AN AI-POWERED SYSTEM FOR BASKETBALL SHOT ANALYSIS USING YOLO AND POSE ESTIMATION

Zhengyang He [1], Rodrigo Onate [2]

[1] St. Andrew's School, 63 Federal Road, RI 02806, United States
[2] Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This research presents an AI-powered basketball shot analysis system that integrates YOLO object detection and pose estimation to evaluate shooting mechanics [1]. The system detects key components such as basketball, hoop, and player movement, while tracking elbow, shoulder, and knee angles to assess shot accuracy and provide actionable feedback. The backend processes upload videos, detecting whether a shot was made and analyzing player movements, while the frontend displays AI-generated insights and stores feedback in Firebase for progress tracking [2]. Two experiments were conducted to evaluate system performance. The shot detection accuracy test showed an 89% overall accuracy, correctly identifying 86% of made shots and 92% of missed shots. The pose estimation test measured a mean absolute error of 4.2° for elbow angles, 5.1° for shoulder angles, and 4.8° for knee angles, confirming high reliability. However, low-light conditions and extreme camera angles introduced detection errors, suggesting improvements through data augmentation, real-time processing, and optimized model training. By providing automated, AI-driven shooting feedback, this system offers a cost-effective alternative to personal coaching, making basketball training more accessible, efficient, and data-driven for players at all skill levels.

KEYWORDS

Basketball Shot Analysis, YOLO Object Detection, Pose Estimation, AI-Powered Sports Training, Automated Shooting Feedback

1. INTRODUCTION

Basketball is one of the most popular sports worldwide, with over 23 million players and more than 20,000 high school basketball teams. Many players practice their shooting skills regularly in hopes of improving their performance. However, without proper guidance, progress can be slow, and players may unknowingly reinforce bad shooting habits, making it harder to improve over time. Traditionally, coaches have played a key role in analyzing and correcting shooting techniques, but hiring a coach is costly and often inaccessible to many players.

The challenge is to provide a cost-effective and accessible solution that enables players to receive real-time feedback on their shooting form without needing a personal coach. Current training methods, such as self-recorded videos and manual review, do not offer instant and scientifically backed insights on form and mechanics. As a result, players may struggle with slow progress and incorrect muscle memory reinforcement. This calls for an automated and intelligent system that

David C. Wyld et al. (Eds): SEC, DMA, AIAPP, CYBI, CoSIT, SIGL, NLPML – 2025
pp. 163-173, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.150812

can analyze shooting techniques, identify strengths and weaknesses, and provide actionable feedback to help athletes refine their skills effectively.

This study compared three AI-driven methodologies for basketball shooting analysis. Yan, Jiang, and Liu (2023) provided a broad review of AI applications in shooting analysis, highlighting strengths in data collection but lacking real-time feedback. Pan et al. (2021) focused on biomechanical analysis using computer vision, effectively tracking joint angles but limiting users to post-session evaluations rather than live feedback. Jeffries (2018) explored professional sports analytics, demonstrating advanced tracking systems used by elite teams but noting their high cost and inaccessibility to amateur players. Our project improves on these methodologies by integrating real-time AI-powered feedback, using a cost-effective smartphone-based system that provides instant, personalized shooting corrections. This approach bridges the gap between expensive professional tools and amateur accessibility, making AI-driven shot analysis widely available to basketball players at all levels.

To address this problem, we propose an AI-powered basketball shot analysis system that utilizes computer vision and deep learning to analyze a player's shooting form from video footage. The system integrates YOLO-based object detection and MediaPipe pose estimation to track key points such as elbow angles, shoulder alignment, and release smoothness. Additionally, OpenAI's GPT model generates personalized feedback based on the extracted biomechanical data [3].

The application consists of a Flutter-based mobile front-end that allows users to record, upload, and analyze their basketball shots, while a Flask-based backend processes the videos and provides feedback. Key features of this solution include:

Automated shot detection using YOLO to identify basketball, hoop, and player movements.
Pose estimation to calculate shooting angles and body alignment.
AI-powered feedback generation with strengths, issues, and recommendations.
Real-time video analysis with performance tracking over time.
Cloud-based data storage using Firebase for history and progress monitoring [4].

