Computer Vision Analysis of Speed Climbing Human Pose Estimation

Manny Cassar
Queen's School of Computing
Kingston, Canada
m.cassar@queensu.ca

Kieran Green
Queen's School of Computing
Kingston, Canada
21kg38@queensu.ca

Mike Stefan
Queen's School of Computing
Kingston, Canada
21mgs11@queensu.com

I. Introduction

The sport of speed climbing made its debut on the world stage in the 2020 Tokyo Olympic Games due to its remarkable surge in popularity in recent years. This rise from a niche discipline to a mainstream competitive activity has underscored the need for sophisticated tools to analyze climbers' movements that will analyze and optimize training methodologies.

The goal of a speed climber is to navigate up a 15-meter vertical wall containing commonly placed climbing holds with precise and swift body positions in the shortest time possible. The primary challenge for coaches in speed climbing lies in separating and grading specific movements and sequences in a rapid and dynamic climbing run. This fast-paced format makes conventional pose estimation techniques less effective, necessitating a specially trained system capable of processing data swiftly to provide immediate feedback. This system's absence hinders the ability of climbers and coaches to enhance performance between runs in the same climbing session. Addressing this gap is critical for advancing training methodologies in speed climbing and can have broader implications for motion analysis in other high-velocity sports and activities.

This project aims to develop a deep neural network model tailored explicitly for speed climbing capable of detecting and overlaying skeletal coordinates onto live video feeds of climbers. The neural network will process individual video frames as input and output the coordinate positions of critical joints, including feet, hips, hands, elbows, and knees. For each frame of the climbing run, a complimentary Python script will then use these coordinates to produce a skeletal overlay, connecting the joints with lines to visualize the climber's real-time posture and movement. The video will then be reconstructed to produce a live skeleton overlay on the climber.

This project leverages high-quality competition data from the 2018-2020 Olympic speed climbing runs to train and validate the neural network. The project seeks to produce a model that can accurately depict pose estimation for a new speed climb run by harnessing advanced neural network architectures and processing techniques tailored to high-speed and intricate movements. This project has the potential to make significant contributions to the field of neural networks and computer vision by precisely tuning to quick, real-time motion analysis using speed climbing, which will enable more precise and immediate feedback mechanisms for climbers and may be adapted to other fast-paced sports. The successful implementation of this system could revolutionize how motion analysis is conducted in speed climbing, offering a scalable and non-invasive solution. The methodologies developed could also be adapted for other sports and activities involving rapid and complex movements, thereby broadening the impact within the broader neural networks and artificial intelligence fields.

II. Motivation

Speed climbing has rapidly evolved, with world record times improving dramatically over the past seven years. At this level of competition, every fractional adjustment in body movement can significantly impact overall performance. However, traditional coaching techniques rely heavily on manual video analysis, which is time-consuming and often subjective. The fast-paced nature of speed climbing renders conventional human pose estimation techniques impractical. Most existing models are tailored for slower, controlled movements, such as bouldering. These systems often require intrusive setups, such as multiple cameras, motion sensors, or physical markers, limiting their applicability for real-time analysis in speed-climbing events. The motivation for this project stems from the need to bridge this gap by developing an accurate quick real time model of a climber's body during their speed climb. We plan to build a model that will capture the skeletal coordinates of the climber and overlay them in a realtime video using a deep neural network. This will allow coaches and climbers to more quickly spot mistakes made during a climb without the need to wait in between climbing sections. It is our goal to develop this revolutionary tool using a deep convolution neural network model which will allow our model to be greatly optimised allowing for real time viewing and application.

III. Contributions

This report represents the collaborative efforts of the group, with each member contributing specific expertise to various aspects of the project:

- Manny Cassar: Developed the data ingestion and preprocessing pipeline, ensuring the dataset was formatted correctly for training and analysis.
- Kieran Green: Implemented the model architecture using Lite-HRNet, focusing on optimization for real-time performance.
- Mike Stefan: Created the video collection and seperation pipeline. Configured the project setup and its dependancies to allow for development. Conducted benchmarking and results analysis, comparing the system's performance against state-of-the-art models and identifying areas for improvement.

Each contribution played a crucial role in achieving the project's objectives, demonstrating the importance of interdisciplinary collaboration in solving complex engineering problems.

IV. Problem Description

Real-time pose estimation for speed climbing presents unique challenges due to the sport's rapid pace and dynamic movements. Existing systems, such as those designed for bouldering, fail to meet the demands of speed climbing for several reasons:

- Speed Constraints: Models optimized for slower activities cannot process video frames quickly enough for real-time feedback.
- Setup Complexity: Many current systems rely on specialized hardware, such as motion capture sensors or multi-camera setups, which could be more practical for field applications.
- Generalization: Standard datasets used for pose estimation, such as COCO, do not account for climbing-specific postures, limiting the accuracy of pre-trained models when applied to speed climbing.The proposed system addresses these challenges by developing a convolutional neural network capable of accurately detecting climbers' joint positions in real time. The project aims to deliver a scalable solution suitable for both professional and amateur use by optimizing for speed and accuracy while minimizing hardware requirements.

