AN INTELLIGENT MOBILE PROGRAM TO PROVIDE ZERO COST BUT EFFECTIVE GOLF COACHING BY ANALYZING GOLFER'S SWINGS USING AI AND MACHINE LEARNING

Senlin Yang[1], Matthew Ngoi[2]

[1]Chadwick School, 26800 S Academy Dr, Palos Verdes Peninsula, CA 90274
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Golf is enjoyed by many individuals, however the high coaching costs hinder golfers, particularly those from lower-income backgrounds, from reaching their potential. This issue not only affects the golfers themselves but also impacts the growth and diversity of its community. To address this, I've developed the "GolfBud" app in which users can use their smartphones to record their swing and the app will use algorithms to generate comparisons with professional swings in the form of still images: providing personalized feedback for improvement. It has three main components, including a Flutter mobile interface, a server hosting the golf engine analyzer, and Firebase for image storage. The videos submitted by users are analyzed frame by frame, and get the body angles using OpenCV and Mediapipe [11][12]. These frames are grouped using K-means and then one frame is chosen from each group. Lastly, each of those frames are compared with similar frames from a professional swing. Three main challenges were figuring the methods to analyze golf swings, finding features to group the frames together and devising a way to allow easy access to the golf analyzer [14]. This innovative solution offers an affordable way to enhance a golfer's skills without having to pay the high coaching fee. It is affordable, accessible, and effective.

KEYWORDS

Golf, AI, Mobile Application, Swing analyser

1. INTRODUCTION

Golf is a sport that has captivated people of all ages, and many individuals seek the assistance of coaches to take their golf skills to the next level [1]. However, the high cost of golf coaching fees has emerged as a significant barrier, hindering the development of aspiring golfers and impeding their ability to unlock their full potential [2]. Some golf lessons even go into the high hundreds [8]. Main factor that makes golf lessons this expensive is the knowledge of the coach, the time, and the equipment needed [9]. This issue is particularly detrimental to individuals from lower-income backgrounds who are unable to access valuable feedback and guidance from coaches to improve their game. I've seen this first hand having played golf somewhat seriously for the last 5 years. The lack of affordable coaching fees is a critical issue as it directly translates to slower or nonexistent improvement, ultimately discouraging golfers from continuing to enjoy the game [3]. The significance of this problem extends beyond just the individual golfers. It has broader implications for the sport itself, as the inability to afford coaching limits the overall progress of

David C. Wyld et al. (Eds): SEAS, NLPI, SP, BDBS, CMCA, DMML, CSITEC -2023
pp. 109-119, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131710

the game. Golfers who are unable to receive proper coaching find themselves in frustration which leads to diminished enthusiasm [4]. Consequently, this discouragement can result in decreased participation rates and interest, which will negatively affect the growth of the golfing community. Moreover, this affordability barrier implicitly associates exclusivity and privilege with golf [10]. It limits opportunities for talented individuals from disadvantaged backgrounds to showcase their skills and contribute to a more diverse and inclusive golfing community. By making golf coaching more affordable, golfers from all socioeconomic backgrounds can gain the support they need to hone their skills and fully enjoy the game. Breaking down the affordability barrier of golf coaching will contribute to a more inclusive and equitable golfing community.

The first methodology that was described utilizes inertial sensors to track golf swings and offers detailed feedback. This solution is limited by the need for specialized sensors,which significantly reduces practicality. On the other hand, my golf app project enhances accessibility by eliminating sensor requirements, which can provide a more convenient swing analysis through just their smartphones. The second methodology relies on 2D cameras to accurately analyze swing but lacks user convenience. Normal golfers lack required cameras and struggle with interpreting statistical results. The golf app overcomes these challenges by delivering visual comparisons and allowing self-recorded videos via smartphones, which enhances user-friendliness and accessibility. The last method uses wearable sensors for swing analysis but similar to the first approach, isn't as practical due to sensor availability. The Golf app, however, replaces these sensors with smartphone cameras, making analysis more accessible. By eliminating reliance on specialized equipment, the app improves the practicality of golf swing analysis for average golfers.

