A Deep-Learning Framework for the Biomechanical Analysis of Lateral Angle and Stance Width Ratio Error

Harman Jani
harman.j@ahduni.edu.in
Kaival Shah
kaival.s@ahduni.edu.in
Divykumar Patel
divykumar.p@ahduni.edu.in
Krutarth Trivedi
krutarth.t@ahduni.edu.in

***Abstract*—Competitive sports like basketball necessitate rapid and intense movements, such as jump landings, which can make athletes susceptible to injuries from neuromuscular control and joint mechanics. Biomechanical features during landings are closely related to the importance of proper movement execution and postural stability. Innovative research introduces a pioneering image analysis framework designed to comprehensively assess jump-landing biomechanics in athletes, enabling the provision of targeted recommendations to coaches for injury prediction and athlete training needs. Our framework offers a distinctive approach to analyzing lateral flexion and stance width ratio errors, crucial for understanding injury mechanisms and formulating effective prevention strategies. This approach significantly enhances injury prevention measures and optimizes personalized training regimens for athletes. A key innovation is the integration of YOLOv8, a state-of-the-art deep learning framework, which streamlines the process by directly outputting images with errors annotated on them. This approach eliminates the sequential process of keypoint detection followed by error calculation, significantly boosting computational efficiency and speed. By leveraging the power of deep learning, our framework simplifies the analysis process, ensures a more accurate and efficient identification of biomechanical errors, and paves the way for more precise and personalized training recommendations, significantly contributing to the prevention of injuries and the optimization of athlete performance.**

***Index Terms*—Pose Estimation, Landing Error, Supervised Learning, Training, YOLO Model, Object Detection, Computer Vision, Machine Learning, Sports Analytics.**

I. Introduction

In many sports, high-intensity activities performed for prolonged periods may increase the risk of injury. Such injuries often result from incorrect posture and movements during gameplay, which can compromise neuro-muscular control. Consequently, these injuries not only impact the performance of individual players but can also affect the overall team's success. Identifying and addressing these incorrect postures and movements is crucial for injury prevention and athlete well-being.

This project aims to tackle the challenge of identifying detrimental postures and movements early on to mitigate the risk of injuries. To achieve this, various techniques are employed. One such approach involves analyzing the ratio of the distance between key points on the shoulders to the distance between key points on the ankles to detect incorrect postures. Additionally, lateral flexion, which assesses the angle formed by specific body landmarks such as the midpoint of shoulder distance, critical points of the hip, and critical points of the ankle, is utilized to recognize flawed movements. Innovatively, we incorporate a YOLO (You Only Look Once) model into our analysis pipeline to automate the process of detecting lateral flexion and stance width errors during jump landings. By harnessing the power of deep learning, the YOLO model can efficiently and accurately identify these errors by processing annotated images. Through a combination of training the YOLO model with annotated images and subsequently testing it on new data, we can achieve rapid and precise detection of biomechanical errors, enhancing injury prevention strategies and optimizing athlete performance.

Advancements in sports biomechanics have revolutionized injury prevention through automated movement analysis. For instance, systems like OpenPose enable real-time pose estimation from video, facilitating the identification of risky movement patterns. Additionally, the PhysiMax system automates scoring of the Landing Error Scoring System (LESS) using 3D Microsoft Kinect, reducing reliance on expert clinicians. These innovations streamline the assessment process, making it more accessible and cost-effective for large-scale screening. Moreover, the integration of deep learning techniques allows for accurate prediction of injury risk factors from 2D video recordings, paving the way for smartphone-based applications for injury risk screening in sports.

II. Methodology

To begin with, we extracted the crucial points of the human pose from an athlete's jumping video. We segmented the video into individual frames and converted each from BGR to RGB format to ensure compatibility with the MediaPipe pose estimation tool. MediaPipe provides 33 body landmarks, but we only used four landmarks (11, 12, 27, 28) to estimate kinematic features. The coordinates we obtained were unnormalised.

Next, we calculated whether the athlete had correct or incorrect body posture. Python was our chosen tool for automating this task. Our code calculates the stance width ratio and lateral flexion angle. The algorithm identified errors if the ratio and angles exceeded the set threshold of ±5

After training the model on architectures like EfficientNet and MoveNet, we found that the YOLOv8 model best satisfied our requirements to predict error movements in athletes. We trained YOLOv8 on our dataset for 50 epochs, after which there was no further improvement in accuracy. We referred to the documentation provided by Ultralytics and Roboflow throughout the training process.

