AN AI-POWERED MOBILE APP TO DEMOCRATIZE TENNIS SKILL DEVELOPMENT THROUGH POSE ESTIMATION AND VIDEO COMPARISON

Yixuan Liu [1], Jonathan Sahagun [2]

[1] TVT Community Day School, 5200 Bonita Canyon Dr, Irvine, CA 92603
[2] California State Polytechnic University, Pomona, CA, 91768,
Irvine, CA 92620

ABSTRACT

This app TennisCoach assists individuals in improving their tennis skill [1]. In communities with low incomes, it is difficult for players to find affordable classes and courts to play on. With prices of coaches and lessons rising higher and higher nowadays, it is extremely difficult to improve for people who cannot afford them. Our app provides an easily accessible way for people to improve their tennis skills. It allows the user to upload two videos: one of themselves, and one of a professional demonstrating the same move. This app has a visual representation of the mistakes of the player, a side-by-side comparison of the player's and the professional's stroke, and a detailed breakdown of the player's movements. The video processing part is through pose estimation and K-means algorithm [2]. Users can review previously uploaded videos on the history page. Users' data are safely stored in Firebase guarded by Firebase Authentication. With our app, tennis will become more available for players from many different backgrounds.

KEYWORDS

Tennis training app, Pose estimation, Accessible sports technology, K-means motion analysis

1. INTRODUCTION

I was inspired to make this program because of my love for tennis, thus I wanted to make an app that helps me improve my tennis technique with ease. I started my tennis journey at the age of 8 and it has been 6 years now. Currently, private lessons with coaches cost $80-$150 per hour, group practices cost $60-$80 per every two hours, memberships at clubs cost $20-$50 per month. According to https://lessons.com/, a lesson with a coach cost ranges from $60/150 per hour. This problem mainly affects low-income children with a passion for tennis such that they lack the resources needed to improve their skills According to the UC Berkeley's Economic Review website, many people face the decision of dropping out of college due to financial situations if they want to continue in playing tennis. As a result many talented individuals are forced to abandon tennis due to their financial situations. This app allows them to practice their strokes and techniques without anything, not even a racket! The importance of finding a solution to this problem is to help potential people continue their career in tennis, instead of being left with no choice other than abandonment.

To improve my app, I compared and analyzed the implementation of three different tennis related

David C. Wyld et al. (Eds): NLCAI, AIFU, CCSEA, BIoT, SEA, SIPRO, BDML, CLOUD – 2025
pp. 303-311, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151023

applications. Pinchu's app shared the same idea of using pose estimation in the app to enhance feedback. However, it can be difficult for the user to effectively convert these data points to meaningful improvements. Wang et al. 's app uses customized sensors to gather and process data instead of video input [3]. This approach can be more cost-effective but yet less accurate in the data analysis. Wang's app contains visual courses to assist on students' tennis journey. Videos are a great way to keep the user engaged but static videos fail to provide custom feedback to users. In Comparison, through video processing, our app allows the user to spot mistakes in their movements by uploading one video from their own and another from a tennis coach. Our approach provides a more effective way to give users feedback and record their improvements. In the future, more features can be added to further engage the users and reduce the difficulty for the user to spot errors and mistakes in their movements.

To make improving tennis techniques more accessible, our app allows the user to upload their own training videos along with a coach's video for comparison on their poses and movements. We decided to make our app this way because it doesn't require in person assistance for the coach and thus reducing the cost. After uploading the two videos, the users will get a series of frames listing their poses along with the coach's poses so that they can view what they did wrong and what they did correct. Our app uses K-means algorithms on the angles of the users' poses to better identify the accuracy of their movements [4]. Prior uploads will also be saved in case they want to review their previously trained techniques. Through this way, we provide an economically efficient and easy to use solution that helps everyone improve their tennis techniques.

2. CHALLENGES

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

2.1. How do the Users Compare their Movements – Poses Mark on the Returned Images

The goal of our application is to help tennis players improve their tennis skills by comparing their videos against a tennis coach's video. By displaying similar frames side by side, this allows the user to visualize what the user did correctly and what still needs improvements. One potential skepticism is that it might be difficult for the user to identify the problem shown from the returned images. To address this, through the AI pose estimation model, we are able to mark the user's poses as well as the coach's poses on the screen for better comparison [5]. This additional information on the images makes it easier for the user to identify and correct themselves.