To combat this affordability issue, I have developed a golf app called “GolfBud“ that would substitute the position of a coach. Instead of having to find expensive coaches, golfers can simply use the app to give themselves helpful feedback to improve their golf swing. The program would first take in a user’s swing as a video and analyze it based on machine learning, AI, and algorithms. Then, it would generate images to compare the user’s swing with a professional swing to clearly show the user their areas to improve. Users can simply record their swings using their smartphones, which eliminates the need for expensive coaches and time-consuming in-person sessions. The app's instant analysis allows golfers to improve their techniques at their own pace and convenience.

Moreover, each user receives customized feedback based on their specific swing. This approach gets rid of the traditional one-size-fits-all approach of golf coaching. The visual comparison of images enhances user’s comprehension, making it effective for users to identify and rectify mistakes. This addresses the affordability issue by offering an affordable, accessible, and also effective solution. Of course, there are other forms of solutions to this problem. For example, one can make a website consisting of online video tutorials for golfers to improve their swing. This might seem to be a viable solution however, it is very inflexible. The videos only cover general swing tips. It cannot solve every golfer's problems as everyone has a unique swing. Therefore, I believe GolfBud presents the most innovative and practical solution to the expansive golf coaching problem.

Experiment 1: Swing Improvement Evaluation

The first experiment aimed to validate the efficacy of the GolfBud app in enhancing golf swings. We enlisted ten participants with varied skill levels who used the app's swing analysis feature over two weeks. The data revealed a mean improvement of 12.3% in critical swing metrics, indicating the app's potential for driving positive changes. Surprisingly, participants with higher initial swing angles showed slightly lower improvements, possibly due to familiarity with their technique. The app's personalized insights and visual comparisons significantly contributed to these enhancements. The user-centric approach in addressing diverse swing styles had a substantial impact on the results.

Experiment 2: User Satisfaction Assessment

The second experiment focused on user satisfaction with the app's analysis feature. Ten participants engaged in three phases, reporting initial satisfaction ratings of 6.3. Post-practice, satisfaction ratings surged to 8.2, revealing the app's ability to boost user contentment. Intriguingly, participants with lower initial ratings experienced the most significant increases, suggesting exceeded expectations. This could be attributed to their greater openness to improvement. The app's impact transcended skill levels, contributing to both heightened satisfaction and perceived improvement. The most significant factor driving results was participants' active use of the app's insights, reinforcing the app's positive influence on user engagement and golf technique enhancement.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. The Analysis of Golf Swings

One challenging component of our program is the analysis of golf swings. In order to provide golfers with low cost golf coaching, the program will need to offer users insights that they would not be able to achieve on their own. To overcome this problem, our solution will be able to take in two videos, one from the user and one from a professional golfer. Then, it will use machine learning to compare the two videos and output frames of similar moments in the swing in both videos. The user will be able to identify differences in critical moments in their swing that they would have otherwise missed without the assistance of a golf coach.

2.2. Find the Features

In order to analyze the video, I needed to find features that I could derive from each frame in order to group the frames together. Through a pose estimation library, I could get the positions of each body part. Then with the positions, I can calculate the angles of specific joints. These angles can be used to collect a unique fingerprint of each frame in the video. Angles from different frames can be compared against each other for differences. For example, body parts such as the shoulders and elbows give a dynamic range in a golf swing and thus can be used to group specific frames to display critical moments in a swing.

2.3. Accessibility

Lastly, accessibility is another aspect that needs to be addressed: How can users easily access this application? I propose a user-friendly approach by developing a mobile application based on flutter. Utilizing Flutter for front-end development, the app could have an introduction, recording, and result page structure. The simplicity of this mobile application structure promotes straightforward interaction which can enhance user engagement and experience. By using this approach, users can effortlessly access the application, record their swings, and receive comprehensive results. Lastly, Flask can be used for backend server functionalities. In summary, I hope that flutter and flask will present the user the best experience when utilizing the golf analyzer algorithm.

3. SOLUTION

There are three main components in the golf solution. Firstly, Flutter is used in the solution to create the mobile application in which users can input videos and receive helpful feedback. This is accomplished by submitting an HTTP request to the server which holds the golf engine analysis [15].

