Using Machine Learning for move sequence visualization and generation in climbing

Thomas Rimbot

EPFL-Semester Project
Supervisor: Martin Jaggi

Tutor: Luis Barba

July 6, 2022

Project Goals

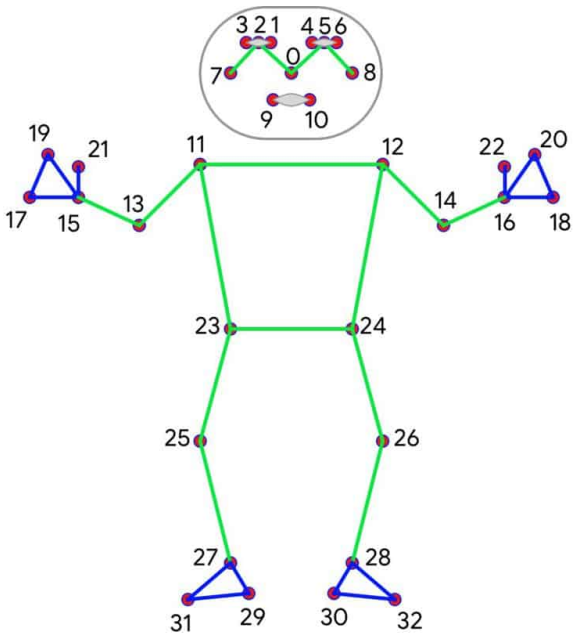
- Implementing a visualization pipeline for move sequence assessment;
- Developing a simple interface for moves and holds selection;
- Experimenting with text-based models for move sequence prediction.

Presentation outline

1. Move sequence detection recap from the previous projects;
2. Move sequence visualization;
3. Selection interface;
4. Move sequence prediction experiments.

Move Sequence Detection: Pose Estimation

- Extract landmarks from a video using `Mediapipe`.



0. nose
1. right eye inner
2. right eye
3. right eye outer
4. left eye inner
5. left eye
6. left eye outer
7. right ear
8. left ear
9. mouth right
10. mouth left
11. right shoulder
12. left shoulder
13. right elbow
14. left elbow
15. right wrist
16. left wrist
17. right pinky knuckle #1
18. left pinky knuckle #1
19. right index knuclke #1
20. left index knuckle #1
21. right thumb knuckle #2
22. left thumb knuckle #2
23. right hip
24. left hip
25. right knee
26. left knee
27. right ankle
28. left ankle
29. right heel
30. left heel
31. right foot index
32. left foot index

Figure: `Mediapipe`'s pose landmarks

Move Sequence Detection: Static Extremity Detection

- Combine landmarks information into single extremities;

- Consider fixed extremities by looking at difference between consecutive frames.

