World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:16, No:10, 2022

Machine Learning-Enabled Classification of Climbing Using Small Data

Nicholas Milburn, Yu Liang, Dalei Wu

***Abstract*—**Athlete performance scoring within the climbing domain presents interesting challenges as the sport does not have an objective way to assign skill. Assessing skill levels within any sport is valuable as it can be used to mark progress while training, and it can help an athlete choose appropriate climbs to attempt. Machine learning-based methods are popular for complex problems like this. The dataset available was composed of dynamic force data recorded during climbing; however, this dataset came with challenges such as data scarcity, imbalance, and it was temporally heterogeneous. Investigated solutions to these challenges include data augmentation, temporal normalization, conversion of time series to the spectral domain, and cross validation strategies. The investigated solutions to the classification problem included light weight machine classifiers KNN and SVM as well as the deep learning with CNN. The best performing model had an 80% accuracy. In conclusion, there seems to be enough information within climbing force data to accurately categorize climbers by skill.

***Keywords*—**classification, climbing, data imbalance, data scarcity, machine learning, time sequence.

I. Introduction

THE sport of climbing has grown substantially in the last few decades. Organized competitive climbing is bigger than ever with the introduction of climbing into the 2021 Olympics. More state of the art climbing gyms are opening around the world, which draws more people into the sport and creates a greater market for coaching, training, and other instruction. However, there is still not a strong understanding of the best way to train for climbing. There is not even a large pool of scientific understanding of climbing. This paper attempts to shed some light on the characterization of climbing athletes by classifying climbers (grading climbers) into skill categories using machine learning techniques.

Grading climbers is not a well defined concept. There have been a number of studies that attempt to characterizis the sport. Researchers have attempted to understand the anthropometrics of climbing by examining height, ape index, body fat, and other factors. It has been fairly well documented that elite level climbers tend to have lower body fat [1]–[3]. Ape index (the ratio of arm span to height) is more controversial. Some papers found that elite climbers tend to have a greater ape index, but others showed no significance. Hand anatomy have shown to be important as well. Specifically, having more bone-to-tip pulp in the finger is a feature of elite climbers [2].

Climbing dynamics is another interesting of area research. Instrumented climbing holds (holds mounted to a force transducer to record force) have provided much insight. Quaine et al. have performed multiple laboratory studies to analyze the forces involved in isolated climbing movements. In one they found that when a climber removes a leg from the wall, the change in force on each hold is actually done before the foot leaves contact from the hold. [4] Thus, the change in force is done in preparation for, rather than as a reaction to, the foot leaving the wall. In another study, they found that a climbing posture where the trunk is closer to the wall to be advantageous [5]. Franz et al. placed an instrumented climbing hold in a competition and found that elite climbers tend to have smaller contact forces, shorter contact time, smaller impulse, better smoothness factor, and a smaller Hausdorff dimension [6].

The works done by Dobles et al. [7] and by Phillips et al. [8] are similar to that of this paper with the key difference that they were classifying the climbs rather than the athletes. Dobles et al. used chaotic variations and machine learning to create a generator to help human route setters. Phillips et al. examined data from a standardized climbing wall called the MoonBoard in order to train a classifier that could assign difficulty ratings to each climb. They explored Naive Bayes, softmax regression, and CNN and were able to achieve a 35% top-1 accuracy with each model.

This type of classification problem is similar to those found in gesture/posture recognition. Typically, the aim of gesture recognition is to identify the position of the hand for some application area such as human-computer interaction [9]. There are a number of methodologies; computer vision and instrumented gloves are two examples [9]. Qi et al. investigated using surface EMG (sEMG) signals, linear discriminate analysis, and extreme learning machine in gesture recognition [10]. The premise of their study is that the electrical signals from the muscles in the arm that control the positioning of the hand should provide enough information to classify the positioning of the hand. They were able to get an accuracy of 79.32% using this technique.

