incorrect form in a repetition, it is likely they will not be in good form when they return to the start of the position. However, the start position is clearly defined in Progress-Up to have good form. Therefore, when a user fails to return to a good start position, repetitions are incorrectly counted.
4. The application's form analysis metrics are reliant on achieving the testing setup indicated in Subsection 2.2. While the testing setup was made to be as consistent as possible with the help of tripods and floor markers, it is possible that slight inconsistencies between recorded sets (such as slight tilt in the camera) resulted in unintended behavior that cannot be clearly seen in the footage.

3.4 Errors Due to Model

Model-related errors account for 25.81% of all errors, with 23.7% attributed to occlusion errors. As indicated by Subsection 2.2, occlusion is a well-known source of error for 2D HPE projects, of which Progress-Up is no exception. Fig. 5 displays an example where the hip further from the camera is being hallucinated

Fig. 4. Examples of errors due to leniency (left) and "fisheyeing" (right)

Fig. 5. Examples of errors due to occlusion (left) and baggy clothing (right)

by the model (left). Other model-related issues include problems with baggy clothing, where the knees begin to bend unnaturally (right).

Unfortunately, there is little Progress-Up can do about model-related issues, other than choose a model with greater accuracy or disregard occluded parts of the body entirely. As expected, however, there will be drawbacks to choosing either option, such as worse mobile performance or less effective form checking.

3.5 External Errors

A small portion of the feedback errors can be visually attributed to the users acting in ways outside of the application's expectation, such as scratching their nose or raising a hand. A visual example can be found in Fig. 6, which shows user movement performed before the set properly began, counting as an incorrect repetition. While the application is correct in that this is incorrect push-up form, it will never be able to tell exactly what is wrong with the push-up in a way that provides meaningful feedback. These minor movements and adjustments result in unpredictable behavior, causing counting or feedback issues.

Beyond what can be accurately categorized by the authors are errors due to the performance of the testing device. Progress-Up is inherently designed under the assumption that the user can be seen all throughout the entire motion. Unpredictable behavior can also be a result of certain key frames being missed due to stutters in the application or hardware limitations of the testing device.

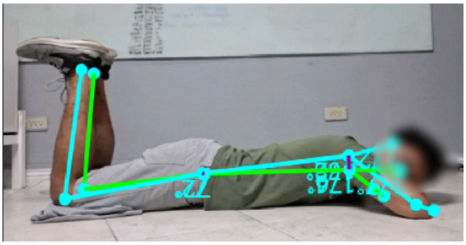

Fig. 6. Example of an error due to user movement

3.6 Domain Expert Feedback

As hinted in Subsection 3.2, not counted in the accuracy of the application are form cues which are not accounted by the application. However, that did not stop expert evaluation from evaluating those form cues anyway, indicating that they are both important and visible from the same camera angle as the application. Common form cues brought into attention by experts include (1) the positioning of the neck, (2) the flexion of the feet, and (3) tightness of the core and glutes.

A big benefit of expert feedback is feedback specificity. Even through a limited camera angle, experts were capable of providing feedback far more detailed than that provided by Progress-Up. Examples include:

– Making the connection between body positions relative to the body's height (i.e., this tester is tall, therefore this wall push-up form seems poor because they should be further from the wall)
– Connections between different form cues (i.e., this tester's range of motion is poor because their neck is incorrectly positioned)
– Justifications for certain form cue errors (i.e., this tester is strong and this progression is too easy, therefore they have no reason to activate their core)

In comparison, the feedback provided by Progress-Up, while also targeting specific aspects of the body, is very simplistic in nature. As an example, Progress-Up may state "Poor range of motion", but the experts can specify that "the neck is too flexed, limiting the range of motion".