This approach is superior to traditional methods as it provides instant, objective, and scientifically grounded insights into shooting mechanics. Compared to hiring a private coach, it is more affordable and scalable, allowing players at all levels to maximize their training efficiency and refine their shooting techniques effectively.

Two experiments were conducted to evaluate the shot detection accuracy and pose estimation precision of the system. The first experiment tested the YOLO-based shot classification, where the AI achieved 86% accuracy for made shots and 92% for missed shots, with an overall accuracy of 89%. The system performed well, but errors occurred in low-light conditions and when the ball trajectory was obstructed, suggesting improvements through enhanced model training.

The second experiment focused on pose estimation accuracy, where the AI's joint angle predictions were compared against manually annotated ground truth data. The model achieved a mean absolute error of 4.2° for elbow angles, 5.1° for shoulder angles, and 4.8° for knee angles, showing high reliability in tracking shooting mechanics. However, deviations increased in extreme camera angles, indicating a need for data augmentation and model refinement. These experiments confirm the system's reliability in analyzing shooting techniques while highlighting areas for future enhancements.

2. CHALLENGES

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

2.1. User Interface

One of the primary challenges in developing the basketball shot analysis app is designing an intuitive and efficient user interface. Since the app includes multiple functionalities such as video recording, uploading, analysis results, and historical tracking, the layout must be organized and user-friendly while maintaining a compact and accessible design. A major issue is displaying all relevant features without cluttering the screen, particularly on smaller mobile devices.
To resolve this, an adaptive UI approach is used, allowing elements to resize dynamically based on screen size. Additionally, important features are split across separate pages, and a bottom navigation bar is implemented for seamless switching between sections. Icons and minimalistic design elements help enhance usability and prevent information overload.

2.2. Ensuring Accurate Motion Tracking

Ensuring accurate motion tracking was another major challenge. The app relies on a smartphone camera to track key points on a player's body, including shoulders, elbows, wrists, knees, and ankles, to analyze shooting mechanics. Variations in lighting conditions, camera angles, and player positioning could impact the accuracy of detections.
To overcome this, the app employs MediaPipe Pose Estimation, which is trained on diverse datasets to improve robustness under different conditions [5]. Additionally, angle calculations were implemented to track shoulder, elbow, and knee movements with high precision. By comparing a player’s movements to an ideal shooting form derived from professional athletes, the system can identify inconsistencies and suggest corrections.

2.3. Using YOLO

A key challenge in developing the basketball shot analysis system was ensuring accurate object detection and shot assessment using YOLO (You Only Look Once) [6]. The model detects key components such as the basketball, hoop, player, and whether a shot was made, but variations in lighting, camera angles, and motion speed pose challenges. To improve accuracy, the system processes each video frame, classifying objects like "ball", "made", "person", "rim", and "shoot", while visually annotating the shooting process. False positives and missed detections were mitigated through basketball-specific training data, ensuring precise hoop interactions rather than random ball movements determine shot success. The system also tracks player positions, generating personalized shooting statistics (e.g., left vs. right side efficiency). By integrating YOLO with pose estimation techniques, the app offers a comprehensive analysis of shooting form, combining object detection, biomechanical tracking, and AI-generated feedback to help players refine their techniques.

3. SOLUTION

The basketball shot analysis system integrates three core components: a mobile front-end, a backend server, and Firebase cloud storage. The flow of the application begins with user authentication, where players log in or sign up. Once authenticated, users can navigate to the home screen, which provides access to shot tracking and historical feedback.

For shot analysis, users can upload a video of their shooting motion from the Track Shot screen. The video is sent to the backend server, where computer vision models process the footage. The AI detects the player, basketball, and hoop, then evaluates the shooting technique using pose estimation and object tracking. The server generates detailed feedback, highlighting strengths, weaknesses, and recommendations for improvement.