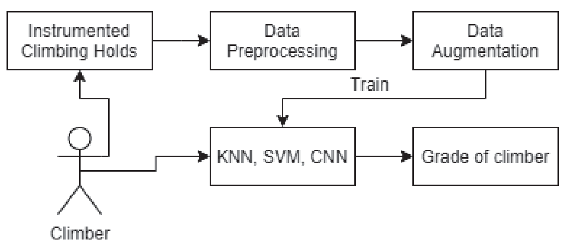

Fig. 1 Climber grading process block diagram

N. Milburn, Y. Liang, and D. Wu are with the Department of Computer Science and Engineering, University of Tennessee at Chattanooga, Chattanooga, TN, 37403 USA (e-mail: {lhs443@mocs, yu-liang@, and dalei-wu@} utc.edu).

The approach by Qi et al. for gesture recognition is similar to the techniques employed in this paper. We explore the viability of using force data taken from an instrumented climbing hold to grade climbers. Section II will elaborate upon the problem statement. Section IV will explain our methodology in data augmentation and machine learning techniques. Section V will discuss and compare the classification accuracy from data preprocessing and augmentation. Finally section VI will offer our conclusion and ideas for future work.

II. Problem Statement

This paper aims to use machine learning techniques to identify features within time series data of climbing reaction forces to correctly grade climbers. This problem is similar to other time series classification problems, such as audio classification, but with an important distinction: there is not an objective ground truth. As of yet, there is not an objective way to assess the performance level of a climber. Other problems to overcome in this study were heterogeneous time series data, data scarcity, and data imbalance.

A. Data Acquisition

A wall (Fig. 2) was set up during the 2018 USA Climbing Open ABS National Championship with a set boulder problem. Athletes competing at the National Championship climbed on the wall on a volunteer basis. The specific problem was perfectly mirrored around the instrumented hold. The instrumented hold was mounted to a 3-axis load cell from Interface Inc (model 3A120) rated to 2KN. The athletes were asked to climb the boulder twice leading with one hand on the first try and then the other on the second. This way data were recorded from the left side and right of the body. The decision of which hand lead first was left to the athlete. There were 37 subjects. The resulting data were multi-channel time series data representing the force on the climbing hold with three channels: one for each direct (X, Y, Z).

The forces were recorded from when the climber made contact with the hold to when the climber moved off of the hold. This time frame varied with each athlete and even within attempts. The attempts of two athletes are shown in Fig. 3.

Other data gathered include anthropometric measurements (weight, height, arm span) and climbing history (hardest grade sent at least three times, hardest grade sent once).

B. Series of Varying Lengths

Due to the nature of practical climbing, the kinetics data are featured with heterogeneous length and offset. This could not be accounted for during data acquisition while still collecting data in a realistic scenario, so it had to be accounted for in data preprocessing. In order to prepare the data to be used as input, the length of each sample had to be adjusted so they were of equal length. It was important that the dynamics of each sample be preserved during this process. Hogg et al. had success in this using interpolation [11], and another common technique is to use padding.

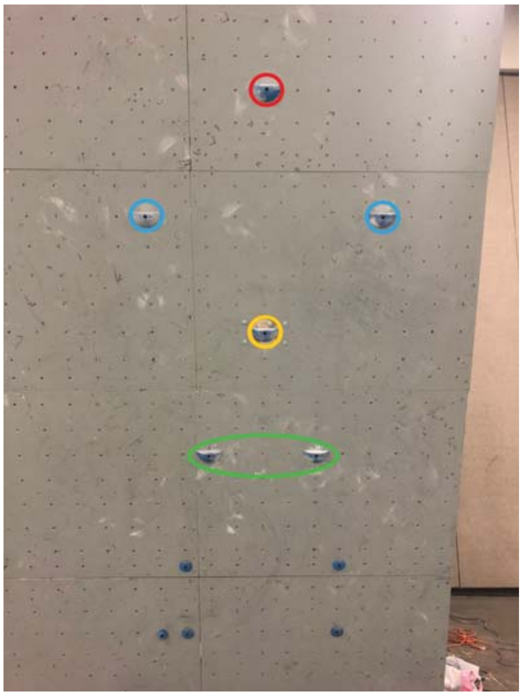

Fig. 2 Boulder: The green circle indicates the start holds, orange is the instrumented hold, blue are hand holds, and the red hold is the finish hold