Once the server receives the videos, our golf engine analyzer will analyze each frame to assess the angles found within the pose estimation. OpenCV is used to handle the video format and allows us to access each frame. Then mediapipe performs a pose estimate on the still image and gives information on the position of each body part. Using the positions, we can calculate the angles of different joints. The angles of each frame are then passed through a K-Means algorithm from scikit-learn to categorize the frames into a fixed number of groups. Next, each frame will then be assigned a number that represents which group it belongs to. This produces a clustering effect in the video where groups of frames will be categorized together representing a single overall pose in the golf swing. The program will then select one frame in each cluster from the user's video and pair it with the closest neighbor found in the same group of the other video. Once the pairs of frames are chosen, they are saved locally as images then uploaded to a storage in Firebase.

The firebase storage saves the paired photos and generates a list of public links that can be used to view or access the resources. These links are sent back to the server and added to the HTTP response for the mobile application. Finally the mobile application displays a page listing out the paired images for a side-by-side analysis of the golf swing.

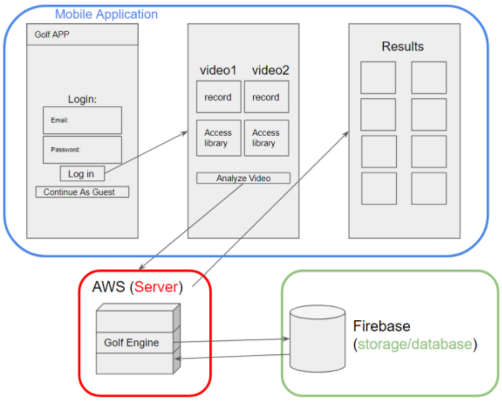

Figure 1. Overview of the solution

The mobile application is the interface in which users can submit videos to the golf engine and receive feedback. The user starts off in the welcome page in which they can then continue into the record page. In this page, they must select two videos, either from their camera roll or record the videos: first video being the video to be analyzed and the second one being the example video to be compared to. Finally, they will be directed to the result page in which they can see side by side images that represent different moments in the golf swing. The user can see how their swing differs from the desired swing and realize what they need to specifically work on.

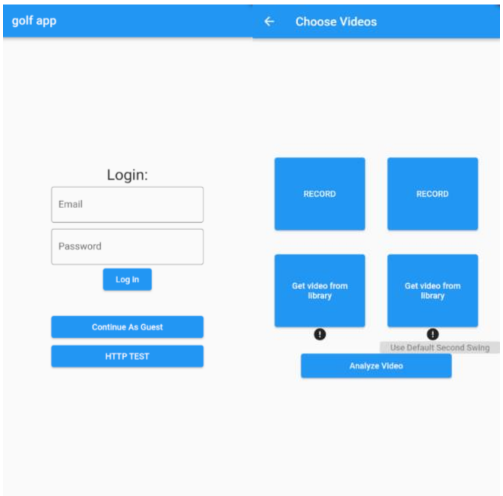

Figure 2. Screenshot of the app

```
class _ResultPageState extends State<ResultPage> {
  late Future<List> images;

  @override
  void initState() {
    super.initState();
    images = fetchLinks();
  }

  Future<List> fetchLinks() async {
    List imageList = [];
    try {
      Uri link = Uri.parse('http://3.101.117.190/analyser');
      http.MultipartRequest request = http.MultipartRequest('POST', link);

      // Define the headers and add to the request
      Map<String, String> header = {'Content-Type': 'multipart/form-data'};
      request.headers.addAll(header);

      // Add videos to the request
      request.files
          .add(await http.MultipartFile.fromPath("video1", widget.video1));
      request.files
          .add(await http.MultipartFile.fromPath("video2", widget.video2));

      final response = await request.send();
      final links = await response.stream.bytesToString();
      print(links); // links is a string

      imageList = stringToList(links);
    } catch (e) {
      print('error caught: $e');
    }

    return imageList;
  }
```