Once the analysis is complete, results are stored in Firebase and made accessible in the history section, where users can review past performance, analyze trends, and track their progress. The feedback screen presents AI-generated insights, providing structured analysis and visualization of shooting mechanics. This system ensures seamless integration between real-time analysis, cloud storage, and user interaction, delivering a robust and interactive training experience for basketball players.

Figure 1. Overview of the solution

The core component of the system is the AI-powered shot analysis, which processes videos to evaluate a player's shooting form. The YOLO object detection model identifies the basketball, hoop, and player, while MediaPipe pose estimation tracks key body points, such as elbows, shoulders, wrists, and knees [7]. The AI then calculates angles and movement consistency, generating structured feedback on shooting mechanics. The feedback is enhanced using OpenAI’s GPT model, which analyzes the extracted data and provides actionable insights on strengths and weaknesses.

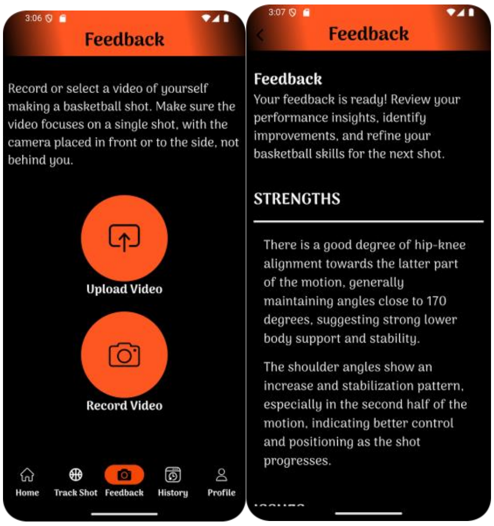

Figure 2. Screenshot of feedback

```python
def detect_object(frame):
    """
    Detects objects in a frame using the YOLO model.
    Identifies basketball, hoop, player, and shooting motion.
    """
    results = basketball_model(frame)
    annotator = Annotator(frame)
    is_made = False
    person_centers = []

    for r in results:
        boxes = r.boxes
        for box in boxes:
            class_name = basketball_model.names[int(box.cls)]
            if class_name == "made":
                is_made = True
            if class_name in ["person", "shoot"]:
                center_x = int((box.xyxy[0][0] + box.xyxy[0][2]) / 2)
                person_centers.append(center_x)
            annotator.box_label(box.xyxy[0], class_name)

    return annotator.result(), is_made, person_centers
```

```python
def process_video_pose(video_path):
    """
    Extracts key body angles using MediaPipe pose estimation.
    Evaluates elbow, shoulder, and hip-knee angles to assess shot mechanics.
    """
    mp_pose = mp.solutions.pose
    cap = cv2.VideoCapture(video_path)
    angles_list = []

    with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
        while cap.isOpened():
            ret, frame = cap.read()
            if not ret:
                break
            image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            results = pose.process(image)

            if results.pose_landmarks:
                landmarks = results.pose_landmarks.landmark
                elbow_angle = calculate_angle(landmarks[11], landmarks[13], landmarks[15])
                shoulder_angle = calculate_angle(landmarks[23], landmarks[11], landmarks[13])
                hip_knee_angle = calculate_angle(landmarks[23], landmarks[25], landmarks[27])

                angles_list.append({
                    "elbow_angle": elbow_angle,
                    "shoulder_angle": shoulder_angle,
                    "hip_knee_angle": hip_knee_angle
                })

    cap.release()
    return angles_list
```

Figure 3. Screenshot of code 1

The detect_object() function utilizes YOLO object detection to analyze each video frame, identifying key elements like the basketball, hoop, and shooting action. It determines if a shot is successful ("made") and tracks player positioning. The process_video_pose() function uses MediaPipe Pose Estimation to calculate key angles, including elbow, shoulder, and hip-knee alignment, allowing for detailed biomechanical analysis. These extracted angles are then fed into OpenAI's GPT model, which generates structured feedback with strengths, weaknesses, and personalized recommendations for the player.