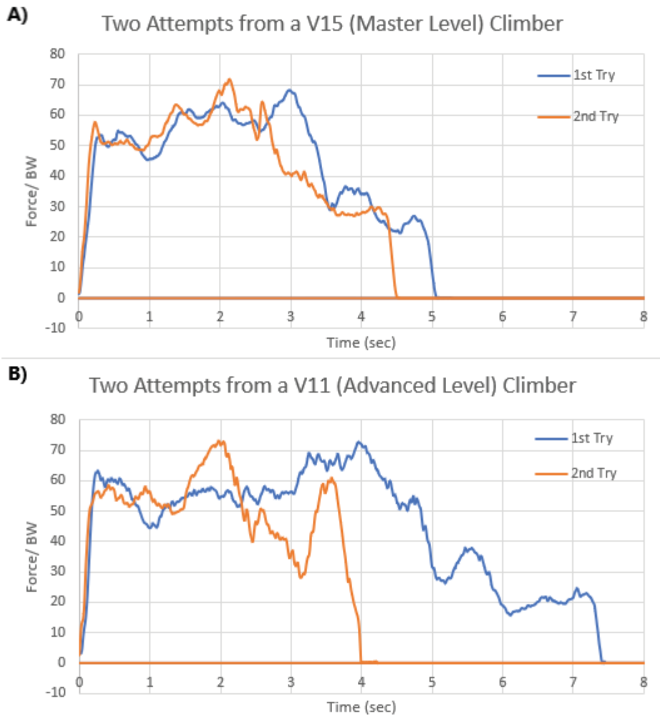

Fig. 3 (A) The first and second tries of a V15 climber; (B) the first and second tries of a V11 climber

C. Data Scarcity and Imbalance

Another significant problem with the dataset is how small it is. Ideally, a vast collection of force data would have been available, but due to the newness of this type of data acquisition, and the cost of the data collection equipment, there were only 37 participants. Creating an accurate model with such little data is more difficult, and over-fitting becomes an issue.

D. Grading

Climbers developed a way to represent the difficulty of a problem called grading. Every climb is assigned a grade, and the grades indicate the difficult of each climb. There are many scales, but in this paper, the V-grade will be used. The V-grade is an open ended alphanumeric continuum that starts at V0 and, at the time of publication, caps out at V17. The higher the v-grade number, the more difficult the climb. An important thing to understand about grading is its subjective nature. The personal difficulty of a problem can vary by as much as several grades between climbers despite its assigned V-grade due to climber specialization and anthropometrics. However, as more climbers complete a problem and give their opinion, it is possible to get a good average v-grade. While this subjectivity is fascinating, it also provides a unique challenge to research because there is not an objective way to quantify performance. Fortunately, self reported grades have shown to be a reliable metric [12].

It is simple enough to examine Fig. 3 to identify that there are differences between an elite level climber and a less skilled one. The most noticeable difference lies in the time variance between attempts. (A) is of a V15 climber and the two attempts show similar signals; whereas, (B) is of a V11 climber, and the signal from the first attempt is nearly twice as long as the signal from the second attempt. Another feature of the less skilled athlete is the second peak at about 3.5 seconds on signal two. (A) has no such second peak.

III. Data Preprocessing

A. Temporal Normalization

Two techniques were explored for temporal normalization: padding with zeros and interpolation. Padding is the simpler of the two techniques. It involves adding zeros to the end of all the smaller samples until they were of equal length to the longest sample. Interpolation involved resampling the data to 101 data points and to 1001 data points using Fast Fourier Transform (FFT). FFT is an algorithm that computes the Discrete Fourier Transform (DFT) much faster. The formula for DFT is displayed in (1). FFT works by first transforming the signal into the frequency domain. A window is used to taper the spectrum and prevent smearing. Once the signal is in the frequency domain, it will either be upsampled or downsampled. If it is upsampled, $N/2$ zeros will be added at the end, and if it is downsampled, the second and third groups of $N/4$ elements will be removed. Finally, the signal is transformed back into the time domain using inverse DFT (IDFT) shown in (refeq:IDFT). These two techniques resulted in three data sets that could then be compared against each other.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}, \quad k = 0, \dots, N-1. \tag{1}$$

$$X_n = 1/N \sum_{k=0}^{N-1} x_k e^{i2\pi kn/N}, \quad n = 0, \dots, N-1. \tag{2}$$

B. Dataset Augmentation

The data scarcity problem and imbalance were solved using data augmentation. Three different augmentation techniques created twenty different datasets to compare against each other. Different parameters for each augmentation technique were established to create five data sets per technique. In this way, the best utilization of each augmentation function could be identified. Then five datasets containing a combination of these techniques were made. In addition to increasing the total number of samples through data augmentation, our data set was balanced. More augmented samples were added to the classes with fewer samples so that each class ended up having an equal number of samples. The number of samples was increased from 74 to 1000.