Algorithm 1 Error Detection Algorithm

1. Load the input image from a file.
2. Initialize an empty list to store the details obtained from the detection and processing steps.
3. For each pose landmark point to be detected:
3.1. Determine the approximate location of the landmark point in the image based on its index.
3.2. Record the pixel coordinates of the landmark point.
4. Calculate the distance between the shoulder points and the ankle points using the Euclidean distance formula.
5. Calculate the stance ratio by dividing the shoulder distance by the ankle distance.
6. Determine the stance error based on the stance ratio:
6.1. If the stance ratio is within the range [0.8, 1.2], mark as ”no error”.
6.2. If the stance ratio is less than 0.8, mark as ”wide stance error”.
6.3. If the stance ratio is greater than 1.2, mark as ”narrow stance error”.
7. Calculate the midpoint of the hips and the top of the frame.
8. Compute the lateral angle formed by the top of the frame, hips, and shoulders.
9. Determine if the lateral angle indicates an error:
9.1. If the lateral angle is within the range [0, 3.1] or [176.9, 180], mark as ”no error”.
9.2. Otherwise, mark as ”error”. And based on the points left or right lateral flexion is also provided.
10. Save all the details obtained from the detection and processing steps.

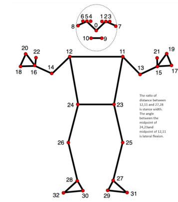
III. Preprocessing of Dataset

In the Dataset Preprocessing part, the process starts with adjusting a tilted video using Adobe Video Editor, ensuring correct orientation for data extraction and annotation. 210 frames from a 35-second video of athletes performing jump landings were extracted and manually annotated using Roboflow. Each image was assigned two classes out of six, representing different types of errors in lateral flexion and stance width. The area of interest in each image was annotated using a bounding box.

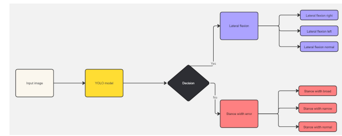
The dataset was initially imbalanced, with fewer images containing athlete errors. Data augmentation was performed to address this imbalance, creating new images by applying transformations like adding Gaussian noise, flipping images horizontally, and applying affine transformations. This augmentation targeted images containing errors.

The data augmentation code iterates over all directories containing images and labels, checking if the class ID corresponds to an error class. If so, it applies augmentations to the image and its bounding box annotations, saving the augmented images with a modified filename. This process significantly increased the dataset size and improved the model's learning capabilities.

In summary, the steps taken to prepare a balanced and comprehensive dataset for the analysis of jump-landing biomechanics, including video adjustment, manual annotation, and data augmentation to address class imbalance and enhance the model's learning capabilities.



IV. Proposed Framework



V. Results

We started by estimating the critical points of human pose in the collegian athlete jump data using MediaPipe. The next step was calculating whether the athlete had correct or incorrect body posture, i.e., errors. Hence, by writing appropriate code, we automated this task by calculating the stance width ratio, i.e., the distance between the shoulder and the distance between ankle points and lateral flexion angle. The algorithm marked errors if the ratio and angles exceeded the set threshold.

Fig. 1. Pose Estimation using MediaPipe

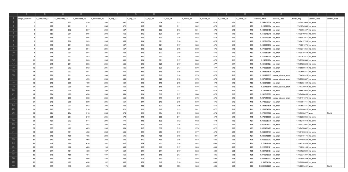


Fig. 2. Error Marked and Saved

Therefore, after building foundations, the next step was to annotate the errors in images. For this purpose, we used tools by Roboflow.

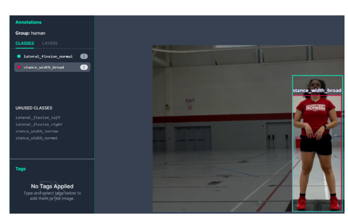


Fig. 3. Annotation using Roboflow

However, after preparing the preliminary data set, we realised our data needed to be more balanced. Therefore, we adopted the method of oversampling the underrepresented samples. From the existing dataset, we generated more images of underrepresented classes by augmenting those images, particularly by applying Gaussian blur, rescaling, rotating, and flipping.

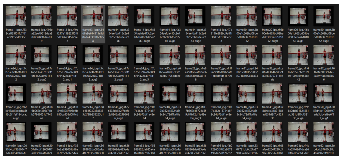


Fig. 4. Data Augmentation

We then exported this dataset to Roboflow to split the data into test, train and validation sets and move the data to the cloud for more straightforward accessibility while training.