V. Related Work

Overlaying skeletons onto human bodies is a widely studied field with a blind spot for algorithms specialized for climbing. One study looks at climbers' body position and motions to spot errors made by the climber while they are bouldering.[1] This is done by mapping a skeleton over the climber, mapping the joints, and then following the motions to determine when the climber makes mistakes. Well, this model takes too long to be used for speed climbing; the training information used in this model will significantly improve our accuracy in recognizing climbing-specific positioning.

In their research, Pieprzycki et al. [2] investigated methods to analyze speed climbers' runs through video recordings. They developed a system that captures spatial and temporal parameters of climbers' movements without requiring intrusive sensors utilizing high-frame-rate cameras and visual markers placed near the climber's center of mass for effective tracking. Their approach employed algorithms such as the Kanade-Lucas-Tomasi (KLT) tracker and the OpenPose convolutional neural network for keypoint detection. This methodology allowed for the extraction of various kinematic parameters, including velocity, acceleration, and movement trajectories, providing valuable insights into climbers' performance. While their work showcased the potential of video analysis in evaluating climbers, its dependence on physical markers and post-processing limits its practicality for real-time applications. There is a clear need for a noninvasive, efficient system capable of real-time pose estimation that can handle the rapid and complex movements characteristic of speed climbing. Our project seeks to address this gap by developing a deep neural network model tailored explicitly for speed climbing. This model aims to enable real-time skeletal overlays on live video feeds without requiring markers, thereby enhancing the applicability and scalability of pose estimation in the sport. Our model will build upon the foundation established by Pieprzycki et al., moving towards a more practical and immediate analysis tool for athletes and coaches alike.

Our application requires our model to be extremely lightweight and able to analyze high-resolution video on standard hardware. Two common lightweight human pose estimation approaches are shuffle blocks [3] and HRNet [4]. Shuffle blocks improve the algorithm's performance by separating the convolutions into a linear combination of depthwise convolutions and 2 other convolutions, drastically reducing the compute time since these convolutions are more computationally efficient than the standard convolutional step. The HRNet architecture starts with high-resolution convolutions and adds high-to-low-resolution streams connected in parallel with their output, eventually being fused. With both of these approaches having drawbacks, Lite-HRNet [5] proposes a novel combination of these two algorithms, which replaces the costly high-resolution convolutions found in the HRNet architecture with the split stream approach derived in shuffle blocks. This approach also provides novel optimizations to the shuffle block algorithm that reduces the number of 1X1 convolutions, an extremely costly operation on video feeds.

VI. Dataset used

The dataset utilized for this project is derived from high-quality competition recordings spanning multiple World Cup events between 2018 and 2020. It is designed to address training challenges and evaluate a pose estimation model tailored for speed climbing. The dataset consists of two primary components:

Video Data:

- Each dataset entry includes a YouTube URL with corresponding timestamps marking the start and end of a climber's run.
- Videos are segmented into frames, capturing climbers' movements in high resolution.

Joint Coordinate Data:

- Each frame includes XY coordinates for 16 critical joints (e.g., feet, hips, hands, elbows, knees)
- Metadata files provide additional context, such as run identifiers, timestamps, and wall orientation.

VII. Challenges in Dataset Preparation

Data Quality:

- Variations in lighting, camera angles, and background complexity introduce noise, complicating joint detection.

Scalability:

- Converting large video datasets into structured, COCO-compliant annotations required efficient preprocessing pipelines.

Specialization:

- Generic pose estimation datasets, such as COCO, lack climbing-specific postures, making domain-specific data augmentation necessary.

VIII. Preprocessing Pipeline

Preprocessing Pipeline A robust preprocessing pipeline was developed to transform raw video data into training-ready formats. The key steps include:

- Video collection
- Clip extraction
- Frame seperation
- Parsing skeleton data files to derive joint coordinates.
- Generating JSON annotations compatible with the MM-Pose training framework.

The video collection was done manually via downloading the competition streams off YouTube. The intial plan for this was to have the pipline download them, but due to changes to YouTube's video cyphering system, the existing Python tooling to do this was rendered non-functional. To still collect the videos, we download them using https://y2meta.tube/. This wasnt too bad as the dataset only contained 17 videos and could trivailly be mass exported.