Worth noting, however, are form cues where experts also struggled. For example, certain testers had clothing which would obscure the midsection, causing the experts to be unsure whether their form errors were due to incorrect form or simply due to the clothing *appearing* as if the form was incorrect. This includes form cues intentionally left out by the application; for example, the experts also found it difficult to check for the cues such as *elbows flaring* and *distance between hands*, given the camera angle. This gives greater confidence to the design choice of leaving such form cues out—if experts struggle to account for such, then it is unlikely that the application can.

The final learning to be gained from the experts are the methods in which they assess form. While experts have access to intensive knowledge of the body's anatomy (which an application is unlikely to have), there are still techniques to be learned from them. For example, experts expressed that the way they check if a push-up is deep enough is by observing the angle of the elbow and whether or not it exceeds 90°, a specific metric Progress-Up does not check for.

3.7 Synthesis

While Progress-Up has displayed its strength through its mobile-first lightweight design, limitations of both the sole use of angle heuristics as well as the choice of model restrict the capability of the application to provide for its intended use case.

Testing has shown that it is difficult to achieve both accessibility and accuracy with the system design when keeping in mind consumer hardware. Additionally, inconsistencies with the model and strictness of the system design make the application unreliable to be used in a real-world setting, particularly when compared to the trained eye of a coach or domain expert.

The primary contribution of Progress-Up is the application of 2D HPE and angle heuristics specifically to push-up progressions. Despite the shortcomings of the application and its implementation, the underlying mechanics behind the system still show great promise, particularly in its potential when combined with other technologies to make up for where it lacks.

4 Recommendations

To increase the application's accuracy, options exist for providing the application with more information on the user's position. A simple example would be upgrading to a better model, one more accurate or which tackles more keypoints, but doing so may result in performance issues. Another consideration is the inclusion of a secondary camera angle to provide greater detail for other form cues. There may also be potential in the use of an initial standing scan of the user to be used as reference for the body's proportions, ensuring they do not stray too far from the norm as seen in Fig. 5. More drastic options include switching model types entirely, such as making the transition to 3D HPE, or using a 2-model approach as suggested by Zhang et al. [13].

Future researchers should keep in mind the feasibility of access to testers. This tackles multiple shortcomings, one of which is the lack of a minimum tester count (as indicated in Subsection 3.1) which affects the study's error bias. Another of which is the lack of functional testing done on the pseudo-planche push-up due to its technicality—most participants were completely unaware of the progression to begin with. Canvassing or sensing for testers may be worth doing as early as possible in the research process, to better select appropriately difficult progressions.

Another direction worth noting is in the inclusion of negative datasets, or pursuing more discussions with experts on safe negative testing. One of the main goals of the application is to correctly determine the error being performed by the user, and so it would be beneficial to be able to test specifically for error distinction. However, this study opted out of negative testing to avoid any risk of injury to the testing participants. With further discussions, there may exist potential compromises to this problem.

Should the system be sufficiently usable, there are significant benefits to be gained in the ability to cover more of the standard form cues, such as that of the scapulae and the pelvis. These form cues, while subtle, are important for the body's positioning, as indicated by expert feedback. Beyond that, the biggest potential for a system such as Progress-Up is to expand to other exercise progressions, covering a variety of motions and holds. In doing so, it is also imperative to ensure further close collaboration with domain experts in the field, particularly personal coaches and trainers, who have the exact experience of analyzing the form of individuals performing exercises.

Acknowledgments The researchers wish to express their deepest gratitude to the following: Mr. Jeriel Manalo from the Department of Sports Science of the UP Diliman College of Human Kinetics; and from the Philippine Sports Commission Medical Scientific Athletes Services (PSC-MSAS): Ysabel De La Rosa, David King Divina, Bettina Gomez, Roxie Narciso, Mark Ng, Ivana Pastrano, and Dea Santos. The insights we received during our consultations were essential to this study, and this would not have been possible without their guidance and feedback.