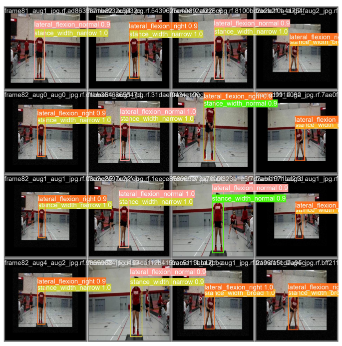
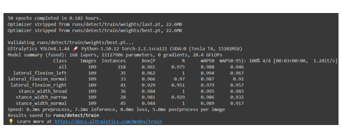


Fig. 5. Input Data for YOLOv8

Furthermore, after much consideration, we used the YOLOv8 model to predict error movements in athletes. There are various image classifiers and pose estimators, some of which we implemented to build our model. However, we found YOLO to best fit our requirements by having a single model to predict and classify the data. After training the model for 50 epochs on the data.



```
50 epochs completed in 0.182 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.6MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.6MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.1.44 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11127906 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 4/4 [00:03<00:00,  1.24it/s]
                   all        109        218      0.961      0.975      0.988      0.946
   lateral_flexion_left        109         35      0.962          1      0.994      0.967
 lateral_flexion_normal        109         33      0.966       0.97      0.987       0.92
  lateral_flexion_right        109         41      0.929      0.951      0.979      0.957
     stance_width_broad        109         36      0.984          1      0.995      0.983
    stance_width_narrow        109         28      0.981      0.929      0.986      0.932
    stance_width_normal        109         45      0.944          1      0.989      0.917
Speed: 0.3ms preprocess, 7.2ms inference, 0.0ms loss, 5.0ms postprocess per image
Results saved to runs/detect/train
💡 Learn more at https://docs.ultralytics.com/modes/train
```

Fig. 6. Model training on YOLOv8

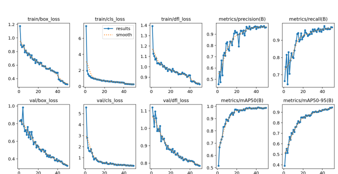


Fig. 7. Model Evaluation

We achieved 96% precision, 97% recall, and 0.98 mAP50 and 0.94 mAP50-95. The training loss after 40 epochs stabilised to a near zero value, and the bounding box value was also near zero. Precision and recall improved and stabilised at 98%. Mean Average Precision (mAP) also stabilised after 40 epochs. The validation loss is also consistent with the above result.

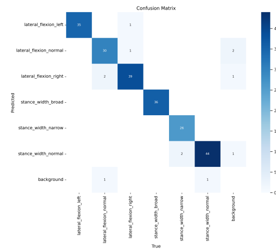


Fig. 8. Confusion Matrix

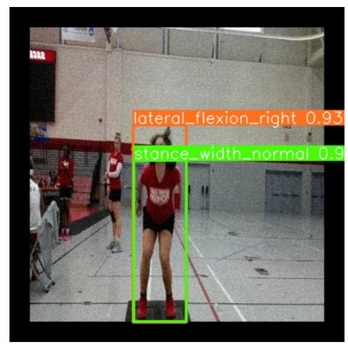
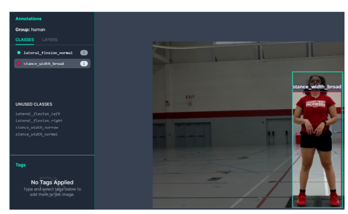
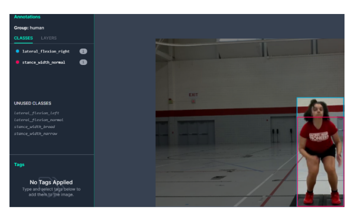


Fig. 9. Prediction by YOLOv8

VI. Images

The below image demonstrate how annotations of stance width and lateral flexion are done.





VII. Discussion

In the Discussion section of your research paper, you delve into the methodologies and outcomes of your project aimed at analyzing jump-landing biomechanics in athletes using advanced image processing and machine learning techniques. The project leverages MediaPipe, a Google-developed framework, to extract keypoints from human body movements captured in videos. These keypoints are then processed to calculate the stance ratio and lateral flexion, which are crucial biomechanical features for understanding athletic performance and injury risk.