An augmentation pipeline was created to input the original data and output the new augmented data. The augmentation pipeline could loop through the original dataset until a predefined number of total samples was reached. This pipeline used three different augmenters taken from tsaug, a python library for time series augmentation, in different combinations, and with adjustments to their parameters, to create a total of twenty new datasets. The augmenters used were noise, drift, and timewarp. This technique increased the sample size from 146 to 1000.

1) Augmentation through adding noise: Equation (3) is a mathematical description of how noise was added, and Table I describes the parameters used to create the datasets. Noise was added to each point with a magnitude between the scaled min and scale max sampled at random. Noise was added to each channel at the probability described in Table I. The range of the level of noise added and the probability with which it was added should help to reduce overfitting.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{3}$$

TABLE I
Parameters Used in Noise Augmentation

Dataset	Scale Min	Scale Max	Probability
1	0.01	0.05	0.5
2	0.05	0.1	0.5
3	0.1	0.15	0.5
4	0.01	0.05	0.25
5	0.01	0.05	0.75

2) Augmentation through signal drifting: The drifting algorithm is described in Algorithm 1. It works by randomly and smoothly drifting X, the original signal, n times with a minimum and maximum drift scale $[l, h]$. Additionally the probability that a signal is drifted can be customized for increased variability. The parameters chosen for this paper are displayed in Table II.

Algorithm 1 Drifting algorithm

Input:
S original signal
$[l, h]$ drift range
n number of drift points
Output:
S_a drifted signal
Procedure:
$y \leftarrow$ random values from normal distribution
$y \leftarrow$ cumulative sum of y
Fit a cubic spline interpolator, $CS()$, to y
$drift \leftarrow$ interpolated series from $CS()$
subtract $drift_0$ from drift
Divide $drift$ by $drift.max(|drift[i]|)$
Multiply $drift$ by random value between $[l, h]$
Normalize $drift$
$S_a \leftarrow S + drift$
Return S_a

TABLE II
PARAMETERS USED IN DRIFT AUGMENTATION

Dataset	Scale Min	Scale Max	Probability
6	0.1	0.5	0.5
7	0.5	0.99	0.5
8	0.1	0.99	0.5
9	0.5	0.99	0.25
10	0.5	0.99	0.75

3) Augmentation through time warping: The time warping algorithm is described in Algorithm 2. It works by randomly changing the speed of X, the original series, n times with a maximum speed ratio of r. As the signal is stretched, or shrunk, the missing values are filled in using interpolation. The result is a smoothly warped series. The parameters chosen for this paper are displayed in Table III.

TABLE III
PARAMETERS USED IN TIMEWARP AUGMENTATION

Dataset	N Speed Change	Max Speed Ratio
11	1	3
12	3	3
13	5	3
14	3	5
15	3	8

4) Multiple Augmentation: The final datasets were created with a combination of all three augmenters. Table IV describes the parameters chosen for each augmenter. For example, dataset 16 used the same parameters for timewarp as dataset 11, the parameters for noise as dataset 1, and the parameters

Algorithm 2 Time warping algorithm

Input:
S original signal
n number of speed changes
r max speed ratio
Output:
S_a drifted signal
Procedure:
$y \leftarrow$ random values from normal distribution
max, $min \leftarrow y.max$, $y.min$
for all i in y **do**
$$i = i - \frac{(max - r * min)}{(1 - r)}$$
end for
$sum \leftarrow y.sum$
$y \leftarrow$ cumulative sum of y
for all i in y **do**
$$i = \frac{i}{sum} * (len(S) - 1)$$
end for
Prepend y with a 0
Fit a PCHIP interpolator, $PCHIP()$, to y
$warp \leftarrow$ interpolated series from $PCHIP()$
Fit a 1-D interpolator, $interp()$, to S
$S_a \leftarrow interp(warp)$
Return S_a

for drift as dataset 6. The parameters for each augmenter were chosen from the datasets whose models performed the best in terms of predictive accuracy. The augmenters were applied to the data in the following order: timewarp, noise, then drift.