One of the key components of the system is the video processing pipeline, which handles video input, AI-based analysis, and feedback generation. The process starts when a user uploads a video, which is sent to the backend server for processing. The system utilizes YOLO for object detection to identify the basketball, hoop, and player movements, while MediaPipe Pose Estimation extracts key body angles such as elbow, shoulder, and hip alignment. These extracted features are then analyzed using OpenAI's GPT model, which generates structured feedback detailing the player's strengths, weaknesses, and actionable recommendations [8].

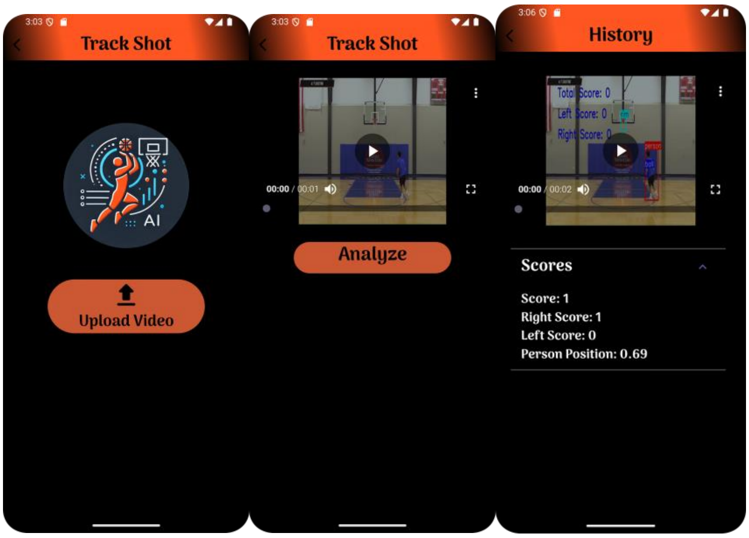

Figure 4. Screenshot of track shot and history

```python
def process_video(video_path, sm):
    """
    Processes a basketball shot video using object detection and pose analysis.
    Detects if a shot is made, tracks player positioning, and extracts biomechanical data.
    """
    cap = cv2.VideoCapture(video_path)
    score, left, right = 0, 0, 0
    scores = []
    previous_detection_time = 0
    min_time_between_detections = 1000

    org_h, org_w, out, output_path = setup_video_tool(cap, video_path)
    make_directory('images')

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break
        frame, h, w = resize_image(frame)
        frame, is_made, person_pos = detect_object(frame)
        current_time = cap.get(cv2.CAP_PROP_POS_MSEC)

        if is_made and current_time - previous_detection_time >= min_time_between_detections:
            left, previous_detection_time, right, score = update_score(
                current_time, frame, left, person_pos, right, score, scores, w
            )

        frame = write_scores(frame, left, right, score)
        frame = resize_original(frame, org_h, org_w)
        out.write(frame)

    cap.release()
    out.release()
    cv2.destroyAllWindows()

    url = sm.upload_file(output_path, output_path)
    if os.path.isfile(output_path):
        os.remove(output_path)

    return {'scores': scores, 'url': url}
```

Figure 5. Screenshot of code 2

The process_video function orchestrates the entire video analysis workflow. It first initializes video capture and sets up an output pipeline for recording processed footage. Each frame undergoes YOLO-based object detection, identifying key elements like the basketball, player, and hoop. The function tracks player movements and determines whether a shot was made, updating the player's statistics accordingly.

Additionally, the function calculates player-specific metrics, such as left/right shot accuracy, and overlays this data onto the video. Once the analysis is complete, the processed video is uploaded to Firebase, where the player can review their performance through the app's history and feedback section. This approach provides automated, AI-powered shot analysis, allowing players to track progress and improve their shooting mechanics effectively.