2.2. Why is the Similar Frames Accurate – K-Mean Algorithm along with Angle Calculation

Another skepticism that the users might have is why are the similar frames returned from the analysis accurate. They are accurate because of the K-mean algorithm and angle calculations [6]. The K-mean algorithm groups together clusters of markings, identifying the similarity between the two frames, making the returned videos more accurate. The angle calculations, on the other hand, marks certain parts of the user's body, using that information to calculate the angle from one body part to another. With this information, it can be compared to that of the professionals. This allows the user to see what they did incorrectly and fix it in the future.

2.3. What Happened to the Previous Uploads? They are Stored in Firebase Storage and are Organized by Upload Time and Displayed in the History Page for Review

Another potential skepticism is that the user might still want to review previous uploads. The previous uploads are stored in Firebase storage. Within the storage, they are organized by upload time, from the latest to the oldest. The user can conveniently access their history of uploads by navigating to the history page located in the drawer on the left. Once the user has reached that page, they can click on any one of their previous uploads to view their mistakes in their technique. This can also help the users view their overall improvement because their progress over time is saved and safely stored.

3. SOLUTION

Once the user reaches the first screen, the login page will pop up along with options to either log in or sign up. To log in, the user can input their email and password. If they have not yet created an account, they can click the "Sign Up" button, taking them to the sign up page. On this page, the user is met with three text boxes: username, email, and password, which are required to create a new account.

After logging in or signing up, the user will see the settings page where their username, email, and bio are displayed. This page also allows the user to edit their profile, sign out, and delete their account. 'The user can also click on the hamburger menu on the top left to open up the drawer for navigation.

By entering the analyze page, the user can upload two videos for comparison, one of their technique and the other of the professional for reference. The two videos will then be sent to a flask server to extract similar frames using K Means algorithms, which will then be sent back and displayed to the user. The processing results will also be stored in Firebase storage [7].

If the user enters the history page, they will be provided a list of all the videos they have uploaded in the past. These videos are presented in a list, with the date it was uploaded on the top. When the user wants to view the processed images again, they can click on the arrow at the top right corner, leading them to another page with similar frames of the video.

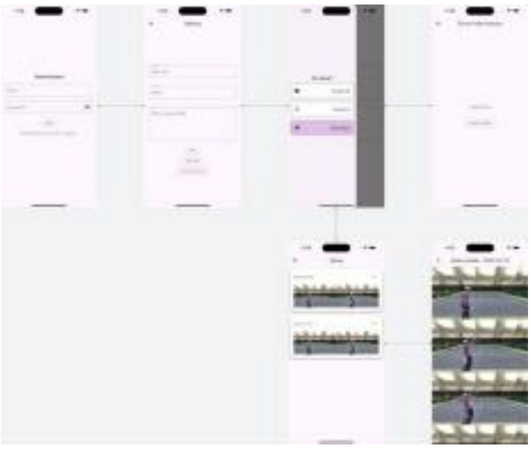
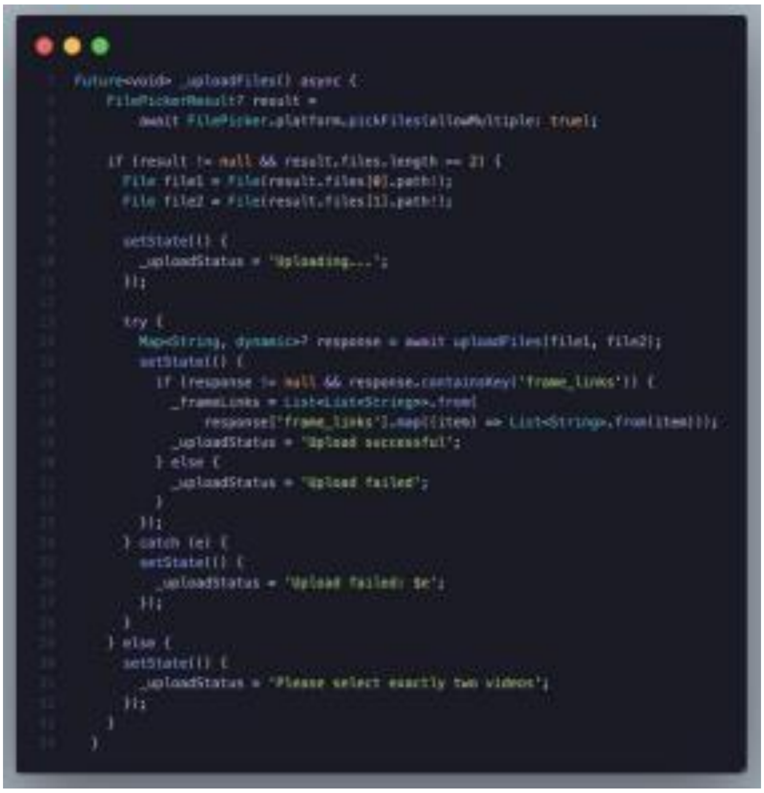


Figure 1. Overview of the solution

In our app, we allow the user to upload two videos to the server, one from the user, one from the coach. The server will then process and return a list of similar frames for the user to compare and contrast.