TABLE IV
COMBINED AUGMENTATION STRATEGIES

Dataset	Timewarp	Noise	Drift
16	11	1	6
17	12	1	9
18	12	1	6
19	11	5	6
20	12	5	9

C. Adjusting Data Input Shape

The datasets were inputted into the machine learning algorithms using three different strategies as shown in Fig. 4. The first (sequential) was to input each sample independently. This way there were 1000 samples with three channels each. The second (parallel) was to pass both tries into the classifier in parallel so as to relate the two attempts with the same climber. This way there were only 500 samples but with six channels each. The third (parallel+) maintained the relationship between the two attempts as in the paired data, but also added another three channels representing the difference between the first and second. This highlighted the differences in time between attempts within the input data. This way there were 500 samples with nine channels each.

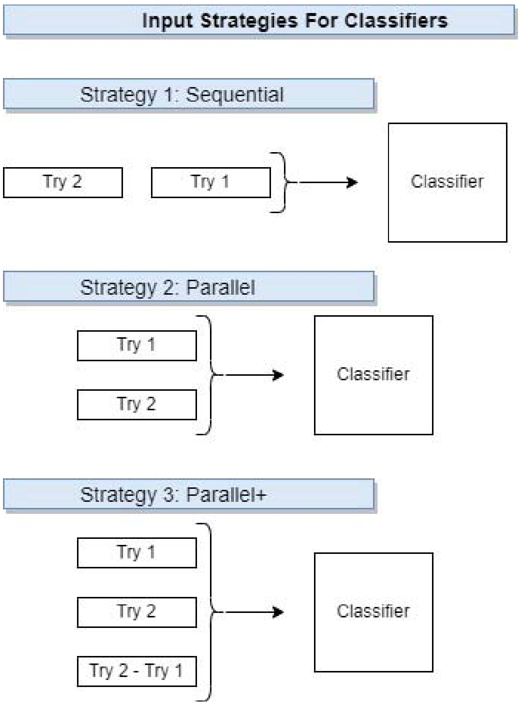

Fig. 4 Three input strategies used for classifiers

IV. Machine Learning-enabled Scoring of Climbing

Because the dataset was small, choosing utilizing classifiers with a smaller number of parameters was ideal. Thus lightweight KNN and SVM classification algorithms were selected to solve this problem as they would be less affected by the scarcity of the dataset. In addition, a CNN classification algorithm was chosen as they are commonly used for time series classification problems.

A. KNN Classifier

The input dataset was built by flattening each sample. The samples were divided into four classes where a single class represents a range of two V-grades. Dividing classes this way helps to overcome the subjectivity of grading, especially when working with a small dataset, by creating a buffer. The dataset was then split into training and testing datasets using a 75/25 split. A grid search technique used to select the KNN's parameters. The grid search values for k were 3, 5, 11, or 19; the grid search values for the weight functions were uniform or distance; and the grid search values for the distance metrics were Euclidean or Manhattan. The best performing parameters are discussed in Section V.

B. SVM Classifier

The input dataset for SVM was the same as for KNN. The samples were flattened, divided into four classes, and split into training and testing datasets using a 75/25 split. There were two groupings of grid search parameters distinguished by the kernel. If the kernel was rbf, then the values for c were either 1, 10, 100, or 1000, and gamma was set to 1e-3 or 1e-4. If the kernel was set to linear, then the values for c were 1, 10, 100, or 1000. The best performing parameters are discussed in Section V.

C. Deep Learning: CNN

A VGG style network was selected [13] (architecture displayed in Fig. 6) as it is well understood and is commonly used for image classification problems. CNNs perform well with image classification problems but are also used for audio classification problems. Audio data are a type of time series data and can easily be converted into the spectral domain via spectrograms. Spectrograms are visual representations of spectra, and time series data are a type of spectrum. Since the force data presented in this paper are time series data, it can also be visually represented as a spectrogram. Multi-channel time series data may also be represented by stacking multiple spectrograms, one per channel, on top of each other. These are called layered spectrograms. Layered spectrograms maintain the relationship between all the channels within a sample. The workflow to prepare the data for input into the CNN is shown in Fig. 5.