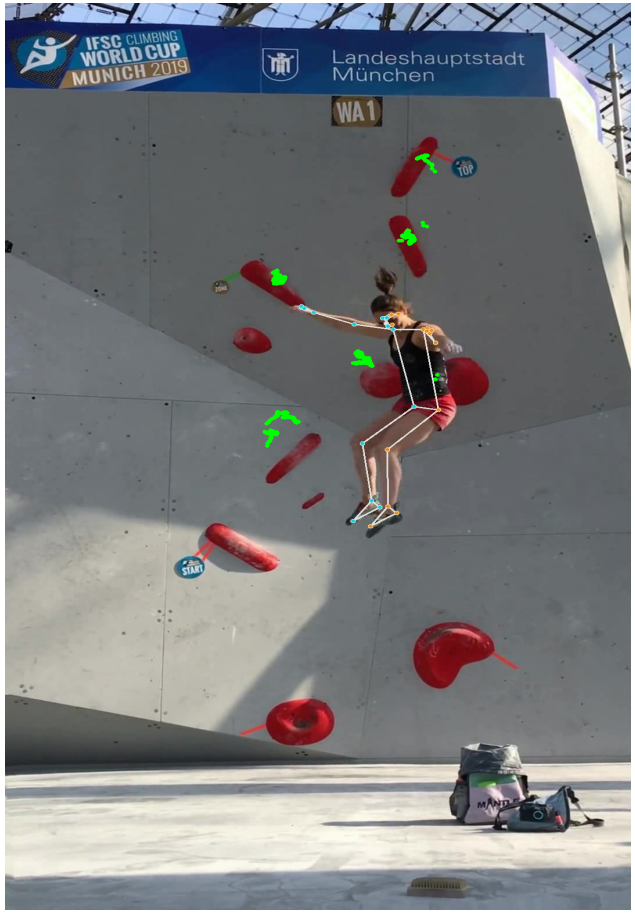


Figure: Point clouds obtained by static extremity detection

Move Sequence Detection: Clustering and Visualization

- Cluster the point clouds with DBSCAN to detect the holds;
- Use the frame information to extract the order;
- Display on a still image using `OpenCV`.

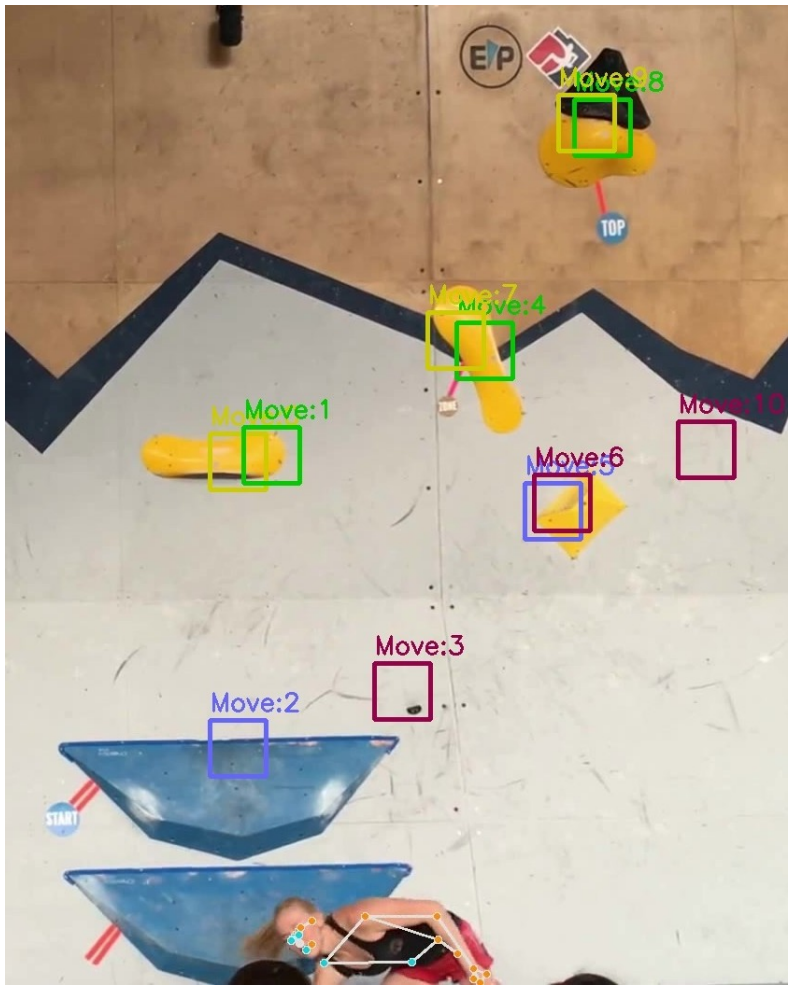


Figure: Move sequence detection example

Move Sequence Visualization: Skeleton Generation

Goal: generate landmarks information from move sequence. This is divided into three steps:

1. Landmarks coordinates for the extremities: interpolate between consecutive positions, weight by the distance and tune for the required time;

2. Landmarks coordinates for the rest of the body: use the `Linear Regression` model from `scikit-learn`;

3. Set the visibility to 1 for all landmarks.

Move Sequence Visualization: Skeleton Generation

We can now draw the generated skeleton with `Mediapipe`.

1. We do get a pretty good idea of how a person would move following this sequence;
2. But the interpolation keeps all extremities static but one, yielding limb stretching: problematic in modern dynamic-style bouldering;
3. Alternative approach: use a ML model to predict all the landmarks coordinates from the move sequence, instead of interpolating the extremities and inferring the rest.