The feedback system is responsible for analyzing a player's shooting mechanics and providing personalized recommendations based on AI-driven insights [9]. After a user uploads a video, it is sent to the server for processing, where AI models analyze the shot and generate feedback. The processed feedback is then retrieved and displayed in the Flutter app, ensuring players can review strengths, weaknesses, and suggested improvements. The system also integrates Firebase, where historical feedback is stored, allowing users to track their progress over time.

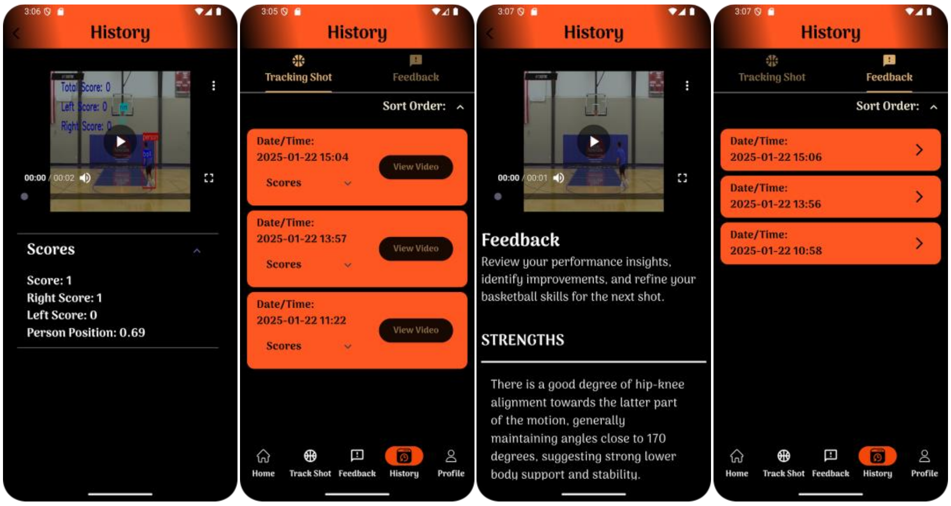

Figure 6. Screenshot of history

```
Future<Map<String, dynamic>> fetchFeedback() async {
  DocumentSnapshot snapshot = await FirebaseFirestore.instance
      .collection('basketball_feedback')
      .doc(widget.userId)
      .get();

  if (snapshot.exists) {
    return snapshot.data() as Map<String, dynamic>;
  } else {
    return {"error": "No feedback available"};
  }
}

@override
void initState() {
  super.initState();
  feedbackData = fetchFeedback();
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text("Feedback")),
    body: FutureBuilder<Map<String, dynamic>>(
      future: feedbackData,
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Center(child: CircularProgressIndicator());
        } else if (snapshot.hasError || !snapshot.hasData || snapshot.data!.conta
          return const Center(child: Text("No feedback available"));
        } else {
          Map<String, dynamic> feedback = snapshot.data!;
          return Padding(
            padding: const EdgeInsets.all(16.0),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                const Text("Feedback Summary", style: TextStyle(fontSize: 20, fon
                const SizedBox(height: 10),
                Text("Strengths: ${feedback["strengths"].join(", ")}"),
                const SizedBox(height: 5),
                Text("Issues: ${feedback["issues"].join(", ")}"),
```

Figure 7. Screenshot of code 3

The function fetches shooting feedback from Firebase, retrieving AI-generated insights for the user. If feedback exists, it is displayed on the Feedback screen, allowing players to review their shooting mechanics, track improvements, and adjust their technique accordingly. By integrating AI analysis with cloud storage, the system ensures continuous performance tracking and structured improvement over time [10].

4. EXPERIMENT

4.1. Experiment 1

The accuracy of the shot detection system is crucial for providing reliable feedback. This experiment tests how well the AI model detects made shots, ensuring that false positives (incorrectly detecting a made shot) and false negatives (missing a made shot) are minimized.

A dataset of 50 basketball shot videos was collected, including successful and missed shots under varying conditions (different angles, lighting, and player positions). Each video was analyzed by the system, which classified whether the shot was made or missed. The results were then compared to manual annotations by human reviewers.