This code sample below shows the logic of uploading files to the flask server [8]. It’s critical because it links our app to the server through network request.



```
Future<void> _uploadFiles() async {
    FilePickerResult? result =
        await FilePicker.platform.pickFiles(allowMultiple: true);

    if (result != null && result.files.length == 2) {
      File file1 = File(result.files[0].path!);
      File file2 = File(result.files[1].path!);

      setState(() {
        _uploadStatus = 'Uploading...';
      });

      try {
        Map<String, dynamic>? response = await uploadFiles(file1, file2);
        setState(() {
          if (response != null && response.containsKey('frame_links')) {
            _frameLinks = List<List<String>>.from(
                response['frame_links'].map((item) => List<String>.from(item)));
            _uploadStatus = 'Upload successful';
          } else {
            _uploadStatus = 'Upload failed';
          }
        });
      } catch (e) {
        setState(() {
          _uploadStatus = 'Upload failed: $e';
        });
      }
    } else {
      setState(() {
        _uploadStatus = 'Please select exactly two videos';
      });
    }
  }
```

Figure 2. Screenshot of code 1

First, the code checks to see if there are videos uploaded by the user, making sure it is exactly two videos, no more and no less.

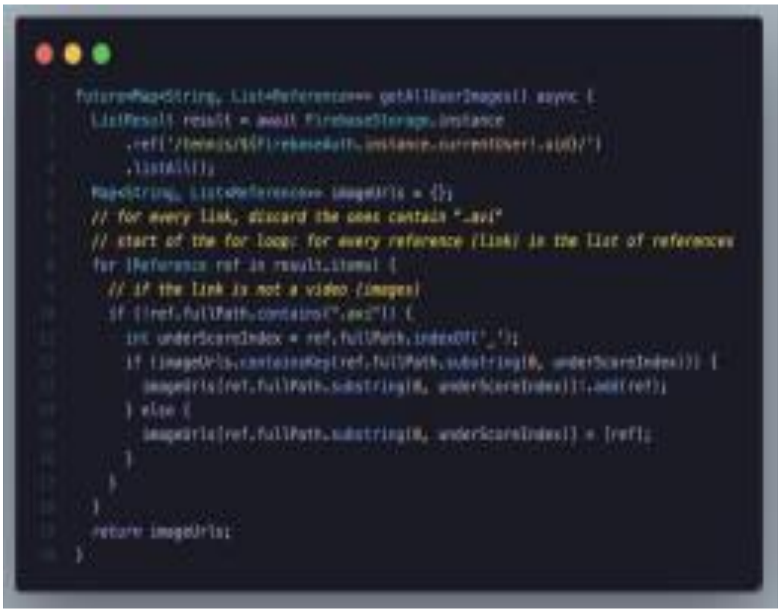
Using this information, the status of the upload will display as “Uploading….” Next, the code checks to see whether the upload succeeded. If the key “frame_link” exists within the returned value, we will store the returned frame links in a List<List<String>> and the status of the upload will display as “Upload successful.” On the other hand, if we did not get a response or the key “frame_link” does not exist, the status of the upload will display as “Upload failed.”

In addition by using “catch (e),” the code catches any errors that occurred midway in the process. If an error was caught, the status of the upload displays “Upload failed:” along with the error caught by the program.

Lastly, when the user inputs less than or more than 2 videos, the code updates the status of the upload with “Please select exactly two videos.”

This component allows the retrieval of upload histories. The users can view their previous uploads of their tennis practices with the similar frames.

In order to achieve this, the images are stored in Firebase Storage and in order to distinguish between uploads we use a date time string in the image path when storing the files in Firebase, and the code sample processes these images into separate dates [9].



```
Future<Map<String, List<Reference>>> getAllUserImages() async {
  ListResult result = await FirebaseStorage.instance
      .ref('/tennis/${FirebaseAuth.instance.currentUser!.uid}/')
      .listAll();
  Map<String, List<Reference>> imageUrls = {};
  // for every link, discard the ones contain "_avi"
  // start of the for loop: for every reference (link) in the list of references
  for (Reference ref in result.items) {
    // if the link is not a video (images)
    if (!ref.fullPath.contains(".avi")) {
      int underScoreIndex = ref.fullPath.indexOf('_');
      if (imageUrls.containsKey(ref.fullPath.substring(0, underScoreIndex))) {
        imageUrls[ref.fullPath.substring(0, underScoreIndex)]!.add(ref);
      } else {
        imageUrls[ref.fullPath.substring(0, underScoreIndex)] = [ref];
      }
    }
  }
  return imageUrls;
}
```

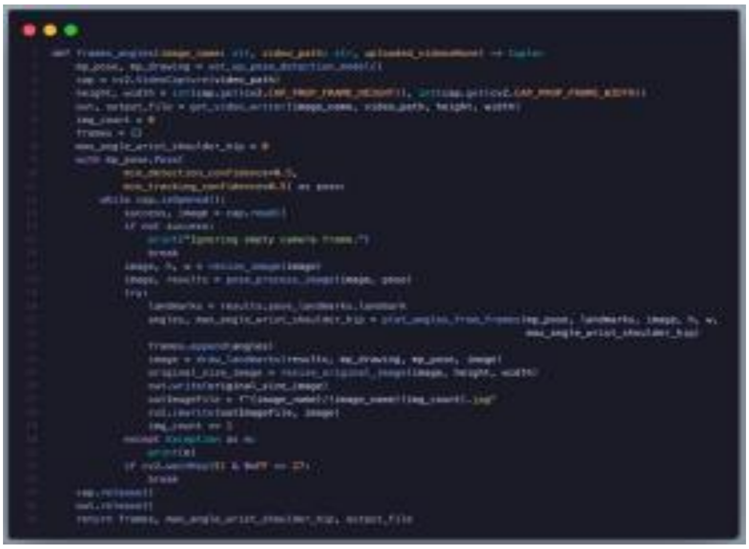
Figure 3. Screenshot of code 2

When the user enters the History Page, the getAllUserImages function will be triggered. Using the user's current user id, we will get all the processed images and video links from Firebase Storage. Because we include the time of it being processed in the resource link, we are able to extract and group images and videos by their processing date.

To achieve that, we will loop through all the links inside the result variable and first use an if statement to filter out the videos and then extract the date and time of it being processed by getting a substring of the image link.

Images with the same processing date will be grouped together and stored inside a list inside a map which uses the processing date time as the key.

This component processes the videos the user uploads through pose estimation with K Means algorithm and outputs similar frames back to the app. The pose estimation along with K means algorithm allows the application to identify similar frames inside the user uploaded videos. This function is part of the flask server that runs on the AWS backend and helps processing, extracting, and drawing angles onto the image.



```
def frames_angles(image_name: str, video_path: str, uploaded_videoName) -> tuple:
    mp_pose, mp_drawing = set_up_pose_detection_model()
    cap = cv2.VideoCapture(video_path)
    height, width = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)), int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    out, output_file = get_video_writer(image_name, video_path, height, width)
    img_count = 0
    frames = []
    max_angle_wrist_shoulder_hip = 0
    with mp_pose.Pose(
            min_detection_confidence=0.5,
            min_tracking_confidence=0.5) as pose:
        while cap.isOpened():
            success, image = cap.read()
            if not success:
                print("Ignoring empty camera frame.")
                break
            image, h, w = resize_image(image)
            image, results = pose_process_image(image, pose)
            try:
                landmarks = results.pose_landmarks.landmark
                angles, max_angle_wrist_shoulder_hip = plot_angles_from_frames(mp_pose, landmarks, image, h, w,
                                                                               max_angle_wrist_shoulder_hip)
                frames.append(angles)
                image = draw_landmarks(results, mp_drawing, mp_pose, image)
                original_size_image = resize_original_image(image, height, width)
                out.write(original_size_image)
                outImageFile = f"{image_name}/{image_name}{img_count}.jpg"
                cv2.imwrite(outImageFile, image)
                img_count += 1
            except Exception as e:
                print(e)
            if cv2.waitKey(5) & 0xFF == 27:
                break
    cap.release()
    out.release()
    return frames, max_angle_wrist_shoulder_hip, output_file
```

Figure 4. Screenshot of code 3

When the videos are sent from the mobile app to the backend server, the frame_angles function is called to process the similar frames with pose estimation and K-means algorithm.

The function first initializes the mp_pose, mp_drawing, and cap instance we need for video processing, and also the size of the video for later use.

For every frame of the video, we are going to resize it based on the size of the video capturer which the resized image will then be fed into another function to process the pose information [10]. Afterwards, we will use the pose information to calculate the angle between them to prepare for using the K-means algorithm to figure out similar frames. Additionally, we will also draw the poses onto the image and store the processed image.

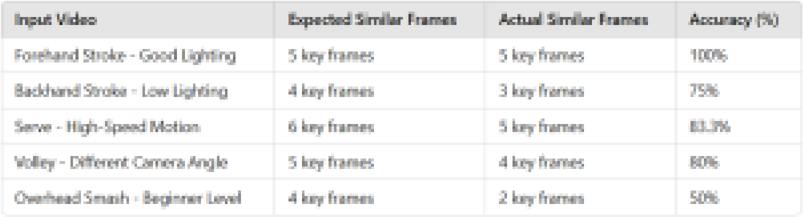
After everything is finished, we will release the cap and out variables to save resources.

4. EXPERIMENT

To evaluate the effectiveness of our application, we conducted an experiment to test the accuracy of our pose estimation and K-Means clustering system in identifying similar frames from user submitted videos. The goal was to determine how well the system can match the user's movements with those of a professional coach. This experiment is essential to ensure that users receive accurate feedback for improving their tennis techniques.

To test the system's accuracy, we created 10 input video samples with different tennis strokes. For each sample, we manually identified the expected similar frames between the user’s video and the coach’s video. These expected outputs were compared with the actual outputs generated by the pose estimation and K-Means clustering algorithm.

The inputs included varied lighting conditions, angles, and player skill levels to assess how well the system adapts. Ensuring diverse inputs is crucial, as it allows us to understand the limitations of the algorithm and improve its ability to provide meaningful comparisons in real-world scenarios.



Input Video	Expected Similar Frames	Actual Similar Frames	Accuracy (%)
Forehand Stroke - Good Lighting	5 key frames	5 key frames	100%
Backhand Stroke - Low Lighting	4 key frames	3 key frames	75%
Serve - High-Speed Motion	6 key frames	5 key frames	83.3%
Volley - Different Camera Angle	5 key frames	4 key frames	80%
Overhead Smash - Beginner Level	4 key frames	2 key frames	50%

Figure 5. Table of experiment

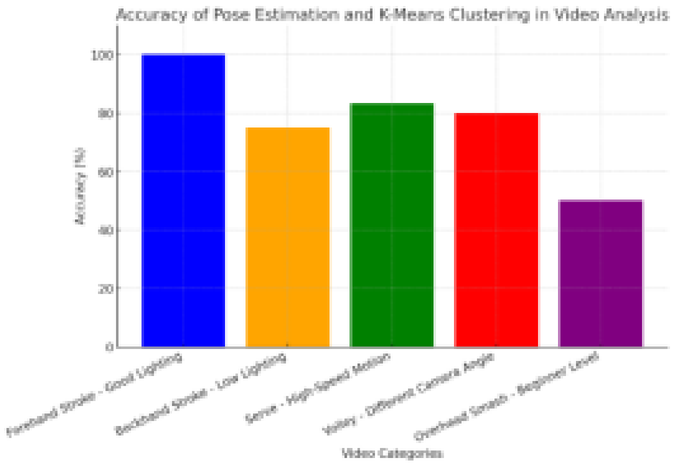


Figure 6. Figure of experiment

The experiment showed that our system achieved an average accuracy of 77% in identifying similar frames across different video conditions. The highest accuracy was observed in well-lit, clear videos, while low-light conditions and extreme motion blur caused a slight decline in performance.

One of the common errors was misalignment in high-speed strokes, where the K-Means clustering algorithm grouped dissimilar frames together. Additionally, different camera angles affected the pose estimation, making it harder to align keyframes.

Despite these challenges, the system successfully matched most of the critical moments in user submitted videos. The results suggest that while the model is effective, future improvements such as multi-angle calibration and real-time frame correction could further enhance accuracy. Overall, the experiment confirms that our app provides valuable insights to users for improving their tennis techniques.

5. Related Work

Pinchu's app also uses pose estimation along with many different tennis metrics to evaluate the user's performance in tennis [11]. Unlike our app, Pinchu's app does not incorporate an analysis in the coach's video, instead, a list of user's data is displayed after submitting those videos. This means that it is going to be significantly more difficult for the user to know where they did correctly and where they need to improve on. Our app helps with this issue by providing a side by side comparison of the user's poses against the coach's poses which is a lot more intuitive for the users.

