Figure Skating Report

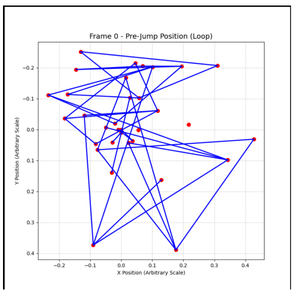
Author	Ⓜ Meghna Yadav
Content Type	
Status	In progress

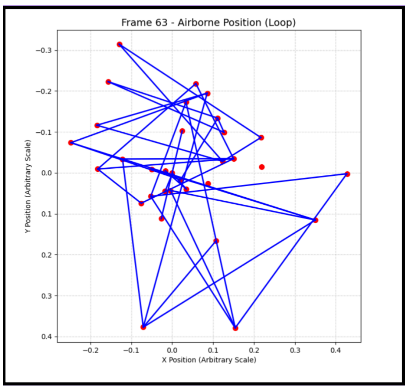
The project is a Figure Skating Jump Analyser, a time-series model built to automatically classify complex aerial manoeuvres (jumps) from raw video footage.

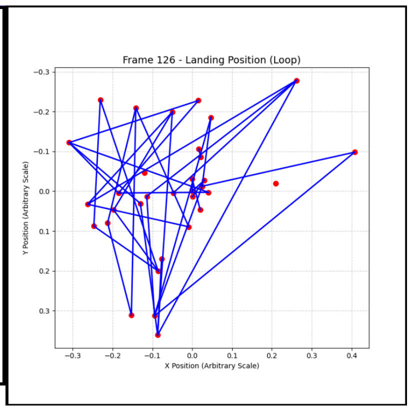
Work done so far:

1. Video Processing Pipeline

- Implemented MediaPipe Pose detection to extract 33 body landmarks per frame (from user input videos- successfully handles mp4 format)







- Process videos frame-by-frame with automatic pose tracking
- Normalise coordinates to account for different video resolutions and camera distances
- Extract x, y, z coordinates plus visibility scores for each landmark

2. Machine Learning Model

- Used Bidirectional LSTM + Encoder only transformer (tried both) architecture

(LSTM was rejected after the initial phase because it achieved only 62% accuracy. Its memory can get diluted over long sequences.) Therefore, poor specificity frequently confuses similar jumps. It's still sequential. While it can remember everything, it still processes frame-by-frame. It can't look back instantly and assign importance without reading through every frame in between. It's slow. Because it must process every frame and update the cell state sequentially, training is slow compared to parallel architectures like the Transformer.)

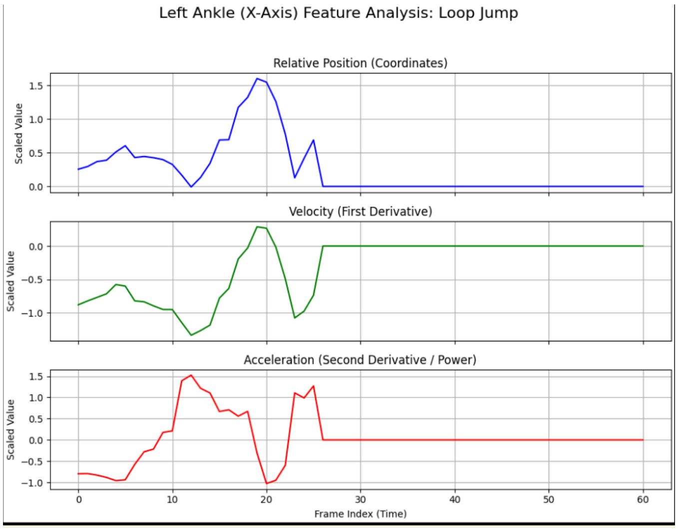
- Implemented sliding window approach (60-frame sequences) for temporal analysis
- Trained models to classify 6 jump types: Axel, Double Axel, Loop, Lutz, Salchow, Toe Loop
- Created voting mechanism that aggregates predictions across multiple windows
- Added dropout layers (30%) for regularisation to prevent overfitting

3. Scoring and Visualisation

- Developed heuristic scoring system (1-10 scale) based on model confidence levels
- Created 3D trajectory visualisation showing jump path through space (shows up as an image)
- Built HTML dashboard displaying predicted jump type and performance score (very basic, inititial format)
- Implemented colour-coded feedback (red/yellow/green) for score interpretation
- Added frame slicing capability to isolate specific jump sequences from longer videos