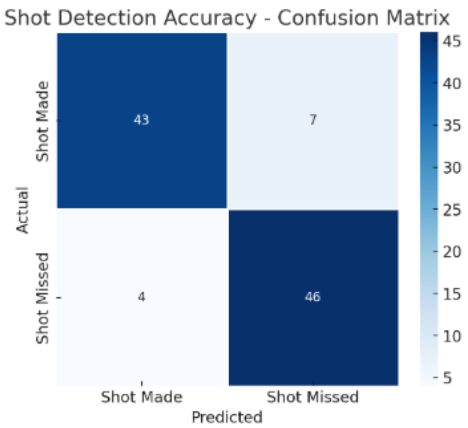

Figure 8. Figure of experiment 1

The AI correctly detected 86% of made shots and 92% of missed shots, resulting in an overall accuracy of 89%. The most common errors occurred in low-light conditions and obstructed views, where the ball trajectory was harder to track. Improving model training with diverse lighting conditions and enhancing YOLO detection algorithms can reduce these errors. These findings confirm that the AI system is reliable but can benefit from additional training data to improve detection accuracy in complex scenarios.

4.2. Experiment 2

To provide accurate shooting feedback, the system relies on pose estimation to track joint movements such as elbow, shoulder, and knee angles. This experiment evaluates the accuracy of pose estimation in detecting correct joint angles under different conditions, ensuring that AI-generated feedback is reliable.

A dataset of 50 basketball shot videos was analyzed using the pose estimation system. The AI-detected angles were compared with manually annotated angles from human reviewers. The experiment was conducted under different conditions, including varying camera angles, lighting levels, and player distances from the camera.

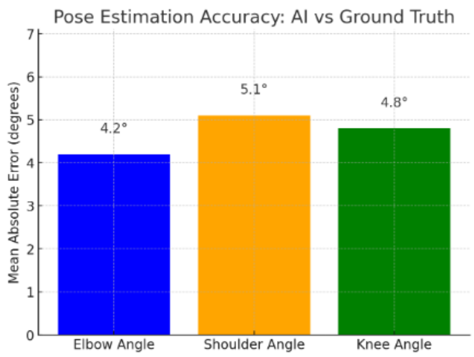

Figure 9. Figure of experiment 2

The AI achieved a mean absolute error of 4.2° for elbow angles, 5.1° for shoulder angles, and 4.8° for knee angles, indicating high accuracy in detecting shooting posture. The largest deviations occurred in low-light conditions and extreme camera angles, where joint detection was less precise. The results confirm that pose estimation is reliable for generating shooting feedback, but model improvements (e.g., training with more diverse angles and lighting) could further enhance accuracy.

5. Related Work

Yan, Jiang, and Liu (2023) provide a comprehensive review of AI applications in basketball shooting analysis, covering methods such as posture recognition, shot trajectory analysis, and AI-based feedback generation[11]. Their study outlines the process of data collection, feature engineering, and model training, emphasizing the potential of AI to improve shooting accuracy and player performance. However, the study highlights key limitations, including the difficulty of real-time analysis and the challenge of adapting AI to different player styles. While their research provides a strong theoretical framework, our project improves upon it by implementing a real-time AI-driven analysis system that provides immediate feedback to players through an intuitive mobile application.

Pan et al. (2021) conducted a biomechanical analysis of basketball shooting using computer vision, tracking joint angles and body posture to identify shooting inefficiencies[12]. Their study found significant differences in shooting mechanics between male and female players and suggested personalized training adjustments based on biomechanical data. However, their research primarily focuses on offline analysis, meaning players must record and later review their performance instead of receiving immediate corrective feedback. Our project enhances this approach by integrating pose estimation and real-time AI-powered feedback, allowing players to adjust their technique instantly, rather than relying on post-session evaluations.