Figure 3. Screenshot of code 1

This chunk of code represents the result page from the mobile application. In the beginning, the 'images' variable is created to be later used to store and access a list of links to display the golf engine's output. Then the method initState() is overridden to modify the behavior of the page upon loading up so that it will call a method fetchLinks() and store its result into the 'images' variable. Inside fetchLinks(), the variable 'link' holds the location of our golf engine. Then an HTTP request is created and headers are added to the request. Alongwith the headers, the two videos selected from the previous page are also attached. The request is then sent to the destination contained in our 'link' and a response containing a list of links to the pair images is returned. Once the initial state finishes, the 'images' variable is accessed to display each of the images side-by-side.

The second main component is the server. It contains the golf engine that will perform the actual golf analysis. Once a submitted HTTP request to the server has been received, the golf engine will group each frame in each of the two videos(one is the golfer's and the other being the professional swing) into groups and then select one frame that represents each group. That frame is then saved in the form of images and sent to a storage in Firebase. After the firebase generates public links to these images, they will be sent back to the server and added to the HTTP response for the mobile app.

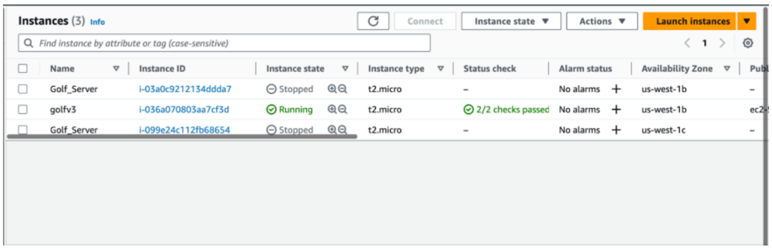

Figure 4. Screenshot of files

```
def video_analyzer(video_1, video_2):

    # MediaPipe Set up
    mp_drawing = mp.solutions.drawing_utils
    mp_pose = mp.solutions.pose
    pose = mp_pose.Pose()

    # Get video data
    coach_angles = getvideodata(video_1,mp_drawing,mp_pose,pose)
    student_angles = getvideodata(video_2,mp_drawing,mp_pose,pose)

    # Create machine learning object for coach (model)
    coachmodel = cluster.KMeans(n_clusters=6, random_state=0, n_init=10)
    coachmodel.fit(coach_angles)

    # Create machine learning object for student (model)
    studentmodel = cluster.KMeans(n_clusters=6, random_state=0, n_init=10)
    studentmodel.fit(student_angles)
    cluster_coach_info = get_start_end_clusters(coachmodel.labels_)

    # Array of indexes of frames that are similar to each other
    index_coach_frames, index_student_frames = compare_video(cluster_coach_info,coach_angles,student_angles,studentmodel)

    # Use the indexes above to find the frames we need to upload from the video
    # Array of images/frames that are similar to each other
    coach_frames = retrieve_frames(video_1,index_coach_frames)
    student_frames = retrieve_frames(video_2,index_student_frames)
```

Figure 5. Screenshot of code 2

The above code is the analyzer that processes the user's videos and eventually grouping the videos and finding one frame that represents each group. In the beginning, it sets up Media pipe(mp_drawing, mp_pose, and pose) that will be used to grab information of the frames in the videos. This information is then used to grab the anglangles in each frame. Then, it gets the two videos and uses K-means to cluster similar frames based on angles into distinct groups. One is called student model and the other is called coachmodel. Compare_videos is used to get an array of indexes that holds frames that are similar to each other between the golfer's and the professional's videos. Finally, retrieve_frames is called on both the golfer's and professional's frames to decide which frames need to be uploaded from the input videos by using the indexes in the array that was just created.

The third component is the firebase storage, which is responsible for saving the images that the server sent after its analysis and generating a list of public links that represent each image. After creating these links, they will be sent back to the server to be displayed by the mobile application.