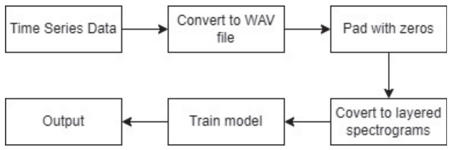

Fig. 5 CNN work flow

V. Results

In all there were sixty-one different datasets. One was the original, unaugmented dataset, and the rest were derived from augmentation and input strategy. There were twenty augmented datasets that were then each formatted in three different ways as presented in Fig. 4. Each model, the KNN, SVM, and CNN, was trained using every data set: the original, the twenty sequential, the twenty parallel, and the twenty parallel+ datasets. The accuracy from each model trained on the original dataset are presented in Fig. 7.

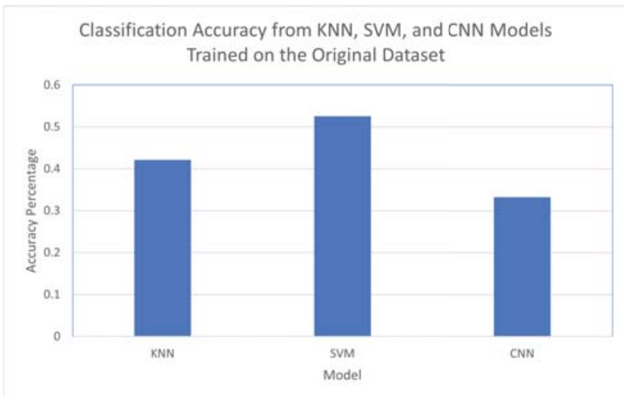

Fig. 7 Classification accuracy for original dataset

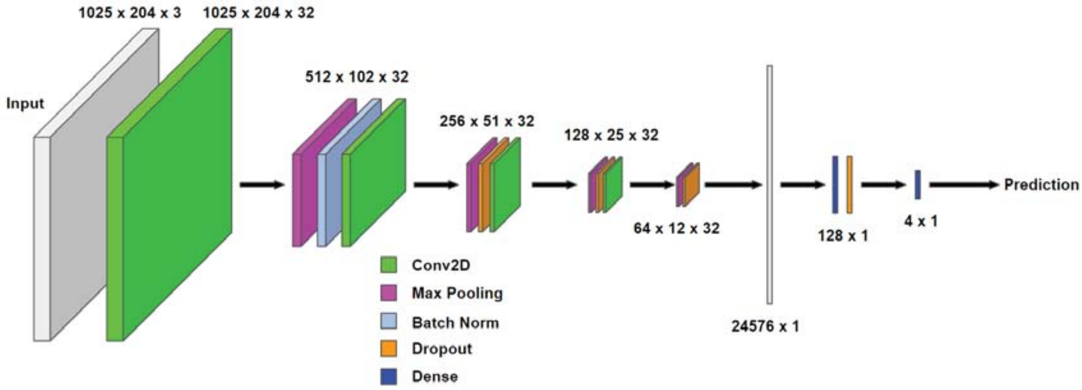

Fig. 6 CNN architecture

A. KNN

Overall, data augmentation greatly increased the classification accuracy of KNN and SVM as depicted in Table V.

With regards to KNN, the best performing model from the sequential input strategy was 77.6% accurate which is an 84% increase from the model trained on the original dataset. The best performing model from the parallel input strategy was 79.2% accurate which is an 88% increase from the original model and a 2% increase from the best sequential model. The best performing parallel+ model was 80.8% accurate which is an 92% increase from the original model, a 2% increase from the best parallel model, and a 4% increase from the best sequential model.

For the best performing parameters for the KNN, the distance weight function and the euclidean distance metric were unanimously the best. A k-value of 11 worked best for most of the higher performing models; however, the model that boasted an accuracy of 80.8% used a k-value of 5.