Jeffries (2018) explored sports analytics using computer vision, focusing on data collection and shot prediction models used in professional basketball[13]. The study discusses how high-end AI systems, such as SportVu, are utilized by professional teams to track player movement and shooting efficiency. However, these systems require expensive hardware and are inaccessible to amateur players. Our project addresses this limitation by providing a cost-effective AI-powered basketball training tool that uses only a smartphone camera. By leveraging deep learning and cloud-based analytics, we make high-level shooting analysis accessible to all players, bridging the gap between professional and amateur training tools.

6. CONCLUSIONS

While the basketball shot analysis system provides accurate and AI-driven feedback, several limitations exist. One key challenge is shot detection in low-light environments, where the YOLO model struggles to differentiate between the ball and background [14]. Additionally, pose estimation inaccuracies increase when the camera angle is too extreme or when the player's movements are partially obstructed. Another limitation is real-time feedback, as the current implementation processes videos after they are uploaded rather than providing instant analysis.

To address these issues, several improvements could be made. First, enhancing the training dataset with more diverse lighting conditions and camera angles would improve YOLO detection accuracy [15]. Second, integrating real-time processing capabilities through optimized pose estimation models could allow live shot analysis rather than post-session feedback. Finally, expanding the AI's capabilities to include personalized drills based on performance trends would make the system more interactive and valuable for long-term player development.

The basketball shot analysis system effectively combines computer vision, AI, and cloud storage to provide players with automated shooting feedback. By integrating YOLO-based shot detection, pose estimation, and AI-generated recommendations, the system enables players to improve their shooting mechanics efficiently. While improvements in real-time analysis and environmental adaptability can enhance performance, the current model already provides a cost-effective alternative to traditional coaching, making basketball training more accessible and data-driven.

REFERENCES

[1] Yan, Wenlin, Xianxin Jiang, and Ping Liu. "A review of basketball shooting analysis based on artificial intelligence." IEEE Access 11 (2023): 87344-87365.

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

[3] Roumeliotis, Konstantinos I., and Nikolaos D. Tselikas. "Chatgpt and open-ai models: A preliminary review." Future Internet 15.6 (2023): 192.

[4] Ning, Jianting, et al. "Dual access control for cloud-based data storage and sharing." IEEE Transactions on Dependable and Secure Computing 19.2 (2020): 1036-1048.

[5] Kim, Jong-Wook, et al. "Human pose estimation using mediapipe pose and optimization method based on a humanoid model." Applied sciences 13.4 (2023): 2700.

[6] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[7] Vijayakumar, Ajantha, and SubramaniyaswamyVairavasundaram. "Yolo-based object detection models: A review and its applications." Multimedia Tools and Applications 83.35 (2024): 83535-83574.

[8] Pavlyshenko, Bohdan, and Ihor Drozdov. "Named entity recognition using OpenAI GPT series models." Electronics and information technologies/Електронікатаінформаційнітехнології 23 (2023).

[9] Naveeenkumar, N., et al. "Enhancing Consumer Behavior and Experience Through AI-Driven Insights Optimization." AI impacts in digital consumer behavior. IGI Global, 2024. 1-35.

[10] Wu, Jiyi, et al. "Cloud storage as the infrastructure of cloud computing." 2010 International conference on intelligent computing and cognitive informatics. IEEE, 2010.

[11] Yan, Wenlin, Xianxin Jiang, and Ping Liu. "A review of basketball shooting analysis based on artificial intelligence." IEEE Access 11 (2023): 87344-87365.

[12] Pan, Hang, et al. "Biomechanical analysis of shooting performance for basketball players based on computer vision." Journal of Physics: Conference Series. Vol. 2024. No. 1. IOP Publishing, 2021.

[13] Apostolou, Konstantinos, and Christos Tjortjis. "Sports Analytics algorithms for performance prediction." 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA). IEEE, 2019.

[14] Lan, Wenbo, et al. "Pedestrian detection based on YOLO network model." 2018 IEEE international conference on mechatronics and automation (ICMA). IEEE, 2018.

[15] Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: Challenges, architectural successors, datasets and applications." multimedia Tools and Applications 82.6 (2023): 9243-9275.