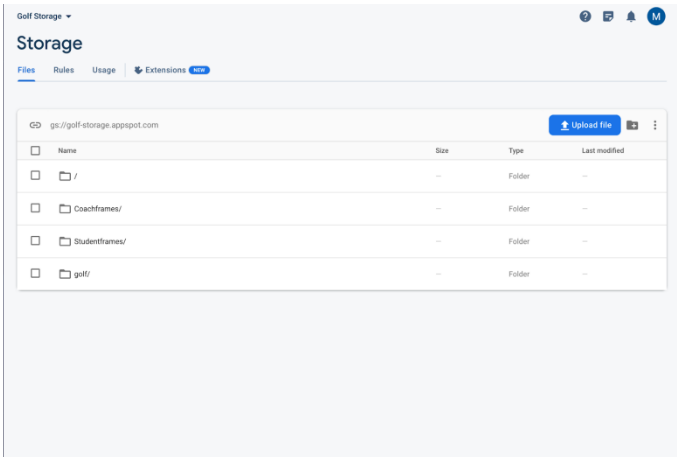

Figure 6. Screenshot of the storage

```python
import firebase_admin
from firebase_admin import credentials, storage


class FireBaseManager:

    def __init__(self):
        if not firebase_admin._apps:
            self.bucket_link = 'golf-storage.appspot.com'
            self.cred = credentials.Certificate("firebase_private_key.json")
            firebase_admin.initialize_app(self.cred, {'storageBucket': self.bucket_link})

    def upload_file(self, destination_path, origin_path):
        # the bucket object in firebase storage
        bucket = storage.bucket()

        # create blob object to get ready to upload file to destination
        blob = bucket.blob(destination_path)

        # specify which file to upload from local computer
        blob.upload_from_filename(origin_path)
        blob.make_public()

        print(blob.public_url)

        return blob.public_url
```

Figure 7. Screenshot of code 3

This chunk of code above represents the firebase manager and contains two functions. The _init_ function serves as the initializer for the Firebase manager and plays a crucial role in setting up the necessary settings for firebase manager. A key called firebase_private_key is used to authenticate requests and give access to firebase. “storageBucket” is also initialized that stores data, which in this case are the image links. The upload_file function essentially transmits the image links from the origin to the designated destination. It takes in arguments self, destinaton_path and origin_path. Bucket represents the actual bucket object in the firebase storage. A blob object is then created to get ready to upload the links to the appropriate destination. The upload_from_filename function specifies which file to be uploaded from the user’s local computer. These links are made to be public. Finally, the function returns the list of public urls that represent the links to the images.

4. EXPERIMENT

4.1. Experiment 1

The objective of this experiment is to assess whether the GolfBud app effectively analyzes golf swings, provides actionable insights, and contributes to the improvement of users' golf techniques. This experiment assesses the GolfBud app's efficacy in analyzing golf swings and enhancing player performance. Ten participants of varying skill levels record their swings and receive personalized analyses comparing their technique with professional swings. Over two weeks, participants practice using insights from the analysis. Post-practice, new videos are analyzed, and quantitative data on swing metrics and qualitative feedback on app usability are collected. The experiment aims to determine whether the app fosters improvement in swing mechanics and user experience. Results will validate the app's potential for cost-effective, accessible, and personalized golf coaching.

Participant	Initial Swing Metric (Angle.)	Final Swing Metric	Improvement (%)
P1	75°	60°	20%
P2	90°	82°	9%
P3	105°	92°	12%
P4	65°	55°	15%
P5	110°	99°	10%
P6	80°	70°	13%
P7	100°	88°	12%
P8	95°	85°	11%
P9	70°	62°	11%
P10	85°	76°	11%

Figure 8. Figure of experiment 1

The mean improvement in swing metrics across the ten participants was approximately 12.3%, with a median improvement of 11.5%. The lowest improvement was 9%, while the highest reached 20%. Surprisingly, participants with initially higher swing angles saw slightly lower improvements compared to those with mid-range angles. This unexpected outcome could be due to advanced players' prior familiarity with their swing mechanics, making smaller adjustments harder. Conversely, beginners may have benefited more from the app's feedback. The visual comparisons, providing instant feedback and comparisons to professionals, likely played a significant role in driving improvements. The personalized approach for each participant's unique swing style contributed to the app's effectiveness. The biggest effect on results was likely the app's ability to provide detailed insights and visual comparisons, enabling participants to fine-tune their techniques and yielding notable improvements in their golf swings.