The extracted keypoints are used to annotate errors in the athletes' movements, a process facilitated by Roboflow, a software designed for creating and managing datasets for machine learning projects. This manual annotation process assigns two classes out of six possible classes to each image, representing different types of errors in lateral flexion and stance width. The area of interest in each image is also annotated using a bounding box, focusing the analysis on the relevant parts of the image.

To address the imbalance in the dataset, where images containing athlete errors were significantly less than errorless images, data augmentation was employed. This involved creating new images by applying various transformations to the existing images, such as adding Gaussian noise, flipping the images horizontally, and applying affine transformations. This process was specifically targeted at images containing errors, as indicated by the presence of certain class IDs in the annotations.

Following the preprocessing of the dataset, the project aims to develop a model that can train on this annotated dataset. The model is designed to take images as input and output whether an error was made or not. To achieve this, transfer learning was utilized, employing the EfficientNetB0 model as the base model. The model was fine-tuned on the training dataset, with a learning rate scheduler implemented to adjust the learning rate during training. The model's performance was evaluated on both the validation and test datasets, with the training accuracy and validation accuracy plotted to visualize the model's learning progress.

The project also explored the use of bounding box information in the dataset to improve the model's output. This was achieved by integrating the YOLO (You Only Look Once) object detection model, which was used to annotate errors on images. The YOLO model was trained on the dataset, and its predictions were used to enhance the model's understanding of the biomechanical errors in athletic movements.

In summary, the Discussion section outlines the methodologies and outcomes of the project, highlighting the use of advanced image processing and machine learning techniques to analyze jump-landing biomechanics in athletes. The project demonstrates the potential of these techniques in identifying and classifying errors in athletic movements, with the goal of providing targeted recommendations for injury prevention and athlete training needs. Code implementation - colab link

VIII. Conclusion

In conclusion, our research presents a comprehensive approach to analyzing jump-landing biomechanics in athletes through the integration of advanced image processing and machine learning techniques. Leveraging MediaPipe for keypoint extraction and Roboflow for dataset annotation, we addressed the critical challenge of identifying errors in athletic movements. Through data augmentation and model training using transfer learning with pretrained image classifiers, we achieved high precision and recall rates, indicating the model's efficacy in error detection. Furthermore, the incorporation of YOLOv8 object detection enhanced the model's performance by leveraging bounding box information. Our results demonstrate the potential of these techniques in not only identifying biomechanical errors but also in providing valuable insights for injury prevention and athlete training optimization. Moving forward, our methodology offers a robust framework for further research in sports biomechanics, with implications for enhancing athletic performance and minimizing injury risks. As technology continues to advance, we anticipate continued refinement and application of these techniques to benefit both athletes and sports science practitioners.

IX. References

1) Sharma, S., Divakaran, S., Kaya, T., Taber, C., Raval, M. (n.d.). A framework for biomechanical analysis of jump ... A Framework for Biomechanical Analysis of Jump Landings for Injury Risk Assessment. *http://dx.doi.org/10.1109/PRDC59308.2023.00052*

2) Zhu, D., Zhang, H., Sun, Y., Qi, H. (2021). Injury risk prediction of aerobics athletes based on big data and computer vision. Scientific Programming, 2021, 1-10. *https://doi.org/10.1155/2021/5526971*.

3) Myer, G. D., Chu, D. A., Brent, J. L., Hewett, T. E. (2008). Trunk and hip control neuromuscular training for the prevention of knee joint injury. Clinics in sports medicine, 27(3), 425-448. *https://doi.org/10.1016/j.csm.2008.02.006*.

4) Fransz, D. P., Huurnink, A., Kingma, I., de Boode, V. A., Heyligers, I. C., van Dieen, J. H. (2018). Performance on a single-legged drop- ¨ jump landing test is related to increased risk of lateral ankle sprains among male elite soccer players: a 3-year prospective cohort study. The American journal of sports medicine, 46(14), 3454-3462. *https://doi.org/10.1177/0363546518808027*.

5) Solawetz, J. (2023, August 3). How to Train A Custom Object Detection Model with YOLO v5. Medium. *https://towardsdatascience.com/how-to-train-a-custom-object-detection-model-with-yolo-v5-917e9ce13208*

6) Google. (n.d.). MediaPipe — google for developers. Google. https://developers.google.com/mediapipe

7) Ultralytics. (2024, March 13). Home. Ultralytics YOLOv8 Docs. https://docs.ultralytics.com/

8) Google. (n.d.). Google colaboratory. Google Colab. https://colab.research.google.com/github/ultralytics/ultralytics/blob/m