Selection Interface

Using `OpenCV`, we created a User Interface to select a move sequence (or holds sequence) and export the data to be used directly by the model.

Demonstration

Move Sequence Prediction: Seq2seq Model

- Text translation using a Seq2seq model;
- Input: sequences of words of the form $x_N\_y_N$;
- Output: sequences of words of the form $limb\_x_N\_y_N$;
- Add permutations to make the model order-invariant and increase the dataset;
- Drawback: define discrete vocabulary by rounding up the coordinates to one decimal, and combining all possible combinations of tokens.

Move Sequence Prediction: Seq2seq Model

Sequence-to-sequence model for translation:

- *Encoder-Decoder* structure combined with an *Attention* mechanism;
- Predict one word at a time from the previous ones;
- Trained using *Teacher forcing*, evaluated with the *NLL* loss and the *Perplexity of Fixed-Length models* metric.

Move Sequence Prediction: Seq2seq Model

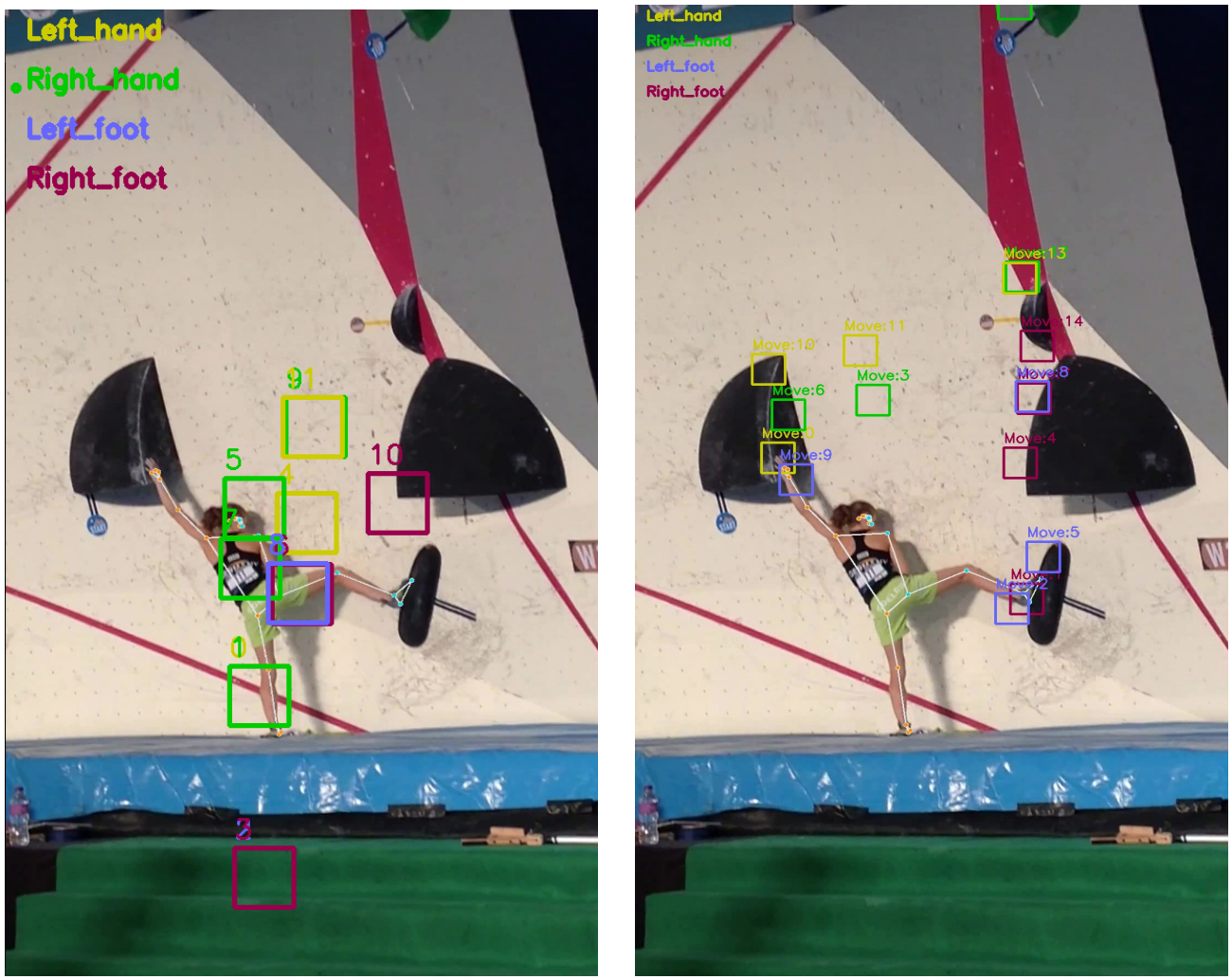


Figure: *Seq2seq* move sequence prediction (left) against the truth (right)

Move Sequence Prediction: Seq2seq Model

Unsatisfying results, which can come from multiple sources:

- Imprecise method for holds sequence creation by clustering;
- Dataset hard to work with, exhibiting a lot of variability and unreliability;
- Huge loss of precision due to the discrete vocabulary construction.

However, it is worth noting that using the visualization pipeline on the generated move sequence, the skeleton's movements closely resemble the true ones.

Move Sequence Prediction: Autoregressive Model with Positional Encoding

- Simplify the goal: now only work with holds and try to sort them in their order of use;
- Now work with a new dataset, consisting of 20 standardized Moonboard videos;
- Permute and pad the sequences;
- Data handled differently: this time the model requires the input and output data to be the same, but the output is shifted one token to the right.

Move Sequence Prediction: Autoregressive Model with Positional Encoding

The model is a *Sequence completion* model, using a *Transformer achitecture*:

- *Encoder-Decoder* structure;
- *Attention Mask* to hide 'future' information, and allow the model to only look at the 'past' at inference;
- *Positional Encoding* to add the coordinates information to the holds token.
- Trained with *Adam* and *Cross Entropy* loss, experimenting with different learning rates and embedding dimensions.

Move Sequence Prediction: Autoregressive Model with Positional Encoding

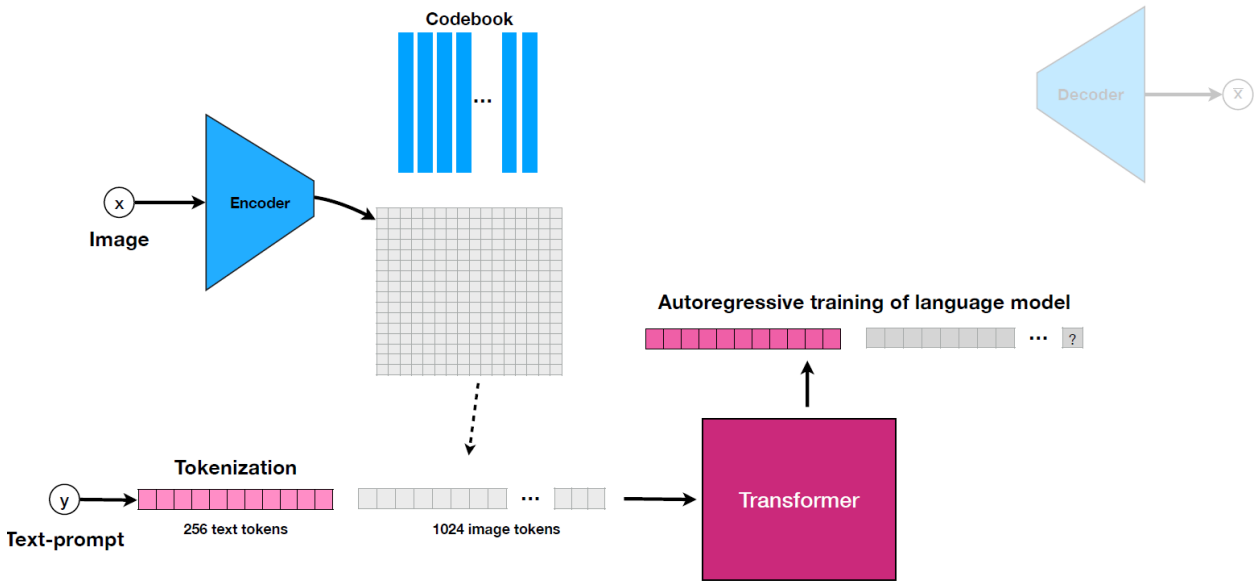


Figure: Autoregressive Transformer functioning

Unusable results: the model always predicts the padding element.