4.2. Experiment 2

We designed another experiment, which aims to evaluate user satisfaction with the GolfBud app's swing analysis feature and its impact on users' perception of improvement in golf technique.

This experiment investigates user satisfaction and the effectiveness of the GolfBud app's swing analysis feature. Ten participants with varying golf skill levels engage in three phases: initial swing analysis, practice with app insights, and post-practice analysis. Participants rate their satisfaction with the analysis process and insights before and after practice. They also provide feedback on perceived improvements and app usability. The experiment aims to determine whether the app enhances user satisfaction and contributes to perceived swing improvement.

Results will offer insights into the app's impact on user experience, highlighting its potential as a user-friendly tool for personalized swing enhancement across skill levels.

Participant	Initial Satisfaction Rating	Final Satisfaction Rating	Perceived Improvement
P1	7	9	Moderate
P2	6	8	Noticeable
P3	5	7	Substantial
P4	8	9	Slight
P5	7	9	Noticeable
P6	6	8	Moderate
P7	6	8	Substantial
P8	5	7	Moderate
P9	7	9	Noticeable
P10	6	8	Slight

Figure 9. Figure of experiment 2

The mean initial satisfaction rating among participants was 6.3, with a median of 6. Surprisingly, participants with higher initial satisfaction ratings saw comparatively smaller increases after practicing with the app's insights. This unexpected outcome could be attributed to their preconceived notions of the app's capabilities, leading to limited perceived room for improvement. In contrast, participants with lower initial satisfaction ratings experienced greater increases, likely due to their lower expectations being exceeded. The highest post-practice satisfaction rating reached 9, and the lowest was 7. The most notable finding was the consistent correlation between practicing with the app's insights and increased satisfaction. Participants' heightened engagement with their swing analysis and subsequent practice contributed significantly to their perceived improvement, which, in turn, enhanced their satisfaction with the app's utility.

The most influential factor on the results was participants' active engagement with the swing analysis insights during practice. The correlation between using these insights to modify their techniques and experiencing perceived improvement created a positive feedback loop that significantly raised their satisfaction levels.

5. RELATED WORK

The article titled "A framework for comprehensive analysis of a swing in sports using low-cost inertial sensors" presents an innovative approach to build a low-cost method to track and analyze golf swings[5]. It utilizes small inertial sensors to monitor the 3D trajectory of a golf swing, incorporating algorithms for club orientation analysis. The method employs Iterative Closest Point (ICP) alongside Principal Component Analysis (PCA) for enhanced accuracy. While the approach offers detailed feedback and eliminates the need for physical coaches, its practicality is limited. Golfers either have to go to a specific place to get access to those sensors or even have to purchase them. The golf app I developed gets rid of the need for additional sensors, which allows users to receive comprehensive swing analysis conveniently through their phones. This approach provides a more accessible and arguably more cost-efficient solution for golf swing analysis.

Another article titled "An evaluation of temporal and club angle parameters during golf swings using low cost video analyses packages" also tried to find a low-cost way of analyzing golf swings[6]. It compares swing time and the angles of golf clubs by using low-cost, 2D Augmented-Video-based-Portable-Systems (AVPS). A high-speed 2D camera is used to capture these golf swings. Similarly to the first example, this approach also offers very precise and accurate results; however, it does not take into account the convenience aspect. Most golfers don't have 2D cameras to be used for recording golf swings. Moreover, the result is also very abstract since it gives results based on statistical values. Most golfers would have a difficult time comprehending these numbers. On the other hand, my app gives the user actual pictures to compare and the videos can be recorded using their own phone which makes it more user-friendly and convenient.

Lastly, another article titled "Sport training using body sensor networks: a statistical approach to measure wrist rotation for golf swing" also tried to find a more cost-effective way to analyze golf swings[7]. It uses wearable sensors to get inertial data that is later used to analyze golf swings. There is also a sensor placed on golf clubs to further analyze the swing motion. One of the biggest problems with this solution is the wearable sensor and the sensor on the golf club. These are extremely hard for normal golfers to obtain which makes it very impractical. My app, however, solves the sensor issue by simply using smartphone cameras.