Our Approach

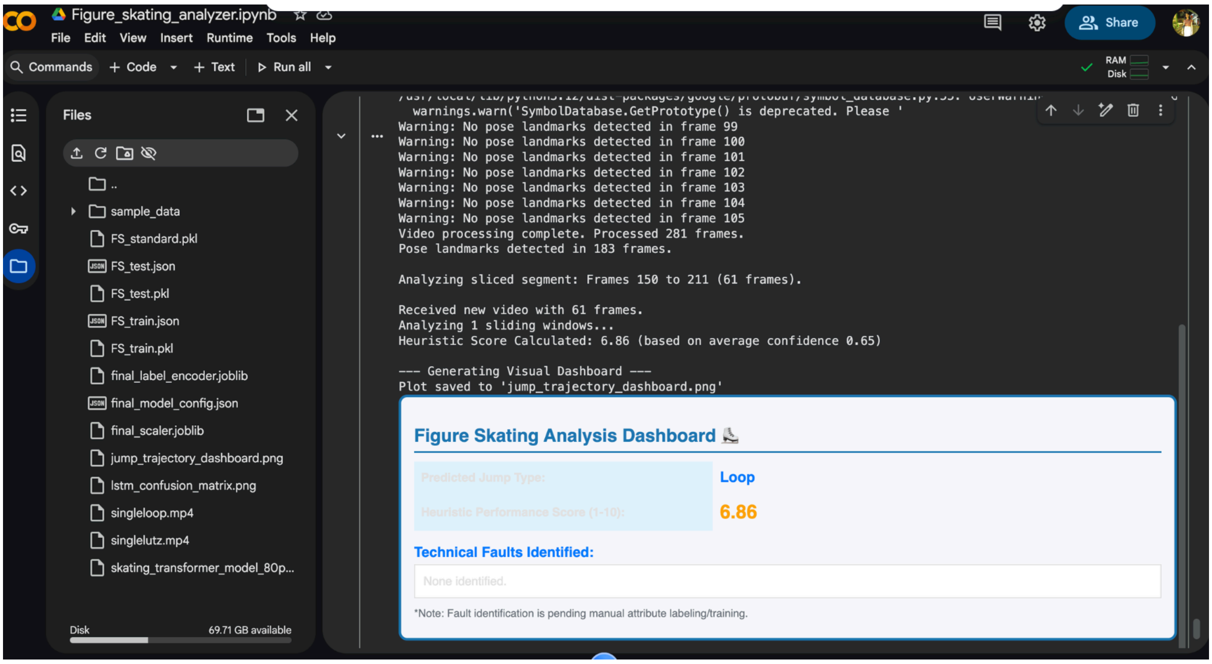
1. We extract the skater's (x, y) body joint coordinates from each video frame.



2. We created features that keep our system's results consistent regardless of the skater's speed or position on screen. This includes:

- **Relative Position:** We subtract the skater's center of mass from all joint points. This recenters the body so the model focuses on body movement rather than location on the ice.
- **Velocity and Acceleration:** We calculate how fast each joint moves and how quickly that speed changes. This helps the model understand the skater's motion and power—distinguishing, for example, between a slow crouch and a powerful jump.

3. The total input is 198 features per frame: 33 Joints × 2 Coordinates (X, Y) × 3 Feature Types (Position, Velocity, Acceleration) = 198 Total Features
4. We apply Standard Scaling (Z-score normalisation: z = (x−mean)/S.D.) to all 198 features. This puts them on the same scale, helping the model train more smoothly and consistently.
5. We used a publicly available dataset containing video links (.json files) and coordinates (.pkl files) for 33 body features. We trained on 287 video sequences across 4 jump classes. After augmenting with Jitter and Scaling, our training set expanded to 861 sequences, helping prevent overfitting due to limited data.



Scoring Metrics

We evaluated our model using these performance metrics:

- Heuristic Score
- Accuracy
- Precision, Recall, and F1-Score (Per-Class): We examine the F1-Score for each jump type (e.g., Double Axel, Lutz) to ensure the model learns distinct patterns rather than guessing. High F1-Scores across core jumps confirm effective learning.