TABLE V
CLASSIFICATION ACCURACY OF KNN AND SVM

Dataset	KNN Seq	KNN Parallel	KNN Parallel+	SVM Seq	SVM Parallel	SVM Parallel+
1	0.752	0.752	0.768	0.752	0.768	0.784
2	0.744	0.760	0.760	0.728	0.768	0.776
3	0.704	0.712	0.704	0.728	0.760	0.776
4	0.716	0.720	0.720	0.752	0.800	0.800
5	0.756	0.744	0.744	0.748	0.736	0.776
6	0.636	0.688	0.736	0.676	0.728	0.760
7	0.528	0.672	0.632	0.608	0.672	0.664
8	0.564	0.664	0.648	0.704	0.648	0.648
9	0.632	0.696	0.680	0.664	0.688	0.616
10	0.412	0.497	0.528	0.636	0.616	0.632
11	0.768	0.768	0.768	0.772	0.800	0.800
12	0.776	0.776	0.808	0.768	0.800	0.800
13	0.768	0.768	0.760	0.768	0.800	0.800
14	0.772	0.792	0.792	0.760	0.792	0.792
15	0.772	0.792	0.784	0.760	0.792	0.792
16	0.612	0.744	0.728	0.660	0.736	0.680
17	0.620	0.736	0.704	0.668	0.696	0.680
18	0.620	0.728	0.688	0.660	0.728	0.744
19	0.572	0.712	0.704	0.692	0.688	0.704
20	0.612	0.640	0.656	0.632	0.680	0.680

B. SVM

The SVM models showed a very similar pattern to that of the KNN models. The best sequential model was 77.2% accurate, the best parallel model was 80.0% accurate, and the best parallel+ model was also 80.0% accurate. However, a greater number of the SVM models boasted an accuracy of greater than 79%. It seems the specific augmentation strategy is less important when working with SVM.

All models performed best with the rbf kernel. The best performing sequential model had a c-value of 10 and gamma equal to 1e-3. The parallel and parallel+ input strategies both had four models with an accuracy of 80%. Of these eight models, six had a c-value of 1 and gamma equal to 1e-3, one had a c-value of 10 and gamma equal to 1e-4, and the final model had a c-value of 1 and gamma equal to 1e-4.

C. CNN

The CNN models also showed improvement from augmentation as shown in Table VI, but the overall classification ability of these models was substantially less than KNN and SVM. The best performing sequential model was 47.5% accurate which is a 43% increase from the model trained on the original dataset. The best performing model from the parallel input strategy was 59.0% accurate which is a 77% increase from the original model and a 24% increase from the best sequential model. The best performing parallel+ model was 53.0% accurate which is a 59% increase from the original model and a 12% increase from the best sequential model, but it was a 10% reduction from the best parallel model. It is likely the CNN was less performant as the large number of parameters did not handle the small dataset well.

VI. CONCLUSION AND FUTURE WORK

To the best of our knowledge, there has not yet been a study utilizing machine learning techniques to grade climbers, but it does seem to be a viable area of research. Grading climbers is a complicated problem due to the large number of variables and the idea that a climber is greater than the sum of his or her parts, but machine learning algorithms are well suited to dealing with large feature sets.

TABLE VI
CLASSIFICATION ACCURACY OF CNN

Dataset	Seq	Parallel	Parallel+
1	0.333	0.424	0.340
2	0.380	0.360	0.480
3	0.415	0.320	0.380
4	0.340	0.270	0.480
5	0.355	0.390	0.270
6	0.475	0.400	0.480
7	0.340	0.320	0.260
8	0.255	0.300	0.400
9	0.310	0.290	0.410
10	0.240	0.360	0.270
11	0.465	0.590	0.390
12	0.415	0.480	0.380
13	0.395	0.430	0.520
14	0.420	0.350	0.470
15	0.325	0.380	0.530
16	0.300	0.360	0.350
17	0.345	0.450	0.290
18	0.345	0.350	0.410
19	0.305	0.300	0.210
20	0.305	0.340	0.280

This area of research would benefit significantly by increased availability of and access to larger datasets. It may be the case that the models will not need to rely so heavily on augmentation if the datasets are large enough, but if that is not the case, the augmentation strategies are still available. More data will likely become available as the sport of climbing grows and it gains more attention from the research community. Another factor that could greatly enhance the performance of athlete grading would be the increased implementation of more instrumented climbing holds. Our data were collected using only one instrumented hold to look at only one limb of a climber, but having at least five instrumented holds (one per limb plus one additional move) would supply much more data and give a more all encompassing view of the athlete.