Move Sequence Prediction: Simplified Model with Direct Forward Pass

The final try was a simplified Transformer model:

- No Positional Encoding, no Autoregressive features;
- Directly tries to sort the holds from their coordinates;
- Trained with *Adam*, evaluated with the *Cross Entropy* loss and the *PPL* metric, saving the best model.

Move Sequence Prediction: Simplified Model with Direct Forward Pass

- Interesting results, better than before: the padding element is not repeated, but the model does output the holds token;
- Nonetheless, the results are pretty random: only 2 non-padding elements were correctly predicted here.

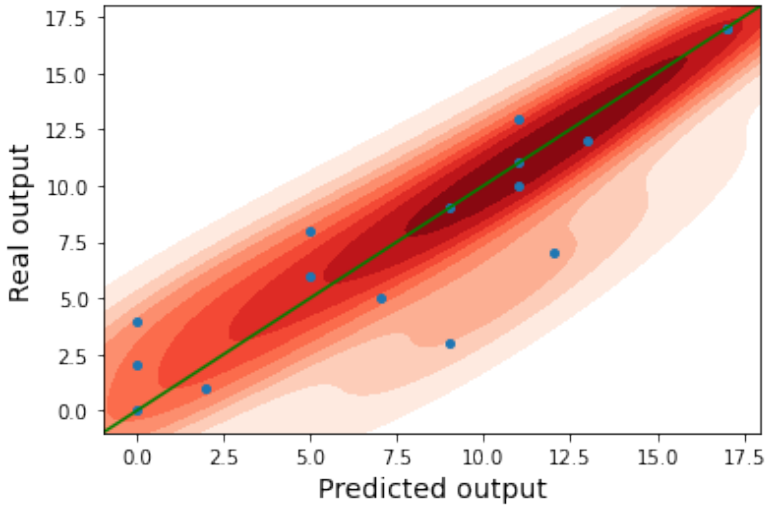


Figure: Evaluation of the model's performances on a test sequence of size 14, padded to a fixed length of 17. Accuracy $\approx 35\%$.

Conclusions

- Implemented a simple pipeline for move sequence visualization;
- Developed a User Interface for move and holds sequence selection allowing for personal experimentation;
- Started building a standardized dataset of moves/holds sequences videos;
- Experimented with different text-based models for move sequence prediction, giving a first insight into the application of ML techniques to bouldering's intellectual challenge;
- The results are still far from satisfactory, but will serve as groundwork for future projects.

Acknowledgements

I would like to thank Martin Jaggi for supervising this project and allowing me to experiment with the combination ML-climbing.

Thanks to Luis Barba for his huge help regarding Transformer models, and to the EPFL students who gave me access to their projects and data.

Main References

al., Camillo Lugaresi et. "Mediapipe: A framework for building perception pipelines.". In: ().

Beauville, Charles, Robin Debalme, and Théo Patron. "Move sequence detection on bouldering problems.". In: ().

Comeliau, Loïc, Julia Heiniger, and Weiran Wang. "Understanding bouldering using ML methods.". In: ().

DALL.E: Creating images from text.

Dobles, Alejandro, Juan Carlos Sarmiento, and Peter Satterthwaite. "Machine Learning Methods for Climbing Route Classification.". In: ().

Ester, Martin et al. "A density-based algorithm for discovering clusters in large spatial databases with noise.". In: ().

Moonboard benchmarks 40 degree masters 2017 beta videos. URL: https://youtube.com/playlist?list=PL2M6xN8TDexz8j-a94S3VX-HiCKhLBdly.

NLP from scratch: translation with a sequence to sequence network and attention. URL: https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html.

OpenCV: Open Source Computer Vision Library. URL: https://github.com/opencv/opencv.

Scikit-learn: Machine Learning in Python. URL: https://scikit-learn.org/stable/.

Transformer architecture: the Positional Embedding. URL: https://kazemnejad.com/blog/transformer_architecture_positional_encoding/.