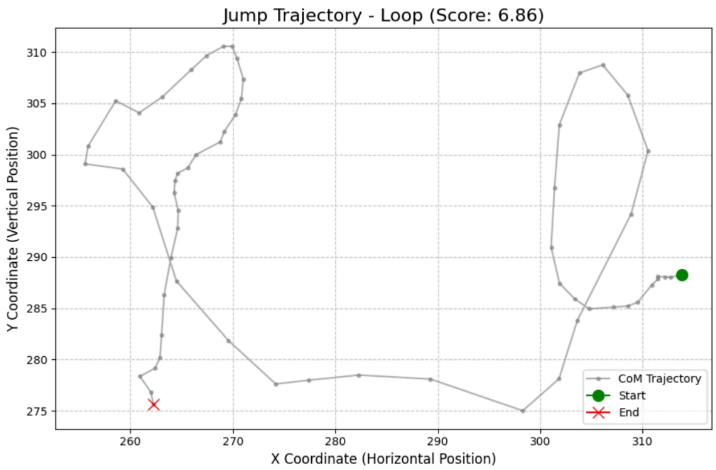
Evaluating model...
2/2 ━━━━━━━━━━━━━━━━━━━━ 1s 308ms/step

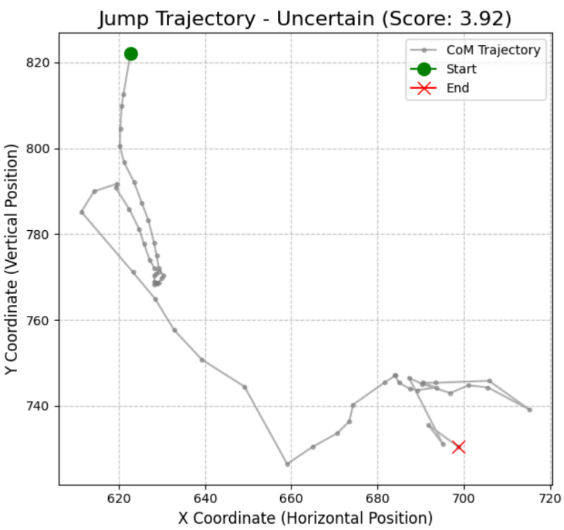
--- Classification Report ---

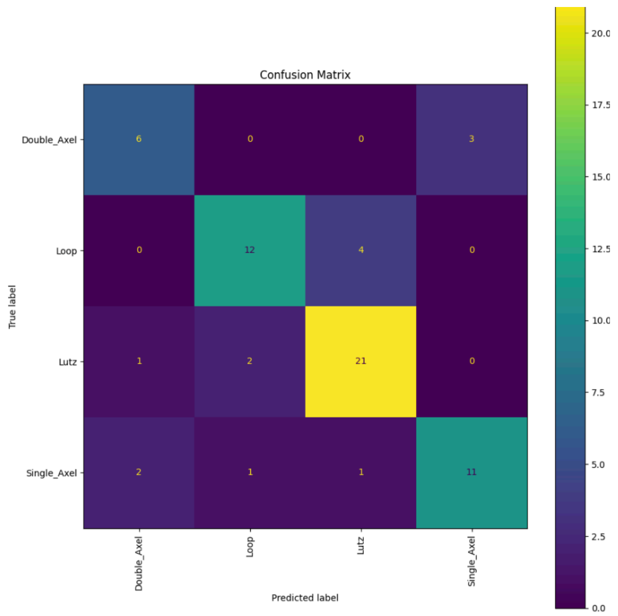
	precision	recall	f1-score	support
Double_Axel	0.67	0.67	0.67	9
Loop	0.80	0.75	0.77	16
Lutz	0.81	0.88	0.84	24
Single_Axel	0.79	0.73	0.76	15
accuracy			0.78	64
macro avg	0.77	0.76	0.76	64
weighted avg	0.78	0.78	0.78	64

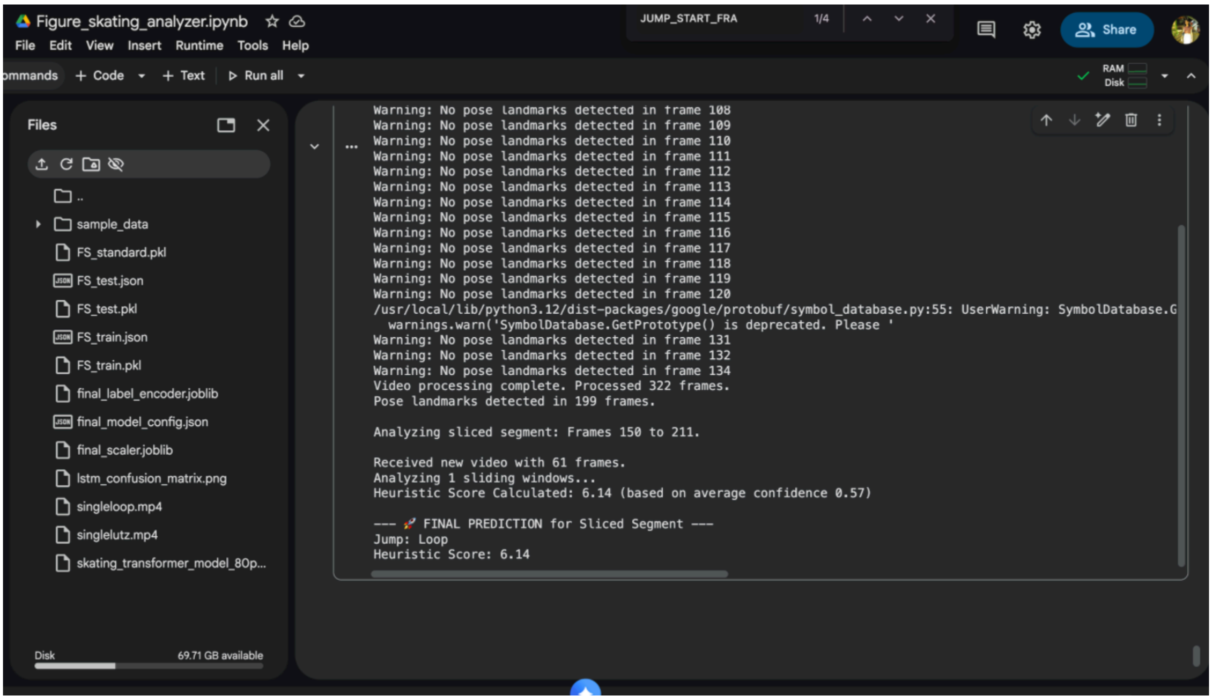
Results/Inference

Jump trajectories were used to understand how the model interprets our feature vectors (coordinates, velocity, acceleration and rotation) and compare test and training datasets









Works that needs to be done

Current Limitations

Incomplete Temporal Data:

- MediaPipe loses tracking during critical rotation phases when skaters move rapidly
- Current implementation skips frames with failed pose detection
- Creates gaps and discontinuities in temporal sequences
- LSTM receives fragmented data instead of continuous motion patterns
- Undermines the model's ability to learn coherent jump trajectories

Insufficient Feature Representation:

- System uses only raw x, y, z landmark coordinates from MediaPipe
- Captures spatial positions but misses biomechanical characteristics
- Missing critical features:
  - Joint angles (hip, knee, ankle flexion)

    - Angular velocities during rotation
    - Edge angles at takeoff
    - Velocity and acceleration profiles
    - Body orientation and rotation speed
- Model classifies based on static snapshots rather than dynamic motion signatures
- Cannot distinguish jumps that differ in execution mechanics rather than body position

Target Capabilities

- **Robust temporal tracking:** Continuous pose estimation throughout all jump phases (approach, takeoff, air, landing)
- **Rich biomechanical features:** Joint kinematics, rotational dynamics, edge mechanics
- **Precise fault detection:** Identify wrong edge, under-rotation, body positioning errors, landing faults
- **Actionable coaching feedback:** Translate detected faults into specific corrections and practice drills

Proposed Solutions

Addressing Missing Data:

- Implement linear interpolation for frames with failed pose detection
- Apply LSTM masking layers to handle variable-length sequences
- Explore alternative pose estimation models (OpenPose, MMPose)
- Use ensemble approaches for more robust tracking during rapid movements

Enhanced Feature Engineering:

- Calculate joint angles through geometric computation between connected landmarks
- Extract temporal derivatives using frame-to-frame differences (velocity, acceleration)
- Measure rotational velocity by tracking torso orientation changes

- Estimate edge angles from ankle-knee-hip alignment and body lean
- Compute body configuration metrics (limb extensions, core compression)
- Normalize spatial relationships accounting for individual skater anthropometrics
- Transform raw positional data into meaningful biomechanical features
- Enable model to learn underlying mechanics rather than memorizing appearance patterns

Immediate Next Steps

1. **Start collecting videos/ other open source datasets**
2. Fix the data leakage bug
3. Add joint angle and edge angle calculations
4. Design app mockup to visualise final experience