References

1. Achapero, J.J., Bolado, J.A., Juayong, R.A., Addawe, J., Caro, J.: Progress-up: Design of a mobile application utilizing angle heuristics to provide real-time feedback for push-up progressions. In: Novel and Intelligent Digital Systems: Proceedings of the 5th International Conference (NiDS 2025), vol. 1, pp. 168–179 (2025). DOI 10.1007/978-3-032-06634-3_13
2. Ben Gamra, M., Akhloufi, M.A.: A review of deep learning techniques for 2D and 3D human pose estimation. Image and Vision Computing **114**, 104,282 (2021). DOI 10.1016/j.imavis.2021.104282. URL https://www.sciencedirect.com/science/article/pii/S0262885621001876
3. Ebben, W.P., Wurm, B., VanderZanden, T.L., Spadavecchia, M.L., Durocher, J.J., Bickham, C.T., Petushek, E.J.: Kinetic analysis of several variations of push-ups. Journal of strength and conditioning research **25**(10), 2891–2894 (2011). DOI 10.1519/JSC.0b013e31820c8587. URL https://doi.org/10.1519/JSC.0b013e31820c8587
4. Gouvali, M., Boudolos, K.: Dynamic and electromyographical analysis in variants of push-up exercise. Journal of Strength and Conditioning Research **19**(1), 146–151 (2005). URL https://journals.lww.com/nsca-jscr/abstract/2005/02000/dynamic_and_electromyographical_analysis_in.25.aspx\%5C
5. Kotte, H., Daiber, F., Kravčík, M., Duong-Trung, N.: FitSight: Tracking and feedback engine for personalized fitness training. In: Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization, UMAP '24, p. 223–231. Association for Computing Machinery, New York, NY, USA (2024). DOI 10.1145/3627043.3659547. URL https://doi.org/10.1145/3627043.3659547

6. Kotte, H., Kravčík, M., Duong-Trung, N.: Real-time posture correction in gym exercises: A computer vision-based approach for performance analysis, error classification and feedback. In: 18th European Conference on Technology Enhanced Learning (ECTEL2023), 3rd International Workshop on Multimodal Immersive Learning Systems, MILeS'23. CEUR Workshop Proceedings (2023). URL https://ceur-ws.org/Vol-3499/paper9.pdf
7. Kritz, M.: Development, reliability and effectiveness of the movement competency screen (MCS) (2012)
8. Kwon, Y., Kim, D.: Real-time workout posture correction using OpenCV and MediaPipe. The Journal of Korean Institute of Information Technology **20**, 199–208 (2022). DOI DOI:10.14801/jkiit.2022.20.1.199
9. Low, S.: Overcoming gravity: A systematic approach to gymnastics and bodyweight strength (second edition). Battle Ground Creative (2016)
10. Pardillo, J.A.S., Bolabola, T.J., Sta. Romana, C.L.C., Cando, J.M.D.: Computer-aided assessment system for calisthenics exercises. In: Proceedings of the 6th International Conference on Digital Technology in Education, ICDTE '22, p. 145–152. Association for Computing Machinery, New York, NY, USA (2023). DOI 10.1145/3568739.3568765. URL https://doi.org/10.1145/3568739.3568765
11. Rahmadani, A., Bayu Dewantara, B.S., Sari, D.M.: Human pose estimation for fitness exercise movement correction. In: 2022 International Electronics Symposium (IES), pp. 484–490 (2022). DOI 10.1109/IES55876.2022.9888451
12. Tharatipyakul, A., Srikaewsiew, T., Pongnumkul, S.: Deep learning-based human body pose estimation in providing feedback for physical movement: A review. Heliyon **10**(17), e36,589 (2024). DOI 10.1016/j.heliyon.2024.e36589. URL https://www.sciencedirect.com/science/article/pii/S2405844024126205
13. Zhang, X., Han, S.Z.H., Lim, K.Y.T.: Designing and prototyping of AI-based real-time mobile detectors for calisthenic push-up exercise. Procedia Computer Science **239**, 445–452 (2024). DOI 10.1016/j.procs.2024.06.192. URL https://www.sciencedirect.com/science/article/pii/S1877050924014352. CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2023