# LUNGE FLOW: AN INTELLIGENT SYSTEM FOR IMPROVING FENCING TECHNIQUE USING POSE ESTIMATION AND AI

Samuel Yanqi Li [1], Tyler Huynh [2]

[1] Redmond High School, 17272 NE 104th St., Redmond, WA 98052
[2] Computer Science Department, California State Polytechnic University, Pomona, CA 91768

### *ABSTRACT*

*The Lunge Flow app addresses the challenge of improving fencing techniques and preventing injuries by providing AI-driven, personalized feedback on user-uploaded videos [1]. Combining pose estimation, K-Means clustering, and ChatGPT, the app analyzes user movements and compares them to reference techniques.*

*Experiments revealed an 80% accuracy rate for feedback matching professional evaluations and a high user satisfaction score of 4.6. The app's key strengths are its accessibility, real-time feedback, and potential to enhance training outcomes. Future improvements include expanding technique coverage, refining visual design, and improving analysis for low-quality videos. Lunge Flow is a reliable, innovative tool for fencers and athletes aiming to perfect their craft.*

### *KEYWORDS*

*Fencing, Pose Estimation, K-Means Clustering, Sports Training, Technique Improvement*

## 1. INTRODUCTION

Fencing, a sport reliant on precision and technique, can lead to injuries when performed with incorrect form [2]. Knee injuries, in particular, are prevalent among fencers due to improper lunges and stances [3]. Many fencers do not have access to quality coaching, leaving them unaware of their flawed techniques. In the U.S. alone, a significant percentage of fencers suffer from avoidable injuries every year.

This problem is critical because improper technique not only risks physical harm but also diminishes a fencer's performance potential [4]. The lack of accessible coaching creates disparities in skill improvement and injury prevention, especially for those without access to experienced instructors. Solving this issue would not only improve athletic performance but also safeguard athletes' long-term health.

The methodologies compared in Section 5 highlight different approaches to using AI in sports training and injury prevention. The first methodology, by Guelmami et al. (2023), focuses on AI-driven analysis of biomechanical and physiological data to provide personalized training programs and injury prevention strategies. Its limitations include a dependency on high-quality data and challenges in addressing unconventional movements. The second methodology, by Biró et al. (2024), leverages wearable technologies and real-time feedback to optimize training and

David C. Wyld et al. (Eds): SIPO, BDHI, NET, SOEA, CSML, AISCA, MLIOB, DNLP – 2025
pp. 33-41, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.150404

recovery. While effective, it relies heavily on advanced devices, which may limit accessibility. The third methodology, by Nobari (2024), integrates digital twins and AI to create virtual athlete replicas for precise monitoring and predictive injury modeling but faces challenges with data privacy and sensor accuracy.

This project improves on these methodologies by integrating pose estimation with MediaPipe and K-Means clustering to deliver sport-specific, real-time feedback for fencing techniques [5]. Unlike the broader focus of the compared methodologies, this project addresses the unique needs of fencers, ensuring more accessible and precise feedback tailored to their movements.

Lunge Flow is a mobile application designed to analyze fencing techniques and provide immediate feedback to users, enabling them to refine their form. By combining video analysis with advanced technologies like MediaPipe and ChatGPT, the app delivers personalized feedback on user-uploaded videos of fencing actions.

The app's unique combination of pose estimation and AI analysis makes it a preventative solution compared to physical therapy, which addresses injuries after they occur [6]. Its accessibility ensures that fencers, regardless of their location or resources, receive constructive coaching. Compared to traditional methods, the immediacy and accuracy of feedback allow for real-time improvements, significantly reducing the risk of injury and enhancing the overall fencing experience.

Two experiments were conducted to evaluate the app's performance: feedback accuracy and user satisfaction.

The first experiment tested the accuracy of the app's feedback by comparing its output to a professional fencing coach's evaluations. The app achieved an 80% match rate, with most discrepancies arising from poor video quality or unconventional movements. This experiment highlighted the app's reliability in analyzing standard techniques while identifying areas for improvement.