The next step is to extract the clippeds as specified in the metadata file. For this we iterated over the data set, finding the start and stop time for each video segment and using FFMPEG [6], the desired clip was extracted. These clips were then split by frame into individual frames, to be associated with the pose data. The pose data found in the data set was converted to math the COCO [7] format. This format was selected because it is an industry standard within human pose estimation. As this standard, all algorithms we researched already have implementations for training on COCO datasets. By conforming our data we make training on our dataset easier.

Below is a snapshot of a preprocessed dataset entry in COCO format:

```
{

"images": [
  {"id": 1, "file_name": "run1_frame1.jpg", "height":
1080, "width": 1920}
],

"annotations": [
  {
    "id": 1,
    "image_id": 1,
    "category_id": 1,
    "keypoints": [320, 480, 2, 400, 520, 2, ...],
    "num_keypoints": 16
  }
],

"categories": [
  {"id": 1, "name": "person", "keypoints": ["nose",
"left_eye", ...]}
]

}
```

Frames:

- Keyframes showing climbers' body positions.

Annotations:

- Overlaid skeletal coordinates highlighting joint positions.

Metadata:

- Descriptive information, including timestamps and wall orientation.

By structuring the dataset, we ensured compatibility with modern machine learning frameworks and enabled seamless integration into the model training process.

IX. Data Preprocessing

The preprocessing pipeline is a cornerstone of this project, enabling the efficient transformation of raw climbing videos into COCO-compliant datasets for model training. The primary tasks included:

- Developing scripts to parse video metadata and skeleton data files.
- Standardizing annotations to ensure consistency across frames.
- Visualizing sample outputs to verify data integrity.

Dataflow Diagram The following diagram summarizes the data preprocessing steps: Raw Videos –> Metadata Extraction –> Skeleton Parsing –> COCO JSON

Challenges and Solutions Challenge:

- Handling large datasets with diverse formats.

Solution:

- Parallelized preprocessing scripts to reduce runtime.

Challenge:

- Noise in joint coordinates from low-quality frames.

Solution:

- Implemented smoothing techniques to interpolate missing or noisy data points.

Outcome

The preprocessing pipeline successfully prepared over 5,000 annotated frames, providing a robust foundation for training the Lite-HRNet model.

X. Model Architecture and Training

The core of the pose estimation system is the Lite-HRNet model, selected for its balance between computational efficiency and high accuracy. Lite-HRNet integrates the lightweight shuffle block architecture with high-resolution networks, making it suitable for real-time applications. The architecture consists of:

- High-to-Low Resolution Streams: Parallel processing of image features at multiple resolutions.
- Stream Fusion: Combining outputs from different resolutions to retain detail while improving efficiency.
- Shuffle Blocks: Reducing computational overhead through efficient depthwise separable convolutions.

A simplified architecture diagram is presented below:

Input Image –> High-Resolution Stream –> Stream Fusion –> Output Keypoints

We are using LiteHRNet as it is implemented in the MMPose [8] project. This project comes with an open-sourced implementation of the algorithm with tooling to test and train the model. This algorithm is implemented in a top down fashion. This means that first an algorithm is run to detect where to humans are in frame, hence forth called the detection layer, then LiteHRNet is run within the bounding boxes to generate our human pose estimation. For our detection layer we selected RT-MDet [9]. This algorithm is an extremely efficient and accurate general object detetion and classifier. It showed to be accurate enough for our application, but further research into this could be warrented. Either further research into lighter models, or training this one on our dataset could pose to be beneficial.

Training Setup

Hardware Configuration:

- GPU: NVIDIA GTX 4070
- CPU: I7-137000H
- Memory: 64 GB RAM

Software Tools:

- Framework: PyTorch with CUDA 11.7
- Libraries: MMPose, MMEngine
- OS: Windows 10
- Dataset: Preprocessed COCO-style annotations derived from climbing video data.

XI. Challenges in Training:

There are significant dependancy issues within MMPose, which made all of their built in training tooling usless. With the way the project is currently structured and implemented, re-training an existing algorithm is non-functional. To get around these issues we attempted to implement the training algorithm ourselves. Through the process of implementing this we found a significantLY larger issue. This issue can be illustrated perfectly in Figure 1.