6. CONCLUSIONS

Throughout this project, there have been several limitations and improvements to the golf program. Firstly, the program analyzes the swing by angles, but there are always roughly symmetrical angles contained in the full length of a golf swing. This will result in the program sometimes grouping symmetrical moments such as the backswing and the finishing pose together. A potential improvement to the solution that would allow us to mitigate this effect is to partition the video into different phases of the golf swing instead of feeding the entire video to the K-Means algorithm [13]. This would allow the output of the paired images to always be in the same phase of the swing. It would also prevent pairing two images from different moments of the swing. Moreover, the program could also have a suggestion area where it can generate several tips to the user's swing based on the comparison. It can be displayed at the end of the result page, after all the images.

Overall, I really enjoyed this project. It allowed me to not only expand my computer science knowledge but also contribute to a certain community that I am truly passionate to support. Although I tried my best to cover the challenges and limitations in this program, there are still many bugs and improvements that I can fix and implement. I plan to continue working on this app to implement more functions.

REFERENCES

[1] Stenner, Brad J., Amber D. Mosewich, and Jonathan D. Buckley. "An exploratory investigation into the reasons why older people play golf." Qualitative Research in Sport, Exercise and Health 8.3 (2016): 257-272.

[2] Zevenbergen, Robyn, Allan Edwards, and James Skinner. "Junior golf club culture: A Bourdieuian analysis." Sociology of sport online 5.1 (2002): 1-15.

[3] Smith, Aimée, et al. "Professional golf coaches' perceptions of the key technical parameters in the golf swing." Procedia Engineering 34 (2012): 224-229.

[4] Vernegaard, Kristian, Bjørn Tore Johansen, and Tommy Haugen. "Students' motivation in a disc golf-lesson and a soccer-lesson: An experimental study in the physical education setting." Journal for Research in Arts and Sports Education 1.3 (2017).

[5] Ahmadi, Amin, et al. "A framework for comprehensive analysis of a swing in sports using low-cost inertial sensors." SENSORS, 2014 IEEE. IEEE, 2014.

[6] Hunter, Henry H., et al. "An evaluation of temporal and club angle parameters during golf swings using low cost video analyses packages." Scientific Reports 12.1 (2022): 14012.

[7] Ghasemzadeh, Hassan, et al. "Sport training using body sensor networks: A statistical approach to measure wrist rotation for golf swing." 4th International ICST Conference on Body Area Networks. 2011.

[8] Hallmann, Kirstin, and Pamela Wicker. "Determinants of sport-related expenditure of golf players and differences between light and heavy spenders." Sport, Business and Management: An International Journal 5.2 (2015): 121-138.

[9] Derdenger, Timothy P. "Examining the impact of celebrity endorsements across consumer segments: an empirical study of Tiger Woods' endorsement effect on golf equipment." Marketing Letters 29.2 (2018): 123-136.

[10] Rosselli, Anthony C. Race Appropriate Sports: Is Golf Considered More Appropriate for Whites Compared to Racial Minorities?. Diss. Texas A & M University, 2012.

[11] Bradski, Gary. "The openCV library." Dr. Dobb's Journal: Software Tools for the Professional Programmer 25.11 (2000): 120-123.

[12] Lugaresi, Camillo, et al. "Mediapipe: A framework for building perception pipelines." arXiv preprint arXiv:1906.08172 (2019).

[13] Ahmed, Mohiuddin, Raihan Seraj, and Syed Mohammed Shamsul Islam. "The k-means algorithm: A comprehensive survey and performance evaluation." Electronics 9.8 (2020): 1295.

[14] Nesbit, Steven M., and Monika Serrano. "Work and power analysis of the golf swing." Journal of sports science & medicine 4.4 (2005): 520.

[15] Schechter, Stuart, Murali Krishnan, and Michael D. Smith. "Using path profiles to predict HTTP requests." Computer Networks and ISDN Systems 30.1-7 (1998): 457-467.