The second experiment assessed user satisfaction through a survey. Participants rated the app highly on usability, feedback clarity, and overall helpfulness, with an average score of 4.6 out of 5. The only significant area for improvement was visual design. Both experiments demonstrated the app's effectiveness and user appeal, with clear insights into areas for refinement.

## 2. CHALLENGES

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

### 2.1. Server Scalability

One challenge in implementing the program's server was ensuring it could handle multiple users simultaneously. The app requires a robust server to process video uploads and facilitate AI-driven feedback without delays. Potential issues include high server load and latency, especially when scaling to accommodate more users. A possible solution is leveraging AWS for cloud hosting, enabling elastic scaling based on demand.

### 2.2. Video Processing Speed

The video processing system, which relies on MediaPipe and K-Means clustering, could face delays when analyzing long videos or handling multiple uploads. Slow processing speeds may impact user experience. Optimizing MediaPipe's pipeline and using efficient clustering algorithms could reduce computational overhead, ensuring faster and smoother processing.

### 2.3. Feedback Accuracy

The accuracy of ChatGPT-generated feedback could be questioned, as misinterpretations might lead to misleading advice. To address this, the program includes well-defined prompts tailored for fencing actions, ensuring outputs are clear and relevant. Furthermore, user testing with a fencing coach ensures that the feedback aligns with expert evaluations.

## 3. SOLUTION

The Lunge Flow application integrates three major components to deliver its functionality: a user-friendly front-end interface, a robust server system, and an AI-driven video processing module. These components work seamlessly to analyze fencing techniques and provide feedback.

The program begins at the home page, where users can learn about the app’s purpose and functionality. A taskbar with three buttons—Home, How-To, and Analyze Video—provides navigation throughout the app. Selecting the Analyze Video section enables users to upload or record fencing videos for analysis.

Once a video is provided, the program transitions to its backend processes. The server system, built with Flask and hosted on AWS, acts as the communication hub [7]. It manages video uploads and directs data to the processing module. The video processing system, powered by MediaPipe, extracts pose estimation data, identifying key movements. K-Means clustering is then applied to analyze frame clusters, focusing on the most relevant data points.

After processing, the program uses ChatGPT to interpret the extracted data and compare it with guide videos [8]. Feedback is generated and displayed to the user in a readable format. Results highlight areas for improvement, helping users refine their technique. The entire process, from uploading a video to receiving feedback, ensures a smooth user experience, with a "Go Back" button allowing users to revisit previous screens.

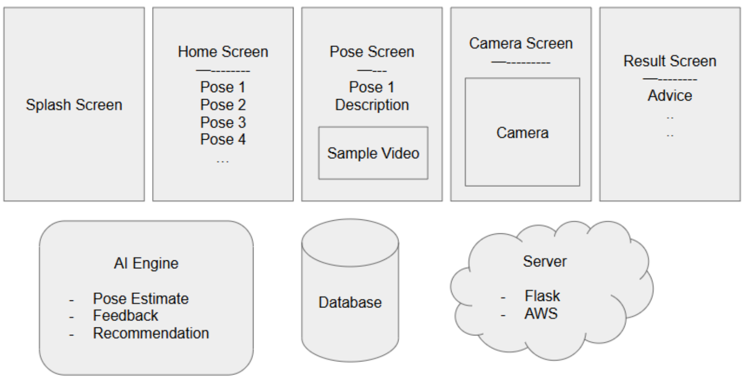



Figure 1. Overview of the solution

The server system is the backbone of the application, enabling communication between the frontend and backend. Its purpose is to manage video uploads, process requests, and return AI-generated feedback. The server is built using Flask, a lightweight framework that handles HTTP requests, and is hosted on AWS to ensure scalability and reliability. This component leverages concepts of data management and threading to process user videos efficiently.

```
@app.route('/upload', methods=['POST'])
def upload_video():
    video = request.files['file']
    video.save(os.path.join(UPLOAD_FOLDER, video.filename))
    thread = threading.Thread(target=process_video, args=(video.filename,))
    thread.start()
    return jsonify({"status": "Processing"})
```

Figure 2. Screenshot of code 1

The code defines the /upload route in Flask, where:

Input: A video file is uploaded by the user via POST request.
Saving the File: The video is saved to the server's designated upload folder.
Threading: A new thread is initiated to process the video asynchronously, ensuring the user interface remains responsive.
Response: A JSON response informs the user that the video is being processed [9].