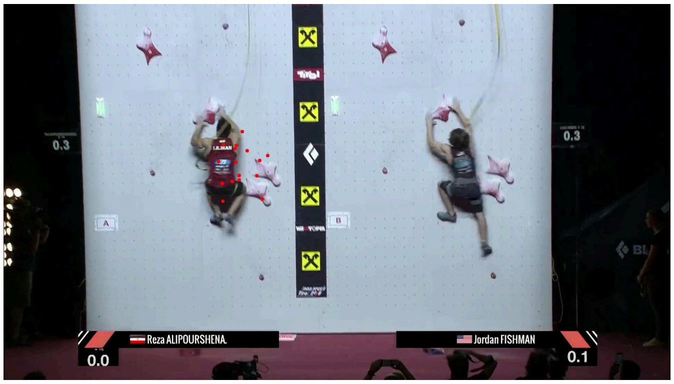

Figure 1: Overlay of the dataset points over the frame

Figure 1 shows the joint data from our dataset overlayed over the frame specified in the dataset. This clearly shows that the pose data contained within our dataset is incorrect. This could be rooted in two different places. Either the dataset is incorrect or our method of frame spliting and then aligning it to the dataset frame number is incorrect. In the process of finding the root of this issue, we compared similar tests to that from from a wide range of videos, as found that all of them had pose data that did not align to the video, not just the frame did not match, but all showed positions that the climber was never in. This leads us to the results that our dataset was fundementally flawed.

Due to the timeline of this project, and when we discouvered this issue we were unable to transition to a different dataset. Since all of our code was based around this dataset, a pivot was virtually impossible, ignoring the fact that a different high quality opensourced dataset is not readily available.

Since the dataset was unable to provide any useful data to compare the output of a pretrained model, or to custom train LiteHRNet, we were unable to produce any quantifiable comparision or validation scores.

XII. Discussion

Our selection of LiteHRNet showed promising results. It was able to generate an output in real time, which visually looks like it could be an accurate representation of the climbers

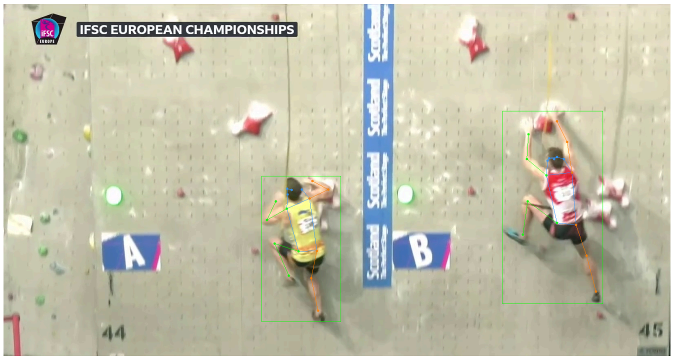

Figure 2: The visual output of LiteHRNet

As shown in Figure 2, this model is able to generate images which appear to work. Due to the afformentioned dataset issues, we lack quantifiable results to compare the models output, we can see some issues which further training would improve our performance. Most of the issues which can be observed qualitatively, including the detection model detecting some of the holds as people. That could be improved in a future iteration of our project by traing the detection algorithm on our dataset. Another issue we observed included estimation confusion for limbs obstructed by the climbers body.

XIII. Conclusion

This project was an attempt to develop a real-time pose estimation system tailored for speed climbing using the LiteHRNet architecture. The model should have balanced accuracy and speed, meeting the requirements for live video analysis while providing actionable feedback for athletes and coaches. We were unable to train the model given the dataset and internal model errors. The future work on this project would be to find or create a dataset which meets our requirements. Most of the challenges which we could not address were consequences of our dataset being incorrect. Once a new dataset is acquired, we can improve our accuracy by trainging both the detection and LiteHRNet models, which should provide better results. The other major gain we would recieve from a functional dataset would be accurate profiling and validation tools. Without a accurate dataset, we were unable to generate valid comparision to see paths of improvement.

References

[1] R. Beltrán, J. Richter, G. Köstermeyer, and U. Heinkel, “Climbing Technique Evaluation by Means of Skeleton Video Stream Analysis,” *Sensors*, vol. 23, p. 8216, 2023, doi: 10.3390/s23198216.

[2] A. Pieprzycki, T. Mazur, M. Krawczyk, D. Król, M. Witek, and R. Rokowski, “Computer-Aided Methods for Analysing Run of Speed Climbers,” *Preprints*, Feb. 2023, doi: 10.20944/preprints202302.0166.v1.

[3] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design,” in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham: Springer International Publishing, 2018, pp. 122–138.

[4] J. Wang *et al.*, “Deep High-Resolution Representation Learning for Visual Recognition,” 2020, [Online]. Available: https://arxiv.org/abs/1908.07919

[5] C. Yu *et al.*, “Lite-HRNet: A Lightweight High-Resolution Network.” [Online]. Available: https://arxiv.org/abs/2104.06403

[6] F. Developers, “FFmpeg.” [Online]. Available: https://ffmpeg.org/

[7] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context.” [Online]. Available: https://arxiv.org/abs/1405.0312

[8] M. Contributors, “OpenMMLab Pose Estimation Toolbox and Benchmark.” 2020.

[9] C. Lyu *et al.*, “RTMDet: An Empirical Study of Designing Real-Time Object Detectors.” [Online]. Available: https://arxiv.org/abs/2212.07784