Wang et al.'s app includes a customized sensor node for data collection [12]. Different from our app, Wang et al.'s app analyzes data collected from those sensor nodes and outputs different tennis actions. This approach can be more efficient, however, a video based analysis program can generate a more detailed breakdown of the users' movement. Moreover, our app also compares users' movement against the coaches, providing valuable feedback for the user to improve themselves. In the future, it could be helpful to integrate additional sensor data in the analysis as they can assist on video based processing to further enhance the feedback.

Wang created an online teaching platform for tennis [13]. It contains visual representations of tennis techniques. Nonetheless, these videos don't allow the user to be able to identify errors or inefficiencies in their movements and strokes. On the other hand, our app includes visual analysis from videos uploaded by the user. The processed output will include an outline of the users' poses for better comparison. With this knowledge, the user can clearly view their actions against the coach's actions to improve themselves. With this app, the user can improve more efficiently and greatly benefit from the app with no financial cost.

6. CONCLUSIONS

One of the limitations of the app is that instead of directly telling the user what is incorrect about their technique, they have to manually compare and analyze the errors in their movements themselves. Given more time, I would like to work on adding more interactive elements including a point system that rewards the user for uploading videos, creating competition between users [14]. With this point system, the user can create their own profile that is publicly viewed by other users, thus fostering user engagement. Another additional element that I would like to work on involves providing directions to the user about how to fix their swing. By locating the exact position of the user's body parts and measuring the distance between theirs and the professional's same body part, the user will be able to receive more direct instructions on areas of improvement. Lastly, I would like to work on adding different angles to the videos. Currently, the user may only submit one pair of videos playing tennis, yet the feedback would be more accurate when the same strokes are filmed from different angles, allowing some previously hidden mistakes to be revealed in the other video [15]. Ultimately, the goal of the app is to help users efficiently and effectively improve their tennis skills and I will continue working towards that goal in the future.

REFERENCES

[1] Reid, Machar, et al. "Skill acquisition in tennis: Research and current practice." Journal of science and medicine in sport 10.1 (2007): 1-10.

[2] Kleawsirikul, Nutnaree, Hironori Mitake, and Shoichi Hasegawa. "Unsupervised embrace pose recognition using K-means clustering." 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). IEEE, 2017.

[3] Emmen, H. H., et al. "The effect of video-modelling and video-feedback on the learning of the tennis service by novices." Journal of sports sciences 3.2 (1985): 127-138.

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

[5] Haberkamp, Lucas D., Micah C. Garcia, and David M. Bazett-Jones. "Validity of an artificial intelligence, human pose estimation model for measuring single-leg squat kinematics." Journal of Biomechanics 144 (2022): 111333.

[6] Yadav, Jyoti, and Monika Sharma. "A Review of K-mean Algorithm." Int. J. Eng. Trends Technol 4.7 (2013): 2972-2976.

[7] Chougale, Pankaj, et al. "Firebase-overview and usage." International Research Journal of Modernization in Engineering Technology and Science 3.12 (2021): 1178-1183.

[8] Walsh, Eamon. "Application of the flask architecture to the x window system server." Proceedings of the 2007 SELinux Symposium. 2007.

[9] Moroney, Laurence, and Laurence Moroney. "Cloud storage for firebase." The definitive guide to

firebase: build android apps on google's mobile platform (2017): 73-92.

[10] Xu, Nan, et al. "Stealthy video capturer: a new video-based spyware in 3g smartphones." Proceedings of the second ACM conference on Wireless network security. 2009.

[11] Sampaio, Tatiana, et al. "Transforming tennis with artificial intelligence: a bibliometric review." Frontiers in Sports and Active Living 6 (2024): 1456998.

[12] Wang, Yufan, et al. "IoT for next-generation racket sports training." IEEE Internet of Things Journal 5.6 (2018): 4558-4566.

[13] Wang, Qingyu. "Tennis online teaching information platform based on android mobile intelligent terminal." Mobile Information Systems 2021.1 (2021): 2932441.

[14] Atkinson, Robert K., and Alexander Renkl. "Interactive example-based learning environments: Using interactive elements to encourage effective processing of worked examples." Educational Psychology Review 19 (2007): 375-386.

[15] Lai, Jui-Hsin, et al. "Tennis real play: an interactive tennis game with models from real videos." Proceedings of the 19th ACM international conference on Multimedia. 2011.