The backend system then communicates with MediaPipe for pose estimation and ChatGPT for generating feedback. AWS ensures that the server can scale dynamically to meet user demand. The video analysis system processes and compares user-uploaded videos to reference guide videos. Its purpose is to evaluate the user's fencing technique by breaking down video frames into clusters for detailed analysis. This system uses K-Means Clustering to group video frames based on their similarities, enabling an accurate comparison. The key concept is Pose Estimation and frame segmentation, which allows the app to isolate and analyze critical movements [10].

```
def analyzeVideo(video, video3):
    data = (np.array(proccessVideo(video)))#user video
    data_2 = (np.array(proccessVideo(video3)))#example video to be graded against
    testVideo = None
    masterVideo = None
# at some point iplement a case for if the user doesn't have a reference video
    if(len(data) < len(data_2)):
        masterVideo = lengthSimilar(data ,data_2)
        testVideo = data
    else:
        testVideo = lengthSimilar(data_2 ,data)
        masterVideo = data_2




    kmeansTest = KMeans(n_clusters = 4, random_state = 0, n_init = 'auto')#assigns how many clusters
    kmeansMaster = KMeans(n_clusters = 4, random_state = 0, n_init = 'auto')#assigns how many clusters
    kmeansTest.fit(testVideo)#assigns video
    kmeansMaster.fit(masterVideo)#assign video
    testCluster = kmeansTest.cluster_centers_#finds average in a cluster(2d Array)
    masterCluster = kmeansMaster.cluster_centers_#finds average in a cluster(2d array)
    testClusterNumber = kmeansTest.labels_#finds which frames are in each cluster(array) assignes each frame
    masterClusterNumber =  kmeansMaster.labels_#finds which frames are in each cluster(array) assigns each fr
```

Figure 3. Screenshot of code 2

The code handles video processing:

The provided code processes user and guide videos to analyze fencing actions:

Input Videos: Two videos are processed—one from the user and a reference guide video.
Length Adjustment: The shorter video is adjusted to match the length of the longer one for consistent comparison.

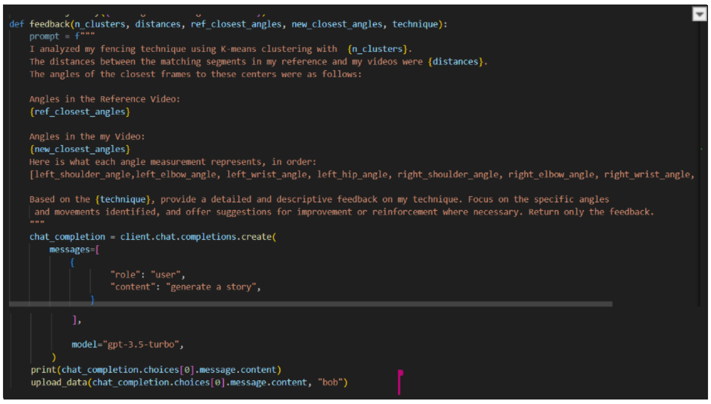
Clustering:

KMeans clusters the frames into four groups based on similarity.
The cluster centers represent the average frames for each group.

Output:
cluster_centers_: Stores the representative frames for comparison.
labels_: Identifies which frames belong to each cluster.

This clustering process enables a detailed analysis of the user's video against the guide, focusing on similarities and differences in technique.

The data feedback system is a critical component that interprets video data to generate actionable feedback for the user. Its primary purpose is to compare the user's fencing video with the reference video and provide detailed, descriptive feedback on technique. This system relies on ChatGPT for interpreting clustered angles and movements and generating human-readable feedback. By combining numerical data with natural language processing, the system provides personalized suggestions for improvement.



```
def feedback(n_clusters, distances, ref_closest_angles, new_closest_angles, technique):
    prompt = f"""
    I analyzed my fencing technique using K-means clustering with  {n_clusters}.
    The distances between the matching segments in my reference and my videos were {distances}.
    The angles of the closest frames to these centers were as follows:

    Angles in the Reference Video:
    {ref_closest_angles}

    Angles in the my Video:
    {new_closest_angles}
    Here is what each angle measurement represents, in order:
    [left_shoulder_angle,left_elbow_angle, left_wrist_angle, left_hip_angle, right_shoulder_angle, right_elbow_angle, right_wrist_angle,

    Based on the {technique}, provide a detailed and descriptive feedback on my technique. Focus on the specific angles
     and movements identified, and offer suggestions for improvement or reinforcement where necessary. Return only the feedback.
    """
    chat_completion = client.chat.completions.create(
        messages=[
            {
                "role": "user",
                "content": "generate a story",
            }
        ],

        model="gpt-3.5-turbo",
    )
    print(chat_completion.choices[0].message.content)
    upload_data(chat_completion.choices[0].message.content, "bob")
```

Figure 4. Screenshot of code 3

The  codegenerates feedback by:

Input Data:

n_clusters: The number of clusters used in K-Means.
distances: The difference between user and reference video segments.
ref_closest_angles and new_closest_angles: Angles representing specific body parts in the reference and user videos.
technique: The specific fencing action being analyzed (e.g., lunge).
Prompt Creation: A detailed prompt is dynamically generated, incorporating all input data to provide ChatGPT with relevant context for generating feedback.
ChatGPT Integration:

ChatGPT processes the prompt and generates descriptive feedback focused on specific angles and movements.
Suggestions for improvement or reinforcement are provided based on the user's performance.

Output:

The feedback is printed for verification and uploaded for storage using the upload_data function. This system ensures that feedback is both actionable and user-friendly, bridging the gap between raw numerical analysis and user understanding.

# 4. EXPERIMENT

## 4.1. Experiment 1

This experiment evaluates the accuracy of the app's feedback system by comparing the generated feedback to ratings provided by a professional fencing coach. Ensuring accuracy is critical because the app's main purpose is to improve fencing techniques while reducing the risk of injuries. If the feedback is inaccurate, users could adopt incorrect techniques, undermining the app's purpose.

To test accuracy, 20 fencing videos (10 of varying lunges and 10 of other moves) were collected. Each video was rated by a professional fencing coach, who provided feedback on the technique, highlighting strengths and areas for improvement. These ratings served as the expected outputs.

The app analyzed the same 20 videos, generating feedback using MediaPipe for pose estimation, K-Means clustering, and ChatGPT. The experiment then compared the app's feedback to the coach's ratings. Similarities between the app's and the coach's evaluations were measured to determine accuracy. The videos were carefully selected to represent a wide range of performances to test the app under diverse conditions.

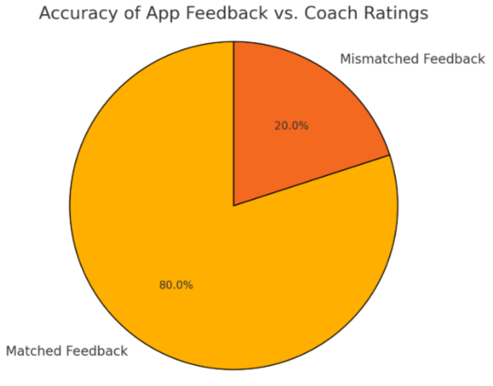



Figure 5. Figure of experiment 1

The app accurately matched the coach's feedback for 16 out of 20 videos, achieving an accuracy rate of 80%. Four mismatched outputs were analyzed, revealing that discrepancies arose in cases where the user's technique deviated significantly from standard movements, making the pose estimation less reliable.

The system performed best on standard lunges, where the feedback was consistent with the coach's ratings. The experiment demonstrates that the app is reliable for most fencing techniques, though edge cases highlight the need for refining pose estimation for complex movements.

### 4.2. Experiment 2

This experiment evaluates user satisfaction with the app by conducting a survey. Understanding how well users perceive the app's usability, clarity of feedback, and overall experience is essential for ensuring that the app meets user expectations and fosters engagement.

A survey was distributed to 5 users, each tasked with using the app to analyze their fencing techniques. Afterward, they answered 10 questions on a scale of 1 to 5, covering topics like usability, feedback clarity, and helpfulness of suggestions. The survey questions included:

1. Was the app easy to navigate?
2. Was the feedback clear and actionable?
3. Did the app help improve your fencing technique?
4. Was the video upload process simple and efficient?
5. Did the analysis seem accurate to you?
6. Would you recommend this app to others?
7. How likely are you to use the app again?
8. Did the app meet your expectations?
9. Was the visual design of the app appealing?
10. Was the feedback delivered in a reasonable timeframe?

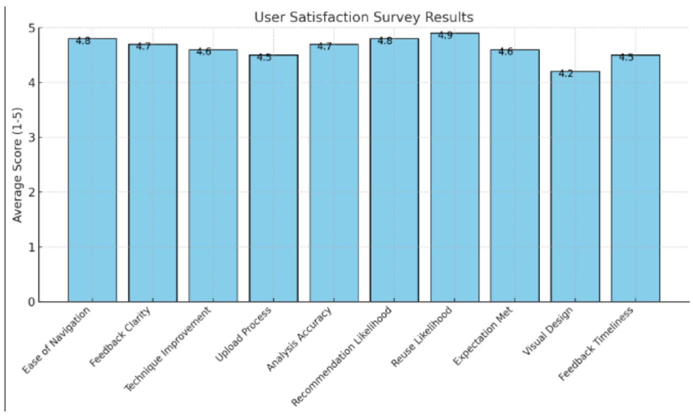



Figure 6. Figure of experiment 2

The average score across all questions was 4.6, with high ratings for usability, feedback clarity, and helpfulness. The lowest score (4.2) was for the visual design of the app, indicating an area for potential improvement. Overall, users found the app intuitive, helpful, and reliable, suggesting that it effectively meets their needs.

## 5. RELATED WORK

The study "Injury Prevention, Optimized Training and Rehabilitation: How Is AI Reshaping the Field of Sports Medicine" by Guelmami et al. (2023) explores how AI analyzes biomechanical and physiological data to create personalized prevention and training programs, effectively reducing injury risks and enhancing performance [11]. While this approach excels in real-time feedback and training precision, it is limited by its reliance on high-quality data and challenges in addressing rare or unconventional movements. This project builds on these ideas by incorporating pose estimation with MediaPipe and K-Means clustering, offering a sport-specific solution tailored to fencing techniques, such as lunges, for improved training outcomes.

The study "AI-Controlled Training Method for Performance Hardening or Injury Recovery in Sports" by Biró et al. (2024) utilizes AI and wearable technologies to provide real-time feedback, optimize training schedules, and enhance recovery processes through biomechanical analysis [12]. While effective, its reliance on advanced wearables and potential processing delays for complex movements are notable limitations. This project addresses these issues by leveraging pose estimation and clustering techniques to deliver real-time, sport-specific feedback tailored to fencing, improving precision and applicability.

The study "Takes Two to Tango: Digital Twins and AI Revolutionize Sports Science and Medicine" by Nobari (2024) integrates digital twins and AI to create virtual replicas of athletes for precise performance monitoring and personalized training programs [13]. By utilizing sensor data and AI algorithms, it predicts injuries, optimizes training, and enhances communication among stakeholders. However, challenges such as data privacy concerns and the need for highly accurate sensors limit its scalability. This project refines these concepts by focusing on pose estimation and clustering specific to fencing techniques, delivering targeted feedback without requiring extensive sensor infrastructure, thereby improving accessibility and user experience.

## 6. CONCLUSIONS

While the app demonstrates strong functionality and high user satisfaction, it does have limitations that can be addressed in future iterations. One major limitation is the app's reliance on clear and properly framed videos for accurate pose estimation. Poor video quality or improper framing can lead to inaccuracies in feedback [14]. To mitigate this, future updates could include a preprocessing step to assess video quality and provide recommendations before analysis.

Another limitation is the app's focus on a limited set of fencing techniques. Expanding the repertoire to include more complex movements and actions would make the app more versatile and valuable to a broader audience.

Lastly, while ChatGPT-generated feedback is generally clear and actionable, refining the prompt to account for edge cases (e.g., unconventional body movements) could further improve feedback accuracy. Additionally, enhancing the visual design and integrating multilingual support would increase the app’s accessibility and appeal.

The app effectively combines pose estimation, clustering, and AI-driven analysis to provide personalized feedback to users, addressing a critical gap in fencing training [15]. With further refinements, the app has the potential to become a vital tool for athletes seeking to enhance their technique and prevent injuries.

## REFERENCES

[1] PAUL, PAVEL LIVIU. "IMPORTANCE OF THE INDIVIDUAL LESSON IN THE TECHNICAL-TACTICAL IMPROVEMENT OF FENCING ATHLETES." Ovidius University Annals, Series Physical Education & Sport/Science, Movement & Health 24.2 (2024).
[2] Roi, Giulio S., and Diana Bianchedi. "The science of fencing: implications for performance and injury prevention." Sports medicine 38 (2008): 465-481.
[3] Thacker, S. B., et al. "Prevention of knee injuries in sports." Journal of Sports Medicine and Physical Fitness 43.165-179 (2003).

[4] Al-Jahdali, Hamdan, et al. "Improper inhaler technique is associated with poor asthma control and frequent emergency department visits." Allergy, asthma & clinical immunology 9 (2013): 1-7.

[5] Kodinariya, Trupti M., and Prashant R. Makwana. "Review on determining number of Cluster in K-Means Clustering." International Journal 1.6 (2013): 90-95.

[6] Brouillette, Christie G., et al. "Structural models of human apolipoprotein AI: a critical analysis and review." Biochimica et Biophysica Acta (BBA)-Molecular and Cell Biology of Lipids 1531.1-2 (2001): 4-46.

[7] Wittig, Andreas, and Michael Wittig. Amazon Web Services in Action: An in-depth guide to AWS. Simon and Schuster, 2023.

[8] An, J., W. Ding, and C. Lin. "ChatGPT." tackle the growing carbon footprint of generative AI 615 (2023): 586.

[9] Shorten, Connor, et al. "Structuredrag: Json response formatting with large language models." arXiv preprint arXiv:2408.11061 (2024).

[10] Chien, Aichi, Bin Dong, and Zuowei Shen. "Frame-based segmentation for medical images." Communications in Mathematical Sciences 9.2 (2011): 551-559.

[11] Guelmami, Noomen, et al. "Injury Prevention, Optimized Training and Rehabilitation: How Is AI Reshaping the Field of Sports Medicine." New Asian Journal of Medicine 1.1 (2023): 30-34.

[12] Biró, Attila, Antonio Ignacio Cuesta-Vargas, and LászlóSzilágyi. "AI-controlled training method for performance hardening or injury recovery in sports." 2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics (SAMI). IEEE, 2024.

[13] Nobari, Hadi. "Takes Two to Tango: Digital Twins and AI Revolutionize Sports Science and Medicine." Acta Kinesiologica 18.2 (2024): 56-57.

[14] Sinno, Zeina, and Alan Conrad Bovik. "Large-scale study of perceptual video quality." IEEE Transactions on Image Processing 28.2 (2018): 612-627.

[15] Chakriswaran, Priya, et al. "Emotion AI-driven sentiment analysis: A survey, future research directions, and open issues." Applied Sciences 9.24 (2019): 5462.