Out future work will focus on the application of transformer [14] to extract the long-term time sequence pattern, which will be used in accurate climbing classification and grading. Stuff about chaos [15].

ACKNOWLEDGMENTS

This work was jointly sponsored by NSF #1924278. Special thanks will go to Dr. Jennifer Hogg at the Department of Health and Human Performance, University of Tennessee at Chattanooga, who provided thorough support to this project. Also, special thanks to Mr. Benjamin Spannuth for his contribution in leading the data collection process.

REFERENCES

[1] G. Laffaye, G. Levernier, and J.-M. Collin, "Determinant factors in climbing ability: Influence of strength, anthropometry, and neuromuscular fatigue," *Scandinavian Journal of Medicine and Science in Sports*, vol. 26, 09 2015.

[2] D. Saul, G. Steinmetz, W. Lehmann, and A. F. Schilling, "Determinants for success in climbing: A systematic review," *Journal of Exercise Science & Fitness*, vol. 17, no. 3, pp. 91–100, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1728869X19300723

[3] C. M. Mermier, J. M. Janot, D. L. Parker, and J. G. Swan, "Physiological and anthropometric determinants of sport climbing performance," *British journal of sports medicine*, vol. 34, no. 5, pp. 359–365, 2000.

[4] F. Quaine, L. Martin, and J. Blanchi, "Effect of a leg movement on the organisation of the forces at the holds in a climbing position 3-d kinetic analysis," *Human Movement Science*, vol. 16, no. 2, pp. 337–346, 1997, 3-D Analysis of Human Movement - II. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167945796000607

[5] F. Quaine, L. Martin, and J.-P. Blanchi, "The effect of body position and number of supports on wall reaction forces in rock climbing," *Journal of Applied Biomechanics*, vol. 13, no. 1, pp. 14–23, 1997.

[6] F. Fuss and G. Niegl, "Instrumented climbing holds and performance analysis in sport climbing," *Sports Technology*, vol. 1, pp. 301 – 313, 03 2009.

[7] A. Dobles, J. C. Sarmiento, and P. Satterthwaite, "Machine learning methods for climbing route classification," *Web link: http://cs229. stanford. edu/proj2017/finalreports/5232206. pdf*, 2017.

[8] C. Phillips, L. Becker, and E. Bradley, "strange beta: An assistance system for indoor rock climbing route setting," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 1, p. 013130, 2012.

[9] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *Journal of Imaging*, vol. 6, no. 8, 2020. [Online]. Available: https://www.mdpi.com/2313-433X/6/8/73

[10] J. Qi, G. Jiang, G. Li, Y. Sun, and B. Tao, "Intelligent human-computer interaction based on surface emg gesture recognition," *IEEE Access*, vol. 7, pp. 61 378–61 387, 2019.

[11] J. A. Hogg, J. Vanrenterghem, T. Ackerman, A.-D. Nguyen, S. E. Ross, R. J. Schmitz, and S. J. Shultz, "Temporal kinematic differences throughout single and double-leg forward landings," *Journal of biomechanics*, vol. 99, p. 109559, 2020.

[12] N. Draper, T. Dickson, G. Blackwell, S. Fryer, S. Priestley, D. Winter, and G. Ellis, "Self-reported ability assessment in rock climbing," *Journal of Sports Sciences*, vol. 29, no. 8, pp. 851–858, 2011, pMID: 21491325. [Online]. Available: https://doi.org/10.1080/02640414.2011.565362

[13] S. H. Hawley, "Panotti: A Convolutional Neural Network Classifier for Multichannel Audio Waveforms," 4 2018. [Online]. Available: https://github.com/drscotthawley/panotti

[14] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2114–2124.

[15] A. Velichko and H. Heidari, "A method for estimating the entropy of time series using artificial neural networks," *Entropy*, vol. 23, no. 11, p. 1432, oct 2021. [Online]. Available: